LocNet: Global localization in 3D point clouds for mobile robots

Huan Yin, Yue Wang, Li Tang, Xiaqing Ding and Rong Xiong

Abstract—Global localization in 3D point clouds is a challenging problem of estimating the pose of robots without priori knowledge. In this paper, a solution to this problem is presented by achieving place recognition and metric pose estimation in the global priori map. Specifically, we present a semi-handcrafted representation learning method for LIDAR point cloud using siamese LocNets, which states the place recognition problem to a similarity modeling problem. With the final learned representations by LocNet, a global localization framework with range-only observations is proposed. To demonstrate the performance and effectiveness of our global localization system, KITTI dataset is employed for comparison with other algorithms, and also on our own multi-session datasets collected by long-time running robot to see the performance under semi-static dynamics. The result shows that our system can achieve both high accuracy and efficiency.

I. INTRODUCTION

Localization aims at estimating the pose of a mobile robot, thus it is a primary requirement for both mapping and navigation. In general, it is applied during the whole operation of the robot. When a map is given and the priori pose is centering on the correct position, pose tracking is enough to handle the localization of the robot [1]. In case the robot misses its pose, which is unavoidable since the map can be out-of-dated in practical scenarios, re-localization tries to re-localize the robot in the map. While for mapping, loop closing is a crucial component for global consistent map building, as it can localize the robot in the previously mapped area. Note that for both loop closing and re-localization, their underlying problems are the same, which requires the robot to localize itself in a given map without any prior knowledge as soon as possible. To unify the naming, we call this common problem as global localization in the sequel of this paper.

Global localization is relatively mature when the robot is equipped with a 2D laser range finder [2] [3]. For vision sensors, the topological loop closure is investigated in recent years [4] [5], and the pose is mainly derived by classical feature keypoints matching [6] [7], which suggests that the metric pose is still hard to be obtained when the changes happen between map and tasking sessions. The success of deep learning raised the learning based methods as new potential techniques for solving visual localization problems [8] [9]. However, for 3D point cloud sensor, e.g. 3D LIDAR, which is popular in many outdoor mobile robots, there are less works focusing on the global localization problem [10] [11], which mainly focused on designing representations for matching between sensor readings and map. Learning representation of point cloud is explored in objects detection [12], but inapplicable in localization problem. In summary, the crucial challenge we consider is the lack of compact and efficient representation for global localization.

In this paper, we propose a semi-handcrafted deep neural network, LocNet, to learn the representation for 3D LIDAR sensor readings, upon which, a Monte-Carlo localization framework is designed for global metric localization. The frame of our global localization system is illustrated in Fig. 1. The sensor readings is first transformed as a handcrafted rotational invariant representation, which is then passed through a network to generate a low-dimensional fingerprint. Importantly, when the representations of two readings are close, the two readings are collected in the same place with high probability. With this property, the sensor readings are organized as a global priori map, on which a particle based localizer is presented with range-only observations to achieve the fast convergence to correct robot pose.

The contributions of this paper are presented as follows:

• A handcrafted representation for structured point cloud is proposed, which achieves rotational invariance, guaranteeing this property in the network learned representation. This step reduces the amount and the variation in the data, possibly simplifying the learning and the network architecture.

• Siamese architecture is introduced in LocNet to model the similarity between two LIDAR readings. As the learning metric is built in the Euclidean space, the learned representation can be measured in the Euclidean space by simple computation, highly accelerating the matching.

• A global localization framework with range-only observations is proposed, which is built upon the priori map made up of poses and learned representations.

• An evaluation of the proposed method is conducted on both public single session dataset and collected multi-session dataset to illustrate the performance and feasibility.

The remainder of this paper is organized as follows: Section II describes related work of global localization in 3D point clouds. In Section III, we introduce the details of our system. We evaluate our system on KITTI benchmark and our own dataset in Section IV. Section V concludes with a brief discussion on our methods and an outlook of future works.

1 All authors are with the State Key Laboratory of Industrial Control and Technology, Zhejiang University, Hangzhou, P.R. China. Yue Wang is with iPlus Robotics Hangzhou, P.R. China. Yue Wang is the corresponding author wangyue@iipc.zju.edu.cn. Rong Xiong is the co-corresponding author.

