Mapping with Dynamic-Object Probabilities Calculated from Single 3D Range Scans

Philipp Ruchti Wolfram Burgard

***Abstract*— Various autonomous robotic systems require maps for robust and safe navigation. Particularly when robots are employed in dynamic environments, accurate knowledge about which components of the robot perceptions belong to dynamic and static aspects in the environment can greatly improve navigation functions. In this paper we propose a novel method for building 3D grid maps using laser range data in dynamic environments. Our approach uses a neural network to estimate the pointwise probability of a point belonging to a dynamic object. The output from our network is fed to the mapping module for building a 3D grid map containing only static parts of the environment. We present experimental results obtained by training our neural network using the KITTI dataset and evaluating it in a mapping process using our own dataset. In extensive experiments, we show that maps generated using the proposed probability about dynamic objects increases the accuracy of the resulting maps.**

I. Introduction

Building maps is a fundamental requirement in many robotic tasks. Maps are typically used to support different navigation tasks including path planning and localization. However, the presence of dynamic objects in the map increases the difficulty of such a task. For this reason, localization is usually done using a map that only represents the static aspects of the environment. The generation of such maps, however, requires a robust detection of dynamic objects or measurements caused by such objects.

In this paper, we propose a novel mapping approach to learning three-dimensional maps from 3D laser data. Our approach predicts the probability of 3D laser points being reflected by dynamic objects to build map of the static components only. In our approach, we first apply a neural network to learn a probability about the fact that a measurement is reflected by a dynamic object. In contrast to many other approaches, this probability is determined using only a single 3D laser scan and does not rely on previous scans or camera images. In a second mapping phase, our approach considers the predicted probability to generate a 3D grid map only containing the static parts of the environment.

Our approach has several features that improve localization and navigation of mobile robots in highly dynamic environment. First, as the probability is calculated from individual scans, it does not require a comparison of pairs or multiple scans to detect moving objects. Rather, it can identify also dynamic objects that are currently not moving like a standing pedestrian or a parked car. In the remainder of this paper we refer to moving and movable objects as dynamic objects. Generating maps without dynamic objects typically yields maps that remain valid for longer periods of time. The maps we generate are 3D grid maps in which cells store the probability that a scan beam is reflected by a static object. Second, as the prediction of the dynamic objects is based on single 3D scans, our approach can also be applied to robots with bad or no IMU or odometry data. Finally our method is highly efficient and can operate online at 20 Hz. Thus there also is a potential to utilize it to avoid dynamic objects while navigating in dynamic environments.

The contribution of this paper is twofold. First we present an approach using deep learning to efficiently predict the probability that points represent dynamic objects in single 3D scans. Second we use the computed probabilities to build a 3D grid map where each cell represents the probability that a beam is reflected by a static object. Fig. 1 shows an example of a proposed dynamic probability together with the corresponding range image. Please note, that we do not use camera images in our approach.

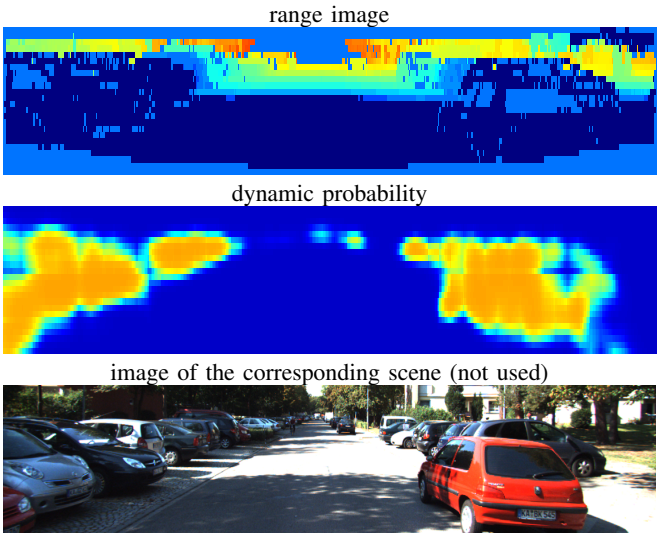

Fig. 1: An example of a range image (blue to red depicts near to far) together with its computed probability of being dynamic (orange shows a high dynamic probability). Note that the camera image is not used by our algorithm.

