OREOS: Oriented Recognition of 3D Point Clouds in Outdoor Scenarios

Lukas Schaupp[1], Mathias Bürki[1,2], Renaud Dubé[1,2], Roland Siegwart[1], Cesar Cadena[1]

***Abstract*— We introduce a novel method for oriented place recognition with 3D LiDAR scans. A Convolutional Neural Network is trained to extract compact descriptors from single 3D LiDAR scans. These can be used both to retrieve near-by place candidates from a map, and to estimate the yaw discrepancy needed for bootstrapping local registration methods. We employ a triplet loss function for training and use a hard-negative mining strategy to further increase the performance of our descriptor extractor. In an evaluation on the NCLT and KITTI datasets, we demonstrate that our method outperforms related state-of-the-art approaches based on both data-driven and handcrafted data representation in challenging long-term outdoor conditions.**

I. Introduction

Global localization constitutes a pivotal component for many autonomous mobile robotics applications. It is a requirement for bootstrapping local localization algorithms and for re-localizing robots after temporarily leaving the mapped area. Global localization can furthermore be used for mitigating pose estimation drift through loop-closure detection and for merging mapping data collected during different sessions. Prior-free localization is especially challenging for autonomous vehicles in urban environments, as GNSS-based localization systems fail to provide reliable and precise localization near buildings due to multi-path effects, or in tunnels or parking garages due to a lack of satellite signal reception. Due to their rich and descriptive information content, camera images have been of great interest for place recognition, with mature and efficient data representations and feature descriptors evolving in recent years. However, visual place recognition algorithms struggle to cope with strong appearance changes that commonly occur during long-term applications in outdoor environments, and fail under certain ill-lighted conditions [1]. In contrast to that, active sensing modalities, such as LiDAR sensors, are mainly unaffected by appearance change [2]. Efficient and descriptive data representations for place recognition using LiDAR point clouds remain, however, an open research question [3]–[5]. In contrast to our work, typical place recognition methods do not always explicitly deal with the full problem of estimating a 3 DoF transformation [6] [7] [8].

This paper addresses the aforementioned issue by presenting a data-driven descriptor for sparse 3D LiDAR point clouds which allows for long-term 3 DoF metric global localization in outdoor environments. Specifically, our method allows us to estimate the relative orientation between scans. Our novel data-driven metric global localization descriptor is fast to compute and robust with respect to long-term appearance changes of the environment, and shows similar place recognition performance compared to other state-of-the-art LiDAR place recognition approaches. Additionally our architecture provides orientation descriptors capable of predicting a yaw angle estimation between two point clouds realizations of the same place. Our contributions can be summarized as follows:

- We present OREOS: an efficient data-driven architecture for extracting a point cloud descriptor that can be used both for place recognition purposes and for regressing the relative orientation between point clouds.
- In an evaluation using two public dataset collections, we demonstrate the capability of our approach to reliably localize in challenging outdoor environments across seasonal and weather changes over the course of more than a year. We show that our approach works even under strong point cloud misalignment, allowing the arbitrary positioning of a robot.
- A computational performance analysis showing that our proposed algorithm exhibits real-time capabilities and performs similarly to other state-of-the-art approaches in place recognition performance while providing ro-

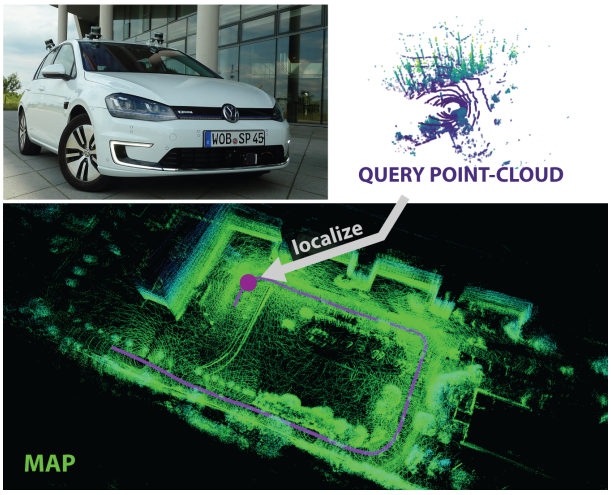

Fig. 1: We aim at accurately localizing our vehicle in a map build from previously collected LiDAR point clouds. A query point cloud scan is fed through the OREOS pipeline, yielding a compact descriptor that allows to retrieve near-by place candidates from the map, and estimate the yaw angle discrepancy. With this information, a local registration method, such as ICP, can be bootstrapped and used for subsequent high accuracy localization along the traversal.