This work was supported by the National Nature Science Foundation of China (Grant No. U1609210 and Grand No. 61473258)
II. RELATED WORKS

Global localization requires the robot to localize itself in a given map using current sensor readings without any priori knowledge of its pose. In general, the global localization consists of two stages, place recognition, which finds the frame in the map that is topologically close to the current frame, and metric pose estimation, which yields the relative pose from the map frame to the current frame, thus localizing the robot finally. We review the related works in two steps, place recognition and metric pose estimation.

As images provide rich information of the surrounding environments, the place recognition phase in the global localization is quite mature in the vision community, as it is similar to the image retrieval problem. Therefore, most image based localization method employed bag of words for description of a frame [4] [5]. Based on this scheme, sequences of images was considered for matching instead of single frame to improve the precision [13]. To enhance the place recognition across illumination variance, illumination invariant description was studied for better matching performance [14]. Besides, experienced-based navigation was proposed to form visual memory so that appearance of the same place under different illumination is connected [15]. However, given the matched frame in the map to the current frame, the pose estimation is still challenging, since the feature points changes a lot across illumination, and the image level descriptor cannot reflect the information of metric pose.

Compared with vision based methods, the place recognition in point cloud does not suffer from various illumination. A direct method of matching current LIDAR with the given map is registration. Go-ICP, a global registration method without initialization was proposed in [16]. But it is relatively computational complex, especially in the outdoor environments. Thus we still refer to the two-step method, place recognition then metric pose estimation. Some works achieved place recognition in the semantic level. SegMatch, presented by Dube et al. [10], tried to match to the map using features like buildings, trees or vehicles. With these matched semantic features, a metric pose estimation was then possible. In [17], planes were detected from 3D point clouds to perform place recognition for indoor scenarios, where sufficient walls, ceilings or tables compose a plane-based map. As semantic level feature is usually environment dependent, point features in point cloud are also investigated. Spin image [18] was a keypoint based method to represent surface of 3D scene. ESF [19] used disatance and angle properties to generate keypoints without computing normal vectors. Bosse and Zlot [20] proposed 3D Gestalt descriptor in 3D point clouds for matching. These works mainly focused on the local features in the frames for matching. In our methods, the frame level descriptor is studied.

For frame level descriptors, there were some works utilizing the handcrafted representation. Magnusson et al. [21] proposed an appearance based loop detection method using NDT surface representation. Röhling et al. [11] proposed a 1-D histogram to describe the range distribution of the entire point cloud. The Wasserstein metric was used to measure the similarity between histograms. For learning based method, Nieto and Ramos [22] used features that capture important geometric and statistical properties of 2D point clouds. The features are used as input to the machine learning algorithm.
Fig. 2: The framework of our siamese network for training including the generation process of handcrafted representation. The two sides of the network are same and share the same parameters. In the frame, \texttt{conv} stands for a convolution layer, \texttt{pool} stands for a MAX pooling layer and \texttt{FC} stands for a fully convolution layer. Pairs of image-like representations \{\textit{R}_1, \textit{R}_2\} and corresponding labels \{\textit{Y}\} are needed to train this network.

- Adaboosts to build a classifier for matching. Based on the deep learning, our method set to develop the learning based representation for 3D point cloud.