All authors are members of the University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany. {ruchtip,burgard}@informatik.uni-freiburg.de This work has been partly supported by the European Commission under the grant numbers ERC-AG-PE7-267686-LifeNav, FP7-610603-EUROPA2 and FP7-610532-SQUIRREL as well as by a grant from the Ministry of Science, Research and the Arts of Baden-Württemberg (Az: 32-7545.24-9/1/1) for the project ZAFH-AAL.

The remainder of this paper is organized as follows. After presenting related work we present our approach to predict the dynamic probabilities which is based on a modified ResNet proposed by Valada *et al.* [1]. The network was presented to be used with camera images. We show how the 3D scans can be transformed into 2D images to be suitable for the network. Afterwards we explain the mapping process which is a modified version of the mapping approach presented by Hähnel *et al.* [2]. This section is followed by the experimental evaluation.

II. Related Work

There has been a tremendous amount of work regarding the detection of dynamic objects in either camera data [1], [3], [4] or laser scans [5], [6], [7].

To detect dynamic objects in camera data Fan *et al.* [4] as well as Reddy *et al.* [8] feed images into a neural network to segment the scene into different classes while also estimating which segments move. In a similar context Vertens *et al.* [9] apply a neural network to jointly detect cars and predict if these are moving. The network gets consecutive camera images as well as optical flow as input. Chabot *et al.* [3] propose a convolutional network to detect cars in color images. They employ a coarse to fine approach to predict bounding boxes for cars and additionally fit 3D shape templates to the detection to even predict object parts that are occluded. Chen *et al.* [10] use camera images as well as different views of 3D scans to predict 3D bounding boxes for different object classes. Xu *et al.* [11] combine camera images with 3D scans for object detection. For each detection in the camera images they assign a segment from the 3D laser scan. Other than these methods our algorithm does not use camera images. We convert the individual 3D laser scans to two 2D images, one for range and one for intensity. These images have a smaller resolution than a camera image and hold less information. While the majority of the previously developed methods for laser range data take more than one scan to determine the measurements caused by dynamic objects our method uses a single 3D scan to predict its dynamic components.

Instead of images previous work also employed 3D range scans together with neural networks for object detection. Similar to our work, Li *et al.* [7] convert 3D scans into range images before applying a neural network for object detection. Engelcke *et al.* [6], [12] propose a fast network based on a sliding window to detect objects directly in 3D scans. In contrast to these works, which generate bounding boxes around detected objects, we predict a pointwise probability of belonging to a dynamic object.

Dewan *et al.* [5] propose a method to detect and distinguish moving and movable points in 3D laser scans. While this approach first computes motion flow between two consecutive scans and seeks to identify entire objects, our method uses a single 3D scan as input.

Compared to other works about detecting dynamics our method has several advantages. First, it only uses single 3D scans to determine a per point dynamic probability and does not need to take previous measurements into account. Thus, it does not require scan matching or tracking methods. Second, our method does not use camera images and thus is not limited to proper lightning conditions. Third, our method can also identify movable objects that are not moving in the current scan.

Hähnel *et al.* [2] introduced a probabilistic approach based on the expectation maximization (EM) algorithm to estimate the beams reflected by moving objects from entire laser scans and to build a map of the static aspects in the scans only. In this paper, instead of the EM-based estimation of static objects, we learn a prior of movable objects and thus can also remove measurements caused by dynamic objects that are non-moving during the data collection process such as parked cars or standing pedestrians. Meyer-Delius *et al.* [13] introduced a variant of occupancy grids in which they utilize a Hidden Markov Model for every cell to better keep track of the potential changes in the occupancy of each cell.

III. Mapping with Dynamic-Object Probabilities

The overall goal of our method is to create a 3D grid map which only contains those components of the environment that are static over a longer period of time. To achieve this, we first use a neural network to compute a per-point probability of being dynamic from range and intensity images as well as other modalities computed from a single 3D laser scan. We then utilize this probability to build a grid map thereby taking the dynamic-object probabilities into account. In the remainder of this section, we will describe the neural network used to compute the dynamic probability, how we apply the network to 3D laser scans and how we build a 3D grid map from the labeled scans.

