SegMap: 3D Segment Mapping using Data-Driven Descriptors

Renaud Dubé*, Andrei Cramariuc*, Daniel Dugas, Juan Nieto, Roland Siegwart, and Cesar Cadena
Autonomous Systems Lab, ETH, Zurich
Emails: {renaudube, andrei.cramariuc}@gmail.com and {dugasd, jnieto, rsiegwart, cesarc}@ethz.ch
*The authors contributed equally to this work.

***Abstract*—When performing localization and mapping, working at the level of structure can be advantageous in terms of robustness to environmental changes and differences in illumination. This paper presents *SegMap*: a *map representation* solution to the localization and mapping problem based on the extraction of segments in 3D point clouds. In addition to facilitating the computationally intensive task of processing 3D point clouds, working at the level of segments addresses the data compression requirements of real-time single- and multi-robot systems. While current methods extract descriptors for the single task of localization, *SegMap* leverages a data-driven descriptor in order to extract meaningful features that can also be used for reconstructing a dense 3D map of the environment and for extracting semantic information. This is particularly interesting for navigation tasks and for providing visual feedback to end-users such as robot operators, for example in search and rescue scenarios. These capabilities are demonstrated in multiple urban driving and search and rescue experiments. Our method leads to an increase of area under the ROC curve of 28.3% over current state of the art using eigenvalue based features. We also obtain very similar reconstruction capabilities to a model specifically trained for this task. The *SegMap* implementation will be made available open-source along with easy to run demonstrations at www.github.com/ethz-asl/segmap. A video demonstration is available at https://youtu.be/CMk4w4eRobg.**

I. Introduction

Being a critical competency for mobile robotics, localization and mapping has been a well-studied topic over the last couple of decades [3]. In recent years, the importance of Simultaneous Localization and Mapping (SLAM) has proven especially relevant in the context of applications with social impact, such as autonomous driving and disaster response. Although cameras and LiDARs are often used in conjunction due to their complementary nature [19], the SLAM problem for 3D LiDAR point clouds still poses open challenges. Moreover, LiDAR based systems rely on structure which can be more consistent than visual appearance across seasons and daylight changes. Despite recent developments, a number of important capabilities are still lacking in many existing 3D LiDAR SLAM frameworks. Perhaps most notably, this includes the absence of global data associations (place recognitions) from almost all 3D LiDAR based SLAM implementations, while contrastingly being a well-studied problem in visual SLAM [17].

This paper presents *SegMap*: a unified approach for *map representation* in the localization and mapping problem for 3D LiDAR point clouds. The *SegMap* approach is formed on the basis of partitioning point clouds into sets of descriptive segments [6], as illustrated in Figure 1. Segments are obtained using clustering techniques which are able to repeatably form similar partitions of the point cloud. The resulting segments provide the means for compact, yet discriminative features to represent the environment efficiently. Global data associations are identified by segment descriptor retrieval, made possible by the repeatable and descriptive nature of segment based features. The use of segment based features facilitates low computational, memory and bandwidth requirements, and therefore makes the approach appropriate for real-time use in both multi-robot and long-term applications. Moreover, as segments typically represent meaningful and distinct elements that make up the environment, a scene can be effectively summarized by a handful of compact feature descriptors.

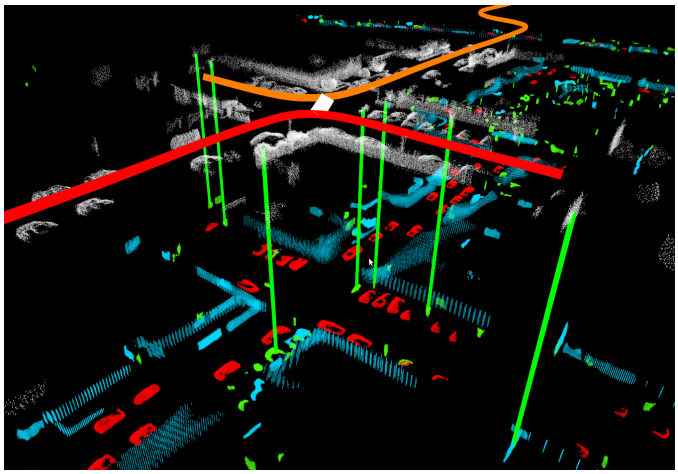

Fig. 1: An illustration of the *SegMap* approach. The red and orange lines represent two robots driving simultaneously in opposite directions through an intersection. In white we show the local segments extracted from the robots' vicinity and characterized using our compact data-driven descriptor. Correspondences are then made with the target segments, resulting in successful localizations depicted with green vertical lines. A reconstruction of the target segments is illustrated below, where colors represent semantic information (cars in red, buildings in light blue, and others in green), all possible by leveraging the same compact representation. We take advantage of the semantic information by performing localization only against static objects, adding robustness against dynamic changes.

Previous work on segment based localization considered hand-crafted features and provided a sparse representation[6]. These features lack the ability to generalize to different environments and only offer limited insights into the underlying

3D structure. In this work, we overcome these shortcomings by introducing a novel data-driven segment descriptor which can offer high retrieval performances, even under variations in point of view. As depicted in Figure 1, these descriptors can be decoded in order to generate 3D reconstructions. These can be used by robots for navigating around obstacles and displayed to remote operators for improved situation awareness. Moreover, we show that semantic information can be extracted by performing classification in the descriptor space. This information can for example lead to increased robustness to changes in the environment. To the best of our knowledge, this is the first work on robot localization proposing to reuse the extracted features for reconstructing environments in three dimensions and for extracting semantic information. This reconstruction is, in our opinion, a very interesting capability for real-world, large-scale applications with limited memory and communication bandwidth.

