Real-time Vehicle Localization and Pose Tracking in High-Resolution 3D Maps

Örkény Zováthi[1,2] Balázs Pálffy[1,2] Csaba Benedek[1,2]

[1]Institute for Computer Science and Control (SZTAKI), Machine Perception Research Laboratory, Budapest, Hungary
[2]Pázmány Péter Catholic University, Faculty of Information Technology and Bionics, Budapest, Hungary
{lastname.firstname}@sztaki.hu

***Abstract*—In this paper we introduce a novel approach for accurate self-localization and pose tracking for Lidar and GPS-equipped autonomous vehicles (AVs) in high-density (more than 5000 points/m^2) 3D localization maps obtained through Mobile Laser Scanning (MLS). Our solution consist of two main steps: First, starting from a poor GPS-based initial position, we estimate the 3DoF pose (planar position and yaw orientation) of the ego vehicle by aligning its sparse (50-500 points/m^2) Lidar point cloud measurements to the MLS prior map, using a novel approach of matching static landmark objects of the scene. Second, to effectively deal with the lack of pairable objects in certain time frames (e.g. due to scene segments occluded by a large moving tram), we track the estimated 3DoF pose of the AVs by a Kalman filter. Comperative test are provided on roads with heavy traffic in downtown city areas with large (5-10 meters) GPS positioning errors. The proposed approach is able to reduce the location error of the vehicle by one order of magnitude and keep the yaw angle error around 1° during its whole trajectory without considerable drift, while running in real-time (20-25 Hz).**

***Index Terms*—lidar, localization, pose estimation, tracking**

I. Introduction

Autonomous vehicles (AVs) would be impossible without sensors as they demand accurate and real-time information about their surroundings, in order to safely navigate with only minor or without any human intervention. In this context, precise and robust localization and pose tracking of the ego vehicle is a key challenge. Although the Global Positioning System (GPS) based position information is usually suitable for helping human drivers, its accuracy is limited in inner-city areas with the presence of several large buildings, making it unfeasible for AVs. Instead, the accurate 3DoF pose (planar position and yaw orientation) of the moving vehicle can be calculated by aligning the onboard measurements (OBM) of the vehicles' visual or range sensors to available 3D city maps [1]. On the other hand, the onboard perception is often effected by temporal occlusions and motion artifacts, which demands robust and accurate tracking of the moving vehicle.

In this paper, we propose a joint solution for estimating and tracking the pose of a moving vehicle in a reference 3D city map. As OBM data, we utilize sparse (50-500 points/m^2) point clouds captured by Rotating Multi-beam (RMB) Lidar sensors, which measure directly the range information, and efficiently perform under different illumination and weather conditions, offering accurate point cloud streams with a large field of view in real-time (5-20 Hz). Beyond the RMB Lidar sensor, we also assume that the vehicle carries a GPS receiver, however, we expect that in various city regions the GPS measurements might be inaccurate due to lack of high quality navigation signals [2], [3], providing position errors up to several (5-10) meters. As reference map, we utilize 3D measurements of recent mobile laser scanning (MLS) platforms equipped with time synchronized Lidar sensors and navigation units, as they provide dense (more than 5000 points/m^2), accurate and feature rich point clouds precisely registered to a geo-referenced global coordinate system [4]. The addressed scenario is displayed in Fig. 1.

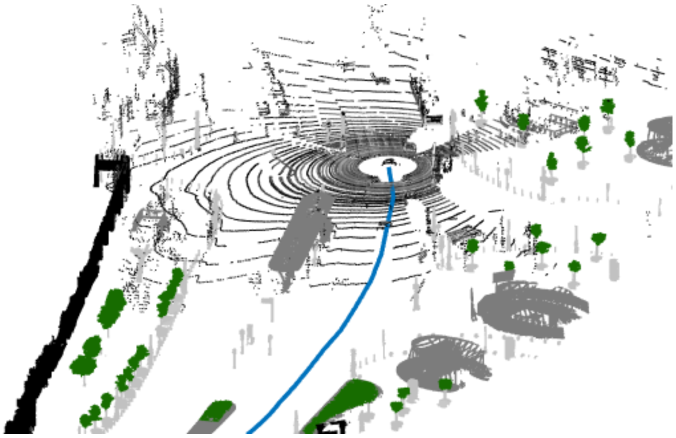

Fig. 1. Aim of the paper: Tracking the trajectory of a moving vehicle (displayed with blue) by positioning the corresponding RMB Lidar measurements to a global MLS map.

