2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

LiDAR-Based Initial Global Localization Using Two-Dimensional (2D) Submap Projection Image (SPI)

Yanhao Li[4] and Hao Li*[1,2,3,4]

Abstract— **Initial global localization is important to mobile robotics in terms of navigation initialization (or re-initialization) and loop closure in SLAM. 3D LiDARs are commonly used for mobile robotics, yet LiDAR-based initial global localization (especially at large scale such as in outdoor environments) is still challenging due to lack of salient features in LiDAR range data. Inspired by visual SLAM oriented initial global localization methods, we propose a method of LiDAR-based initial global localization using 2D submap projection image (SPI). For this, global descriptors from SPIs are extracted for place recognition; pose estimation is realized by feature point matching between the queried SPI and SPIs from a global map database. The proposed initial global localization module runs at 2.4 Hz with precision of 1.2 m and for translation and 1.2° for rotation, which can serve as a suitable initial estimate for subsequent pose estimation refinement via existing mature point cloud registration methods.**

I. Introduction

Initial global localization is important to mobile robotics; it enables a robot to localize itself in a map without temporal *a priori* information. It is important to robot navigation initialization (or reinitialization) and loop closure in SLAM. LiDARs (Light Detection And Ranging) are commonly adopted in mobile robotics for its high measurement precision and robustness to environment conditions. However, LiDAR-based initial global localization (especially at large scale such as in outdoor environments) is still challenging due to lack of salient features in LiDAR range data.

Normally for visual SLAM, global localization is based on visual features and consists of two steps: place recognition and pose estimation. Place recognition aims at locating the robot roughly, typically by querying online visual frames in an image database. Pose estimation aims at computing a refined robot pose, typically by visual feature matching.

In contrast, using LiDAR point cloud features for initial global localization is still hard. Point cloud local descriptors such as NARF [1], FPFH [2] and IRON [3] suit dense output from RGBD or stereo cameras, but perform poorly on sparse LiDAR point clouds.

Inspired by visual SLAM oriented initial global localization methods, we exploit the idea of treating LiDAR range data as images and propose a method of LiDAR-based initial global localization using 2D submap projection image (SPI). The proposed method consists of two phases: place recognition and pose estimation. First, SPIs are formed by projecting scans of LiDAR point clouds onto the horizontal plane with each pixel storing the maximum height of points in its corresponding vertical bin. Then, the NetVLAD [4] that takes a SPI as input and outputs an associated global descriptor is trained; Extracted global descriptors are used for place recognition. Second, a subsequent neural network that integrates the SuperPoint [5] and the SuperGlue [6] performs feature point extraction on both the queried SPI and the candidate SPIs retrieved from the database. Then in the RANSAC spirit, the optimal (or semi-optimal) matching between the queried SPI and the candidate SPIs is searched and its associated pose estimate is computed.

Research supported by SJTU Young Talent Funding (WF220426002).

1.Department of Automation, Shanghai Jiao Tong University (SJTU), Shanghai, 200240, China. 2.Key Laboratory of System Control and Information Processing, Ministry of Education of China. 3.Shanghai Engineering Research Center of Intelligent Control and Management.

4.École d'Ingénieurs SJTU-ParisTech (SPEIT), Shanghai, 200240, China.

* Correspondence: Hao Li, Assoc. Prof. (Email: haoli@sjtu.edu.cn)

II. Related Works

For LiDAR-based initial global localization, one main category of methods rely on point cloud segmentation and segment descriptors [7]–[9]. However, such methods do not work when point cloud segmentation is difficult and not reliable, which often happens in complex outdoor environments.

Another category of methods rely on certain global descriptor that is aggregate of local point cloud features. Combining Bag of Words (BoW) [10] and local features, which is a common idea for visual global localization, is applicable to LiDAR-based cases if 3D point features can be extracted; examples include the SHOT [11], the NARF [1], and the FPFH [2]. Cop *et al.* propose the Delight [12] which forms descriptors from point intensity histograms. Uy and Lee *et al.* present the PointNetVLAD [13] that extracts local point features using PointNet and aggregate them into global descriptors using NetVLAD [4]. Based on the PointNetVLAD, Zhang *et al.* present the PCAN [14] which adds the point contextual attention network to learn task-relevant features. Liu *et al.* present the LPD-Net [15] that adds graph-based neighbourhood aggregation to the PointNetVLAD with local point structures fed as input.