This research has received funding from the EU H2020 research project under grant agreement No 688652, the Swiss State Secretariat for Education, Research and Innovation (SERI) No 15.0284

[1]Autonomous Systems Lab (ASL), ETH Zürich, Switzerland {firstname.lastname}@ethz.ch

[2]Sevensense Robotics AG, Switzerland {firstname.lastname}@sevensense.ch

bustness and better performance in the metric global localization.

The paper is structured as follows. After an overview over related work, we describe the OREOS metric global localization pipeline in Section III, before presenting our evaluation results in Section IV.

II. RELATED WORK

We subdivide the related work into two categories. First we discuss related approaches for solving the place recognition problem. Then we show related work on pose estimation for 3D point clouds.

Place Recognition: Early approaches to solving the place recognition problem with LiDAR data have, analogous to visual place recognition, focused on extracting keypoints on point clouds and describing their neighborhood with structural descriptors [9]. Along this vein, Bosse *et al.* used a 3D Gestalt descriptor [10], Steder *et al.* [11] and Zhuang *et al.* [12] transformed a point cloud to a range or bearing-angle based image by extracting local-invariant ORB [13] features for database matching. The strength of these approaches is the explicit extraction of low-dimensional data representations that can efficiently be queried in a nearest neighbor search. The representation, however, is handcrafted, and may thus not capture all relevant information efficiently in every application scenario. Furthermore, the dependence on good repeatability inherent to the keypoint detection constitutes an additional challenge for these approaches, especially if the sensor viewpoint is slightly varying. The drawbacks of keypoint-based approaches can be tackled by employing a segmentation of the point clouds, and computing place dependent data representations on these segments for subsequent place recognition [5], [14]–[16]. As a requirement for a proper segmentation, giving the sparsity of the data, these methods require the subsequent point clouds to be temporarily integrated and smoothed. In contrast to that, our data representation for place recognition can be computed directly from a single point cloud scan, which obviates any assumption on how the LiDAR data is collected and processed, and even allows to obtain localization without movement. Related approaches that compute handcrafted global descriptors for place recognition from aggregated point clouds are presented by Cop *et al.* [8], who generate distributed histograms of the intensity channel. Along a similar vein, Rohling *et al.* [17] represent each point cloud with a global 1D histogram, and Magnusson *et al.* [14] use the transform-based surface feature NDT (Normal Distribution Transformation). Further global descriptors such as GASD [18], and the extended FPFH - VFH [19] can be also used for the task of place recognition. Recent advances in machine learning have opened up new possibilities to deal with the weaknesses of handcrafted data presentation for place recognition with LiDAR data. Employing a DNN (Deep Neural Network) to learn a suitable data representation from point clouds for place recognition allows for implicitly encoding and exploiting the most relevant cues in the input data. Within this field of research Yin *et al.* LocNet [6] use semi-handcrafted range histogram features as an input to a 2D CNN (Convolutional Neural Network), while Uy *et al.* use a NetVLAD [20] layer on top of the PointNet [21] architecture [7]. Furthermore, Kim *et al.* [22] recently presented the idea to transform point clouds into scan context images [23] and feed them into a CNN for sovling the place recognition problem. Apart from the work by Uy *et al.*, all these approaches depend on a precomputed handcrafted descriptor, which may not represent all relevant information in an optimal, in this case most compact, manner. In contrast to that, we refrain from any pre-processing of the point clouds and directly employ our DNN on the raw 2D projected LiDAR data. In comparison to LocNet, our data-driven method is capable of learning a descriptor that is used both for fetching a nearest neighbour place and for estimating the orientation, which is not possible after the computation of the inherently rotation invariant histogram representation.

Pose Estimation: Common approaches to retrieve a 3 DoF pose from LiDAR data employ either local features extraction such as FPFH [24] and feature matching using RANSAC [25], or use handcrafted rotation variant global features such as VFH [19] or GASD [18]. An overview of recent research on 3D pose estimation and recognition is given by Han *et al.* [26]. Velas *et al.* [27] propose to use a CNN to estimate both translation and rotation between successive LiDAR scans for local motion estimation. In contrast to this, we aim for solving the metric global localization problem, and demonstrate that the best performance is obtained by a combination of learning and classical registration approaches.

III. METHODOLOGY

We first define the problem addressed in this paper, and outline the pipeline we propose for solving it, before elaborating in detail on our Neural Network architecture and the training process.

