

SuMa++: Efficient LiDAR-based Semantic SLAM

Xieyuanli Chen Andres Milioto Emanuele Palazzolo Philippe Giguère Jens Behley Cyrill Stachniss

Abstract—Reliable and accurate localization and mapping are key components of most autonomous systems. Besides geometric information about the mapped environment, the semantics plays an important role to enable intelligent navigation behaviors. In most realistic environments, this task is particularly complicated due to dynamics caused by moving objects, which can corrupt the mapping step or derail localization. In this paper, we propose an extension of a recently published surfel-based mapping approach exploiting three-dimensional laser range scans by integrating semantic information to facilitate the mapping process. The semantic information is efficiently extracted by a fully convolutional neural network and rendered on a spherical projection of the laser range data. This computed semantic segmentation results in point-wise labels for the whole scan, allowing us to build a semantically-enriched map with labeled surfels. This semantic map enables us to reliably filter moving objects, but also improve the projective scan matching via semantic constraints. Our experimental evaluation on challenging highways sequences from KITTI dataset with very few static structures and a large amount of moving cars shows the advantage of our semantic SLAM approach in comparison to a purely geometric, state-of-the-art approach.

I. INTRODUCTION

Accurate localization and reliable mapping of unknown environments are fundamental for most autonomous vehicles. Such systems often operate in highly dynamic environments, which makes the generation of consistent maps more difficult. Furthermore, semantic information about the mapped area is needed to enable intelligent navigation behavior. For example, a self-driving car must be able to reliably find a location to legally park, or pull over in places where a safe exit of the passengers is possible – even in locations that were never seen, and thus not accurately mapped before.

In this work, we propose a novel approach to simultaneous localization and mapping (SLAM) able to generate such semantic maps by using three-dimensional laser range scans. Our approach exploits ideas from a modern LiDAR SLAM pipeline [2] and incorporates semantic information obtained from a semantic segmentation generated by a Fully Convolutional Neural Network (FCN) [21]. This allows us to generate high-quality semantic maps, while at the same time improve the geometry of the map and the quality of the odometry.

The FCN provides class labels for each point of the laser range scan. We perform highly efficient processing of the

X.Chen, A. Milioto, E. Palazzolo, J. Behley, and C. Stachniss are with the University of Bonn, Germany. Philippe Giguère is with the Laval University, Québec, Canada. This work has partly been supported by the German Research Foundation under Germany’s Excellence Strategy, EXC-2070 - 390732324 (PhenoRob) and under grant number BE 5996/1-1 as well as by the Chinese Scholarship Committee.



Fig. 1: Semantic map of the KITTI dataset generated with our approach using only LiDAR scans. The map is represented by surfels that have a class label indicated by the respective color. Overall, our semantic SLAM pipeline is able to provide high-quality semantic maps with higher metric accuracy than its non-semantic counterpart.

point cloud by first using a spherical projection. Classification results on this two-dimensional spherical projection are then back-projected to the three-dimensional point cloud. However, the back-projection introduces artifacts, which we reduce by a two-step process of an erosion followed by depth-based flood-fill of the semantic labels. The semantic labels are then integrated into the surfel-based map representation and exploited to better register new observations to the already built map. We furthermore use the semantics to filter moving objects by checking semantic consistency between the new observation and the world model when updating

the map. In this way, we reduce the risk of integrating dynamic objects into the map. Fig. 1 shows an example of our semantic map representation. The semantic classes are generated by the FCN of Milioto *et al.* [21], which was trained using the SemanticKITTI dataset by Behley *et al.* [1].

The main contribution of this paper is an approach to integrate semantics into a surfel-based map representation and a method to filter dynamic objects exploiting these semantic labels. In sum, we claim that we are (i) able to accurately map an environment especially in situations with a large number of moving objects and we are (ii) able to achieve a better performance than the same mapping system simply removing possibly moving objects in general environments, including urban, countryside, and highway scenes. We experimentally evaluated our approach on challenging sequences of KITTI [11] and show superior performance of our semantic surfel-mapping approach, called *SuMa++*, compared to purely geometric surfel-based mapping and compared mapping removing all potentially moving objects based on class labels. The source code of our approach is available at:

https://github.com/PRBonn/semantic_suma/

II. RELATED WORK

Odometry estimation and SLAM are classical topics in robotics with a large body of scientific work covered by several overview articles [6], [7], [28]. Here, we mainly concentrate on related work for semantic SLAM based on learning approaches and dynamic scenes.

Motivated by the advances of deep learning and Convolutional Neural Networks (CNNs) for scene understanding, there have been many semantic SLAM techniques exploiting this information using cameras [5], [31], cameras + IMU data [4], stereo cameras [10], [15], [18], [33], [38], or RGB-D sensors [3], [19], [20], [26], [27], [30], [39]. Most of these approaches were only applied indoors and use either an object detector or a semantic segmentation of the camera image. In contrast, we only use laser range data and exploit information from a semantic segmentation operating on depth images generated from LiDAR scans.

There exists also a large body of literature tackling localization and mapping changing environments, for example by filtering moving objects [14], considering residuals in matching [22], or by exploiting sequence information [34]. To achieve outdoor large-scale semantic SLAM, one can also combine 3D LiDAR sensors with RGB cameras. Yan *et al.* [37] associate 2D images and 3D points to improve the segmentation for detecting moving objects. Wang and Kim [35] use images and 3D point clouds from the KITTI dataset [11] to jointly estimate road layout and segment urban scenes semantically by applying a relative location prior. Jeong *et al.* [12], [13] also propose a multi-modal sensor-based semantic 3D mapping system to improve the segmentation results in terms of the intersection-over-union (IoU) metric, in large-scale environments as well as in environments with few features. Liang *et al.* [17] propose a novel 3D object detector that can exploit both LiDAR and camera

data to perform accurate object localization. All of these approaches focus on combining 3D LiDAR and cameras to improve the object detection, semantic segmentation, or 3D reconstruction.