Instead of extracting features directly from 3D point clouds, one may convert 3D point clouds to 2D structures for convenient feature extraction. He *et al.* present the M2DP [16] that projects a LiDAR point cloud onto multiple 2D planes and generates a density signature for points on each plane. Kim *et al.* advocate the Scan Context [17] that distributes a point cloud to sectors and rings with different directions to form a global descriptor with rotation-invariant distance metric, based on which the authors further present the Scan Context Image (SCI) [18] that transforms Scan Contexts into RGB images and treats global localization as

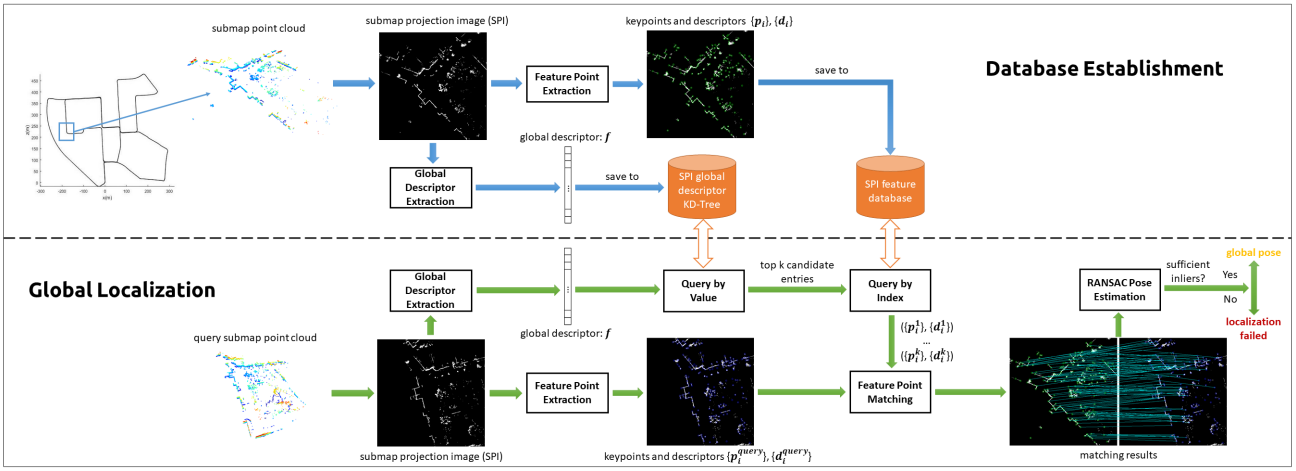


Fig. 1. Overall pipeline of SPI based global localization (including database establishment).

SCI classification using convolutional network. Chen *et al.* present the OverlapNet [19] that processes a point cloud as a 2D range image and then estimates the overlap ratio between scans for heading angle prediction.

Taking advantage of aforementioned feature aggregation and 2D space conversion methods, our proposed LiDAR-based initial global localization method performs NetVLAD-based global descriptor extraction on 2D submap projection images (SPI) and combines the SuperPoint [5] and the SuperGlue [6] to estimate the pose of a queried SPI with respect to the database map. Key points include:

- A method of LiDAR-based initial global localization using 2D submap projection image (SPI), which consists of two phases i.e. place recognition and pose estimation, is proposed.
- A NetVLAD-based method is proposed to extract global descriptors from SPIs for place recognition.
- The SuperPoint and SuperGlue are used in SPI feature matching.

III. Framework of Initial Global Localization Using 2D Submap Projection Image

The proposed LiDAR-based initial global localization framework (see Fig. 1) includes: 1) global descriptor extraction, 2) feature point extraction, 3) feature point matching, 4) pose estimation. Database establishment involves only global descriptor extraction and feature point extraction. For global localization, a query submap point cloud is converted to a global descriptor for place recognition, then candidate submaps are retrieved from the database. Feature points of the query submap projection image (SPI) are extracted and matched with candidate feature points to determine the query submap pose. The RANSAC mechanism is adopted to filter out false feature matching pairs for refined pose estimation.

IV. Submap Projection Image (SPI) Generation

Our Submap Projection Image (SPI) generator inputs a 3D submap point cloud $C = \{P_i \in \mathbb{R}^3\}$, divides the horizontal plane into square pixels in a *cartesian coordinate system*, and forms a projection image $I \in \mathbb{R}^{L \times L}$. Each pixel represents a vertical *bin* and stores the maximum height of points in its corresponding vertical bin. As a submap point cloud may contain a sequence of LiDAR scans $\{S_0, ..., S_n\}$, we select the middle scan $S_{n/2}$ as anchor and transform all scans in the frame of $S_{n/2}$ to form a consistent submap point cloud.

