Multi-object detection in urban scenes utilizing 3D background maps and tracking

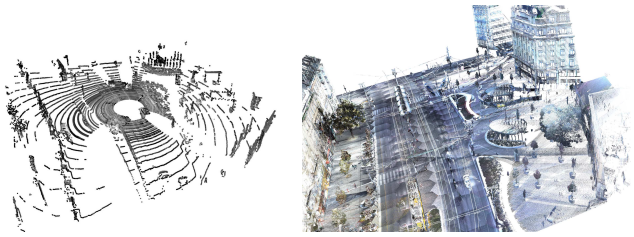
Örkény Zováthi[1,2] Lóránt Kovács[1,3] Balázs Nagy[1,3] Csaba Benedek[1,3]
[1]Institute for Computer Science and Control (SZTAKI), Machine Perception Research Laboratory
[2]Budapest University of Technology and Economics, Department of Control Engineering and Information Technology
[3]Pázmány Péter Catholic University, Faculty of Information Technology and Bionics
{lastname.firstname}@sztaki.hu

***Abstract*—In this paper we propose a novel approach for upgrading real time 3D dynamic object detection methods operating on rotating multi-beam (RMB) Lidar measurements using 3D background city maps stored in new generation geographic information systems (GIS) and previously detected dynamic objects propagated by tracking. First, we apply a state-of-the-art object detection method and distinguish the predicted dynamic object candidates and the remaining static regions of the current Lidar measurement. Next we find an optimal transformation between the static part of the RMB Lidar measurements and the background city map using a multimodal point cloud registration algorithm operating in the Hough space. After the accurate alignment, we filter false-positively detected object candidates in the RMB Lidar data based on the map. To find additional objects missed by the object detector on the current measurement, we apply a Kalman-filter based object tracking. Hereby we first predict the current state of the previously detected and tracked objects. Next, we apply a Hungarian matcher based assignment between the tracked and the current objects and update the object list according to the result. For better accuracy, we keep all predictions through a couple of frames. We evaluated our method qualitatively and quantitatively in crowded urban scenes of Budapest, Hungary, and the results showed that with background map based filtering we can achieve a 26,52% improvement detecting vehicles and 9,38% for pedestrians in *precision*, while via tracking, a 12,84% improvement for vehicles and 14,34% for pedestrians in *recall* against the state-of-the-art object detection method relying purely on a single Lidar time frame.**

***Index Terms*—Lidar, Object detection, Tracking, Background map**

I. Introduction

Detecting and tracking dynamic objects relying *purely* on sparse real time Lidar point clouds is an active research area in autonomous driving. In the recent years several deep learning-based approaches have emerged in the literature, which are solely based on Lidar measurements [1], [2], [3], [4], [5], [6], [7] and show promising results on the available public datasets [8], [9]. In terms of average precision, the state-of-the-art PointPillars network [1] also outperforms methods based on the fusion of the onboard Lidar and camera data [10], [11], [12], [13], [14]. However, due to the limitations of the spatial resolution of the Lidar measurements (see Figure 1(a)), the state-of-the-art object detection method is still not reliable alone in complex and crowded urban scenarios and produces two sorts of errors. On one hand, its predictions are sometimes false in static regions that have similar appearance to dynamic objects. On the other hand, it temporarily misses dynamic objects which are only partly sensed on the current measurement due to cover or occlusion by other parts or objects of the scene. The main objective of this paper is to introduce a robust method in order to eliminate these two sorts of errors and improve the accuracy of the state-of-the-art object detectors using *purely* Lidar data utilizing 3D background maps and tracking.



(a) Sparse Lidar measurement (b) Dense 3D city map

Fig. 1. Measurement (left) and map (right) on Kálvin square, Budapest

Nowadays several new generation geographic information systems (GIS) contain high resolution and geo-referred 3D point cloud maps (Figure 1(b)) of cities obtained by Mobile Laser Scanning. To utilize object level information of this point cloud, we need its semantic interpretation in order to distinguish ground, dynamic (moving vehicle, people) and static (street furniture, column, wall) object regions [15]. This process needs a lot of computation, although, we can calculate it *offline* in order to construct a semantic background map containing only static objects of the scene. Hence, this background map can be accessed in real time to validate the predicted object candidates, which process enables the removal of the false-positive predictions.