In our work, we only rely on the vehicles' onboard RMB Lidar and inaccurate GPS position measurements, and we do not use any Inertial Measurement Units (IMUs). We propose two significant contributions to improve the state-of-the-art:

1) Fast and accurate *pose estimation* by matching static objects of the sparse RMB data frames to the prior MLS map, starting from a poor GPS-based position estimation

This work was funded by the National Research, Development and Innovation (NRDI) Office within the framework of the Autonomous Systems National Laboratory, and the project no. SZTAKI-NVA-01 under the TKP2021 funding scheme. The work of Ö. Zováthi was partially funded by the KDP-2020 Cooperative Doctoral Program (KDP-977852) and the ÚNKP-21-3 New National Excellence Program (ÚNKP-21-3-I-PPKE-14) from the source of the NRDI Fund.

(5-10 meters error)

2) Efficient and robust *tracking* of the vehicle movement by fusing the estimated poses in a constant velocity model-based position-only-measured (POM) Kalman filter

II. Related work

Robot or vehicle localization given a prior map is a hot topic in the literature [5]. In general, we can distinguish methods addressing global localization or pose estimation (when no prior pose is available), and pose tracking, when the vehicle starts from a known pose which is updated over time.

In the field of pose tracking, the majority of existing methods uses visual odometry information [6] by incrementally aligning consecutive Lidar point cloud measurements mostly using a variant of the Iterative Closest Point (ICP) [7] algorithm, and determining the relative pose of the moving vehicle at each iteration. Some methods integrate visual odometry with IMU sensors for more accurate results [8], [9]. As these methods integrate small incremental motions over time, they are bound to drift-effect in large-scale scenarios, which is typically reduced by loop closure detection.

Tackling the problem of global localization in large-scale urban environment with poor GPS coverage, the pose estimation problem can be described as a point cloud registration between the OBM and map data, starting from a poor initial alignment [10]. Among the point-level registration methods, the ICP-based methods need sufficient initial alignment, which can not be performed in poor GPS-covered areas. Plenty of methods apply keypoint based data matches [10], [11], however, these approaches are sensitive to the characteristics of the alignable point clouds. [12] solves ICP-based data mapping at a higher level, by explicitly matching segments across point cloud frames. This method proved to be efficient for matching consecutive RMB Lidar point clouds, however the computational time remained a few seconds per point cloud pair, which significantly increases using dense MLS data. Instead of using raw point clouds, [2] integrates a range-image based observation model into a Monte Carlo localization framework to estimate the 3DoF pose of a vehicle.

The closest solution to our approach is the SegMap [13] technique, which detects wall segments of the corresponding OBM and map regions, and describes these regions using geometric or data-driven features. On the other hand, similar features are hard to extract for segments captured with different sensor modalities, while the lack of close wall segments (i.e in large square regions, or due to occlusions) can mislead this method as well. Following a different approach, we aim to match pillar-like objects (poles, traffic lights, signs, etc.) of the scenes in real-time, which sorts of objects are typically present in urban regions. In addition, we also introduce an efficient pose tracking method to tackle with featureless measurement frames (e.g. due to temporal occlusions by moving objects).

III. The proposed method

We propose a real-time, robust pose estimation and tracking technique for AVs using sparse RMB Lidar and low accuracy GPS measurements, with respect to prior high density localization maps obtained from MLS point clouds. As a preliminary step, we efficiently extract and describe the static objects of the MLS data by their geometric and semantic properties. Next, for estimating the optimal pose of the vehicle, we apply an object-based matching transform between the pillar-like objects detected from the RMB Lidar data and the extracted static objects of the MLS map. Here we adopt a generalized Hough transform based algorithm, applied first for fingerprint minutia matching [14], which is able to find a robust transformation between two point sets even if the size of the point clouds are significantly different. Finally, from the optimal transformation – assuming planar movements – we extract the 3DoF pose of the vehicle (x,y,θ), which parameters we track by a constant velocity model-based position-only-measured (POM) Kalman filter, effectively compensating temporal occlusions.