III. METHODS

The proposed global localization system includes two components as shown in Fig. 1, map building and localization, to estimate the correct metric pose of the robot. In the map building component, frames collected in the mapping session are transformed through LocNet to generate the corresponding fingerprints, forming a kd-tree based vocabulary for online matching as a global priori map. In addition, the global priori map also includes the metric pose graph and the global point cloud built by SLAM. The localization component is utilized in the online session, which transforms the current frame to the fingerprint using LocNet for searching similar frames in the global priori map. Besides, to estimate the metric pose, a particle filter is proposed with the matching result as a range-only observation, and the pose computed by the particle filter is refined using ICP and yielded as the final odometry as the motion information. The pose computed by LocNet is duplicated to form the siamese network for training. When being deployed in the localization system, the LocNet is extracted and the global priori map also includes the metric pose graph as shown in Fig. 2, map building and localization session are transformed through LocNet to generate the corresponding fingerprints, forming a kd-tree based vocabulary for online matching as a global priori map. In addition, the proposed global localization system includes two components as shown in Fig. 1, map building and localization, to estimate the correct metric pose of the robot. In the map building component, frames collected in the mapping session are transformed through LocNet to generate the corresponding fingerprints, forming a kd-tree based vocabulary for online matching as a global priori map. In addition, the global priori map also includes the metric pose graph and the global point cloud built by SLAM. The localization component is utilized in the online session, which transforms the current frame to the fingerprint using LocNet for searching similar frames in the global priori map. Besides, to estimate the metric pose, a particle filter is proposed with the matching result as a range-only observation, and the odometry as the motion information. The pose computed by particle filter is refined using ICP and yielded as the final global pose of the robot. One can see that the crucial module in both components is the LocNet which is shown in Fig. 2. The LocNet is duplicated to form the siamese network for training. When being deployed in the localization system, LocNet, i.e. one branch of the siamese network, is extracted and the index of rings from 1 to \(N\). In each ring, the points are sorted by azimuth angle \(\phi\), of which each point is denoted by \(p_k\), where \(k\) is the index determined by \(\phi\). As a result, in each LIDAR frame, the range measurement \(r\) of each point is indexed by ring index \(S^i_N\) and \(\phi\) index \(k\).

Given a ring \(S^i_N\), the 2D distance between two consecutive points \(p_k\) and \(p_{k-1}\) is calculated as

\[
d(p_k, p_{k-1}) = \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}
\]

which lies in a pre-specified range interval \(d \in [d_{\text{min}}, d_{\text{max}}]\). For a whole ring, we calculate all the distances between each pair of consecutive points. Then, we set a constant bucket count \(b\) and a range value \(I = [d_{\text{min}}, d_{\text{max}}]\), and divide \(I\) into subintervals of size:

\[
\Delta I_b = \frac{1}{b} \left(d_{\text{max}} - d_{\text{min}}\right)
\]

Each bucket corresponds to one of the disjunct intervals, show as follows:

\[
I^b_k = [d_{\text{min}} + m \cdot \Delta I_b, d_{\text{min}} + (m + 1) \cdot \Delta I_b]
\]

So all \(d\) in the ring \(S^i_N\) can find which bucket it belongs to. And the histogram for a ring \(S^i_N\) can be written as

\[
H^i_b = \{h^0_b, \cdots, h^{b-1}\}
\]

with

\[
h^m_b = \frac{1}{|S^i_N|} \left|\left\{p_k \in S^i_N : d(p_k, p_{k-1}) \in I^m_b\right\}\right|
\]

Theoretically, the number of LIDAR measurements per frame is a constant value. While in practice, some surface types (e.g. glass) may not return any meaningful measurements for some reason. The normalization ensures that the histograms remain comparable under these unexpected conditions.

Finally, we stack \(N\) histograms \(H^i_b\) together in the order of rings from top to bottom. Then a \(N \times b\) one-channel image-like representation \(R = (H^0_b, \cdots, H^{b-1})^T\) is produced.
from the point cloud $P$ using our transformation method. Using this representation, if the robot rotates at the same place, the representation keeps constant, thus rotational invariant.

One disturbance in global localization is the moving objects, as it may cause unpredictable changes of range distribution. By utilizing the ring information, this disturbance can be tolerated to some extent, since the moving objects usually occur near the ground, the rings corresponding to higher elevation angles are decoupled from these dynamics. Besides, using the difference of the ranges between consecutive points improves the translational invariance of the representation compared with the absolute range based descriptors [11].

### B. Learned representation

With the handcrafted representation $R$, we transform the global localization problem to an identity verification problem. Siamese neural network is able to solve this problem and transform the 2D representations to the feature vectors that we need to build the global map. The entire neural network for training is shown in Fig. 2.

In the siamese convolution neural network, the most critical part is contrastive loss function, proposed by Lecun et al. [23], shows as follows:

$$L(Y, R_1, R_2) = Y \frac{1}{2} (D_W)^2 + (1 - Y) \frac{1}{2} \max(0, m - D_W)^2$$

In the training step, the loss function needs a pair of samples to compute the final loss. Let label $Y$ be assigned to a pair of image-like representations $R_1$ and $R_2$ : $Y = 1$ if $R_1$ and $R_2$ are similar, representing the two places are close, i.e. the two frames are matched; and $Y = 0$ if $R_1$ and $R_2$ are dissimilar, representing the two places are not the same which is the most common situation in localization or mapping system.