The recent work by Parkison *et al.* [23] develops a point cloud registration algorithm by directly incorporating image-based semantic information into the estimation of the relative transformation between two point clouds. A subsequent work by Zaganidis *et al.* [40] realizes both LiDAR combined with images and LiDAR only semantic 3D point cloud registration. Both approaches use semantic information to improve pose estimation, but they cannot be used for online operation because of the long processing time.

The most similar approaches to the one proposed in this paper are Sun *et al.* [29] and Dubé *et al.* [9], which realize semantic SLAM using only a single LiDAR sensor. Sun *et al.* [29] present a semantic mapping approach, which is formulated as a sequence-to-sequence encoding-decoding problem. Dubé *et al.* [9] propose an approach called SegMap, which is based on segments extracted from the point cloud and assigns semantic labels to them. They mainly aim at extracting meaningful features for the global retrieval and multi-robot collaborative SLAM with very limited types of semantic classes. In contrast to them, we focus on generating a semantic map with an abundance of semantic classes and using these semantics to filter outliers caused by dynamic objects, like moving vehicles and humans, to improve both mapping and odometry accuracy.

III. OUR APPROACH

The foundation of our semantic SLAM approach is our Surfel-based Mapping (SuMa) [2] pipeline, which we extend by integrating semantic information provided by a semantic segmentation using the FCN RangeNet++ [21] as illustrated in Fig. 2. The point-wise labels are provided by RangeNet++ using spherical projections of the point clouds. This information is then used to filter dynamic objects and to add semantic constraints to the scan registration, which improves the robustness and accuracy of the pose estimation by SuMa.

A. Notation

We denote the transformation of a point \mathbf{p}_A in coordinate frame A to a point \mathbf{p}_B in coordinate frame B by $\mathbf{T}_{BA} \in \mathbb{R}^{4 \times 4}$, such that $\mathbf{p}_B = \mathbf{T}_{BA}\mathbf{p}_A$. Let $\mathbf{R}_{BA} \in \text{SO}(3)$ and $\mathbf{t}_{BA} \in \mathbb{R}^3$ denote the corresponding rotational and translational part of transformation \mathbf{T}_{BA} .

We call the coordinate frame at timestep t as C_t . Each variable in coordinate frame C_t is associated to the world frame W by a pose $\mathbf{T}_{WC_t} \in \mathbb{R}^{4 \times 4}$, transforming the observed point cloud into the world coordinate frame.

B. Surfel-based Mapping

Our approach relies on SuMa, but we only summarize here the main steps relevant to our approach and refer for more details to the original paper [2]. SuMa first generates a spherical projection of the point cloud \mathcal{P} at timestep t , the so-called vertex map \mathcal{V}_D , which is then used to generate

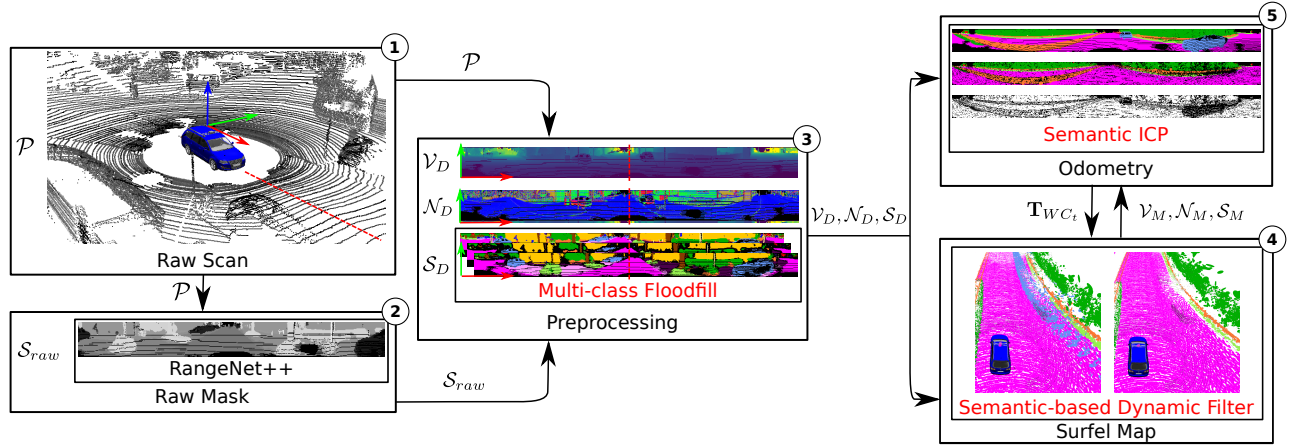


Fig. 2: Pipeline overview of our proposed approach. We integrate semantic predictions into the SuMa pipeline in a compact way: (1) The input is only the LiDAR scan \mathcal{P} . (2) Before processing the raw point clouds \mathcal{P} , we first use a semantic segmentation from RangeNet++ to predict the semantic label for each point and generate a raw semantic mask \mathcal{S}_{raw} . (3) Given the raw mask, we generate an refined semantic map \mathcal{S}_D in the preprocessing module using multi-class flood-fill. (4) During the map updating process, we add a dynamic detection and removal module which checks the semantic consistency between the new observation \mathcal{S}_D and the world model \mathcal{S}_M and remove the outliers. (5) Meanwhile, we add extra semantic constraints into the ICP process to make it more robust to outliers.

a corresponding normal map \mathcal{N}_D . Given this information, SuMa determines via projective ICP in a rendered map view \mathcal{V}_M and \mathcal{N}_M at timestep $t-1$ the pose update $\mathbf{T}_{C_{t-1}C_t}$ and consequently \mathbf{T}_{WC_t} by chaining all pose increments.