During SPI conversion, only the points within the horizontal square region $[-R_{max}, R_{max}]^2$ in the frame of $S_{n/2}$ are preserved to form a SPI of size $L \times L$, with pixel size $2R_{max}/L$. In practice, R_{max} is slightly smaller than the maximum LiDAR scanning range such that a SPI can cover most of submap points. The maximum height of each pixel is saturated in $[h_{min}, h_{max}]$, then normalized in $[0, 1]$ as:

$$F_{norm}(h) = \frac{h - h_{min}}{h_{max} - h_{min}} \tag{1}$$

Where h_{min} should be above the ground level so that annular textures of ground points are not mistaken for SPI features; h_{max} should surpass the maximum LiDAR scanning height.

V. Global Descriptor Extraction

The global descriptor extraction process aims at generating a NetVLAD descriptor for each SPI. We follow a typical approach of NetVLAD: first we process convolutional operations on raw image to form a D-channel embedding image, then we aggregate the vectors in D-dimension of embedding image pixels using NetVLAD to form a $(D \times K)$-dimensional VLAD vector. The NetVLAD is a layer that learns K cluster centers from a set of D-dimensional vectors with a differentiable soft-assignment machinery. In details, the VLAD vector $V \in \mathbb{R}^{D \times K}$ is computed by

$$\mathbf{V_k} = \sum_{i=1}^{n} \mathbf{a}_k(\mathbf{x}_i)(\mathbf{x}_i - \mathbf{c}_k) \tag{2}$$

Where $\mathbf{a}_k(\mathbf{x}_i) = e^{\mathbf{w}_k^T \mathbf{x}_i + b_k} / \sum_{k'} e^{\mathbf{w}_{k'}^T \mathbf{x}_i + b_{k'}}$. n is the number of pixels in the D-channel embedding image, $\mathbf{w}_k \in \mathbb{R}^D$ and $b_k \in \mathbb{R}$ are the weights and biases that determine the contribution of $\mathbf{x}_i \in \mathbb{R}^D$ to the k-th cluster, $\mathbf{c}_k \in \mathbb{R}^D$ is the origin of k-th cluster. The $\{\mathbf{w}_k\}$, $\{b_k\}$ and $\{\mathbf{c}_k\}$ are all learnable parameters.

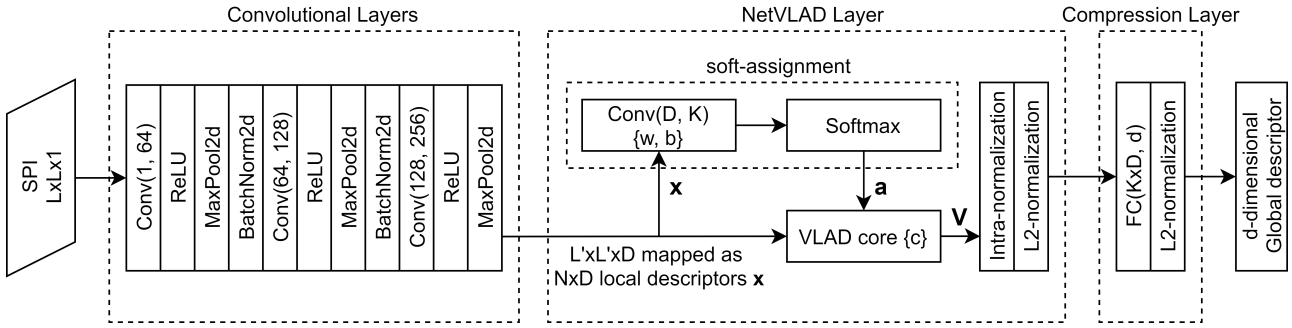


Fig. 2. SPI-NetVLAD: use of NetVLAD to generate descriptor for each SPI.