To summarize, this paper presents the following contributions:

- A novel data-driven 3D segment descriptor achieving increased localization performance.
- A technique for reconstructing the environment based on the same compact features used for localization.
- An extensive evaluation of the *SegMap* approach using real-world, multi-robot automotive and disaster scenario datasets.

The remainder of the paper is structured as follows: Section II provides an overview of the related work in the fields of localization and machine learning based descriptors for 3D point clouds. The *SegMap* approach and our novel descriptor enabling environment reconstruction are respectively detailed in Section III and Section IV. The method is evaluated in Section V, and Section VI finally concludes with a short discussion.

II. RELATED WORK

An overview of the related work on localization in 3D point clouds was presented in [3] and [6]. In this section, we review learning based techniques with applications to 3D points clouds.

In recent years, Convolutional Neural Networks (CNNs) have become the state-of-the-art method for generating learning based descriptors, due to their ability to find complex patterns in data [15]. When working with 3D point clouds, methods based on CNNs achieve impressive performances in applications such as object detection [10, 16, 18, 22, 23, 29, 32], semantic segmentation [16, 22, 23, 25], and 3D object generation [31].

Recently, a handful of works proposing the use of CNNs for localization in 3D point clouds have started to appear [9, 33]. First, Zeng et al. [33] propose to extract data-driven 3D keypoint descriptors (3DMatch) which are robust to changes in point of view. Although impressive retrieval performances are demonstrated using an RGB–D sensor in indoor environments, it is not clear whether this method is applicable in real-time in large-scale outdoor environments. Elbaz et al. [9] propose to describe local subsets of points using a deep neural network autoencoder. The authors state that the implementation has not been optimized for real-time operation and no timings have been provided. Contrastingly, our work here presents a data-driven segment based localization method that can operate in real-time and that allows map reconstruction and semantic extraction capabilities.

To achieve this reconstruction capability, the architecture of our descriptor was inspired by autoencoders in which an encoder network compresses the input to a small dimensional representation, and a decoder network attempts to decompress the representation back into the original input. The compressed representation can be used as a descriptor for performing 3D object classification [1]. Brock et al. [1] also present successful results using variational autoencoders for reconstructing voxelized 3D data. Different configurations of encoding and decoding networks have also been proposed for reconstructing and completing 3D shapes and environments [5, 13, 26].

While autoencoders present an interesting opportunity of simultaneously accomplishing both compression and feature extraction tasks, optimal performance at both is not guaranteed. As will be shown in Section V-E, encoding and feature extraction can have conflicting goals when robustness to changes in point of view is desired. In this work, we combine the advantages of the encoding-decoding architecture of autoencoders with a technique proposed by Parkhi et al. [21]. The authors address the face recognition problem by first training a CNN to classify people in a training set and afterwards use the second to last layer as a descriptor for new faces. This classification based method is an alternative to training networks using contrastive loss [2] or triplet loss [27]. We use the resulting segment descriptors in the context of SLAM to achieve better performance, as well as significantly compressed maps that can easily be stored, shared, and reconstructed.

III. THE *SegMap* APPROACH

This section presents our *SegMap* approach to localization and mapping in 3D point clouds. It is composed of five core modules: segment extraction, description, localization, map reconstruction, and semantics extraction. These modules are detailed in this section and together allow single and multi-robot systems to create a powerful unified representation which can conveniently be communicated.

Segmentation The stream of point clouds generated by a 3D sensor is first accumulated in a dynamic voxel grid. A circular section of radius R around the robot is efficiently segmented with an incremental region growing algorithm [8]. This results in a handful of local segments, which are each associated to a set of past observations i.e. $S_i = \{s_1, s_2, \ldots, s_n\}$. Each observation $s_j \in S_i$ is a 3D point cloud representing a snapshot of the segment as points are added to it.

Description Compact features are then extracted from these 3D segment point clouds using the data-driven descriptor presented in Section IV. A global segment map is created online by accumulating these segment descriptors and the

segment centroids. In order for the global map to most accurately represent the latest state of the world, we only keep the descriptor associated with the last and most complete observation.

Localization In the next step, candidate correspondences are identified between global and local segments using k-Nearest Neighbors (k-NN) in feature space. Localization is finally performed by filtering these candidate correspondences using an incremental geometric verification strategy based on the segment centroids [8]. When a geometrically consistent set of correspondence is identified, a 6 Degrees of Freedom (DoF) transformation between the local and global maps is estimated. This transformation is fed to an incremental pose-graph SLAM solver which in turn estimates, in real-time, the trajectories of all robots. More details about this pose-graph approach can be found in our previous work [7].

Reconstruction & Semantics The compressed representation can at any time be used to reconstruct a map and to extract semantic information. Thanks to the compactness of the *SegMap* descriptor which can conveniently be transmitted over wireless networks with limited bandwith, any agent in the network can reconstruct and leverage this 3D information. On the other hand, the semantic information can for example be used to discern between static and dynamic objects which can improve the robustness of the localization.

IV. The *SegMap* Descriptor

In this section we present our main contribution: a data-driven descriptor for 3D segment point clouds which allows for localization, map reconstruction and semantic extraction. The descriptor extractor's architecture and the processing steps for feeding the point clouds to the network's input are first introduced. We then describe our technique for training this descriptor to accomplish both tasks of segment retrieval and map reconstruction. We finally show how the descriptor can further be used to extract semantic information from the point cloud.