The map is represented by surfels, where each surfel is defined by a position $\mathbf{v}_s \in \mathbb{R}^3$, a normal $\mathbf{n}_s \in \mathbb{R}^3$, and a radius $r_s \in \mathbb{R}$. Each surfel additionally carries two timestamps: the creation timestamp t_c and the timestamp t_u of its last update by a measurement. Furthermore, a stability log odds ratio l_s is maintained using a binary Bayes Filter [32] to determine if a surfel is considered stable or unstable. SuMa also performs loop closure detection with subsequent pose-graph optimization to obtain globally consistent maps.

C. Semantic Segmentation

For each frame, we use RangeNet++ [21] to predict a semantic label for each point and generate a semantic map \mathcal{S}_D . RangeNet++ semantically segments a range image generated by a spherical projection of each laser scan. Briefly, the network is based on the SqueezeSeg architecture proposed by Wu *et al.* [36] and uses a DarkNet53 backbone proposed by Redmon *et al.* [25] to improve results by using more parameters, while keeping the approach real-time capable. For more details about the semantic segmentation approach, we refer to the paper of Milioto *et al.* [21]. The availability of point-wise labels in the field of view of the sensor makes it also possible to integrate the semantic information into the map. To this end, we add for each surfel the inferred semantic label y and the corresponding probability of that label from the semantic segmentation.

D. Refined Semantic Map

Due to the projective input and the blob-like outputs produced as a by-product of in-network down-sampling of RangeNet++, we have to deal with errors of the semantic labels, when the labels are re-projected to the map. To reduce these errors, we use a flood-fill algorithm, summarized

Algorithm 1: flood-fill for refining \mathcal{S}_D .

Input: semantic mask \mathcal{S}_{raw} and the corresponding vertex map \mathcal{V}_D

Result: refined mask \mathcal{S}_D

Let \mathcal{N}_s be the set of neighbors of pixel $s \in \mathcal{S}$ within a filter kernel of size d .

θ is the rejection threshold.

0 represents an empty pixel with label 0.

```

foreach  $\mathbf{s}_u \in \mathcal{S}_{\text{raw}}$  do
  Let  $\mathcal{S}_{\text{raw}}^{\text{eroded}}(\mathbf{u}) = \mathbf{s}_u$ 
  foreach  $\mathbf{n} \in \mathcal{N}_{\mathbf{s}_u}$  do
    if  $y_{\mathbf{s}_u} \neq y_{\mathbf{n}}$  then
      Let  $\mathcal{S}_{\text{raw}}^{\text{eroded}}(\mathbf{u}) = 0$ 
      break
    end
  end
end
foreach  $\mathbf{s}_u \in \mathcal{S}_{\text{raw}}^{\text{eroded}}$  do
  Let  $\mathcal{S}_D(\mathbf{u}) = \mathbf{s}_u$ 
  foreach  $\mathbf{n} \in \mathcal{N}_{\mathbf{s}_u}$  do
    if  $\|\mathbf{u} - \mathbf{u}_n\| < \theta \cdot \|\mathbf{u}\|$  then
      Let  $\mathcal{S}_D(\mathbf{u}) = \mathbf{n}$  if  $y_{\mathbf{s}_u} = 0$ 
      break
    end
  end
end

```

in Alg. 1. It is inside the preprocessing module, which uses depth information from the vertex map \mathcal{V}_D to refine the semantic mask \mathcal{S}_D .

The input to the flood-fill is the raw semantic mask \mathcal{S}_{raw} generated by the RangeNet++ and the corresponding vertex map \mathcal{V}_D . The value of each pixel in the mask \mathcal{S}_{raw} is a semantic label. The corresponding pixel in the vertex map contains the 3D coordinates of the nearest 3D point in the

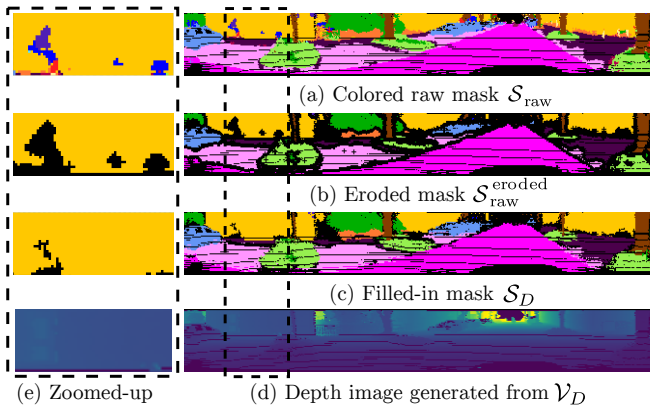


Fig. 3: Visualization of the processing steps of the proposed flood-fill algorithm. Given the (a) raw semantic map \mathcal{S}_{raw} , we first use an erosion to remove boundary labels and small areas of wrong labels resulting in (b) the eroded mask $\mathcal{S}_{\text{raw}}^{\text{eroded}}$. (c) We then finally fill-in eroded labels with neighboring labels to get a more consistent result \mathcal{S}_D . Black points represent empty pixels with label 0. (d) Shows the depth and (e) the details inside the areas with the dashed borders.

LiDAR coordinate system. The output of the approach is the refined semantic mask \mathcal{S}_D .

Considering that the prediction uncertainty of object boundaries is higher than that of the center of an object [16], we use the following two steps in the flood-fill. The first step is an erosion that removes pixels where the neighbors within a kernel of size d show at least one different semantic label resulting in the eroded mask $\mathcal{S}_{\text{raw}}^{\text{eroded}}$. Combining this mask with the depth information generated from the vertex map \mathcal{V}_D , we then fill-in the eroded mask. To this end, we set the label of an empty boundary pixel to the neighboring labeled pixels if the distances of the corresponding points are consistent, i.e., less than a threshold θ .