A. Dynamic Probability

We apply a neural network to compute a probability for each point in a single 3D laser scan that this point belongs to a dynamic object. Our approach does not only consider moving objects as dynamic objects but also movable objects as they might move in future. In this work we apply the *neural expert network* proposed by Valada *et al.* [1] to 3D laser scans. It is a network for semantic segmentation of images and builds upon a modified ResNet50 network. The network follows the general principle of an encoder-decoder network. In the first half it aggregates the image features while in the second half it upscales the feature maps to the original image size to get the segmentation. Compared to ResNet50, the network employed in this work uses *multi-scale blocks* to detect objects of different sizes. By applying dilation instead of down-sampling the network allows for a segmentation of higher resolution. For a more detailed network description please refer to Valada *et al.* [1]. This network was proposed to be used with RGB-color camera images. To apply the neural network to a 3D laser scan, we first have to transform it to a 2D image. We investigate different modalities to fill the image channels, including range and intensity generated from the 3D laser scan (see Sec. IV-A.2). For a more robust learning process we compute

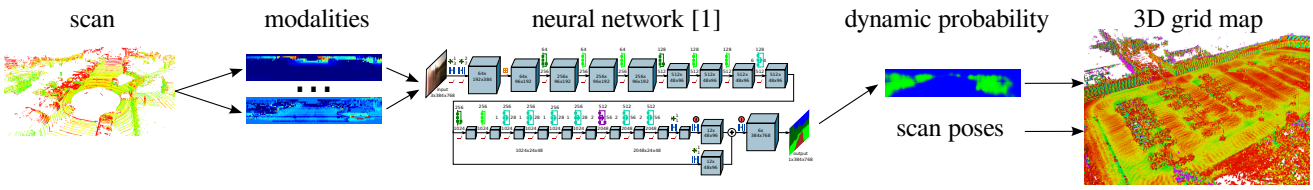

Fig. 2: Overview of our proposed system. We first convert the 3D scan into 2D images, which we then feed into the network. We then utilize the resulting dynamic probability with the scan poses to generate the 3D grid map of the static aspects of the environment.

the mean for each channel over the whole training dataset and use this to generate zero mean training data. The original network predicts binary class labels only. In this work, however, we are interested in obtaining a probability that a point is dynamic. To achieve this, we remove the final argmax-layer of the network and interpret the output of the softmax-layer as an approximation of the desired probability. After applying the trained network to our 2D representation of the 3D scans we need to project the prediction back into the 3D scan. To do so, we project each 3D scan point into the range image and assign the corresponding predicted dynamic probability to it.

B. Mapping

To compute a 3D grid map from the set of scans we adapt the map building method proposed by Hähnel *et al.* [2]. This approach employs an expectation maximization (EM) framework to decide which beams of a range scan are reflected by static objects. These beams are then used to compute α- and β-values for each cell of the map. Here, α corresponds to the number of beams which end in this cell and β counts the beams that pass through a cell without ending in it. These values are then employed to compute the reflectance probability of a grid cell according to

$$m = \frac{\alpha}{\alpha + \beta}. \tag{1}$$

In contrast to their work we do not need EM as we compute a dynamic probability with our trained neural network. Instead we directly incorporate our continuous dynamic probability into the calculation of α and β.

Let p be the dynamic probability calculated by our network for a beam that is not a maximum range measurement. For the cell, in which that beam ends, we add $1 - p$ to the α-value. In addition, we add p to the β-value of that cell. If the beam was a maximum-range measurement, we update neither the α- nor the β-value. Independent of maximum range measurements, we increment the β-value of every cell traversed by the beam by one. More formally, for a beam that has a predicted dynamic probability p and passes through the cells $j = 1, \ldots, k-1$ and ends in cell k we calculate

$$\beta_j \leftarrow \beta_j + 1. \tag{2}$$

If the beam is not a maximum range measurement, we calculate

$$\alpha_k \leftarrow \alpha_k + (1 - p) \tag{3}$$

$$\beta_k \leftarrow \beta_k + p. \tag{4}$$

IV. Experimental Evaluation

In this section we provide experiments carried out with the KITTI dataset to test the performance of the dynamic probability prediction. We also present how our approach is able to segment moving and movable objects using our dynamic probability. We furthermore present results indication that our probability can be applied to create 3D grid maps of static aspects only.