A. Descriptor extractor architecture

The architecture of the descriptor extractor is presented in Fig. 2. Its input is a 3D binary voxel grid of fixed dimension $32 \times 32 \times 16$ which was determined empirically to offer a good balance between the descriptiveness and size of the network. The description part of the CNN is composed of three 3D convolutional layers with max pool layers placed in between and two fully connected layers. Unless otherwise specified,Rectified Linear Unit (ReLU) activation functions are used for all layers. The original scale of the input segment is passed as an additional parameter to the first fully connected layer to increase robustness to voxelization at different aspect ratios. The descriptor is obtained by taking the activations of the extractor's last fully connected layer. This architecture was found by grid searching through different depths and sizes for the layers and filters.

B. Segment alignment and scaling

A pre-processing stage is required in order to feed the 3D segment point clouds for description. First, an alignment step is applied such that segments extracted from the same objects are similarly presented to the descriptor network. This is performed with the assumption that the z-axis is roughly aligned with gravity and by applying a 2D Principal Components Analysis (PCA) of all points located within a segment. The segment is then rotated so that the x-axis of its frame of reference aligns with the eigenvector corresponding to the largest eigenvalue. We choose to solve the ambiguity in direction by rotating the segment so that the lower half section along the y-axis of its frame of reference contains the highest number of points. From the multiple alignment strategies we evaluated, the presented strategy worked best.

The network's input voxel grid is applied to the segment so that its center corresponds to the centroid of the aligned segment. By default the voxels have minimum side lengths of $0.1\,\mathrm{m}$. These can individually be increased to exactly fit segments having one or more larger dimension than the grid. Whereas maintaining the aspect ratio while scaling can potentially offer better retrieval performance, this individual scaling with a minimum side length better avoids large errors caused by aliasing. We also found that individually scaling the dimensions offers the best reconstruction performance, with only a minimal impact on the retrieval performance when the original scale of the segments is passed as an additional parameter to the network.

C. Training the SegMap descriptor

In order to achieve both a high retrieval performance and reconstruction capabilities, we propose a customized learning technique. The two desired objectives are imposed on the network by the *softmax cross entropy loss* L_c for retrieval and the reconstruction loss L_r. We propose to simultaneously apply both losses to the descriptor and to this end define a combined loss function L which merges the contributions of both objectives:

$$L = L_c + \alpha L_r \tag{1}$$

where the parameter α weighs the relative importance of the two losses. The value $\alpha = 200$ was empirically found to not significantly impact the performance of the combined network, as opposed to training separately with either of the losses. Weights are initialized based on Xavier's initialization method [12] and trained using the Adaptive Moment Estimation (ADAM) optimizer [20] with a learning rate of 10^{-4}. In comparison to Stochastic Gradient Descent (SGD), ADAM maintains separate learning rates for each network parameter, which facilitates training the network with two separate objectives simultaneously. Regularization is achieved using dropout [24] and batch normalization [14].

Classification loss L_c For training the descriptor to achieve better retrieval performance, we use a learning technique similar to the *N-ways classification problem* proposed by Parkhi

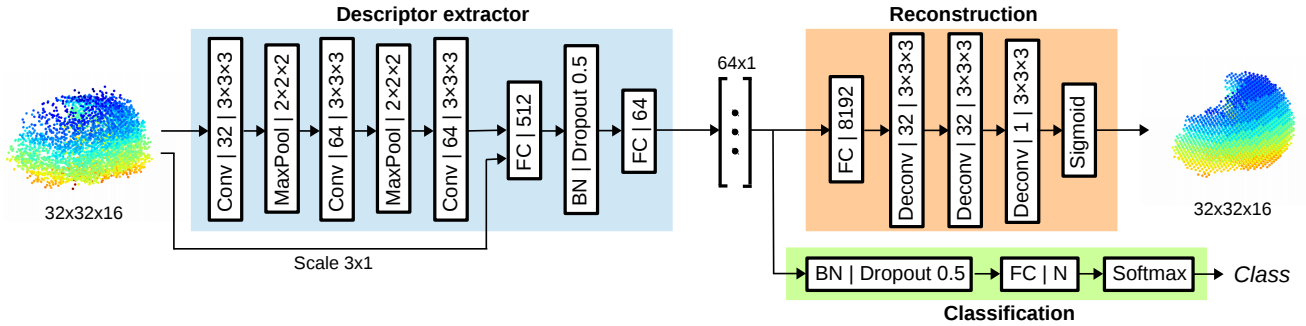

Fig. 2: The descriptor extractor is composed of three convolutional and two fully connected layers. The 3D segments are compressed in a representation of dimension 64×1 which can be used for localization, map reconstruction and semantic extraction. Right of the descriptor we illustrate the classification and reconstruction layers which are used for training. In the diagram the convolutional (Conv), deconvolutional (Deconv), fully connected (FC) and batch normalization (BN) layers are abbreviated respectively. As parameters the Conv and Deconv layers have the number of filters and their sizes, FC layers have the number of nodes, max pool layers have the size of the pooling operation, and dropout layers have the ratio of values to drop. Unless otherwise specified, ReLU activation functions are used for all layers.

et al. [21].[1] Specifically, we organize the training data into N classes where each class contains all observations of a segment or of multiple segments that belong to the same object or environment part. Note that these classes are solely used for training the descriptor and are not related to the semantics presented in Section IV-D. As seen in Fig 2, we then append a classification layer to the descriptor and teach the network to associate a score to each of the N predictors for each segment sample. These scores are compared to the true class labels using *softmax cross entropy loss*:

$$L_c = -\sum_{i=1}^{N} y_i \log \frac{e^{l_i}}{\sum_{k=1}^{N} e^{l_k}} \tag{2}$$

where y is the one hot encoded vector of the true class labels and l is the layer output.