A. Problem Formulation

Our aim is to develop a metric global localization algorithm, yielding a 3 DoF (x, y, θ) in the map reference frame from a single 3D LiDAR point cloud scan C. This can formally be expressed with a function f as follows:

$$x, y, \theta \coloneqq f(C), \textit{ with } x, y, \theta \in \mathbb{R} \tag{1}$$

To solve this problem, we divide function f, as depicted in Figure 2, into the following four sequential components: a) Point Cloud Projection b) Descriptor Extraction c) Yaw Estimation, and d) Local Point Cloud Registration. The Point Cloud Projection module converts the input LiDAR point cloud scan C, given by a list of point coordinates p_x, p_z and p_z within the sensor frame, onto a 2D range image using a spherical projection model:

$$\varphi = atan(\frac{p_y}{p_x}) \tag{2}$$

$$\rho = asin(\frac{y_s}{\sqrt{p_x^2 + p_y^2 + p_z^2}}) \tag{3}$$

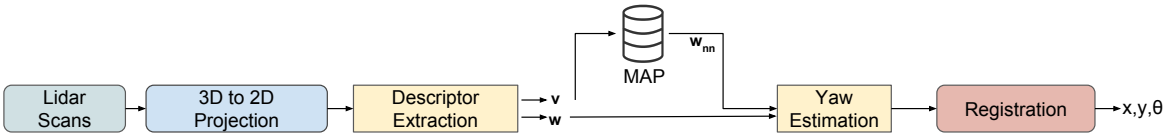

Fig. 2: In a first step, we project the current 3D LiDAR scan onto a 2D range image. In a second stage, a Convolutional Neural Network is leveraged for extracting two compact descriptors v, and w. The former is used in a k-Nearest-Neighbor (kNN) search to retrieve the w_{nn} from the closest place candidate in the map. Both descriptors w and w_{nn} are then fed to into a Yaw Estimation Neural Network to estimate the yaw angle discrepancy between the query point cloud and the point cloud of the nearest place in the map. Finally, an accurate 3 DoF pose estimate is obtained by applying a local registration method, using the orientation estimate and the planar x and y coordinates from the nearest map place as an initial guess.

The zenith θ and azimuth φ angles are directly mapped onto the image plane, yielding a 2D range image. For our work we use the whole 360 degree field of view given by one point cloud scan and the range information of the sensor.

The Descriptor Extraction module aims at deriving a compact representation for place-, and orientation related information from the input data. This is achieved by employing a Convolutional Neural Network, taking the normalized 2D range image as an input and generating two compact descriptors vectors v, and w respectively. While v represents rotation invariant place dependent information, w encodes rotation variant information used for determining a yaw angle discrepancy in a latter stage of the pipeline.

The place specific vector v can be used to query our map for nearby place candidates, yielding a map position x_{nn}, y_{nn}, and orientation descriptor w_{nn} of a nearest neighbor place candidate. In the subsequent step, the Yaw Estimation module estimates a yaw angle discrepancy $\delta\theta$ between the query point cloud C, and the point cloud associated with the retrieved nearest place in the map. For this, the two orientation descriptors w, and w_{nn} are fed into a small, fully connected Neural Network, which directly regresses a value for $\delta\theta$.

The position of the map place candidate x_{nn}, and y_{nn}, together with the yaw discrepancy $\delta\theta$, can then be used as an initial condition for further refining the pose estimation, yielding the desired highly accurate 3DoF pose estimate x, y, and θ of the point cloud C in the map coordinate frame.

Note that in our map, the place dependent descriptors v extracted from point cloud scans of a map dataset are organized in a kd-tree for fast nearest neighbor search. Retrieving the orientation descriptor w_{nn} of a map place candidate can be achieved by a simple look-up table.

B. Network Architecture

The network architecture of the CNN used for the descriptor extraction is based on the principles described in [28], [29]. We use a combination of 2D Convolutional and Max Pooling Layers for feature extraction. Subsequent fully connected layers map the features into a compact descriptor representation as depicted Figure 3. As proposed by Simonyan *et al.* [28], we use smaller filters rather than larger filters as well as designed the network around the receptive field size. Additionally, we use asymmetric pooling layers at the beginning of the architecture to further increase the descriptor retrieval performance.

In contrast to that, our Yaw Estimation network is composed of two fully-connected layers.

C. Training the OREOS descriptor

The two neural networks pursue two orthogonal goals, namely finding a compact place dependent descriptor representation for v, and finding a compact orientation dependent descriptor representation for w. For each of these two goals, a loss term is defined, denoted by the place-recognition loss L_{pr}, and orientation loss L_θ, respectively.