Fig. 3 shows the intermediate steps of this algorithm. Note that the filtered semantic map does contain less artifacts compared to the raw predictions. For instance, the wrong labels on the wall of the building are mostly corrected, which is illustrated in Fig. 3(e).

E. Filtering Dynamics using Semantics

Most existing SLAM systems rely on geometric information to represent the environment and associate observations to the map. They work well under the assumption that the environment is mostly static. However, the world is usually dynamic, especially when considering driving scenarios, and several of the traditional approaches fail to account for dynamic scene changes caused by moving objects. Therefore, moving objects can cause wrong associations between observations and the map in such situations, which must be treated carefully. Commonly, SLAM approaches use some kind of outlier rejection, either by directly filtering the observation or by building map representations that filter out changes caused by moving objects.

In our approach, we exploit the labels provided by the semantic segmentation to handle moving objects. More specifically, we filter dynamics by checking semantic consistency between the new observation \mathcal{S}_D and the world model \mathcal{S}_M , when we update the map. If the labels are inconsistent,

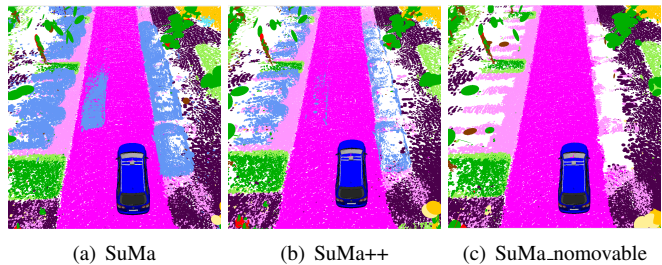


Fig. 4: Effect of the proposed filtering of dynamics. For all figures, we show the color of the corresponding label, but note that SuMa does not use the semantic information. (a) Surfels generated by SuMa; (b) our method; (c) remove all potentially moving objects.

we assume those surfels belong to an object that moved between the scans. Therefore, we add a penalty term to the computation of the stability term in the recursive Bayes filter. After several observations, we can remove the unstable surfels. In this way, we achieve a detection of dynamics and finally a removal.

More precisely, we penalize that surfel by giving a penalty odds(p_{penalty}) to its stability log odds ratio l_s , which will be updated as follows:

$$l_s^{(t)} = l_s^{(t-1)} + \text{odds} \left(p_{\text{stable}} \exp \left(-\frac{\alpha^2}{\sigma_\alpha^2} \right) \exp \left(-\frac{d^2}{\sigma_d^2} \right) \right) - \text{odds}(p_{\text{prior}}) - \text{odds}(p_{\text{penalty}}), \quad (1)$$

where $\text{odds}(p) = \log(p(1-p)^{-1})$ and p_{stable} and p_{prior} are probabilities for a stable surfel given a compatible measurement and the prior probability, respectively. The terms $\exp(-x^2\sigma^{-2})$ are used to account for noisy measurements, where α is the angle between the surfel's normal \mathbf{n}_s and the normal of the measurement to be integrated, and d is the distance of the measurement in respect to the associated surfel. The measurement normal is taken from \mathcal{N}_D and the correspondences from the frame-to-model ICP, see [2] for more details.

Instead of using semantic information, Pomerleau *et al.* [24] proposes a method to infer the dominant motion patterns within the map by storing the time history of velocities. In contrast to our approach, their method requires a given global map to estimate the velocities of points in the current scan. Furthermore, their robot pose estimate is assumed to be rather accurate.

In Fig. 4, we illustrate the effect of our filtering method compared to naively removing all surfels from classes corresponding to movable objects. When utilizing the naive method, surfels on parked cars are removed, even though these might be valuable features for the incremental pose estimation. With the proposed filtering, we can effectively remove the dynamic outliers and obtain a cleaner semantic world model, while keeping surfels from static objects, e.g., parked cars. These static objects are valuable information for the ICP scan registration and simply removing them can lead to failures in scan registration due to missing correspondences.

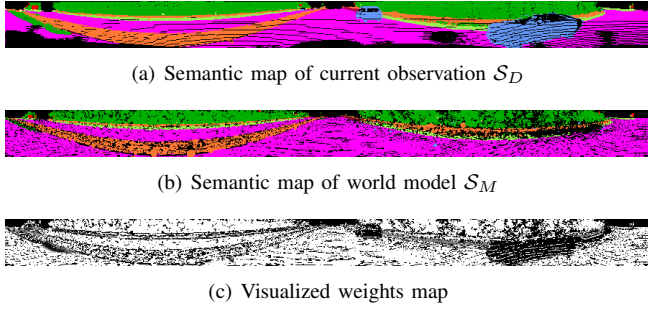


Fig. 5: Visualization of Semantic ICP: (a) semantic map \mathcal{S}_D for the current laser scan, (b) corresponding semantic map \mathcal{S}_M rendered from the model, (c) the weight map during the ICP. The darker the pixel, the lower is the weight of the corresponding pixel.