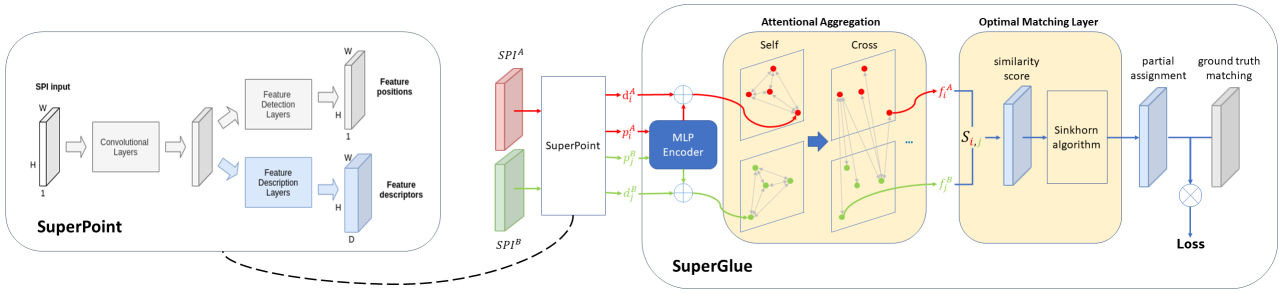


Fig. 3. End-to-end network with SuperPoint + SuperGlue for feature extraction and matching: A pair of SPIs are fed to the SuperPoint to get feature positions $\{p_i^A\}$, $\{p_j^B\}$, and feature descriptors $\{d_i^A\}$, $\{d_j^B\}$, respectively. Then all feature positions and descriptors are fed to the SuperGlue. The layers related to feature detection in the SuperPoint are fixed, the feature description layers in the SuperPoint and the whole SuperGlue are trained.

However, the $(D \times K)$-dimensional VLAD vector is in high dimension which is computationally expensive and which also degrades the global descriptor search with Nearest Neighbour (NN) model. To alleviate this problem, we follow the idea of PointNetVLAD to use a fully connected layer and a subsequent L2-normalization layer to compress the $(D \times K)$-dimensional VLAD vector to a final d-dimensional global descriptor.

We train our end-to-end global descriptor extraction network such that similar SPIs correspond to similar global descriptors. For this purpose, we use the "Lazy quadruplet" loss function [13] to learn the discriminative and generalizable descriptors. We generate a set of training tuples $(SPI_a, SPI_{pos}, \{SPI_{neg}\}, SPI_{neg^*})$ from dataset. SPI_a, SPI_{pos} and $\{SPI_{neg}\}$ are respectively denoted as anchor SPI, positive SPI and negative SPI. Anchor SPI and positive SPI are originated from two similar submaps, anchor SPI and negative SPI from dissimilar submaps. SPI_{neg^*} is from a random submap that is dissimilar to all submaps related to $\{SPI_{neg}\}$. The tuple of SPIs are then converted to a tuple of global descriptors $(f_a, f_{pos}, \{f_{neg}\}, f_{neg^*})$ through the neural network. The loss function is designed to minimize the squared point distance $\delta_{pos} = ||f_a - f_{pos}||_2^2$, and maximize the squared *normal set distances* [1] $\delta_{neg} = ||f_a - \{f_{neg}\}||_2^2$ and $\delta_{neg^*} = ||\{f_{neg}\} - f_{neg^*}||_2^2$. The "Lazy quadruplet" loss function is expressed as:

$$\begin{aligned} &\mathcal{L}_{lazyQuad}(f_a, f_{pos}, \{f_{neg}\}, f_{neg^*}) \\ &= [\alpha + \delta_{pos} - \delta_{neg_j}]_+ + [\beta + \delta_{pos} - \delta_{neg_k^*}]_+ \end{aligned} \tag{3}$$

with α a constant parameter indicating the margin.

Besides, the place recognition task with SPI strongly requires the global descriptors to be independent of rotation w.r.t Z-axis, since the wheeled-robot may visit a same place with different heading. To add the rotational invariance in our network, we augment our tuple SPIs by adding random rotation in the training phase.

VI. Feature-based Pose Estimation

A. Feature Extraction

Feature extraction includes both feature detection and feature description. We select the SuperPoint [5] for two-fold reasons: First, its feature detection performance is stable. Second, its feature description model is differentiable and hence suitable for SPI feature point description oriented training. We retain the original SuperPoint in general: It first accumulates local information of pixels with convolutional layers to get an intermediate layer, then performs feature detection and feature description with two separate networks respectively. The SuperPoint is illustrated in Fig. 3.

Since we do not have synthetic dataset of SPI features, and feature detection performance is good enough with the pretrained SuperPoint[2] network, all the layers related to feature detection are blocked during training.