A. Dynamic-Object Probability

1) Training data from the KITTI dataset: We trained and evaluated our neural network to predict the dynamic probabilities using the publicly available KITTI object dataset created by Geiger *et al.* [14]. This dataset contains camera images with labeled object bounding boxes and 3D LiDAR scans. To apply this data to our framework, we projected the laser scans into the camera frame and transfered the labels that fall into a bounding box to the 3D points. Each bounding box encloses an object that can move such as cars, vans, trucks, pedestrians, sitting persons, cyclists or trams. For all these movable objects we treat points falling into the corresponding bounding boxes as dynamic and all others as static. Unfortunately, the provided ground truth labels were limited to the field-of-view of the camera. Therefore, we only use the part of a 3D scan that overlaps with the camera view.

As we found out during our experiments, the data set contains a substantial amount of errors. Several bounding boxes are missing and others are either displaced or too small. Two examples are shown in Fig. 3. To reduce the impact of the inaccuracies of the bounding boxes, we increase the box sizes in both horizontal directions (not up and down) for the entire dataset by 0.4 m. For our experiments we split the labeled training data into a test and a validation dataset, each with roughly 3,700 scans, as proposed by Chen *et al.* [15].

2) Modalities: To apply our neural network, which works on 2D images, to 3D laser scans we first need to transform the data into 2D images. We thus generate 2D images filled with modalities such as range or intensity obtained from the 3D scan. In our experiments we test different modalities and evaluate combinations of modalities.

When recording 3D scans with a moving robot one has to correct the 3D points based on IMU or odometry data to compensate the motion of the robot. Accordingly, the back-projection of the 3D points into the range image is approximate and multiple or no points might fall into a single pixel. In our current system, we use a laser scanner with

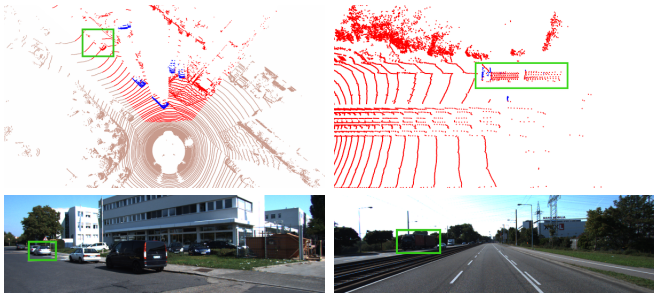

Fig. 3: Two scans (top) and the corresponding images (bottom) with incorrect labels. The blue dots in the scan are the labeled points. The marked car in the left is completely missing, while for the truck on the right only the points of the front side of the truck have been labeled. Brown dots correspond to points not present in the camera image.

64 individual laser beams and the back-projection leads to images with a size of 2,000 $\times$ 24 pixels.

To generate the different modalities we first collect all 3D points that are projected into the pixel at position (x, y):

$$\left\{ \left(P_{(x,y,j)}, I_{(x,y,j)} \right) \right\},$$

where $P = (X, Y, Z)$ is the 3D position and I is the measured intensity of the 3D point. Using these we then generate multiple modalities: First we calculate the minimum distance of all 3D points falling into a cell:

$$r_{(x,y)} = \min_j \|P_{(x,y,j)}\|, \tag{5}$$

which we denote as *range*. Furthermore, we calculate the mean *intensity* of all points falling into a pixel:

$$i_{(x,y)} = \overline{I_{(x,y,j)}}. \tag{6}$$

For the *height* we compute the mean z-value (up):

$$h_{(x,y)} = \overline{Z_{(x,y,j)}}. \tag{7}$$

To get rid of the absolute distance value we compute the *rangeDiff* modality given by the standard deviation from the range of the pixel (x, y). More precisely, using all eight neighboring 2D-pixels (x', y') of a cell (x, y) we calculate

$$m_{(x,y)} = \frac{1}{8-1} \sum_{(x',y')} \left(r_{(x,y)} - r_{(x',y')} \right)^2. \tag{8}$$