F. Semantic ICP

To further improve the pose estimation using the frame-to-model ICP, we also add semantic constraints to the optimization problem, which helps to reduce the influence of outliers. Our error function to minimize for ICP is given by:

$$E(\mathcal{V}_D, \mathcal{V}_M, \mathcal{N}_M) = \sum_{\mathbf{u} \in \mathcal{V}_D} w_{\mathbf{u}} \mathbf{n}_{\mathbf{u}}^\top \underbrace{\left(\mathbf{T}_{C_{t-1}C_t}^{(k)} \mathbf{u} - \mathbf{v}_{\mathbf{u}} \right)}_{r_{\mathbf{u}}}, \quad (2)$$

where each vertex $\mathbf{u} \in \mathcal{V}_D$ is projectively associated to a reference vertex $\mathbf{v}_{\mathbf{u}} \in \mathcal{V}_M$ and its normal $\mathbf{n}_{\mathbf{u}} \in \mathcal{N}_M$ via

$$\mathbf{v}_{\mathbf{u}} = \mathcal{V}_M \left(\Pi \left(\mathbf{T}_{C_{t-1}C_t}^{(k)} \mathbf{u} \right) \right), \quad (3)$$

$$\mathbf{n}_{\mathbf{u}} = \mathcal{N}_M \left(\Pi \left(\mathbf{T}_{C_{t-1}C_t}^{(k)} \mathbf{u} \right) \right), \quad (4)$$

$r_{\mathbf{u}}$ and $w_{\mathbf{u}}$ are the corresponding residual and weight, respectively.

For the minimization, we use Gauss-Newton and determine increments δ by iteratively solving:

$$\delta = (\mathbf{J}^\top \mathbf{W} \mathbf{J})^{-1} \mathbf{J}^\top \mathbf{W} \mathbf{r}, \quad (5)$$

where $\mathbf{W} \in \mathbb{R}^{n \times n}$ is a diagonal matrix containing weights $w_{\mathbf{u}}$ for each residual $r_{\mathbf{u}}$, $\mathbf{r} \in \mathbb{R}^n$ is the stacked residual vector, and $\mathbf{J} \in \mathbb{R}^{n \times 6}$ the Jacobian of \mathbf{r} with respect to the increment δ . Besides the hard association and weighting by a Huber norm, we add extra constraints from higher level semantic scene understanding to weight the residuals. In this way, we can combine semantics with geometric information to make the ICP process more robust to outliers.

Within ICP, we compute the weight $w_{\mathbf{u}}^{(k)}$ for the residual $r_{\mathbf{u}}^{(k)}$ in iteration k as follows:

$$w_{\mathbf{u}}^{(k)} = \rho_{\text{Huber}} \left(r_{\mathbf{u}}^{(k)} \right) C_{\text{semantic}}(\mathcal{S}_D(\mathbf{u}), \mathcal{S}_M(\mathbf{u})) \mathbb{I} \left\{ l_s^{(k)} \geq l_{\text{stable}} \right\}, \quad (6)$$

where $\rho_{\text{Huber}}(r)$ corresponds to the Huber norm, given by:

$$\rho_{\text{Huber}}(r) = \begin{cases} 1 & , \text{ if } |r| < \delta \\ \delta |r|^{-1} & , \text{ otherwise.} \end{cases} \quad (7)$$

For semantic compatibility $C_{\text{semantic}}((y_{\mathbf{u}}, P_{\mathbf{u}}), (y_{v_{\mathbf{u}}}, P_{v_{\mathbf{u}}}))$, the term is defined as:

$$C_{\text{semantic}}(\cdot, \cdot) = \begin{cases} P(y_{\mathbf{u}} | \mathbf{u}) & , \text{ if } y_{\mathbf{u}} = y_{v_{\mathbf{u}}} \\ 1 - P(y_{\mathbf{u}} | \mathbf{u}) & , \text{ otherwise.} \end{cases}, \quad (8)$$

which is using the certainty of the predicted label to weight the residual. By $\mathbb{I}\{a\}$, we denote the indicator function that is 1 if the argument a is true, and 0 otherwise.

Fig. 5 shows the weighting for a highway scene with two moving cars visible in the scan, see Fig. 5(a). Note that our filtering of dynamics using the semantics, as described in Sec. III-E, removed the moving cars from the map, see Fig. 5(b). Therefore, we can also see a low weight corresponding to lower intensity in Fig. 5(c), since the classes of the observation and the map disagree.

IV. EXPERIMENTAL EVALUATION

Our experimental evaluation is designed to support our main claims that we are (i) able to accurately map even in situations with a considerable amount of moving objects and we are (ii) able to achieve better performance than simply removing possibly moving objects in general environments, including urban, countryside, and highway scenes.

To this end, we evaluate our approach using data from the KITTI Vision Benchmark [11], where we use the provided point clouds generated by a Velodyne HDL-64E S2 recorded at a rate of 10 Hz. To evaluate the performance of odometry, the dataset proposes to compute relative errors in respect to translation and rotation averaged over different distances between poses and averaging it. The ground truth poses are generated using pose information from an inertial navigation system and in most sequences the GPS position is referenced to a base station, which makes it quite accurate, but still often only locally consistent.

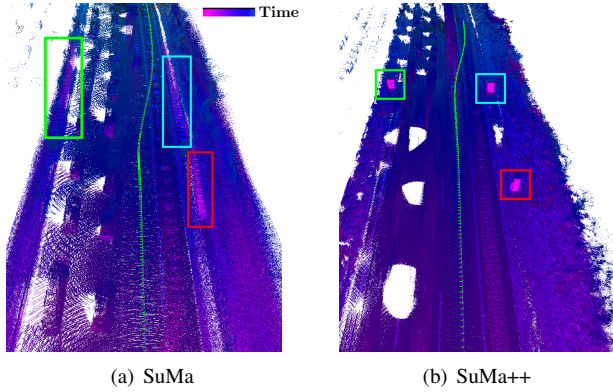
In the following, we compare our proposed approach (denoted by SuMa++) against the original surfel-based mapping (denoted by SuMa), and SuMa with the naive approach of removing all movable classes (cars, buses, trucks, bicycles, motorcycles, other vehicles, persons, bicyclists, motorcyclist) given by the semantic segmentation (denoted by SuMa_nomovable).

The RangeNet++ for the semantic segmentation was trained using point-wise annotations [1] using all training sequences from the KITTI Odometry Benchmark, which are the labels available for training purposes. This includes sequences 00 to 10, except for sequence 08 which is left out for validation.