B. Feature Matching

We leverage SuperGlue[2] for feature matching purpose. SuperGlue is characterized by its ability to learn topological information between features within an image and across two images. It incorporates an attentional graph neural network to propagate spatial information p_i and descriptor information

[1]Normal set distance: $||\mathbf{X} - \mathbf{Y}|| \equiv \inf\{||x - y|| : x \in \mathbf{X}, y \in \mathbf{Y}\}$

[2]https://github.com/magicleap/SuperGluePretrainedNetwork

d_i across the feature points inside an image with self-attention layers, and across two images with cross-attention layers, with output representations f_i^A and f_j^B containing topological information to form a similarity score matrix $S \in \mathbb{R}_+^{M\times N}$ of feature pairs. Then an optimal matching layer is leveraged to calculate a partial assignment matrix $P \in \mathbb{R}_+^{M\times N}$ by solving an optimal transport problem. To give the network the ability of dropping unmatched pairs, the score matrix and partial assignment matrix are respectively augmented to $\bar{S} \in \mathbb{R}_+^{(M+1)\times(N+1)}$ and $\bar{P} \in \mathbb{R}_+^{(M+1)\times(N+1)}$ to introduce *dustbins* for null matching, which is detailed in [6]. The loss function is designed to maximize the partial assignments for true matches and minimize for false matches, which is expressed by:

$$Loss = -\sum_{i=1}^{M+1}\sum_{j=1}^{N+1} \mathbf{M}_{i,j}^{gt} \ln \bar{P}_{i,j} \tag{4}$$

where $\mathbf{M}_{(M+1)\times(N+1)}^{gt}$ is the ground truth matching matrix.

$$\begin{aligned}
&\forall (i,j) \in \{1,...,M\} \times \{1,...,N\} \\
&\mathbf{M}_{i,j}^{gt} = \begin{cases} 1, \text{feature pair (i, j) is matched} \\ 0, \text{otherwise} \end{cases} \\
&\mathbf{M}_{i,N+1}^{gt} = \begin{cases} 1, \text{i-th feature in A has no match in B} \\ 0, \text{otherwise} \end{cases} \\
&\mathbf{M}_{M+1,j}^{gt} = \begin{cases} 1, \text{j-th feature in B has no match in A} \\ 0, \text{otherwise} \end{cases} \\
&\mathbf{M}_{M+1,N+1}^{gt} = 0
\end{aligned} \tag{5}$$

C. Self-Supervised Training

The end-to-end network is trained in self-supervised manner with ground truth matching labels generated on-the-fly. The SPI dataset contains tuples of (SPI_{tgt}, SPI_{src}, ${}^{tgt}\mathbb{T}_{src}$) representing pairs of (target SPI, source SPI) and 6-DoF transform between the target submap and source submap. SuperPoint module ingests a pair of images (SPI_{tgt}, SPI_{src}) and outputs keypoints in target SPI, $\mathcal{A} = \{\mathbf{p}_i^{tgt}, i = 1,...,M\}$, keypoints in source SPI, $\mathcal{B} = \{\mathbf{p}_j^{src}, j = 1,...,N\}$. Although the keypoints $\mathbf{p}_i$ are 2-dimensional pixel coordinates, with additional pixel values indicating height information we can easily restore the height value by $height = {F_{norm}}^{-1}(value)$ in equation 1 and thus retrieve the corresponding augmented 3D coordinates $\mathbf{X}_i$ in submap frame. Two keypoints are regarded as a matched pair if their corresponding 3D coordinates are close enough. The i-th keypoint in SPI_{tgt} and j-th keypoint in SPI_{src} are regarded as matched features if:

$$||\mathbf{X}_i^{tgt} - {}^{tgt}\mathbb{T}_{src}\mathbf{X}_j^{src}||_2 \leq \delta \tag{6}$$

with δ as a threshold for keypoint closeness verification.

D. Pose Estimation

In the inference stage, we feed a candidate SPI and a query SPI (SPI_{cdd}, SPI_{qry}) to the feature matching module and get matched keypoints $\{(\mathbf{p}_i^{cdd}, \mathbf{p}_i^{qry})\}$ representing the projection on OXY plane in a submap, which are converted augmented *2D coordinates* $\{(\mathbf{P}_i^{cdd}, \mathbf{P}_i^{qry})\}$. Then we calculate the relative 3-DoF pose (x, y, θ) between candidate and query submaps:

$$(x,y,\theta) = \underset{x,y,\theta}{\arg\min} \sum_i ||\mathbf{P}_i^{cdd} - T(x,y,\theta)\mathbf{P}_i^{qry}||_2^2 \tag{7}$$

with

$$T(x,y,\theta) = \begin{bmatrix} cos\theta & sin\theta & x \\ -sin\theta & cos\theta & y \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