Place-Recognition Loss: To train our network for the task of place recognition, we use the triplet loss method [30]. The loss-function is designed to steer the network towards pushing similar and dissimilar point-cloud pairs close together and far apart in the resulting vector space. Let N_{DE} denote our descriptor extraction network, and let I_A denote an anchor range image, I_S a range image from a similar place, and I_D a range image from a dissimilar place. The Neural Network N_{DE} transforms these input images into three place dependent output descriptors v_A, v_S, v_D as depicted in Figure 3. We further define δ_S as the euclidean distance between descriptors of the anchor and similar place, and δ_D as the distance between descriptors of the anchor and the dissimilar one, and m as a margin parameter for separating similar and dissimilar pairs. The triplet loss can then be defined as follows:

$$\begin{aligned} L_{pr}(D_p, D_n) &= D_p^2 - D_n^2 + m \\ \textit{with } D_p &= \|f(I_i^A) - f(I_i^S)\|_2^2 \\ \textit{and } D_n &= \|f(I_i^A) - f(I_i^D)\|_2^2 \end{aligned} \tag{4}$$

Orientation Estimation Loss: As we want to predict an orientation estimate, we are implementing a L_θ regression loss function. For this task, we add an additionally fully-connected layer at the end of the triplet network. In this case we make only use of the anchor image I_A and the similar image I_S and obtain our rotation dependent descriptors w_A and w_S from our descriptor extraction network N_{DE}. We then feed the obtained descriptors w_A and w_S into a additional orientation estimation network that yields the yaw angle discrepancy descriptor y_{yaw} between both given point clouds which is then compared to our ground truth yaw discrepancy angle $\delta\theta_{gt}$. By transforming the ground truth yaw angle $\delta\theta_{gt}$ into the euclidean space, the ambiguity between 0 and 360 degree angles is avoided, which would result in false corrections during training. The orientation

loss term is defined as follows:

$$L_\theta(y_{yaw}, \delta\theta_{gt}) = \frac{1}{2}((y_{yaw,0} - cos(\delta\theta_{gt}))^2 + (y_{yaw,1} - sin(\delta\theta_{gt}))^2) \quad (5)$$

where $y_{yaw,i}$ represents the i-th index of our yaw angle discrepancy descriptor y_{yaw}.

Joint Training: As it is the goal of our proposed metric localization algorithm to both achieve a high localization recall with an accurate yaw angle estimation, we learn the weights of both Neural Network architectures in a joint training process. For this, both loss terms are combined into a joined loss L as follows:

$$L = L_{pr} + L_\theta \quad (6)$$

The joint training consists of a three-tuple network, whereas we sample point clouds based on the euclidean distance of their associated ground truth poses and a predefined distance threshold ρ. The three point clouds are fed after the 2D projection into the Descriptor Extractor network, and the corresponding three place dependent output vectors v_A, v_S, and v_D are fed into the Place-Recognition Loss L_{pr}. In contrast to that, the two orientation specific vectors w_A, and w_S from the two close-by point clouds are fed into the Orientation Loss L_θ. The combined loss L is then evaluated as described in Equation 6. We use ADAM [31] as a learning optimizer and use a learning rate of alpha = 0.001. We convert our range data to 16 bit and normalize the channel before training. To achieve rotation invariance for our place recognition descriptor v and generate training data for our yaw angle discrepancy descriptor w, we employ data augmentation by randomly rotating the input image around its yaw-axis.

IV. EXPERIMENTS

Our experimental evaluation pursues the following goals: a) In a comparison of our proposed metric global localization algorithm with related state-of-the-art techniques, we demonstrate that our approach not only outperforms existing feature-based algorithms, but that it is also computationally less expensive. b) In addition to that, we provide valuable insights of the place-recognition and orientation estimation performance by performing a separate in depth analysis of two core modules of our pipeline dedicated at deriving a compact place dependent, and a compact orientation dependent descriptor, respectively.

Before addressing these two evaluation foci in detail, a brief overview of the two dataset collections used and the respective sensor setups is provided.

A. Dataset Collections

We use the following two dataset collections for our experiments:

1) KITTI: The KITTI dataset collection contains recordings from several drivings through the urban areas of Karlsruhe [32]. The point clouds are recorded by a Velodyne 64 HDL sensor at 10 Hz, placed on the center of the car's roof. Ground truth poses are provided by a RTK GPS sensor.

2) NCLT: The University of Michigan North Campus Long-Term Vision and LIDAR Dataset [33] consists of 27 recordings collected by driving a Segway platform through the indoor and outdoor of the University campus over the course of 14 months. A Velodyne HDL-32 sensor provides point clouds at 10 Hz, and ground truth trajectories are provided by a globally optimized SLAM solution fusing RTK GPS with co-registered LiDAR point clouds.