In the contrastive loss function, $m > 0$ is a margin value, which has an influence on the role that dissimilar pairs play in the training step. The greater the margin value is, the harder the neural network is trained. In this paper, most dissimilar pairs of representations cannot be differentiated from similar pairs easily, so we set $m$ with a relatively high value.

The output of any side (Side 1 or Side 2 in Fig. 2) of siamese neural network is a $d$ dimensional feature vector $G_W(R) = \{g_1, \ldots, g_d\}$. The parameterized distance function to be learned from the neural network between $R_1$ and $R_2$ is $D_W(R_1, R_2)$, which represents the Euclidean distance between the outputs of $G_W$. Show as follows:

$$D_W(R_1, R_2) = \|G_W(R_1) - G_W(R_2)\|_2$$

Actually, the purpose of contrastive loss is try to decrease the value of $D_W$ for similar pairs and increase it for dissimilar pairs in the training step.

After the network is trained, in the testing step, we assume all places are the same to the query frame. So in order to achieve the similarity between a pair of input representations $R_1$ and $R_2$, we manually set the label $Y = 1$, then the contrastive loss is as following:

$$L(R_1, R_2) = \frac{1}{2} (D_W)^2$$

As one can see, if the matching of two places is real, the calculated contrastive loss should be a very low value. If not real, the loss should be a higher value, and the second part of original contrastive loss should be low to minimize $L(Y, R_1, R_2)$. So it is easy to judge the place recognition using a binary classifier $C_\tau$ with threshold $\tau$:

$$C_\tau(R_1, R_2) = \begin{cases} \text{true ,} & D_W \leq \tau \\ \text{false ,} & D_W > \tau \end{cases}$$

Parameter $\tau$ is critical for place recognition. In this paper, $D_W$ is integrated into observation model in a probabilistic way, so the thresholding process of $\tau$ is unnecessary in the approach of global localization. An important fact is that the final determination of the matching is simply based on the Euclidean distance between the learned representations of the current and the map frames.

In summary, the advantages of using neural network are obvious. Firstly, the low-dimensional representation, the fingerprint $G_W(R)$, is learned to be a Euclidean space, so it is convenient to build the priori map for global localization; secondly, the similarity is learned from the data, which is statistically better than artificially defined distance metrics.

### C. Global localization implementation

As previously described, global localization of mobile robots is based on global maps and place recognition algorithm. Accordingly, it is necessary to build a reliable priori global map first, especially for long-time running robots.

In this paper, we utilize both 3D LIDAR based SLAM technology to produce the map, which provides a metric pose graph $\{X\}$ with each node indicating a LIDAR frame, whose corresponding representations $\{G_W(R)\}$ is also included by passing the frame through LocNet, as well as the entire corrected global point cloud $P$. So, our priori global map $M$ is made up as follows:

$$M = \{P, \{G_W(R)\}, \{X\}\}$$

In order to improve the searching efficiency for real-time localization, a kd-tree $K$ is built based on set $\{G_W(R)\}$ in Euclidean space. Thus the priori global map is as follows:

$$M = \{P, K, \{X\}\}$$

The priori global map represents the memory of robot about the running environment essentially.

After the map is built, when a new point cloud $P_t$ comes at time $t$, it is firstly transformed to handcrafted representation $R_t$ and then to the dimension reduced feature $G_W(R_t)$ by LocNet. Finally, the matching result of it in $K$ are $R_K$ and the relative distance $D_K$. Suppose the closest pose in memory that $P_t$ matched with is $X_K$. Thus, the observation $Z$ of the matching can be regarded as a range only observation:

$$Z = \{X_K, p_K\}$$

where $p_K$ is the observation probability, defined as
\[ p_K = 1 - \frac{D_K}{D_M} \]

\( D_M > 0 \) is an tuned parameter, which determines how reliability the observation is. The greater \( D_K \) is, or the smaller \( D_M \) is, the more uncertain \( Z \) will be. Actually, we determine the value of \( D_M \) during the training and testing step of LocNet.