We assume that the OXY plane in the frame of each submap is strictly horizontal, which can practically be ensured by transforming the original LiDAR point cloud submaps into a new frame with Z axis parallel to the gravity direction during LiDAR SLAM. Thus, the query and candidate submaps which are close to each other differ with a 3-DoF relative pose on the OXY plane. Given the 6-DoF pose of the candidate SPI in world frame ${}^w\mathbb{T}_{cdd}$, the 6-DoF pose matrix of the query SPI in world frame is

$${}^w\mathbb{T}_{qry} = {}^w\mathbb{T}_{cdd} \begin{bmatrix} cos\theta & sin\theta & 0 & x \\ -sin\theta & cos\theta & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

To further filter inlier feature matches given by SuperGlue, RANSAC approach is leveraged to select the best pose with maximum inlier matches. Besides, a query will be rejected if no candidate SPI provides sufficient feature matches with the query SPI.

VII. Experiments

The proposed method is evaluated using the KITTI odometry dataset in which ground-truth trajectories are provided by RTK-GPS. We train both SPI-NetVLAD place recognition and SuperPoint+SuperGlue modules using sequences 00 and 01, and evaluate the performance on sequences 02, 05 and 08. The submap we use for SPI generation consists of 3 consecutive LiDAR scans on KITTI dataset. Two adjacent submaps are 3 meters apart from each other on average.

To assess the generalization ability of our approach across different LiDARs and scenes, we further evaluate the performance of global localization modules on two custom datasets, which contain LiDAR scans from Velodyne HDL-32E captured in street blocks and an underground parking (fig. 4). No additional training is processed on this custom dataset. The ground truth trajectories in our custom dataset are provided by our SLAM module fusing RTK, IMU and LiDAR sensors.

The range of submap point cloud centered for SPI generation is $(x,y) \in [-50m, +50m]^2$, $z \in [h_{min}, h_{max}] = [0m, 10m]$. The size of SPI for SPI-NetVLAD is 1000 $\times$ 1000 pixels, and for feature-based pose estimation is 400 $\times$ 400 pixels. Parameters in SPI-NetVLAD: intermediate descriptor dimension $D = 256$, number of clusters in VLAD $K = 64$, final SPI global descriptor dimension $d = 256$, lazy-quadruplet loss margins $\alpha = \beta = 0.8$. Threshold for feature closeness verification in SuperPoint+SuperGlue ground truth matching matrix $\delta = 0.5m$. The minimum inlier

matched pairs of features to accept a localization result is 20. Number of candidates retrieved from SPI global descriptor database is $k = 3$. All experiments are processed on a machine with Intel Core i7-8700K CPU and NVIDIA GTX 1080 GPU.

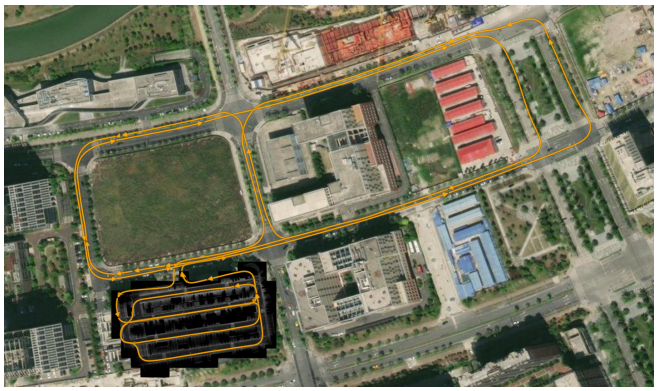


Fig. 4. Bird-eye's view of our custom dataset trajectory covering streets on the ground and an underground parking shown in perspective projected image.

A. Place Recognition

In our first experiments, we investigate the performance of SPI-NetVLAD-based place recognition approach, and compare it to existing methods including M2DP [16], Scan Context [17] and PointNetVLAD [13]. We exhaustively take all existing submap pairs in a test sequence, e.g. 1553×1553 for sequence 02, as positive and negative samples, and calculate AUC (Area Under the Curve) score based on the descriptor distance of each submap pair. A pair of submaps are regarded as positive if their centers are within the range of 5m. As can be seen in tab. I, our place recognition approach outperforms other methods.