3) Data augmentation: To increase the diversity of our training set we use data augmentation. When creating a range image from a given 3D laser scan one has to provide the sensor origin, which usually is $(0, 0, 0)$. We create augmented scans by moving the origin in a radius of 1 m in the horizontal plane while generating the range image. We then further augment the resulting range images by applying augmentations on image level. More precisely, for each image and augmentation we select the corresponding augmentation with a given pre-defined probability. Thereby we enforce that at least one augmentation is chosen. Thus we apply between on and all six augmentations to each image. In our current system, we used the following augmentations with parameters sampled from the denoted intervals (probability of choosing that augmentation in brackets):

- Rotate the image by $[-2, 2]$ degrees ($p = 0.4$).
- Scale the image by a factor of $[0.8, 1)$ ($p = 0.4$).
- Translate the image by $[(-50, -5), (50, 5)]$ pixels ($p = 0.2$).
- Flip the image horizontally ($p = 0.3$).
- Crop the image by a factor of $[0.8, 0.9]$ ($p = 0.4$).
- Skew the image by $[0.025, 0.05]$ ($p = 0.3$).

For each scan we generate three range images by shifting the origin. We then four times augment each of these images plus the original range image by applying the augmentations described above. Together with the non-augmented images this yields 20 images per scan. For the training we generate a multi-channel image where the different channels are filled with all tested modalities.

4) Training the neural network: To train the neural network we use the labels 0 = static, 1 = dynamic and the ignore label 2 for unseen/unlabeled points. We pad the 2D images with zeros such that width and height are a power of two. We also crop the images from a size of 2,048 $\times$ 64 pixels to the field of view of the camera (512 $\times$ 64 pixels).

5) IoU results: We use the validation dataset to compute the per-class intersection over union (IoU) for our learned neural networks. We train our network based on different modalities and compare the results with and without augmentation.

In our first experiment we demonstrate how well different modalities perform individually and how they can be combined to improve the prediction result. Tab. I shows the intersection over union (IoU) score on the KITTI dataset for different modalities trained using augmented data. As can be seen, *range* and *rangeDiff* perform well as standalone modalities. By combining modalities the result further improves. The combination of *rangeDiff*, *intensity* and *height* yields the best results. It performs better than the same combination using *range*. This is due to the fact that *rangeDiff* shows more contrast which seems to help the network.

We also evaluate how much the augmentation boosts the performance of the network. Tab. II shows that the result improves especially if we only use single modalities.

	intersection over union		
modalities	static	dynamic	mean
range	0.956	0.632	0.794
rangeDiff	0.958	0.633	**0.795**
intensity	0.945	0.526	**0.735**
height	0.943	0.477	**0.710**
range, *intensity*	0.961	0.664	0.813
rangeDiff, *intensity*	0.964	0.688	0.826
range, *intensity*, *height*	0.961	0.667	0.814
rangeDiff, *intensity*, *height*	0.965	0.695	**0.830**

TABLE I: Prediction quality of our learned dynamic probability trained on augmented data using different modalities. The highlighted values show how the combination of modalities increases the performance.

modalities	intersection over union static	dynamic	mean	mean IoU increase
range	0.945	0.573	0.506	**0.288**
rangeDiff	0.949	0.592	0.514	**0.281**
intensity	0.939	0.509	0.483	**0.252**
range, *intensity*, *height*	0.949	0.599	0.774	0.040
rangeDiff, *intensity*, *height*	0.957	0.647	0.802	0.028

TABLE II: Prediction quality of our learned dynamic probability trained without augmented data using different modalities. The last column shows the increase of the mean IoU when using augmentation. The highlighted values indicate that the augmentation increases the performance especially for single modalities.

In the above experiments we compared modalities that use only one scan and cannot sense the real motion of objects in the dataset. In this experiment we test as to whether additionally considering a second previously acquired scan (1 m in robot motion) can improve our dynamic probability. We evaluate a *motion* heuristic for how much parts of the scan moved relative to the previous scan, accounting for occlusions. For each point in the scan we search the nearest point in the transformed previous scan. We then remove motion over 60 km/h as this is most likely the result of an occlusion. Tab. III shows that only using the *motion* heuristic performs worse than *range* or *intensity* (compare to Tab. I) as it only has nonzero values for actually moving objects. By combining the *motion* heuristic with *range* and *intensity* we can improve the performance, but not over similar combinations using a single scan (see Tab. I).