Given a large number of classes and a small descriptor dimensionality, the network is forced to learn descriptors that better generalize and prevent overfitting to specific segment samples. Note that when deploying the system in a new environment the classification layer is removed, as the classes are no longer relevant. The activations of the previous fully connected layer are then used as a descriptor for segment retrieval through k-NN.

Reconstruction loss L_r As depicted in Fig. 2, map reconstruction is achieved by appending a decoder network and training it simultaneously with the descriptor extractor and classification layer. This decoder is composed of one fully connected and three deconvolutional layers with a final sigmoid output. Note that no weights are shared between the descriptor and the decoder networks. Furthermore, only the descriptor extraction needs to be run in real-time on the robotic platforms, whereas the decoding part can be executed any time a reconstruction is desired.

As proposed by Brock et al. [1], we use a specialized form of the *binary cross entropy loss*, which we denote by L_r:

$$L_r = -\sum_{x,y,z} \gamma t_{xyz} \log(o_{xyz}) + (1-\gamma)(1-t_{xyz}) \log(1-o_{xyz}) \tag{3}$$

where t and o respectively represent the target segment and the network's output and γ is a hyperparameter which weighs the relative importance of false positives and false negatives. This parameter addresses the fact that only a minority of voxels are activated in the voxel grid. In our experiments, the voxel grids used for training were on average only 3% occupied and we found $\gamma = 0.9$ to yield good results.

[1]Note that in our previous work three training techniques were evaluated for achieving better segment descriptor retrieval performances [4].

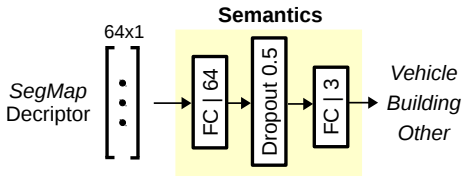

Fig. 3: A simple fully connected network that can be appended to the *SegMap* descriptor (depicted in Fig. 2) in order to extract semantic information. In our experiments, we train this network to distinguish between vehicles, buildings, and other objects.

D. Knowledge transfer for semantic extraction

As can be observed from Fig. 1, segments extracted by the *SegMap* approach for localization and map reconstruction often represent objects or parts of objects. It is therefore possible to assign semantic labels to these segments and use this information to improve the performance of the localization process. As depicted in Fig. 3, we transfer the knowledge embedded in our compact descriptor by training a semantic extraction network on top of it. This last network is trained with labelled data using the *softmax cross entropy loss* and by freezing the weights of the descriptor network.

In this work, we choose to train this network to distinguish between three different semantic classes: *vehicles*, *buildings*, and *others*. Section V-H shows that this information can be used to increase the robustness of the localization algorithm

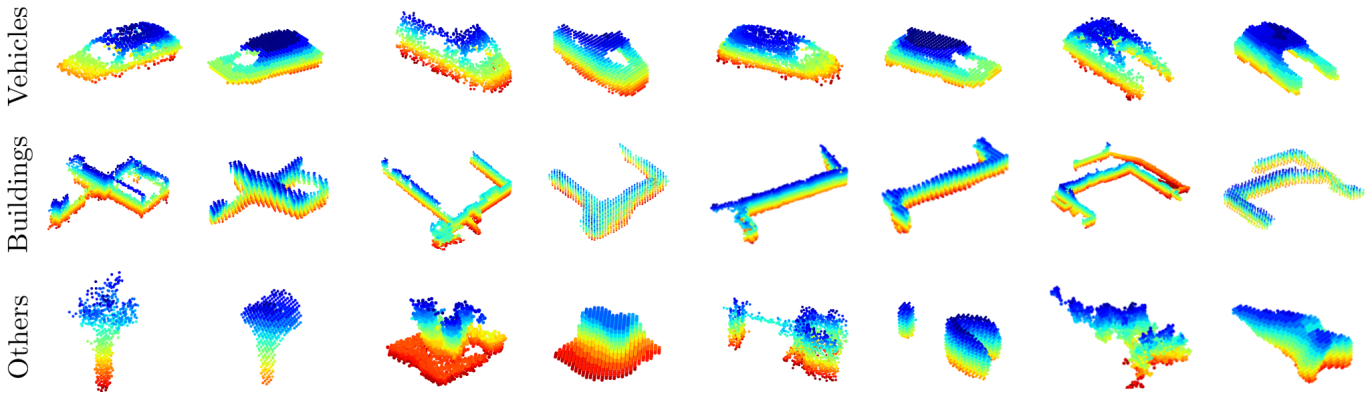

Fig. 4: An illustration of the *SegMap* reconstruction capabilities. The segments are extracted from sequence 00 of the KITTI dataset and represent, from top to bottom respectively, vehicles, buildings, and other objects. For each segment pair, the reconstruction is shown at the right of the original. The network manages to accurately reconstruct the segments despite the high compression to only 64 values. Note that the voxelization effect is more visible on buildings as larger segments necessitate larger voxels to keep the input dimension fixed.

to changes in the environment and to yield smaller map sizes. This is achieved by rejecting segments associated with potentially dynamic objects, such as vehicles, from the list of segment candidates.

V. EXPERIMENTS

This section presents the experimental validation of our approach. We first present a procedure for generating training data and detail the performances of the *SegMap* descriptor for localization, reconstruction and semantics extraction. We finally demonstrate the performance of the *SegMap* approach in two large scale experiments.

A. Experiment setup and implementation

All experiments were performed on a system equipped with 32GB of RAM, an Intel i7-6700K processor, and a Nvidia GeForce GTX 980 Ti Graphics Processing Unit (GPU). The models were developed using the TensorFlow[2] python interface whereas the C++ interface is used for computing the forward pass during real-time execution. The library *libnabo*[3] is used for descriptor retrieval with fast k-NN search in low dimensional space. The incremental optimization back-end is based on the iSAM2 implementation of the GTSAM library[4]. Finally, the system has a full ROS interface with integration to the TF tree for publishing the estimated robot poses.