B. Data Sampling for Training

Training triplet network structures requires sampling three-tuples of anchor, similar, and dissimilar pairs, as described in Section III-C. Two point clouds are considered similar, if their ground-truth poses are within $1.5m$. In the first training stage, dissimilar point clouds are sampled randomly from outside the $1.5m$ radius around the anchor sample. This is followed by a second training stage, where dissimilar point clouds are sampled from within a $2-5m$ radius around the anchor sample. This hard-negative mining strategy is able to boost the network performance by training with three-tuples that are harder to distinguish in the later stage of convergence. For the NCLT dataset collection, we train our model with data from a subarea of the campus using the 2012-01-08 and 2012-01-15 datasets. Validation has been done on 2012-12-01, while we use six different datasets (2012-01-22, 2012-02-04, 2012-03-25, 2012-03-31, 2012-10-28 and 2012-11-17) for our final evaluation. The campus subarea used in the six validation datasets is different from the area used for training. Furthermore, we have downsampled the data, such that for each query point cloud, there is exactly one true-positive map point cloud, and any two query point clouds in the same dataset are at least 3 meters apart. In case of the KITTI dataset collection, only Sequence 00 revisits the same places again, and can thus be used for proper localization evaluation. Sequences 03-08 are used for training, while Sequence 02 is used for validation. For the evaluation, point clouds from the first $170s$ of Sequence 00 are used to generate the map, i.e., to populate the KD-tree. The remaining point clouds are used for localization queries. This split of Sequence 00 prevents any self-localization, as the vehicle starts to revisit previously traversed areas after $170s$. Analogous to the NCLT datasets, the query point clouds are sampled to be at least $3m$ apart.

C. Baselines

We compare our metric global localization algorithm with two versions of LocNet [6]:

- **LocNet (base)**: feeds handcrafted rotation invariant histogram-based range images into a CNN. We have reimplemented LocNet with the network architecture as described in Yin et al. for the base model of LocNet.
- **LocNet++**: We retrained the original LocNet model following our training procedure, i.e., by using the triplet loss and hard negative mining.

In contrast to our work, LocNet is only able to provide a nearest place candidate in a map, but no metric pose estimate.

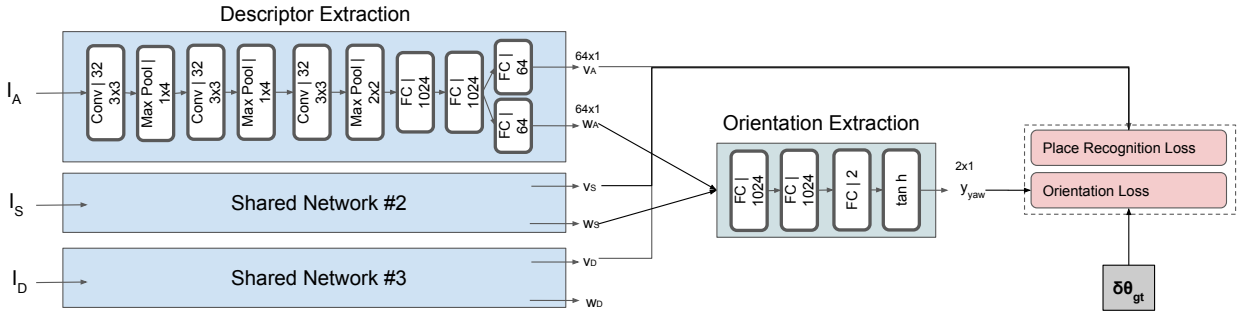

Fig. 3: Our proposed network architecture for descriptor extraction is composed of three convolutional and two fully connected layers. The projected 2D range images (I_A, I_S, I_D) representing an anchor, a similar and a dissimilar point cloud sample, are compressed into a descriptor of dimension 2 x 64 x 1 which can be used for global localization and orientation estimation. The three place recognition vectors v_A, v_S, v_D are used to compute the Place Recognition Loss L_{pr}, while the two orientation vectors from the point clouds from similar places, that is w_A and w_S, are fed into the Orientation Extraction network. The latter estimates a yaw angle discrepancy $\delta\theta$, which is used to compute the Orientation Loss and compared with the yaw angle discrepancy ground truth label $\delta\theta_{gt}$. We abbreviated all layers of our network, where FC represents a fully connected, and Conv represents a convolutional layer. Note that PreLU activation functions are used unless otherwise stated.

An orientation estimate can, however, be generated using local handcrafted features together with RANSAC:

- **FPFH + RANSAC** [34]: we generate for each point a local feature and use RANSAC to obtain the prior pose estimate from the inlier set.

Both (FPFH and RANSAC) are implemented using the PCL library [35], while LocNet's histogram generation is implemented using Matlab. Pose estimates generated by our metric global localization algorithm, and by LocNet in combination with FPFH and RANSAC, are further refined with point-to-plane ICP, yielding accurate 3 DoF pose estimates.

D. Metric Global Localization Performance

We evaluate the localization recall of OREOS with the recall attained by the two versions of LocNet combined with FPFH and RANSAC, for increasing discrepancies in the yaw angle between the query point cloud and the point cloud of the nearest place in the map. For this, the query point clouds are rotated along the yaw-axis in 10 deg steps. The localization of a query point cloud is considered successful, if the following two criteria are met: a) The nearest place candidate retrieved from the map lies within 1.5 m of the ground-truth query pose. b) After running ICP, the refined yaw angle θ is within 2.5 deg of the ground-truth yaw angle. Note that in this evaluation, $k = 1$, that is, only the first place candidate from the map is retrieved and processed.