We also test how the network performs if we transform the *previous* scan into the frame of the current one and add its range image as modality. The result is shown in the last line of Tab. III. As can be seen, adding it to *range* and *intensity* decreases the performance (Tab. I line 5). This is due to the fact, that differences between *range* and *previous* are mainly due to occlusions while only small parts differ due to the motion in the scene.

6) Runtime: In this experiment we demonstrate how much time is spent on the individual components of our approach. For this experiment carried out based on KITTI dataset, we used a computer equipped with an i7-2700K and a GeForce GTX 980 and ran the detection in a single thread. To transform the laser scans into a range image we used the PCL implementation [16] which requires 13.4 ms per scan. Generating the different modalities takes between 0.3 ms (*height*) and 5.2 ms (*rangeDiff*). Finally, predicting the dynamic probability given the modalities takes 28.7 ms. Our approach allows to perform the detection of movable objectes at a rate of 20 Hz so that every scan can be processed. The majority of the time required to create the range image can be reduced to less than 1 ms by using ordered laser scans, which are normally provided by Velodyne laser scanners.

B. Mapping

In this experiment we show how the dynamic probability predicted by our trained network can be used to generate a 3D grid map that contains only the static parts of the environment. We also show that our proposed mapping algorithm is able to remove moving objects as well as movable objects.

modalities	intersection over union static	dynamic	mean
motion	0.923	0.357	0.640
range, *intensity*, *motion*	0.959	0.661	0.810
rangeDiff, *intensity*, *motion*	0.953	0.621	0.787
range, *intensity*, *previous*	0.956	0.625	0.791

TABLE III: Prediction quality of our learned dynamic probability trained on augmented data using modalities computed with a previous scan.

We used our robot *Viona* equipped with an Velodyne HDL-64E LiDAR and an Applanix PosLV (IMU and GPS) to record datasets on our campus parking lot. We applied a SLAM system to correct the scan poses reported by the Applanix system. Following the results of the previous experiments we use the combination of *rangeDiff*, *intensity*, *height* for dynamic probability prediction in the rest of this work as it performs best.

To apply the neural network trained on the KITTI dataset to our data we had to correct the intensities that are different on both datasets. To generate data for training and testing we compute the per channel mean of the training dataset and subtract it channel-wise to get zero mean data. This mean value per channel stays the same for training and testing. To compensate for the different intensity values in both datasets we recomputed the mean for this modality on one of our parking lot dataset and used this value during testing. For this experiment we use one of our recorded campus parking lot datasets to build a map as proposed above. The mean of the predicted probability per cell on the used dataset is shown in Fig. 4.

We generate two different maps from the same dataset. The first map incorporates points given their dynamic probability as well as a second map were we assume all points are static. We choose a cell size of 0.25 m.

To show that our mapping process successfully removes dynamic objects we manually generated a ground truth labeling of static and dynamic objects for the dataset. Then, we determined if a dynamic objects is represented in the map by a cell with a reflectance value of at least 0.5. The ground truth labeling as well as the two maps annotated with the not included (green) and included dynamic cells (red) are shown in Fig. 5. We can see that objects like the moving person recording the dataset are not included in either map. On the other hand the parked cars are only removed by our mapping process using the dynamic probabilities. Our mapping method using the dynamic probability is able to remove 95.66% of all dynamic cells while the map generated assuming all points are static is able to remove the moving objects such as the pedestrians and cyclists but not the cars, it removes 78.25% of all dynamic cells.

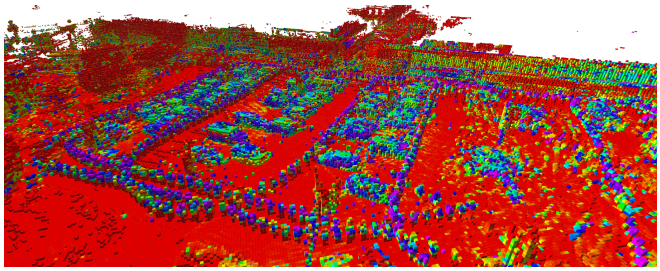

Fig. 4: Mean predicted probability per cell on our parking lot dataset. Dynamic objects are shown in blue (parked cars, walking pedestrians) while static objects are red.