We tested our approach on an Intel Xeon(R) W-2123 with 8 cores @3.60 GHz with 16 GB RAM, and an Nvidia Quadro P4000 with 8 GB RAM. The RangeNet++ needs on average 75 ms to generate point-wise labels for each scan and the surfel-mapping needs on average 48 ms, but we need at most 190 ms to integrate loop closures in some situations (on sequence 00 of the training set with multiple loop closures).

A. KITTI Road Sequences

The first experiment is designed to show that our approach is able to generate consistent maps even in situations with many moving objects. We show results on sequences from the road category of the raw data of the KITTI Vision Benchmark. Note that these sequences are not part of the odometry benchmark, and therefore no labels are provided

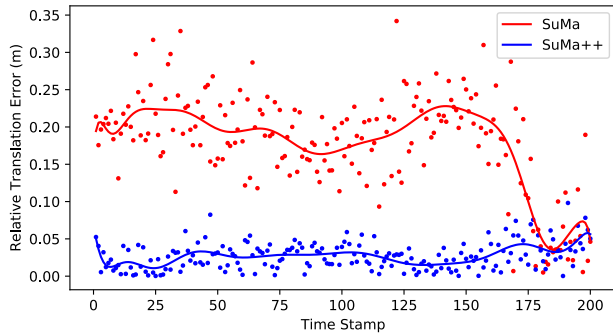


(a) SuMa

(b) SuMa++



(c) Corresponding front-view camera image



(d) Relative translation error plot for each time step

Fig. 6: Qualitative results. (a) SuMa without semantics fails to correctly estimate the motion of the sensor due to the consistent movement of cars in the vicinity of the sensor. The frame-to-model ICP locks to consistently moving cars leading to the map inconsistencies, highlighted by rectangles. (b) By incorporating semantics, we are able to correctly estimate the sensor’s movement and therefore get a more consistent map of the environment and a better estimate of sensor pose via ICP. The color of the 3D points refers to the timestamp when the point has been recorded for the first time. (c) Corresponding front-view camera image, where we highlight the traffic signs. (d) Corresponding relative translation error plot for each time step. The dots are the calculated relative translational errors in each time stamp and the curves are polynomial fitting results of those dots.

for the semantic segmentation, meaning that our network learned to infer the semantic classes of road driving scenes, and it is not simply memorizing. These sequences, especially the highway sequences, are challenging for SLAM methods, since here most of the objects are moving cars. Moreover, there are only sparse distinct features on the side of the road, like traffic signs or poles. Building corners or other more distinctive features are not available to guide the registration process. In such situations, wrong correspondences on consistently moving outliers (like cars in a traffic jam) often lead to wrongly estimated pose changes and consequently

TABLE I: Results on KITTI Road dataset

Sequence	Environment	Approach		
		SuMa	SuMa_nomovable	SuMa++
30	country	0.38/0.96	0.39/0.97	0.38/0.90
31	country	1.54/2.02	1.66/2.13	1.19/2.02
32	country	1.38/1.70	1.63/1.76	1.00/1.57
33	highway	1.61/1.79	1.72/1.80	1.67/1.87
34	highway	0.79/1.17	0.70/1.14	0.60/1.09
35	highway	5.11/26.8	3.20/1.22	2.90/1.11
36	highway	0.93/1.31	0.95/1.30	0.93/1.40
37	country	0.65/1.51	0.62/1.36	0.60/1.48
38	highway	1.07/1.66	1.04/1.46	0.89/1.42
39	country	0.46/1.04	0.47/0.98	0.44/1.05
40	country	1.09/18.0	0.79/1.92	0.75/1.95
41	highway	1.24/15.6	0.92/1.46	1.14/1.67
Average		1.35/6.13	1.17/1.46	1.04/1.46

Relative errors averaged over trajectories of 5 to 400 m length: relative rotational error in degrees per 100 m / relative translational error in % . Bold numbers indicate top performance for laser-based approaches.

We rename the KITTI road raw dataset '2011_09_26_drive_0015_sync' - '2011_10_03_drive_0047_sync' into sequences 30-41.

inconsistencies in the generated map.

Fig. 6 shows an example generated with SuMa and the proposed SuMa++. In the case of the purely geometric approach, we clearly see that the pose cannot be correctly estimated, since the highlighted traffic signs show up at different locations leading to large inconsistencies. With our proposed approach, where we are able to correctly filter the moving cars, we instead generate a consistent map as evident by the highlighted consistently mapped traffic signs. We also plot the relative translational errors of odometry results of both SuMa and SuMa++ in this example. The dots represent the relative translational errors in each timestamp and the curves are polynomial fitting results given the dots. It shows that SuMa++ achieves more accurate pose estimates in such a challenging environments with many outliers caused by moving objects.

Tab. I shows the relative translational and relative rotational error and Fig. 7 shows the corresponding trajectories for different methods tested on this part of the dataset. Generally, we see that our proposed approach, SuMa++, generates more consistent trajectories and achieves in most cases a lower translational error than SuMa. Compared to the baseline of just removing all possibly moving objects, SuMa_nomovable, we see very similar performance compared to SuMa++. This confirms that the major reason for the worse performance of SuMa in such cases is the inconsistencies caused by actually moving objects. However, we will show in the next experiments that removing all potentially moving objects can also have negative effects on the pose estimation performance in urban environments.

B. KITTI Odometry Benchmark

The second experiment is designed to show that our methods performs superior compared to simply removing certain semantic classes from the observations. This evaluation is performed on the KITTI Odometry Benchmark.