We also study the place recognition performance on recall at top k retrieval candidates by comparing with PointNetVLAD (fig. 5). Both SPI-NetVLAD and PointNetVLAD are trained on sequence 00 and 01 for fair comparison. Submaps in each sequence are split in database submaps and query submaps for evaluation.

TABLE I

COMPARISON OF AUC VALUE FOR DIFFERENT PLACE RECOGNITION METHODS WITH ACCEPTANCE DISTANCE THRESHOLD = 5M

Methods	KITTI 02	KITTI 05	KITTI 08
M2DP [16]	0.935	0.894	0.900
Scan Context [17]	0.975	0.959	0.982
PointNetVLAD [13]	0.989	0.977	0.988
SPI-NetVLAD	**0.990**	**0.978**	**0.990**

B. Feature Extraction and Matching

For feature extraction evaluation, we analyze the *repeatability* of extracted features, representing the ratio of the number of existing matched pair formed to the average number of features detected in two images. For feature matching evaluation, precision and recall metrics are utilized. As is shown in tab. II, the blocked feature detection module in SuperPoint predicts slightly less repeatable features, while the overall feature matching scores of SuperPoint+SuperGlue are greatly superior to those of ORB [20] and SIFT [21]. We also evaluate the performance on our custom datasets for generalization ability analysis, which shows a slightly degraded but sufficient performance for subsequent RANSAC based pose estimation.

TABLE II

PERFORMANCE COMPARISON OF FEATURE EXTRACTION AND MATCHING WITH ARTIFICIAL DESCRIPTORS

feature	matcher	feature repeatability [%]	matching precision [%]	matching recall [%]
ORB	brute force	9.2	36.1	31.6
SIFT	brute force	**53.3**	51.4	11.1
SuperPoint	SuperGlue	50.2	**71.9**	**78.5**

TABLE III

PERFORMANCE OF SUPERPOINT+SUPERGLUE ON DIFFERENT DATASETS

dataset	avg. number of existing feature pairs	precision [%]	recall [%]
KITTI 02	64	71.8	80.3
KITTI 05	88	77.9	82.2
KITTI 08	87	68.0	73.9
Dataset 1	71	46.7	46.2
Dataset 2	78	58.4	57.9

C. Overall Performance

Finally, we evaluate the performance of our overall pipeline for LiDAR global localization. In addition to datasets with database and query data generated from same sequence, we also test our pipeline in a cross dataset (C-dataset) where database and query data are respectively from our datasets 1 and 2 recorded in a same trajectory with 1-week gap. An attempt of localization is considered successful if the number of final inlier matched pairs of features given by RANSAC pose estimation meets the minimum acceptance condition.

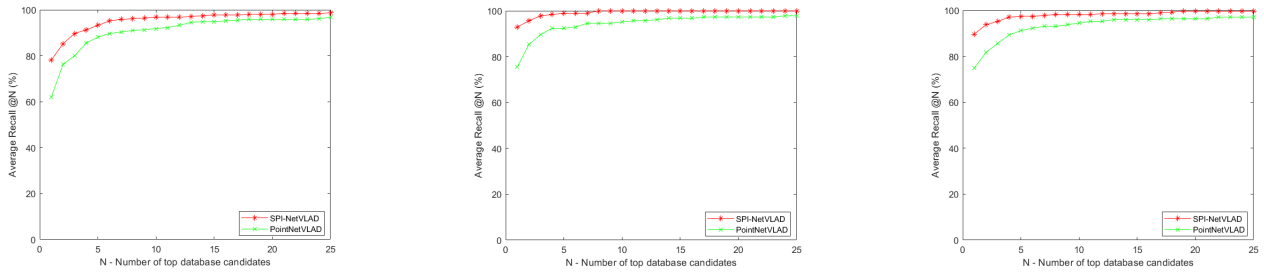
As can be seen in tab. IV, our pipeline is able to achieve the global localization with precision within 0.5 m for highly overlapped trajectories (KITTI 02, 05, 08, our datasets 1 and 2) and for normally overlapped trajectory (cross dataset), with rotation error less than 1.5 degrees, which is sufficient for subsequent point cloud registration methods (e.g. ICP) to refine the localization result.