B. Baselines

In the following experiments, our *SegMap* descriptor is compared with eigenvalue based point cloud features [28] and with a CNN trained specifically for compressing and reconstructing segment point clouds. This purely autoencoder model has the exact same architecture presented in Fig. 2. The single difference is that it is trained solely for reconstructing segment point clouds, i.e. by using only the reconstruction loss L_r. We will refer to these two baselines as Eigen for the eigenvalue based features and AE for the autoencoder model. For reference, previous work proposed to describe 3D segments using the ensemble of shape histograms [6, 30]. However, this descriptor was not included in our evaluation as its high dimensionality is not well suited to our goals of map compression and efficient k-NN retrieval in large maps.

[2]https://www.tensorflow.org/
[3]https://github.com/ethz-asl/libnabo
[4]https://research.cc.gatech.edu/borg/gtsam

C. Training data

The *SegMap* descriptor is trained using real-world data from the KITTI odometry dataset [11]. Sequences 05 and 06 are used for generating training data whereas sequence 00 is solely used for evaluating the descriptor performances. For each sequence, segments are extracted using an incremental Euclidean distance based region growing technique [8]. This training data is filtered by removing segments with too few observations, or training classes (as described in Section IV-C) with too few samples. In this manner, 3300, 1750, and 810 segments are respectively generated from sequences 00, 05, and 06 with an average of 12 observations per segment over the whole dataset.

1) Data augmentation: To further increase robustness and make the descriptor extraction process less sensitive to alignment and point of view, the dataset is augmented by using multiple copies of the same segment rotated at different angles. Also, in order to simulate the effect of occlusion we generate artificial copies of each segment by removing all points which fall on one side of a randomly generated slicing plane. In this process, we ensure that not more than 50% of the points are removed. Note that these two data augmentation steps are performed prior to voxelization.

2) Ground-truth generation: In the following step, we use GPS information in order to identify ground truth correspondences between segments extracted in areas where the vehicle performed multiple visits. For each possible pair of segments, we check whether the centroids of their last observations lie within a maximum distance of $3.0\ \mathrm{m}$. When this applies, we compute the 3D convex hull of each segment observation s_1 and s_2 and create a correspondence when the following

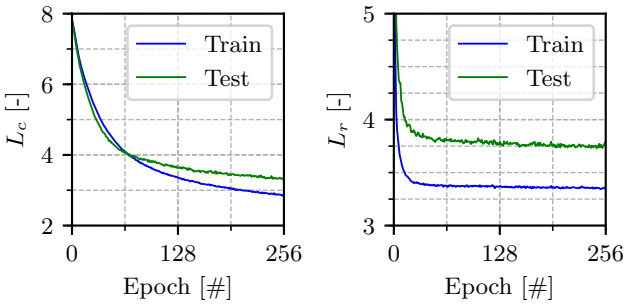

Fig. 5: The classification loss L_c (left) and the reconstruction loss L_r (right) when training the descriptor extractor along with the reconstruction and classification networks. The depicted reconstruction loss has already been scaled by α.

condition, inspired from the Jaccard index, holds:

$$\frac{\text{Volume}(\text{Conv}(s_1) \cap \text{Conv}(s_2))}{\text{Volume}(\text{Conv}(s_1) \cup \text{Conv}(s_2))} \geq p \tag{4}$$

In our experiments we found $p = 0.3$ to generate a sufficient number of correspondences while preventing false labelling. This procedure is performed on each drive, respectively generating 150, 260, and 320 ground truth correspondences in sequences 00, 05, and 06. These correspondences are used during training by merging segments into classes as presented in Section IV-C. We use two-thirds of the correspondences for augmenting the training data and one-third for creating validation samples. Finally, the ground-truth correspondences extracted from sequence 00 are used in Section V-E for evaluating the retrieval performances.

D. Training the models

The descriptor extractor and the decoding part of the reconstruction network are trained using all segments extracted from drive 05 and 06. Training lasts three to four hours on a Nvidia GeForce GTX 980 GPU and produces the classification and scaled reconstruction losses depicted in Fig. 5. The total loss of the model is the sum of the two losses as describe in Section IV-C. We note that for classification the validation loss follows the training loss before converging towards a corresponding accuracy of 41% and 43% respectively. In other words, 41% of the validation samples were correctly assigned to one of the $N = 2500$ classes. This accuracy is expected given the large quantity of classes and the challenging task of discerning between multiple training samples with similar semantic meaning but few distinctive features, e.g. flat walls. The reconstruction loss converges more rapidly at the start, after which it continues slowly decreasing for the remainder of the training process.

E. Retrieval performance

The retrieval performances of the *SegMap*, eigenvalue based, and autoencoder descriptors are depicted in Fig 6. The ROC curves are obtained by generating 90k labelled pairs of segment descriptors from sequence 00 of the KITTI odometry dataset [11]. For each ground-truth correspondence, a positive sample is created for each possible segment observation pair.