Tab. II shows the relative translational and relative rotational errors. IMLS-SLAM [8] and LOAM [41] are state-of-the-art LiDAR-based SLAM approaches. In most sequences, we can see similar performance of SuMa++ compared to

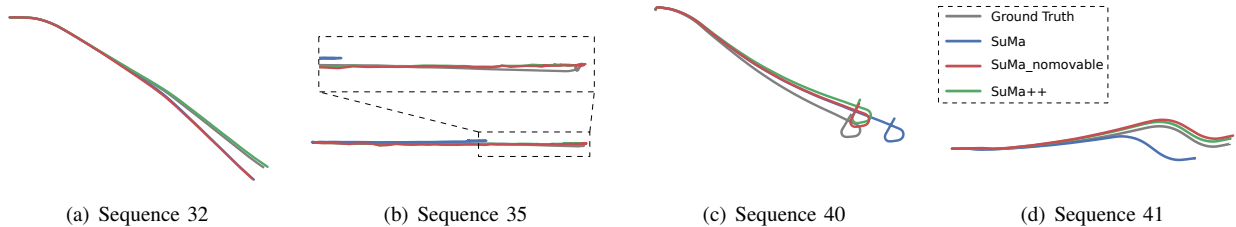


Fig. 7: Trajectories of different methods test on KITTI road dataset.

TABLE II: Results on KITTI Odometry (training)

Approach	00* urban	01 highway	02* urban	03 country	04 country	Sequence 05* country	06* urban	07* urban	08* urban	09* urban	10 country	Average
SuMa	0.23/0.68	0.54/1.70	0.48/1.20	0.50/0.74	0.27/0.44	0.20/0.43	0.3/0.54	0.54/0.74	0.38/1.20	0.22/0.62	0.32/0.72	0.36/0.83
SuMa_nomovable	22.0/58.0	0.57/1.70	25.0/63.0	0.45/0.67	0.26/0.37	14.0/36.0	0.22/0.47	0.21/0.34	13.0/32.0	13.0/45.0	12.0/19.0	23.3/9.24
SuMa++	0.22/0.64	0.46/1.60	0.37/1.00	0.46/ 0.67	0.26/0.37	0.20/0.40	0.21/0.46	0.19/0.34	0.35/1.10	0.23/ 0.47	0.28/0.66	0.29/0.70
IMLS-SLAM [8]	-0.50	-0.82	-0.53	-/0.68	-0.33	-0.32	-0.33	-0.33	-0.80	-/0.55	-0.53	-0.55
LOAM [41]	-/0.78	-/1.43	-/0.92	-/0.86	-/0.71	-/0.57	-/0.65	-/0.63	-/1.12	-/0.77	-/0.79	-/0.84

Relative errors averaged over trajectories of 100 to 800 m length: relative rotational error in degrees per 100 m / relative translational error in %. Sequences marked with an asterisk contain loop closures. Bold numbers indicate best performance in terms of translational error.

the state-of-the-art. More interestingly, the baseline method SuMa_nomovable diverges, particularly in urban scenes.

This might be counter-intuitive since these environments contain a considerable amount of man-made structures and other more distinctive features. But there are two reasons contributing to this worse performance that become clear when one looks at the results and the configuration of the scenes where mapping errors occur. First, even though we try to improve the results of the semantic segmentation, there are wrong predictions that lead to a removal of surfels in the map that are actually static. Second, the removal of parked cars is problems as these are good and distinctive features for aligning scans. Both effects contribute to making the surfel map sparser. This is even more critical as parked cars are the only distinctive or reliable features. In conclusion, the simple removal of certain classes is at least in our situation sub-optimal and can lead to worse performance.

To evaluate the performance of our approach in unseen trajectories, we uploaded our results for server-side evaluation on unknown KITTI test sequences so that no parameter tuning on the test set is possible. Thus, this serves as a good proxy for the real-world performance of our approach. In the test set, we achieved an average rotational error of 0.0032 deg/m and an average translational error of 1.06%, which is an improvement in terms of translational error, when compared to 0.0032 deg/m and 1.39% of the original SuMa.

C. Discussion

During the map updating process, we only penalize surfels of dynamics of movable objects, which means we do not penalize semantically static objects, e.g. vegetation, even though sometimes leaves of vegetation change and the appearance of vegetation changes with the viewpoint due to laser beams that only get reflected from certain viewpoints. Our motivation for this is that they can also serve as good landmarks, e.g., the trunk of a tree is static and a good feature

for pose estimation. Furthermore, the original geometric-based outlier rejection mechanism employing the Huber norm often down-weights such parts.

There is an obvious limitation of our method: we cannot filter out dynamic objects in the first observation. Once there is a large number of moving objects in the first scan, our method will fail because we cannot estimate a proper initial velocity or pose. We solve this problem by removing all potentially movable object classes in the initialization period. However, a more robust method would be to backtrack changes due to the change in the observed moving state and thus update the map retrospectively.

Lastly, the results of our second experiment shows convincingly that blindly removing a certain set of classes can deteriorate the localization accuracy, but potentially moving objects still might be removed from a long-term representation of a map to also allow for a representation of otherwise occluded parts of the environment that might be visible at a different point of time.

V. CONCLUSION

In this paper, we presented a novel approach to build semantic maps enabled by a laser-based semantic segmentation of the point cloud not requiring any camera data. We exploit this information to improve pose estimation accuracy in otherwise ambiguous and challenging situations. In particular, our method exploits semantic consistencies between scans and the map to filter out dynamic objects and provide higher-level constraints during the ICP process. This allows us to successfully combine semantic and geometric information based solely on three-dimensional laser range scans to achieve considerably better pose estimation accuracy than the pure geometric approach. We evaluated our approach on the KITTI Vision Benchmark dataset showing the advantages of our approach in comparison to purely geometric approaches. Despite these encouraging results, there are several avenues

for future research on semantic mapping. In future work, we plan to investigate the usage of semantics for loop closure detection and the estimation of more fine-grained semantic information, such as lane structure or road type.