On NCLT it can be observed that for small discrepancy in yaw angles, OREOS and LocNet++ perform similarly, achieving approximately 60% localization recall, while the original LocNet implementation performs significantly worse. For increasing yaw discrepancies, only OREOS is able to maintain a high localization recall, demonstrating its ability to both predict accurate nearest places in the map, and estimate the yaw angle discrepancies between the query and map point clouds. As expected, LocNet without an additional yaw estimation fails for increased yaw discrepancies, while using FPFH and RANSAC are able to achieve a localization recall between $20\% - 40\%$ for misaligned point clouds.

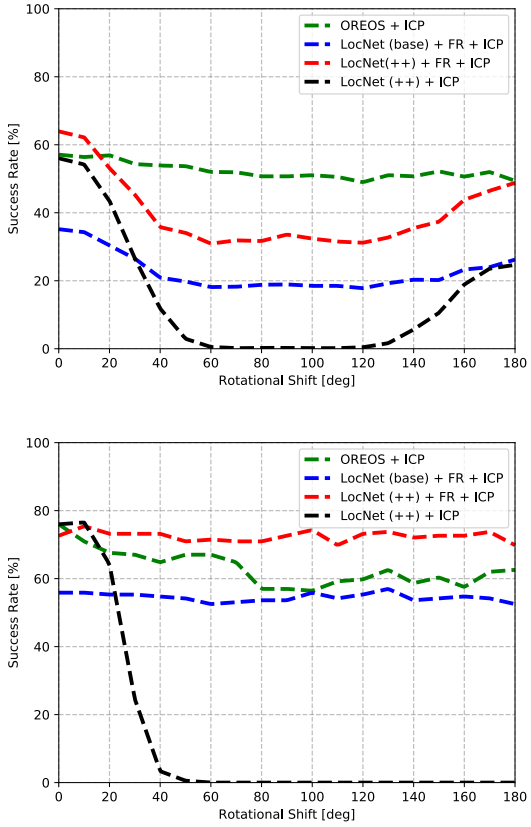

Fig. 4: Metric global localization performance of OREOS on the NCLT (top) and KITTI (bottom) datasets. We compare our approach to LocNet (base) and Locnet(++), whereas the latter uses Fast Point Feature Histograms (FPFH) and RANSAC (abbreviated with FR) for the initial orientation estimation. We vary the rotational shift of the query point cloud in 10 degree steps in order to evaluate the orientation estimation success rate.

Approach	Preprocessing [ms]	Feature Extraction [ms]	CNN [ms]	NN Loc [ms]	FCN [ms]	RANSAC [ms]	ICP [ms]	Total [ms]
FPFH	-	414	-	-	-	3149	27	3590
LocNet (base/++)	56.5	-	1.0	1.0	-	-	-	58.5
OREOS	12	-	2.37	1.0	1.0	-	25	41.37

Approach	Preprocessing [ms]	Feature Extraction [ms]	CNN [ms]	NN Loc [ms]	FCN [ms]	RANSAC [ms]	ICP [ms]	Total [ms]
FPFH	-	564	-	-	-	2124	24	2712
LocNet (base/++)	79.4	-	1.0	1.0	-	-	-	81
OREOS	19	-	2.89	1.0	1.0	-	15	39

TABLE I: Average computational execution times (NCLT (top) and KITTI (bottom)) in [ms].

Towards 180 deg there is an increase of the success rate of ICP for some of the methods. This is due to the fact that in some NCLT datasets, the campus is traversed in the opposite direction. Augmenting point clouds from these datasets by 180 deg thus results in the point clouds being already well-aligned with the map point cloud, without the need of a yaw discrepancy estimation. In addition to a decreased localization recall for large yaw angle discrepancies, the runtime of LocNet combined with FPHF and RANSAC is significantly higher than for OREOS, as can be seen in Table I. On the KITTI dataset, the localization recall of all methods is in general higher than in case of NCLT. This is due to the fact that the KITTI scenario is considerably simpler, with very similar driving trajectories, and without any significant environmental change. While OREOS still performs better than LocNet (base), in this case LocNet(++) takes the lead in overall performance. As the in depth-analysis in Section IV-E and Section IV-F will later reveal, this performance gain is mostly due to FPFH/RANSAC which almost reaches a recall of 100 $\%$. OREOS on the other hand is significantly better with the predicted orientation estimation as compared to FPFH and RANSAC, and computationally more efficient as depicted in Table I and Table II. All approaches are evaluated on a GTX 980 Ti and an i7-4810MQ CPU @ 2.80GHz. Preprocessing the 3D pointcloud to a 2D range image and LocNet's histograms are computed single threaded, while FPFH is implemented using PCL's multithreaded OPM version.