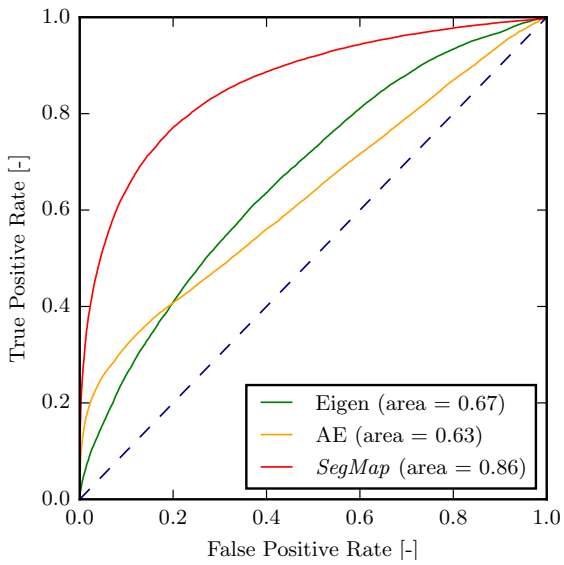

Fig. 6: ROC curves for the descriptors considered in this work. This evaluation is performed using ground-truth correspondences extracted from sequence 00 of the KITTI odometry dataset [11].

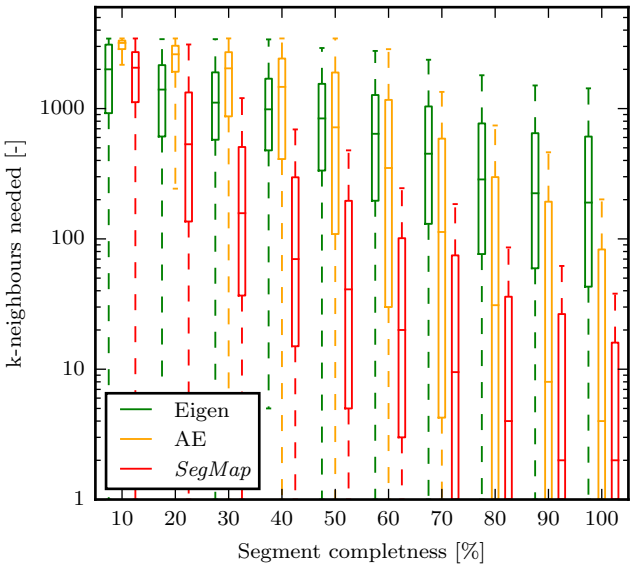

Fig. 7: This figure presents how *quickly* descriptors extracted from incrementally grown segments contain relevant information that can be used for localization. The x-axis represents the growing status of a segment until all its measurements have been accumulated (here termed *complete*). The log-scaled y-axis represents how many neighbours in the target map need to be considered in order to include the correct target segment (the lower the better). The *SegMap* descriptor offers one order of magnitude better retrieval performance for over 40% of the growing process.

Negative samples are generated by randomly sampling segment pairs whose centroids are further than 20 m apart. We choose to adopt a 1:1 ratio between the number of positive and negative samples. The ROC curves are finally obtained by varying the threshold applied on the l^2 distance between the two segment descriptors.

As introduced in Section III, correspondences are made

between segments from the local and global maps by using k-NN retrieval in feature space. In order to avoid false localizations, one aims to reduce the number k of neighbours that need to be considered. Therefore, as a segment grows with time, it is critical that its descriptor converges as quickly as possible towards the descriptor of the corresponding segment in the target map, which in our case is extracted from the last and most complete observation. This behaviour is evaluated in Fig. 7 which relates the number of neighbours which need to be considered to find the correct association as a function of segment completeness. We note that the *SegMap* descriptor offers the best retrieval performance at every stage of the growing process. In practice, this increase in performance makes an important difference, allowing us to close challenging loops such as the one presented in Fig. 1. Interestingly, the autoencoder has the worst performance at the early growing stages whereas good performance is observed at later stages. This is in accordance with the capacity of autoencoders to precisely describe the geometry of a segment, without explicitly aiming at gaining robustness to changes in point of view.

F. Reconstruction performance

In addition to offering high retrieval performances, the *SegMap* descriptor allows us to reconstruct 3D maps using the decoding CNN described in Section IV-C. Some examples of the resulting reconstructions are illustrated in Fig 4, for various objects captured during sequence 00 of the KITTI odometry dataset. Experiments done at a larger scale are presented in Fig. 9 where buildings of a powerplant and a foundry are reconstructed by fusing data from multiple sensors. Overall, the reconstructions are well recognizable despite the high compression to a low descriptor dimensionality. We note that the quantization error resulting from the voxelization step mostly affects larger segments as they have to be downscaled to fit into the voxel grid.

Since most segments only sparsely model real-world surfaces, they occupy on average only 3% of the voxel grid. To obtain a visually relevant comparison metric, we calculate for both the original segment and its reconstruction the ratio of points having a corresponding point in the other segment, within a distance of one voxel. The tolerance of one voxel means that the shape of the original segment must be preserved while not focusing on reconstructing each individual point. Results calculated for different descriptor sizes are presented in Table I, in comparison with the purely reconstruction focused baseline detailed in Sec. V-B. The *SegMap* descriptor with a size of 64 has on average 91% correspondences between the points in the original and reconstructed segments. In terms of reconstruction performance, we note that our descriptor is only slightly outperformed by the AE baseline. Contrastingly, the significantly higher retrieval performances of the *SegMap* descriptor makes it a clear all-rounder choice for achieving both localization and map reconstruction.

TABLE I: Average ratio of corresponding points within one voxel distance between original and reconstructed segments. Statistics are detailed for *SegMap* and the AE baseline using different descriptor sizes.

Descriptor size	AE	*SegMap*
16	0.87	0.86
32	0.91	0.89
64	0.93	0.91
128	0.94	0.92

G. Semantic extraction performance

For training the semantic extractor network (Fig. 3), we manually labelled the last observation of all 1750 segments extracted from KITTI sequence 05. The labels are then propagated to each observation of a segment for a total of 20k labelled segment observations. We use 70% of the samples for training the network and 30% for validation. Given the low complexity of the semantic extraction network and the small amount of labelled samples, training takes only a few minutes. We achieve an accuracy of 89% and 85% on the training and validation data respectively. Note that our goal is not to improve over other semantic extraction methods [16, 22], but rather to illustrate that our compressed representation can additionally be used for gaining robustness to dynamic changes and for reducing the map size (Section V-H1).