Utilizing background maps is a good solution for filtering

This work was supported by the EFOP 3.6.1.-16-2016-00014 program (Research, development, and educational integration of disruptive technologies in the field of e-mobility) and by the Hungarian National Research, Development and Innovation Fund (NKFIA K-120233). The work was also supported by European Union and the Széchenyi 2020 Program, co-financed by the European Social Fund. (grants EFOP-3.6.2-16-2017-00013, EFOP-3.6.2-16-2017-00015, EFOP-3.6.3-VEKOP-6-2017-00002). For Balázs Nagy and Lóránt Kovács, the contribution was supported by the New National Excellence Program of the Ministry for Innovation and Technology (ÚNKP-19-3-I-PPKE-33, ÚNKP-19-3-I-PPKE-86).

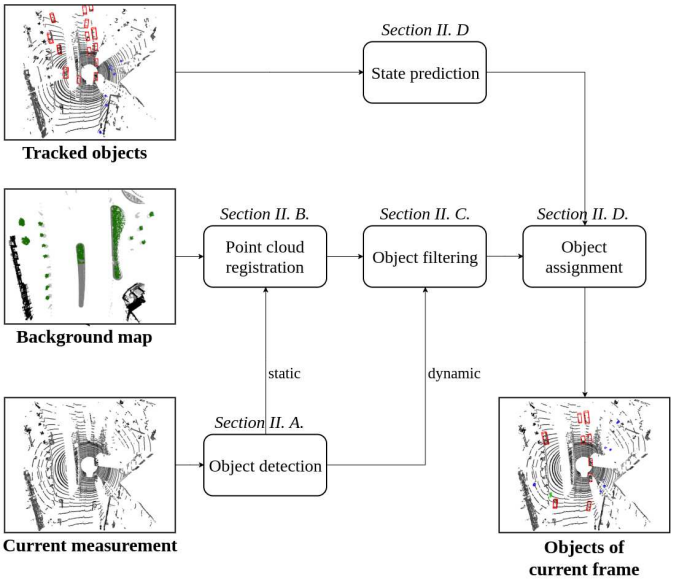


Fig. 2. The workflow of the proposed method

and removing erroneous object candidates but this step does not contribute to the elimination of missing objects. Hereby, if we consider previously observed data by object tracking, we can infer to their current position by a Kalman-filter and keep them through a few time frames which resolve the temporal errors. For reference, we apply the state-of-the-art PointPillars network [1] that we trained on the KITTI [8] dataset and some additional annotated scenes from Budapest. As dynamic objects, we distinguish vehicles, pedestrians and cyclists.

II. Proposed method

The workflow of the proposed algorithm is shown in Figure 2. First, we apply the PointPillars [1] network and distinguish the dynamic object candidates from the remaining static part of the scene. Then we register the static regions of the current measurement to the background map. After accurate alignment, we remove false dynamic object candidates based on their overlapping ratio with the map. Owing to this filtering, we keep and track all remaining candidates which are considered as dynamic objects of the scenario.

A. Object detection

Taking the current Lidar measurement, first we apply the PointPillars [1] network for initial dynamic object detection in the scene. The network determines for each object (o) the 3D bounding box position (P) and orientation (θ), label (c) and prediction score (p) value. To achieve the best results, we keep all objects with a score higher than 0.3, according to [1]. Based on the detection results, we split the point cloud measurement into dynamic object candidates and remaining static regions (see on Figure 3.).