REFERENCES

- [1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.
- [2] J. Behley and C. Stachniss. Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. In *Proc. of Robotics: Science and Systems (RSS)*, 2018.
- [3] B. Bescos, J.M. Fcail, J. Civera, and J. Neira. DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):4076–4083, 2018.
- [4] S. Bowman, N. Atanasov, K. Daniilidis, and G.J. Pappas. Probabilistic Data Association for Semantic SLAM. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.
- [5] N. Brasch, A. Bozic, J. Lallemand, and F. Tombari. Semantic Monocular SLAM for Highly Dynamic Environments. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 393–400, 2018.
- [6] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser. Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, 2017.
- [7] 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 Trans. on Robotics (TRO)*, 32:1309–1332, 2016.
- [8] J. Deschaud. ImSls-slam: scan-to-model matching based on 3d data. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.
- [9] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena. SegMap: 3D Segment Mapping using Data-Driven Descriptors. In *Proc. of Robotics: Science and Systems (RSS)*, 2018.
- [10] P. Ganti and S.L. Waslander. Visual slam with network uncertainty informed feature selection. *arXiv preprint*, 2018.
- [11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.
- [12] J. Jeong, T. S. Yoon, and J. B. Park. Multimodal Sensor-Based Semantic 3D Mapping for a Large-Scale Environment. *Expert Systems with Applications*, 105:1–10, 2018.
- [13] J. Jeong, T.S. Yoon, and J.B. Park. Towards a Meaningful 3D Map Using a 3D Lidar and a Camera. *Sensors*, 18(8), 2018.
- [14] R. Kümmeler, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard. A Navigation System for Robots Operating in Crowded Urban Environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, Karlsruhe, Germany, 2013.
- [15] P. Li, T. Qin, and S. Shen. Stereo Vision-based Semantic 3D Object and Ego-motion Tracking for Autonomous Driving. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018.
- [16] X. Li, Z. Liu, P. Luo, C.C. Loy, and X. Tang. Not All Pixels Are Equal: Difficulty-Aware Semantic Segmentation via Deep Layer Cascade. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [17] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 641–656, 2018.
- [18] K.N. Lianos, J.L. Schonberger, M. Pollefeys, and T. Sattler. VSO: Visual Semantic Odometry. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 234–250, 2018.
- [19] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger. Fusion++: Volumetric Object-Level SLAM. In *Proc. of the Intl. Conf. on 3D Vision (3DV)*, pages 32–41, 2018.
- [20] J. McCormac, A. Handa, A.J. Davison, and S. Leutenegger. SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.
- [21] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.
- [22] E. Palazzolo, J. Behley, P. Lottes, P. Giguere, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.
- [23] S.A. Parkison, L. Gan, M.G. Jadidi, and R.M. Eustice. Semantic Iterative Closest Point through Expectation-Maximization. In *Proc. of British Machine Vision Conference (BMVC)*, page 280, 2018.
- [24] F. Pomerleau, P. Krusiand, F. Colas, P. Furgale, and R. Siegwart. Long-term 3d map maintenance in dynamic environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2014.
- [25] J. Redmon and A. Farhadi. YOLOv3: An Incremental Improvement. *arXiv preprint*, 2018.
- [26] M. Rünz, M. Buffier, and L. Agapito. MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In *Proc. of the Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, pages 10–20, 2018.
- [27] R.F. Salas-Moreno, R.A. Newcombe, H. Strasdat, P.H.J. Kelly, and A.J. Davison. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1352–1359, 2013.
- [28] C. Stachniss, J. Leonard, and S. Thrun. *Springer Handbook of Robotics, 2nd edition*, chapter 46: Simultaneous Localization and Mapping. Springer Verlag, 2016.
- [29] L. Sun, Z. Yan, A. Zaganidis, C. Zhao, and T. Duckett. Recurrent-OctoMap: Learning State-Based Map Refinement for Long-Term Semantic Mapping With 3-D-Lidar Data. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):3749–3756, 2018.
- [30] N. Sünderhauf, T.T. Pham, Y. Latif, M. Milford, and I. Reid. Meaningful maps with object-oriented semantic mapping. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5079–5085, 2017.
- [31] K. Tateno, F. Tombari, I. Laina, and N. Navab. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 6565–6574, 2017.
- [32] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
- [33] V. Vineet, O. Miksik, M. Lidegaard, M. Niessner, S. Golodetz, V. Prisacariu, O. Kahler, D. Murray, S. Izadi, P. Perez, and P. Torr. Incremental Dense Semantic Stereo Fusion for Large-Scale Semantic Scene Reconstruction. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2015.
- [34] O. Vysotska and C. Stachniss. Lazy Data Association for Image Sequences Matching under Substantial Appearance Changes. *IEEE Robotics and Automation Letters (RA-L)*, 2016.
- [35] J. Wang and J. Kim. Semantic segmentation of urban scenes with a location prior map using lidar measurements. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017.
- [36] B. Wu, A. Wan, X. Yue, and K. Keutzer. SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 1887–1893, 2018.
- [37] J. Yan, D. Chen, H. Myeong, T. Shiratori, and Y. Ma. Automatic Extraction of Moving Objects from Image and LiDAR Sequences. In *Proc. of the Intl. Conf. on 3D Vision (3DV)*, 2014.
- [38] S. Yang, Y. Huang, and S. Scherer. Semantic 3D occupancy mapping through efficient high order CRFs. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017.
- [39] C. Yu, Z. Liu, X. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei. DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1168–1174, 2018.
- [40] A. Zaganidis, L. Sun, T. Duckett, and G. Cielniak. Integrating Deep Semantic Segmentation Into 3-D Point Cloud Registration. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):2942–2949, 2018.
- [41] J. Zhang and S. Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, 41:401–416, 2017.