TABLE IV

PERFORMANCE OF OUR PROPOSED LIDAR GLOBAL LOCALIZATION PIPELINE

dataset	translation mean [m]	translation std [m]	rotation mean [deg]	rotation std [deg]	recall [%]
KITTI 02	0.31	0.58	0.47	0.40	90.8
KITTI 05	0.19	0.16	0.43	0.36	97.1
KITTI 08	0.19	0.21	0.46	0.38	96.3
Dataset 1	0.40	0.37	0.54	0.70	69.5
Dataset 2	0.33	0.33	0.67	0.77	76.1
C-dataset	1.24	0.38	1.19	0.62	66.0

The average computational time of each module is shown in tab. V. Note that the computational time of feature matching and pose estimation correspond to one pair of



(a) KITTI 02 (b) KITTI 05 (c) KITTI 08

Fig. 5. Average recalls of PointNetVLAD and our SPI-NetVLAD

SPIs, which means the overall time consumption for feature matching and pose estimation needs to be multiplied by k if we retrieve k candidates from SPI global descriptor database. In our setting $k = 3$, thus the total computational time is about 424 ms and the proposed global localization system can run at 2.4 Hz.

TABLE V

COMPUTATIONAL TIME OF EACH MODULE IN THE PIPELINE

SPI generation	global descriptor generation	feature extraction	feature matching	pose estimation
66 ms	27 ms	7 ms	88 ms	20 ms

VIII. CONCLUSION

We have proposed a method of LiDAR-based initial global localization using 2D submap projection image (SPI), which consists of place recognition and feature-based pose estimation. Place recognition is based on global descriptors extracted by the proposed SPI-NetVLAD. The SuperPoint and the SuperGlue can accomplish SPI feature point extraction and matching, and provide sufficient and reliable matched pairs for RANSAC-based pose estimation. Thanks to good performance and low computational cost of each sub-module, the proposed initial global localization function can run at high rate with high precision.

In future, we intend to improve the proposed initial global localization method by taking local structures (linearity, planarity, etc.) and LiDAR point intensity into account to generate more discriminative global and local descriptors. Besides, the proposed initial global localization method will be evaluated with even extensive and diverse experiments.

ACKNOWLEDGEMENT

Thanks to Wayz.ai for providing partial experimental data.

REFERENCES

[1] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, "Narf: 3d range image features for object recognition," in *Workshops of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 44, 2010.

[2] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 3212–3217.

[3] T. Schmiedel, E. Einhorn, and H.-M. Gross, "Iron: A fast interest point descriptor for robust ndt-map matching and its application to robot localization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 3144–3151.

[4] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *IEEE Conf. on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[5] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Workshops of IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 224–236.

[6] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 4938–4947.

[7] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *IEEE Int. Conf. on Robotics and Automation*, 2017, pp. 5266–5272.

[8] R. Dubé, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, "Segmap: Segment-based mapping and localization using data-driven descriptors," *Int. J. of Robotics Research*, vol. 39, no. 2-3, pp. 339–355, 2020.

[9] G. Tinchev, A. Penate-Sanchez, and M. Fallon, "Learning to see the wood for the trees: Deep laser localization in urban and natural environments on a cpu," *IEEE Robotics and Automation Lett.*, vol. 4, no. 2, pp. 1327–1334, 2019.

[10] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," in *IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 3921–3926.

[11] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *European Conf. on computer vision*. Springer, 2010, pp. 356–369.

[12] K. P. Cop, P. V. Borges, and R. Dubé, "Delight: An efficient descriptor for global localisation using lidar intensities," in *IEEE Int. Conf. on Robotics and Automation*, 2018, pp. 3653–3660.

[13] M. Angelina Uy and G. Hee Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 4470–4479.

[14] W. Zhang and C. Xiao, "Pcan: 3d attention map learning using contextual information for point cloud based retrieval," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 12 436–12 445.

[15] Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li, and Y.-H. Liu, "Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis," in *IEEE Int. Conf. on Computer Vision*, 2019, pp. 2831–2840.

[16] L. He, X. Wang, and H. Zhang, "M2dp: A novel 3d point cloud descriptor and its application in loop closure detection," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 231–237.

[17] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 4802–4809.

[18] G. Kim, B. Park, and A. Kim, "1-day learning, 1-year localization: Long-term lidar localization using scan context image," *IEEE Robotics and Automation Lett.*, vol. 4, no. 2, pp. 1948–1955, 2019.

[19] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, C. Stachniss, and F. Fraunhofer, "Overlapnet: Loop closing for lidar-based slam," in *Proc. Robotics: Science and Systems*, 2020.

[20] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Int. Conf. on Computer Vision*, 2011, pp. 2564–2571.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.