B. Point cloud registration

After removing the dynamic object candidates from the measurement, we align the remaining static part to the 3D

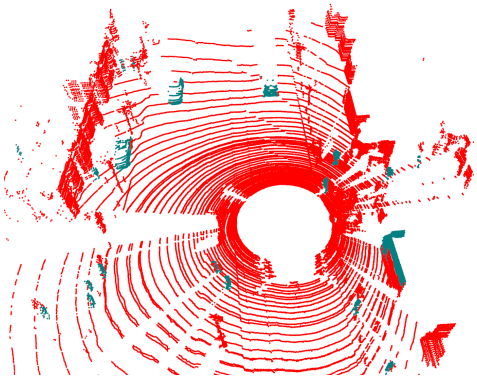
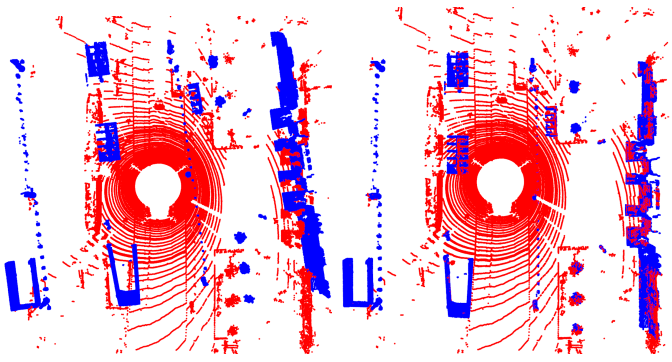


Fig. 3. Initial static-dynamic segmentation on raw Lidar measurement. Color codes: static parts (red), dynamic object candidates (dark green)

background map. Hereby assuming that the vehicle uses a GPS receiver and the map is geo-referred, we transform the vehicle's local Lidar measurements into the global world coordinate system as an initial alignment (Figure 4(a)). Next we cut the surrounding environment of the measurement's GPS position with 35 meter radius as a basis for the precise alignment. Since the density characteristics of the two point clouds are significantly different, point level approaches [16], [17], [18] can not converge because of the initial error or they need heavy computation thus they cannot fulfill the real time algorithm execution requirement. Instead of using point level alignment techniques, we apply our previously proposed multimodal point cloud registration approach [19]. Here we first apply abstract grid based blob extraction on the static measurements and assign keypoints to each blob. Then we find the optimal transformation between compatible keypoints in the following form:

$$\mathbf{T}_{dx,dy,dz,\alpha} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & dx \\ -\sin\alpha & \cos\alpha & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



(a) GPS based alignment (b) Registrated point clouds

Fig. 4. Before (left) and after (right) the registration. Color codes: onboard measurement (red), background map (blue)

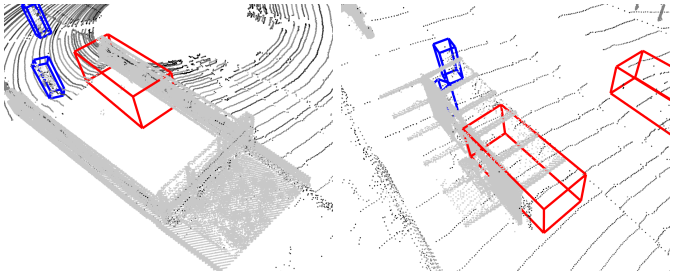


Fig. 5. Falsely detected object samples overlapping with the background map. Color codes: predicted vehicle (red box), predicted pedestrian (blue box)

where dx, dy, dz are the GPS position error and α is the rotation angle around the z axis. Finally we search for the optimal parameter quartet in the generalized Hough space through a voting process. The registration results are shown in Figure 4(b).

C. Object filtering based on background map

By applying the previously introduced registration algorithm, the current measurement is accurately aligned to the background map. Therefore we can validate the object candidates in the following way: at first, for each object candidate of the current measurement (o_c^i) we calculate the exact bounding box parameters in the geo-referred coordinate system. Then we check whether the 3D bounding box contains static points of the map or not. If an object candidate has an intersection with a part of the static background map (Figure 5.) and it includes at least 10 points of the map, we mark it as a false detection and assign its points to the static part of the measurement. Otherwise we keep and consider o_c^i as a dynamic object. The 10 point threshold ensures robustness and keeps candidates which have only a tiny intersection with a static object, considering that we might have to deal with a noisy segmentation and a noisy map.