We utilize Monte Carlo Localization to estimate our robot 2D pose \( X_t \). The probability \( p_K \) is integrated into the weight of particles, shows as follows:

\[ P_{\text{weight}} = p_K \cdot p(X_K | \overline{X}_t) \]

In the expression, \( \overline{X}_t \) represents the estimated pose of one particle after fusing the odometry. \( p(X_K | \overline{X}_t) \) is calculated just based on the distance between the particle and the observed pose \( X_K \) according to the Gaussian distribution. In this way, particles with less weights are filtered in the resampling step, which can estimate the rotation of the robot after some steps.

Based on the produced \( X_t \), we set it as the initial value of ICP algorithm, which help the robot achieve a more accurate 6D pose by registering current point cloud \( f_t \) with the entire point cloud \( P \). In overall, the whole localization process is from coarse to fine.

IV. EXPERIMENTS

The proposed global localization system is evaluated in three aspects: the performance of place recognition that find correct matches, the feasibility and convergence of the localization, as well as the computational time. In the experiments, two datasets are employed. First, KITTI dataset [24], which is a public and popular robotic method benchmarking dataset, has the odometry datasets with both Velodyne 64 rings LIDAR sensor readings and the ground truth for evaluation. We pick the odometry benchmark 00, 02, 05, 06 and 08, since they contains loop closures. Second, we collect our own 21-session dataset with a length of 1.1km in each session across 3 days, called YQ21, from the university campus using our own robot to validate the performance of the localization under the temporally semi-static change. The ground truth of our datasets is built using DGPS aided SLAM and localization with handcrafted initial pose. Our robot equips a Velodyne 16 rings LIDAR sensor, which also tests the generalization when the methods are applied to different 3D LIDAR sensors. We select 6 sessions over 3 days for evaluation. As shown in Fig. 5, the paths in sessions are not strictly the same due to unpredictable obstacle avoidances on the road.

A. Training of LocNet

We implement our siamese network using caffe\(^1\). Traditional training step of place recognition need labeling pairs of samples \([10] [22]\). These positive samples are very few in a dataset. So we prefer to collect training materials in a more unsupervised way. For relatively high frequency of LIDAR sensor and low speed of mobile robots, two consecutive frames are almost collected at the same place, so we take them as a pair of positive sample; in a sequence of laser data and poses, if there is a distance \( p \) between two poses, we can take the corresponding point clouds as a pair of negative samples, shown as follows:

\[ \text{sample} = \begin{cases}  \text{positive} , & \| X_1 - X_2 \|_2 \leq p \\  \text{negative} , & \| X_1 - X_2 \|_2 > p \end{cases} \]

Thus, we can easily generate both positive and negative samples from a sequence of laser data using the dense poses to train the LocNet.

On different datasets, different strategies are applied to train LocNet. In KITTI dataset, we generate 11000 positive samples and 28220 negative samples from sequences without loop closures (01, 03, 04, 07 and 10), and set margin value \( m = 8 \) in the contrastive loss. While in YQ21 dataset, the first session in Day 1 is used to build the target global map \( M \). The 3 sessions in Day 2 are used to train LocNet and the last two sessions in Day 3: Day 3-1 and Day 3-2 are used to test the LocNet. The former is collected in the morning and the latter is in the afternoon, verifying that no illumination variance can affect the localization performance when point cloud is used. On YQ21, we finally generate 21300 positive samples and 40700 negative samples for training and set \( m = 12 \).

B. Place recognition performance

We compare LocNet with other three algorithms, mainly the 3D point cloud keypoints based method, and the frame descriptor based method: Spin Image [18], ESF [19] and Fast Histogram [11]. Spin Image and ESF are based on local descriptors, and we can use bag-of-words methods to match point clouds. Besides, we can transform the entire point cloud as a global descriptor, and the LIDAR sensor is the center of the descriptor. In this paper, we use the second way and use the C++ implementation of them in the PCL\(^2\). As for Fast Histogram and LocNet, the buckets of histograms are set with the same value \( b = 80 \). The dimensions of these final

\[^1\text{http://caffe.berkeleyvision.org/}\]
\[^2\text{http://pointclouds.org/}\]
Fig. 4: (a) results of Fast Histogram (FH) and LocNet with different \( p \) on sequence 00. (b) results of four methods with \( p = 3m \) on sequence 06. The proposed LocNet performs better than the other methods.

representations are: \( d_{SI} = 153 \), \( d_{ESF} = 640 \), \( d_{FH} = 80 \) and \( d_{LocNet} = 20 \). Note that the proposed learned representation has the lowest dimension.