E. Place Recognition Analysis

In a practical application, it may be possible to test more than one place recognition candidate retrieved from the kd-tree. In this section, we thus analyze the performance of the OREOS place recognition module in comparison with LocNet, for increasing values of k. The respective localization recall results are shown in Figure 5. OREOS outperforms the LocNet base model, and attains similar performance as LocNet++ for higher values of k. However, for small values of k, the rotation invariant histogram representation of LocNet, together with a model trained using hard negative mining, appears to exhibit an edge over the our place recognition module learned directly from the 2D range images. As shown in Section IV-D, using the 2D range images does, however, has the advantage of allowing to also estimate a yaw angle discrepancy.

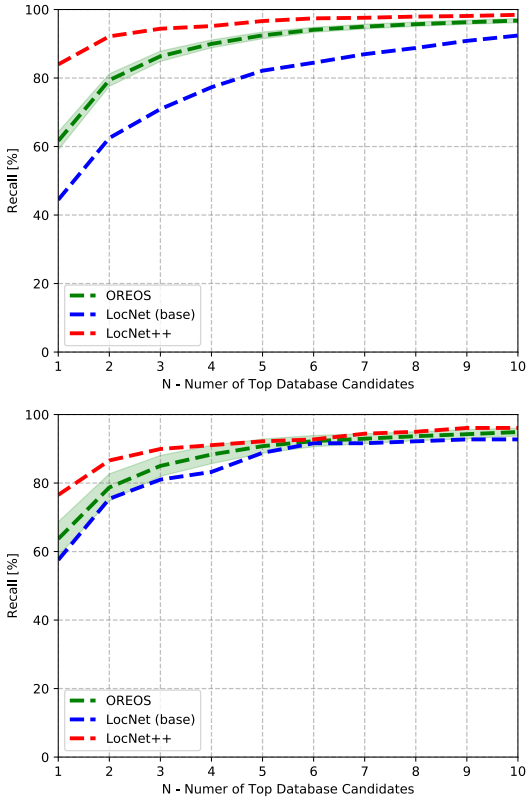

Fig. 5: Place recognition performance on NCLT (top) and KITTI (bottom) of OREOS, and the two variations of LocNet, for an increasing number of nearest place candidates retrieved from the map. For our approach, the standard deviation over augmented rotated point clouds is shown in shaded green.

F. Yaw Estimation Analysis

To investigate the accuracy of the OREOS yaw angle estimation, we analyze and compare the yaw angle discrepancy estimates of our Yaw Estimation network, with the estimates generated by FPFH in combination RANSAC. Using the ground-truth orientations of the point clouds, we can assess the estimation errors, and the respective mean and standard deviations are listed in Table II. Both OREOS and FPFH with RANSAC exhibit similar yaw discrepancy estimation

accuracy. However, OREOS per design attains 100% recall, while RANSAC is prone to fail to provide a yaw estimate in many cases.

Approach in NCLT	**Mean** [deg]	**Std** [deg]	**Recall** [%]
FPFH + RANSAC	9.47	26.65	58.0
OREOS	15.95	21.31	100.0

Approach in KITTI	**Mean** [deg]	**Std** [deg]	**Recall** [%]
FPFH + RANSAC	13.28	32.19	97.0
OREOS	12.67	15.23	100.0

TABLE II: Absolute orientation estimation errors without ICP (NCLT (top) and KITTI (bottom)) with mean and standard deviation in degrees, and recall in $\%$.

As seen in Table II our approach shows a better standard deviation in degree than FPFH + RANSAC while yielding a higher recall.

V. CONCLUSIONS

We have presented a data-driven descriptor that can be used to both retrieve near-by place candidates from a map, and estimate the yaw angle discrepancy between 3D LiDAR scans in challenging outdoor environments. A deep Neural Network architecture is employed to learn a mapping from a range image encoding of the 3D point cloud onto a feature vector representation, which effectively encodes place and orientation dependent cues. Using our learning approach consisting of a triplet loss approach, hard negative mining, we obtain a novel descriptor which resulting 3 DoF pose estimates set a new state-of-the-art in metric global localization for outdoor environments using only single 3D LiDAR scans. At the same time, our learned descriptor mapping function can be computed efficiently in real-time without discarding any useful information through handcrafted intermediate representations. An extensive analysis of the performance of our proposal in two different outdoor environments and sensor setups has revealed a high robustness on the orientation estimates and high place recognition recall.

REFERENCES

[1] S. Lowry, N. Sunderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *Trans. Rob.*, vol. 32, no. 1, pp. 1–19, Feb. 2016. [Online]. Available: https://doi.org/10.1109/TRO.2015.2496823

[2] C. McManus, P. T. Furgale, and T. D. Barfoot, "Towards appearance-based methods for lidar sensors," in *ICRA*. IEEE, 2011, pp. 1930–1935.