D. Finding additional dynamic objects via tracking

The previously introduced steps can accurately assure the validity of all object candidates. However, they operate solely on a single Lidar frame. In cases when an object is temporarily occluded, the current detection fails. By applying tracking on the detected objects and their histories, we can still follow dynamic objects of the scene missed by the object detector temporarily. Based on a given object's previous state we predict its current position with a Kalman-filter. Note that besides the elimination of missing objects, the application of the Kalman-filter yields smoothed object trajectories as well.

For the Kalman-filter, we determine on the current measurement three different types of objects (Figure 6.): previously *tracked* and currently missed, *tracked and also found* or previously missed and *newly found*. To categorize each object of the scene, we make an optimal assignment between the current dynamic objects ($\mathcal{O}_c$) and previously detected and tracked dynamic objects ($\mathcal{O}_t$) applying the Hungarian algorithm with a custom cost matrix C. For each object $o_c^i \in \mathcal{O}_c$ and $o_t^j \in \mathcal{O}_t$ we define the following *pairing cost* value C:

$$C(o_c^i, o_t^j) = w_1 \cdot \Delta P(o_c^i, o_t^j) + w_2 \cdot \Delta\theta(o_c^i, o_t^j) + w_3 \cdot \Delta c(o_c^i, o_t^j)$$

where ΔP is the distance between the three dimensional center points and $\Delta\theta$ is the orientation difference of the two objects calculated from their position in the geo-referred coordinate system. Note that since all measurements are aligned to the background map, we get the exact position and orientation of all dynamic objects in the same, global coordinate system and therefore we can ignore the motion effect of the ego vehicle. The weight parameters w_1, w_2, w_3 were experimentally optimized. The factor of the object labels Δc can be defined as follows:

$$\Delta c(o_c^i, o_t^j) = \begin{cases} 0 & \text{if } c_c^i = c_t^j \\ 1 & \text{otherwise} \end{cases}$$

This supplement ensures pairing objects belonging to the same class. In cases of matching objects belonging to different classes, we check the confidence of the object detector and keep the label with the higher confidence. Applying the Hungarian algorithm to the cost matrix, it delivers a globally optimal assignment. Then for each object pair, we update the

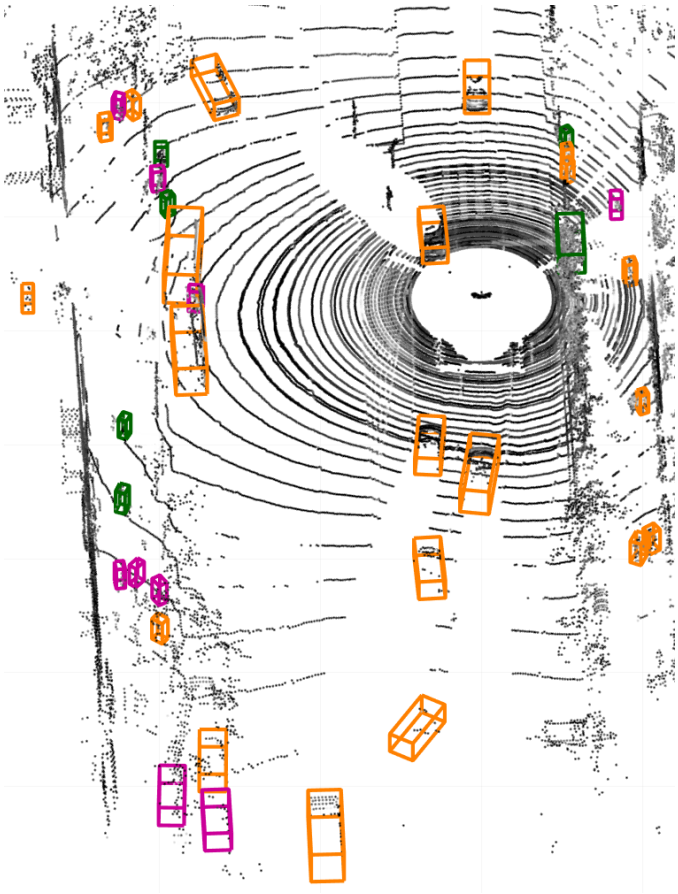


Fig. 6. Three types of objects on one Lidar frame. Color codes: tracked and also found (orange box), newly found (green), temporarily missed but tracked (purple) objects