In some papers, the test set is manually selected [21] [22], while we select to compute the performance by exhaustively matching any pairs of frame in the dataset. Based on the vectorized outputs of algorithms, we generate similarity matrix using kd-tree, then evaluate performance by comparing to the ground truth based similarity matrix. It’s a huge calculation, 4541 \( \times \) 4541 comparison times of sequence 00 for instance. However, the resulting performance can act as a lower bound.

**TABLE I: \( F_1 \) max score on sequence 00 with different \( p \)**

| Methods  | \( p = 2m \) | \( p = 3m \) | \( p = 5m \) | \( p = 10m \) |
|----------|---------------|---------------|--------------|--------------|
| Spin Image | 0.600 | 0.506 | 0.388 | 0.271 |
| ESF       | 0.432 | 0.361 | 0.285 | 0.216 |
| Fast Histogram | 0.612 | 0.503 | 0.384 | 0.273 |
| LocNet    | **0.717** | **0.642** | **0.522** | **0.385** |

**TABLE II: \( F_1 \) max score on other sequences with \( p = 3m \)**

| Methods  | Seq 02 | Seq 05 | Seq 06 | Seq 08 | Day 3-1 |
|----------|--------|--------|--------|--------|---------|
| Spin Image | 0.547 | 0.550 | 0.630 | 0.566 | 0.607 |
| ESF       | 0.461 | 0.371 | 0.439 | 0.423 | 0.373 |
| Fast Histogram | 0.513 | 0.569 | 0.597 | 0.521 | 0.531 |
| LocNet    | **0.702** | **0.689** | **0.714** | **0.664** | **0.614** |

The results of a place recognition algorithm are typically described using precision and recall metrics and their relationship via a precision-recall curve [25]. We use the maximum value of \( F_1 \) score to evaluate different precision-recall curves. With the ground truth provided by KITTI dataset, we consider that two places are the same if their Euclidean distance is below \( p \). So different values of threshold \( p \) determines the performances of algorithms. We test different values of \( p \) on sequence 00 and corresponding curves (Fast Histogram (FH) and LocNet) are shown in Fig. 4. Obviously, the higher threshold \( p \) is, the harder for detection will be, for the differences between 3D point clouds will be harder to describe.

Besides, we set \( p = 3m \) and test the four methods on other sequences and the results of sequence 06 is in Fig. 4. \( F_1 \) scores of these methods of different tests are shown in TABLE I and TABLE II. The proposed LocNet achieves the best performance with the lowest dimensional representation compared to other methods. For precision-recall curves, the area under curve is the biggest using the proposed method. In addition, the place recognition result on sequence 00 is shown as Fig. 5. These experiments all support that our system is effective to yield the correct topologically matching frame in the map.

**C. Localization probability**

In order to evaluate the performance of localizing, [10] shows the probability of traveling a given distance without successful localization in the target map, which is defined as follows:

\[
P(x) = \frac{\text{Sum of distance travelled without localization for greater or equal to } x \text{ meters}}{\text{Total distance travelled}}
\]

**TABLE III:**

| Methods  | Seq 02 | Seq 05 | Seq 06 | Seq 08 | Day 3-1 |
|----------|--------|--------|--------|--------|---------|
| Spin Image | 0.547 | 0.550 | 0.630 | 0.566 | 0.607 |
| ESF       | 0.461 | 0.371 | 0.439 | 0.423 | 0.373 |
| Fast Histogram | 0.513 | 0.569 | 0.597 | 0.521 | 0.531 |
| LocNet    | **0.702** | **0.689** | **0.714** | **0.664** | **0.614** |

Thanks to the open source code of SegMatch\(^3\) we run 10 times in sequence 00 on the entire provided target map and present the average result. We record all the segmentation

\(^3\)https://github.com/ethz-asl/segmatch
Fig. 6: Localization probability of traveling a given distance before localizing on the target global map of sequence 00. SegMatch runs 10 times and presents the average result; and others run once.

points as the successful localizations from the beginning to the end. As for the other four methods, we build the kd-tree based on the vectorized representations and look for the nearest pose when the current frame comes. If it is a true positive matching, we record it as a successful localization. There are no random factors in these methods, so we run them only once.