H. Large scale experiments

We evaluate the *SegMap* approach on three large-scale multi-robot experiments: one in urban-driving environment and two in search and rescue scenarios. In order to realize these experiments on one single machine, the approach is implemented with multiple threads, simulating a centralized system. One thread per robot is used for accumulating the 3D measurements, extracting segments, and performing the descriptor extraction. These descriptors are transmitted to a separate thread which localizes the robots, through descriptor retrieval and geometric verification, and runs the pose-graph optimization. In all experiments, sufficient global associations need to be made, in real-time, for linking the trajectories and merging the maps. Moreover, in such a centralized setup it can be crucial to limit the data to transmit over the wireless network with potentially limited bandwidth.

1) Multi-robot SLAM in urban scenario: In order to simulate a multi-robot setup, we split sequence 00 of the KITTI odometry dataset into five sequences which are simultaneously played back on a single computer for a duration of 114 seconds. In this experiment, we consider 40 neighbours when performing segment retrieval and require a minimum of 7 correspondences which are altogether geometrically consistent to output a localization. These parameters were chosen empirically using the information presented in Fig. 6 and 7 as a reference. Additionally, the semantic information extracted from the *SegMap* descriptors is used for rejecting segments classified as *vehicles* from the retrieval process.

With this setup, 113 global associations were discovered,

TABLE II: Statistics resulting from the three experiments.

Statistic	KITTI	Powerplant	Foundry
Duration (s)	114	850	1086
Number of robots	5	3	2
Number of segmented local cloud	557	784	689
Average number of segments per cloud	42.9	36.9	48.0
Bandwidth for transmitting local clouds (kB/s)	4814.7	1312.7	756.8
Bandwidth for transmitting segments (kB/s)	2626.6	226.2	186.6
Bandwidth for transmitting descriptors (kB/s)	60.4	9.8	8.8
Final map size with the *SegMap* descriptor (kB)	386.2	172.5	123.0
Number of successful localizations	113	32	93

allowing to link all the robot trajectories and to create a common representation. Localization and map reconstruction was performed at an average frequency of 10.5 Hz and segment description was responsible for 30% of this computing share with an average duration of 28.4 ms per local cloud (1.6 ms per segment). A section of the target map which has been reconstructed from the descriptors is depicted in Fig. 1.

Table II presents the results of this experiment. The required bandwidth is estimated by considering that each point is defined by three floats and assuming a typical size of 4 bytes per float. Note that, six floats and two unsigned integer are additionally required to link each descriptor to the trajectories. We also only treat the *useful data* and do not consider the data transfer overhead. The final map of the KITTI sequence 00 contains 1341 segments out of which 284 were classified as vehicles. A map composed of all the segments' point clouds would be 16.8 MB whereas the full map can be represented with only 386.2 kB using our descriptor. This compression ratio of 43.5x can be increased to 55.2x if one decides to remove vehicles from the map. This suggests that our approach could be used for mapping much larger environments.

2) Multi-robot SLAM in disaster environments: For the two following experiments, we use data collected by Unmanned Ground Vehicles (UGVs) equipped with multiple motor encoders, an Xsens MTI-G Inertial Measurement Unit (IMU) and a rotating 2D SICK LMS-151 LiDAR. First, three UGVs were deployed at the decommissioned Gustav Knepper powerplant: a large two-floors utility building measuring 100 m long by 25 m wide. The second mission took place at the Phoenix-West foundry in a semi-open building made of steel. A section measuring 100 m by 40 m was mapped using two UGVs. The buildings are shown in Fig 8.

For these two experiments, we used an incremental smoothness based region growing algorithm which extracts plane-like segments [8]. The number of nearest neighbours considered for segment retrieval is set to $k = 25$ and a localization occurs when at least 6 matches are geometrically consistent. The resulting *SegMap* reconstructions are shown in Fig. 9 and detailed statistics are presented in Table II. Although these planar segments have a very different nature than the ones used for training the descriptor extractor, multiple localizations have been made in real-time so that consistent maps could be reconstructed in both experiments.

Fig. 8: Buildings of the Gustav Knepper powerplant (left) and the Phoenix-West foundry (right).

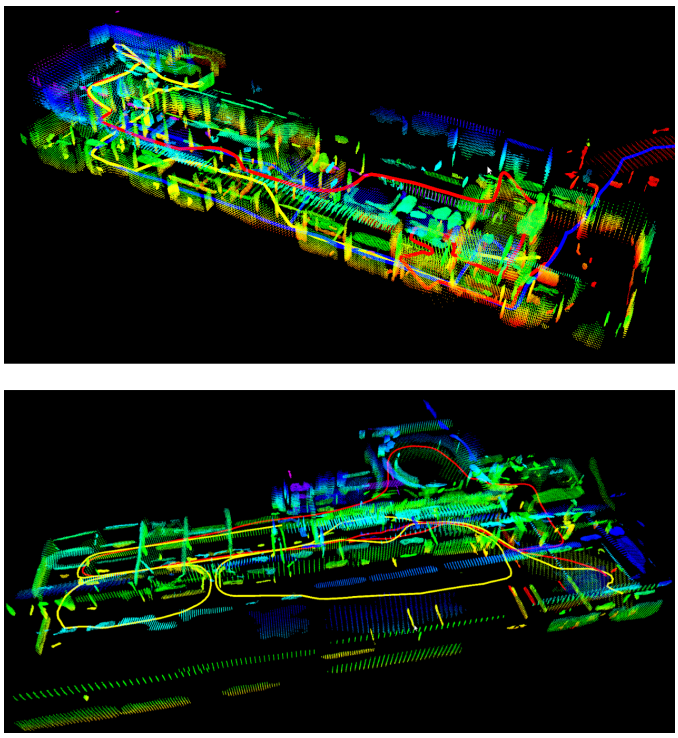

Fig. 9: This figure illustrates a reconstruction of buildings of the Gustav Knepper powerplant (top) and Phoenix-West foundry (bottom). The point clouds are colored by height and the estimated robot trajectories are depicted with colored lines.