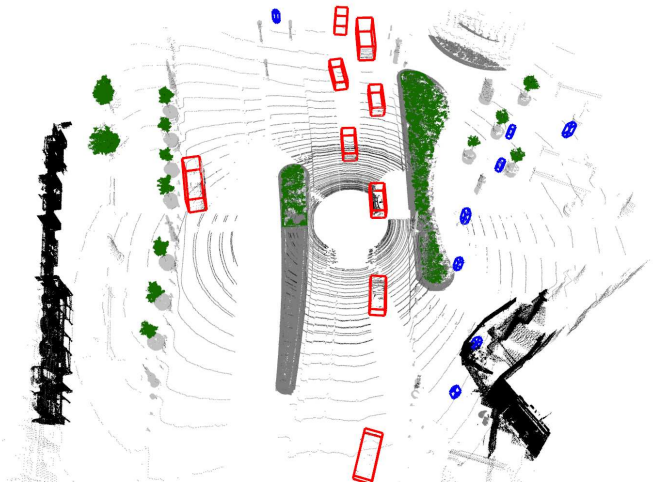


Fig. 7. Qualitative results on Kálvin square, Budapest

states of the *tracked* objects based on the current Lidar measurement. Otherwise, for *newly detected* objects we add them to the tracked object list, and for *not found*, but previously tracked objects, we keep the predicted state and label for the next Lidar measurement. In general, we keep all tracked objects for the next 10 frames after their last appearance. This parameter was empirically set, according to our road scene experience.

III. Evaluation

We evaluated the proposed method on real point cloud data sequences recorded on different crowded roads of Budapest (Deák and Fővám square). The measurements were taken by a Velodyne HDL-64E sensor, mounted to the top of a test vehicle. The reference city map was provided by a Riegl VMX-450 Mobile Laser Scanning system, recorded by the Budapest road management company (Budapest Közút Zrt.). For qualitative evaluation, in Figure 7. we show that the proposed method detects and tracks all dynamic objects of the vehicle's environment. Thanks to the exploitation of static objects in the background map, there are not any false predictions on the scene. In Figure 8. we show an example of the same scene where a false prediction was filtered by the proposed method. In Figure 9. we show an other scenario where the detector misses pedestrians due to occlusion, but those are still recognisable using the proposed tracking algorithm.

Method	Class	Precision	Recall	F-score
PointPillars	Vehicle	64,15%	82,93%	72,34%
	Pedestrian	84,37%	72,19%	77,81%
PointPillars with map	Vehicle	90,67%	82,93%	86,62%
	Pedestrian	93,75%	72,19%	81,57%
PointPillars with map and tracking	Vehicle	**93,15%**	**95,77%**	**94,44%**
	Pedestrian	**94,41%**	**86,53%**	**90,30%**

TABLE I
Quantitative evaluation of the proposed algorithm

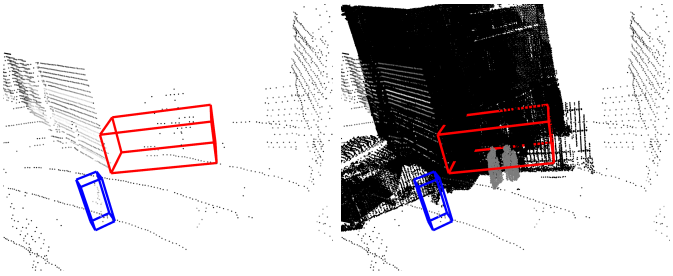


Fig. 8. False object prediction without (left) and with (right) map, Kálvin square, Budapest

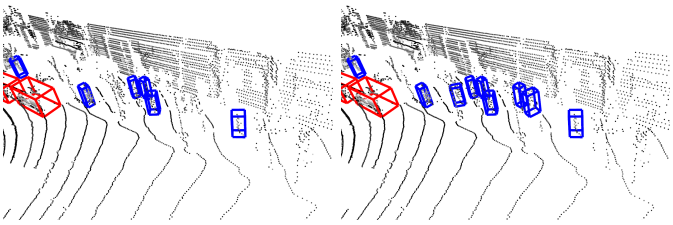


Fig. 9. Missed predictions without (left) and with (right) tracking information, Deák square, Budapest