As shown in Fig. 6, the robot can localize itself within 5 meters 100% of the time using our method, outperforming the other methods. It is nearly 90% for Fast Histogram and Spin Image. SegMatch is based on the semantic segmentation of the environments, so it needs an accumulation of 3D point clouds. The other four methods we test are based on one frame of LIDAR data, so they can achieve a higher percentage of the timing within 10 meters. There are too many false positive matches using ESF, which makes the robot localized failed above 30 meters on several sections.

D. Global localization Performance

We demonstrate the performance of our global localization system in two ways: re-localization and position tracking. If a robot loses its location, re-localization ability is the key to help it localize itself; and the coming position tracking achieve the localization continually.

1) Re-localization: The convergence rate of re-localization actually depends on the number of particles and the first observation they get in Monte-Carlo Localization. So it is hard to evaluate the convergence rate. We give a case study of re-localization of our robot in YQ21 dataset (see Fig. 7). The error of yaw angle and the positional angle are presented, together with the root mean square (rmse) error of registration, which is the Euclidean distance between the aligned point clouds $P$ and $P_t$.

Obviously the re-localization process can converges to a stable pose within 5m operation of the robot. And a change in location makes a change in the registration error. In multi-session datasets, the point clouds for localization are different from those frames forming the map, so the registration error can not be decreased to zero using ICP.

2) Position tracking: Once the robot is re-localized, we use our global localization to achieve the position tracking of the robot. The range-only observations provide reliable information of robot global pose, and ICP algorithm can help the robot localize more accurately. We test both Day 3-1 and Day 3-2 and the continuous localization results are presented in Fig. 8.

Based on the statistics analysis, our global localization can achieve high accuracy of position tracking. 93.0% of location errors are below 0.1m in Day 3-1, and 88.9% in Day 3-2; for rotational errors, 93.1% of heading errors are below 0.2° in Day 3-1, and 90.7% in the afternoon of Day 3-2. The negligible errors cause no effect for further navigation actions of the robot.

3) Computational time: We run the localization modules on the target global map 200 times and compute the average time cost of each modules. There are more than 10000 poses in the map. The computational device of the proposed method is on an Intel i5-5200U @ 2.20GHz. The whole system is implemented using Matlab. Since the robot in practice may not have the GPU equipped, we only use CPU for deep network computation. As shown in TABLE II LocNet module consists of the produce of handcrafted representation and passing through the neural network. Matching the nearest pose on the kd-tree takes the minimum amount of time because the learned representation is assigned with Euclidean distance and with the lowest dimensional representation. Monte-Carlo Localization requires times for computing the states of particles. ICP is time consuming, but not run for each new frame, as the odometry can be used for motion propagation, the whole system runs in real-time.

TABLE III: Average timing of each localization module

| Module   | LocNet | Match | MCL | Total |
|----------|--------|-------|-----|-------|
| Time Cost (s) | 0.207  | 0.001 | 0.004 | 0.212 |

V. CONCLUSION

This paper presents a global localization system in 3D point clouds for mobile robots. The global localization...
method is based on learning representations by LocNet in Euclidean space. The final representations are used to build the priori map for global localization. We demonstrate our place recognition performance and global localization system on KITTI dataset and YQ21 dataset, compared to other algorithms.

Our method is easy to implement and does not require labeling matched samples. And we take full use of every frame of laser data to build the map. Besides, the learned metric distance by LocNet make the similarity measure get rid of handcrafted functions. In the future, we would like to use the continuous similarity value for re-localization planning, that can help the robot to find its location using shortest time.