V. Conclusion

In this paper we presented a method to generate a 3D grid map of the static aspects of the environment of a mobile robot. We first predict a pixelwise dynamic probability that a point is part of a dynamic object for range images generated from single 3D laser scans using a neural network. Despite we only use single scans we are capable of detecting moving objects as well as parked cars and other movable objects. We demonstrated the performance of our approach using the publicly available KITTI dataset. In the experiments we also demonstrated that the proposed dynamic probability can be used to generate accurate maps of the static parts of the environment.

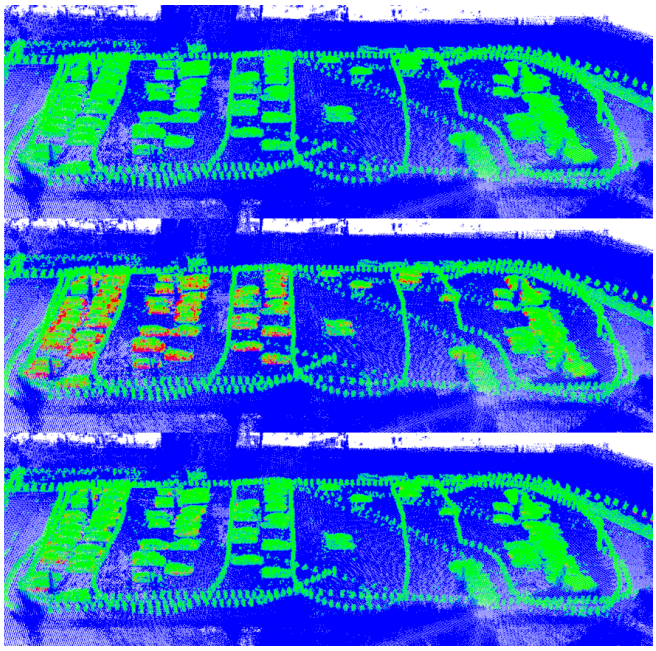

Fig. 5: Our approach robustly removes dynamic objects from the generated maps. The top image shows the manually labeled dynamic objects in green. The second image shows the map built under the assumption that everything is static. While the moving pedestrians as well as the cyclists are not included in the map (green), the parking cars at least partially are (red). The bottom image demonstrates how effectively our approach removes parked cars.

References

[1] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "Adapnet: Adaptive semantic segmentation in adverse environmental conditions," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4644–4651.

[2] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 1557–1563.

[3] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, "Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image," *arXiv preprint arXiv:1703.07570*, 2017.

[4] Q. Fan, Y. Yi, L. Hao, F. Mengyin, and W. Shunting, "Semantic motion segmentation for urban dynamic scene understanding," in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*. IEEE, 2016, pp. 497–502.

[5] A. Dewan, G. L. Oliveira, and W. Burgard, "Deep semantic classification for 3d lidar data," *arXiv preprint arXiv:1706.08355*, 2017.

[6] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1355–1361.

[7] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," *arXiv preprint arXiv:1608.07916*, 2016.

[8] N. D. Reddy, P. Singhal, and K. M. Krishna, "Semantic motion segmentation using dense crf formulation," in *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing*. ACM, 2014, p. 56.

[9] J. Vertens, A. Valada, and W. Burgard, "Smsnet: Semantic motion segmentation using deep convolutional neural networks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2017.

[10] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," *arXiv preprint arXiv:1611.07759*, 2016.

[11] J. Xu, K. Kim, Z. Zhang, H.-w. Chen, and Y. Owechko, "2d/3d sensor exploitation and fusion for enhanced object detection," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. IEEE, 2014, pp. 778–784.

[12] D. Z. Wang and I. Posner, "Voting for voting in online point cloud object detection." in *Robotics: Science and Systems*, 2015.

[13] D. Meyer-Delius, M. Beinhofer, and W. Burgard, "Occupancy grid models for robot mapping in changing environments." in *AAAI*, 2012.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[15] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems*, 2015, pp. 424–432.

[16] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.