Quantitative results are shown in Table I. We evaluated 10 sequences in two heavy traffic roads of Budapest, each sequence contains 30 consecutive Lidar time frames. We observed that using the background map, the average *precision* is improved with $26,52\%$ for vehicles and $9,38\%$ for pedestrians. Applying tracking, the *recall* is also improved with $12,84\%$ for vehicles and $14,34\%$ for pedestrians. By tracking, we also achieved a little increase in precision as well. Thus, we could obtain a balanced F-score value over 90% for all classes. This result is significantly better and produces more robust results than the frame-wise applied state-of-the-art PointPillars (Table II.) network.

Prediction confidence		**GT label**		
		Car	**Pedestrian**	**Cyclist**
Prediction	**Car**	$94,44\%$	$4,17\%$	$1,39\%$
	Pedestrian	0%	$97,12\%$	$2,88\%$
	Cyclist	0%	25%	75%

TABLE II
Confidence matrix of predicted classes

IV. Conclusion

We introduced a novel method to improve the accuracy and robustness of the current state-of-the-art object detection methods operating on purely sparse Lidar point clouds. First, we proved that using background city maps, the falsely predicted dynamic objects can be efficiently filtered in real time. Second, we showed that using tracking and past information of the dynamic objects, we can still find and track additional object that are temporarily missed by the object detector through a few time frames.

REFERENCES

[1] A.H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast Encoders for Object Detection from Point Clouds," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] M. Simon, S. Milz, K. Amende, and H-M. Groß, "Complex-YOLO: Real-time 3D Object Detection on Point Clouds," *ArXiv*, vol. abs/1803.06199, 2018.

[3] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely Embedded Convolutional Detection," *Sensors*, vol. 18, pp. 3337, 10 2018.

[4] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D Object Detection from Point Clouds," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7652–7660.

[5] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4490–4499.

[6] A. Borcs, B. Nagy, and C. Benedek, "Instant Object Detection in Lidar Point Clouds," *IEEE Geoscience and Remote Sensing Letters*, vol. PP, pp. 1–5, 05 2017.

[7] Z. Rozsa and T. Sziranyi, "Object detection from a few lidar scanning planes," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 4, pp. 548–560, Dec 2019.

[8] 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.

[9] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[10] C. Xiaozhi, M. Huimin, W. Ji, L. Bo, and X. Tian, "Multi-view 3D object detection network for autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6526–6534.

[11] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, 2017.

[12] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *ECCV*, 2018.

[13] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3D object detection from RGB-D data," 2017.

[14] K. Shin, Y.P. Kwon, and M. Tomizuka, "RoarNet: A Robust 3D Object Detection based on Region Approximation Refinement," *IEEE Intelligent Vehicles Symposium*, pp. 2510–2515, 2018.

[15] B. Nagy and C. Benedek, "3D CNN-Based Semantic Labeling Approach for Mobile Laser Scanning Data," *IEEE Sensors Journal*, vol. 19, no. 21, pp. 10034–10045, Nov 2019.

[16] P. Biber and W. Strasser, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," in *IEEE International Conference on Intelligent Robots and Systems*, 11 2003, vol. 3, pp. 2743 – 2748 vol.3.

[17] M. Magnusson, *The Three-Dimensional Normal-Distributions Transform - an Efficient Representation for Registration, Surface Analysis, and Loop Detection*, Ph.D. thesis, Örebro University, December 2009.

[18] H. Men, B. Gebre, and K. Pochiraju, "Color point cloud registration with 4D ICP algorithm," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 1511–1516.

[19] B. Nagy and C. Benedek, "Real-time point cloud alignment for vehicle localization in a high resolution 3D map," in *Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving at ECCV'18*, vol. 11129 of *LNCS*, pp. 226–239. Munich, Germany, 2019.