**REFERENCES**

[1] S. Thrun, “Probabilistic robotics,” Communications of the Acm, vol. 45, no. 3, pp. 52–57, 2005.
[2] E. Olson, “M3sm: Many-to-many multi-resolution scan matching,” in Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015, pp. 5815–5821.
[3] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2d lidar slam,” in IEEE International Conference on Robotics and Automation, 2016, pp. 1271–1278.
[4] D. Filliat, “A visual bag of words method for interactive qualitative localization and mapping,” in IEEE International Conference on Robotics and Automation, 2011, pp. 3921–3926.
[5] M. J. Cummins and P. M. Newman, “Fab-map: Probabilistic localization and mapping in the space of appearance,” International Journal of Robotics Research, vol. 27, no. 6, pp. 647–665, 2008.
[6] M. J. M. Mur-Artal, Raul and J. D. Tardos, “ORB-SLAM: a versatile and accurate monocular SLAM system,” IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147–1163, 2015.
[7] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in European Conference on Computer Vision. Springer, 2014, pp. 834–849.
[8] Z. Chen, O. Lam, A. Jacobson, and M. Milford, “Convolutional neural network-based place recognition,” Computer Science, 2014.
[9] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A convolutional network for real-time 6-dof camera relocalization,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 2938–2946.
[10] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, “Segmatch: Segment based loop-closure for 3d point clouds,” arXiv preprint arXiv:1609.07720, 2016.
[11] T. Röhring, J. Mack, and D. Schulz, “A fast histogram-based similarity measure for detecting loop closures in 3-d lidar data,” in Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. IEEE, 2015, pp. 736–741.
[12] B. Li, T. Zhang, and T. Xia, “Vehicle detection from 3d lidar using fully convolutional network,” arXiv preprint arXiv:1608.07916, 2016.
[13] W. Wohlkinger and M. Vincze, “Ensemble of shape functions for 3d object classification,” in IEEE International Conference on Robotics and Automation, 2012, pp. 1643–1649.
[14] T. Naseer, M. Ruhnke, C. Stachniss, and L. Spinello, “Robust visual slam across seasons,” in IEEE/rsj International Conference on Intelligent Robots and Systems, 2015, pp. 2529–2535.
[15] C. Linegar, W. Churchill, and P. Newman, “Work smart, not hard: Recalling relevant experiences for fast-scale but time-constrained localisation,” in IEEE International Conference on Robotics and Automation, 2015, pp. 90–97.
[16] J. Yang, H. Li, and Y. Jia, “Go-ipc: Solving 3d registration efficiently and globally optimally,” in IEEE International Conference on Computer Vision, 2013, pp. 1457–1464.
[17] E. Fernandez-Moral, W. Mayol-Cuevas, V. Arevalo, and J. Gonzalez-Jimenez, “Fast place recognition with plane-based maps,” in IEEE International Conference on Robotics and Automation, 2013, pp. 2719–2724.
[18] A. E. Johnson and M. Hebert, “Using spin images for efficient object recognition in cluttered 3d scenes,” IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 24, no. 5, pp. 433–449, 2002.
[19] W. Wohlkinger and M. Vincze, “Ensemble of shape functions for 3d object classification,” in IEEE International Conference on Robotics and Biomimetics, 2012, pp. 2987–2992.
[20] M. Bosse and R. Zlot, “Place recognition using keypoint voting in large 3d lidar datasets,” in IEEE International Conference on Robotics and Automation, 2013, pp. 2677–2684.
[21] M. Magnusson, H. Andreason, A. Nuchter, and A. J. Lilienthal, “Appearance-based loop detection from 3d laser data using the normal distributions transform,” Journal of Field Robotics, vol. 26, no. 11–12, pp. 892–914, 2009.
[22] J. I. Nieto and F. T. Ramos, “Learning to close loops from range data,” International Journal of Robotics Research, vol. 30, no. 14, pp. 1728–1754, 2011.
[23] R. Hadsell, S. Chopra, and Y. Lecun, “Dimensionality reduction by learning an invariant mapping,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 1735–1742.
[24] R. Urtasun, P. Lenz, and A. Geiger, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354–3361.
[25] S. Lowry, N. Snderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, “Visual place recognition: A survey,” IEEE Transactions on Robotics, vol. 32, no. 1, pp. 1–19, 2016.