VI. CONCLUSION

This paper presented *SegMap*: a segment based approach for *map representation* in localization and mapping with 3D sensors. In essence, the robot's surroundings are decomposed into a set of segments, and each segment is represented by a distinctive, low dimensional learning based descriptor. Data associations are identified by segment descriptor retrieval and matching, made possible by the repeatable and descriptive nature of segment based features. The descriptive power of *SegMap* outperforms hand-crafted features as well as the evaluated autoencoder alternative with access to the same training data.

In addition to enabling global localization, the *SegMap* descriptor allows us to reconstruct a map of the environment and to extract semantic information. The ability to reconstruct the environment while achieving a high compression rate is one of the main features of *SegMap*. This feature allows

performing SLAM with 3D LiDARs at a large scale requiring low communication bandwidth between the robots and a central computer. These capabilities have been demonstrated through experiments with real-world data in urban driving and search and rescue scenarios. The reconstructed maps could allow performing navigation tasks such as, for instance, multi-robot global path planning or increasing situational awareness.

In future work, we would like to extend the *SegMap* approach to different sensor modalities and different point cloud segmentation algorithms. Furthermore, whereas the present work performs segment description in a discrete manner, it would be interesting to investigate incremental updates of learning based descriptors that could make the description process more efficient, such as the voting scheme proposed by Engelcke et al. [10]. Moreover, it could of interest to learn the usefulness of segments as a precursory step to localization, based on their distinctiveness and semantic attributes.

ACKNOWLEDGMENTS

This work was supported by the European Union's Seventh Framework Programme for research, technological development and demonstration under the TRADR project No. FP7-ICT-609763. The authors would like to thank Hannes Sommer, Mark Pfeiffer, Mattia Gollub, Helen Oleynikova, Abel Gawel, Dr. Philipp Krüsi, Dr. Elena Stumm, and Alexander Winkler for their valuable collaboration and support.

REFERENCES

[1] Andrew Brock, Theodore Lim, JM Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. In *Workshop on 3D Deep Learning, NIPS*, 2016.

[2] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744, 1994.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J.J. Leonard. Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[4] Andrei Cramariuc, Renaud Dubé, Hannes Sommer, Roland Siegwart, and Igor Gilitschenski. Learning 3d segment descriptors for place recognition. In *LLM-IROS*, 2017.

[5] Angela Dai, Charles Ruizhongtai Qi, and Matthias Niessner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

[6] Renaud Dubé, Daniel Dugas, Elena Stumm, Juan Nieto, Roland Siegwart, and Cesar Cadena. SegMatch: Segment based place recognition in 3D point clouds. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5266–5272. IEEE, 2017.

[7] Renaud Dubé, Abel Gawel, Hannes Sommer, Juan Nieto, Roland Siegwart, and Cesar Cadena. An online multi-robot slam system for 3d lidars. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 1004–1011. IEEE, 2017.

[8] Renaud Dubé, Mattia Gollub, Hannes Sommer, Igor Igilitschenski, Roland Siegwart, Cesar Cadena, and Juan Nieto. Incremental segment-based localization in 3d point clouds. *IEEE Robotics and Automation Letters*, 3(1):1–8, 2018.

[9] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

[10] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1355–1361. IEEE, 2017.

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[12] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[13] Vitor Guizilini and Fabio Ramos. Learning to reconstruct 3d structures for occupancy mapping. In *Proceedings of Robotics: Science and Systems (RSS)*, 2017.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[16] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3D lidar using fully convolutional network. In *Robotics: Science and Systems (RSS)*, 2016.

[17] Stephanie Lowry, Niko Sunderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 2016.

[18] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D convolutional neural network for real-time object recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[19] Paul Newman, Gabe Sibley, Mike Smith, Mark Cummins, Alastair Harrison, Chris Mei, Ingmar Posner, Robbie Shade, Derik Schroeter, Liz Murphy, et al. Navigating, recognizing and describing urban spaces with vision and lasers. *The International Journal of Robotics Research*, 28(11-12):1406–1433, 2009.

[20] Kingma D. P. and Ba J. L. Adam: a method for stochastic

optimization. *International Conference on Learning Representations, 113*, 2015.

[21] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[22] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

[23] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. OctNet: Learning deep 3D representations at high resolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[24] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

[25] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *International Conference on 3D Vision (3DV)*, 2017.

[26] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen. Shape completion enabled robotic grasping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2442–2447, 2017.

[27] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006.

[28] Martin Weinmann, Boris Jutzi, and Clément Mallet. Semantic 3D scene interpretation: a framework combining optimal neighborhood size selection with relevant features. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3):181, 2014.

[29] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3109–3118, 2015.

[30] Walter Wohlkinger and Markus Vincze. Ensemble of shape functions for 3D object classification. In *IEEE International Conference on Robotics and Biomimetics*, 2011.

[31] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

[32] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D shapenets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[33] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning local geometric descriptors from rgb-d reconstructions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.