[3] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.

[4] M. Bosse and R. Zlot, "Place recognition using keypoint voting in large 3d lidar datasets," 05 2013.

[5] R. Dubé, A. Cramariuc, D. Dugas, J. I. Nieto, R. Siegwart, and C. Cadena, "Segmap: 3d segment mapping using data-driven descriptors," *CoRR*, vol. abs/1804.09557, 2018.

[6] H. Yin, Y. Wang, L. Tang, X. Ding, and R. Xiong, "LocNet: Global localization in 3D point clouds for mobile robots," *Arxiv*, 2017. [Online]. Available: http://arxiv.org/abs/1712.02165

[7] M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," *CoRR*, vol. abs/1804.03492, 2018. [Online]. Available: http://arxiv.org/abs/1804.03492

[8] K. P. Cop, P. V. Borges, and R. Dubé, "Delight: An efficient descriptor for global localisation using lidar intensities," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3653 – 3660, 2018 IEEE International Conference on Robotics and Automation (ICRA); Conference Location: Brisbane, Australia; Conference Date: May 21-25, 2018.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *ImageNet Classification with Deep Convolutional Neural Networks*, pp. 1097–1105, 2012.

[10] M. Bosse and R. Zlot, "Place recognition using keypoint voting in large 3D lidar datasets," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013, pp. 2677–2684.

[11] B. Steder, G. Grisetti, and W. Burgard, "Robust place recognition for 3D range data based on point features," *2010 IEEE International Conference on Robotics and Automation*, pp. 1400–1405, 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5509401

[12] F. Cao, Y. Zhuang, H. Zhang, and W. Wang, "Robust Place Recognition and Loop Closing in Laser-Based SLAM for UGVs in Urban Environments," 2018.

[13] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 2564–2571. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2011.6126544

[14] M. Magnusson, H. Andreasson, A. Nüchter, and A. J. Lilienthal, "Appearance-based loop detection from 3D laser data using the normal distributions transform," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, vol. 3, no. 2, 2009, pp. 23–28. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5152712

[15] R. Dubé, M. G. Gollub, H. Sommer, I. Gilitschenski, R. Siegwart, C. Cadena, and J. I. Nieto, "Incremental-segment-based localization in 3-d point clouds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1832–1839, 2018.

[16] R. Dubé, D. Dugas, E. Stumm, J. I. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based loop-closure for 3d point clouds," *CoRR*, vol. abs/1609.07720, 2016.

[17] T. Rohling, J. Mack, and D. Schulz, "A fast histogram-based similarity measure for detecting loop closures in 3-D LIDAR data," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, 2015, pp. 736–741.

[18] J. Do Monte Lima and V. Teichrieb, "An efficient global point cloud descriptor for object recognition and pose estimation," in *Proceedings - 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2016*, 2017.

[19] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 2010, pp. 2155–2162.

[20] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," 2016.

[21] A. Garcia-Garcia, F. Gomez-Donoso, J. Garcia-Rodriguez, S. Orts-Escolano, M. Cazorla, and J. Azorin-Lopez, "PointNet: A 3D Convolutional Neural Network for real-time object class recognition," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2016-Octob, 2016, pp. 1578–1584.

[22] 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 Letters (RA-L) (with ICRA)*, 2019, accepted. To appear.

[23] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," 10 2018, pp. 4802–4809.

[24] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217. [Online]. Available: http://ieeexplore.ieee.org/document/5152473/

[25] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[26] X. Han, J. S. Jin, J. Xie, M. Wang, and W. Jiang, "A comprehensive review of 3d point cloud descriptors," *CoRR*, vol. abs/1802.02297, 2018. [Online]. Available: http://arxiv.org/abs/1802.02297

[27] M. Velas, M. Spanel, M. Hradis, and A. Herout, "Cnn for imu assisted odometry estimation using velodyne lidar," in *Autonomous Robot Systems and Competitions (ICARSC), 2018 IEEE International Conference on*. IEEE, 2018, pp. 71–77.

[28] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *International Conference on Learning Representations (ICRL)*, pp. 1–14, 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[29] S. Appalaraju and V. Chaoji, "Image similarity using Deep CNN and Curriculum Learning," *Proceedings of the 2017 Grace Hopper India Annual Conference (GHCI)*, pp. 1–9, 2017. [Online]. Available: http://arxiv.org/abs/1709.08761

[30] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *SIMBAD*, 2015.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[32] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[33] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and lidar dataset," *International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2016.

[34] B. Li, "Vehicle Detection from 3D Lidar Using Fully Convolutional Network," *Robotics Science and Systems*, 2016.

[35] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011.