A. Extraction of static objects from the MLS prior map

The raw, noisy and dense MLS point clouds may include several measurement segments which do not contain relevant information for vehicle navigation (e.g. many ground points, regions of large building facades, re-located or dynamic objects). First, to reduce the map's size and redundancy, we semantically segment the raw MLS point clouds and keep only regions of static object classes (*pillar-like*, *street furniture*, *vegetation* and *facade*), whose appearance do not vary significantly over time, and should be also present in *empty* street segments. Note that this step can be performed in an offline pre-processing stage, either in manual or in automatic manner [15]. Next, we extract object samples from these static class regions by 3D Euclidean clustering [16] and we describe each static landmark object with the following parameters:

- Global coordinates (x,y,z) of the object's 3D bounding box corner points (24 parameters)
- Yaw orientation of the object (1 parameter)
- Label of the object class (1 parameter)
- Size of the object's 3D bounding box: width, depth, height and volume (4 parameters)

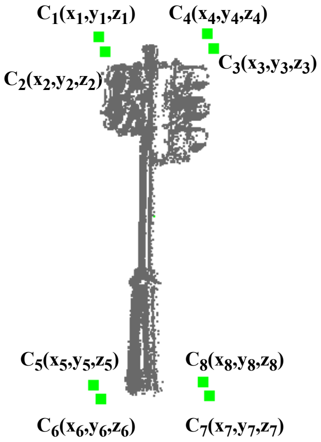

Fig. 2. Extracted corner points of a *pillar-like* object sample from the MLS map.

With this feature extraction, we reduce the number of points describing an MLS object from a few tens of thousands to 30 parameters (see Fig. 2), enabling quick access to the map objects.

B. Pose estimation of the moving vehicle

Next, we estimate the global pose of the vehicle in the MLS map by matching its perceived OBM data to the reference model. As a key idea, instead of aligning the original point clouds, we separate *pillar-like* objects and *wall segments* in the RMB Lidar frames (see Fig. 3) by a grid-based fast scene segmentation technique [17], and we aim to match the detected pillars of each frame to the previously extracted static landmarks from the MLS map. Here, as a key challenge, we should expect many falsely detected object candidates in the automatically segmented RMB Lidar frames [18] (e.g. traffic participants, partially occluded objects), which may result in a possibly large ratio (up to 80%) of outlier matches. To handle their effect, we turned to the voting schema of the generalized Hough transform, motivated by complex assignment problems such as fingerprint minutiae matching [14].

First, we use the available, usually notably inaccurate GPS signal for initially positioning the actual RMB point cloud frame's center in the global coordinate system of the MLS map. Assuming that the local ground height information is available from the map, we search for an optimized rigid transformation with a 2D translation and a rotation component between the point clouds. The translation component $(\Delta x, \Delta y)$ compensates for the originally unknown position error of the GPS sensor, while – based on experiments – the rotation component can be approximated by the yaw rotation angle $(\Delta\theta)$. In summary, we model the optimal transform as follows:

$$\mathbf{T}_{\Delta x,\Delta y,\Delta\theta}\begin{pmatrix}x\\y\end{pmatrix} = \begin{bmatrix}\cos\Delta\theta & -\sin\Delta\theta\\ \sin\Delta\theta & \cos\Delta\theta\end{bmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix}$$

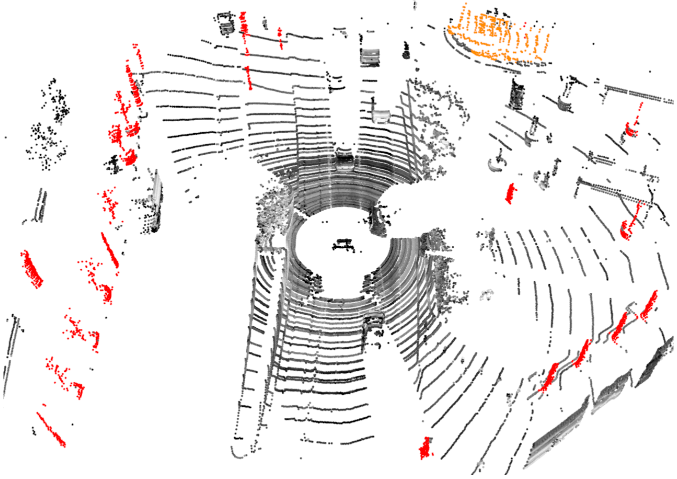

Fig. 3. A sample RMB Lidar frame. Detected *pillar-like* objects [17] are displayed with red, while *wall segments* are showed by orange.

Next, we find the optimal parameters of this $\mathbf{T}_{\Delta x,\Delta y,\Delta\theta}$ transformation via extracted keypoint pairs. First, we collect the extracted pillar-like *RMB* Lidar and *MLS* map objects into two sets denoted by $\mathcal{O}_{\text{RMB}}$ and $\mathcal{O}_{\text{MLS}}$. Then, we describe each *RMB* object candidate by the same 8 keypoints (i.e. bounding box corners) as previously with the map landmarks, and adopt the generalized 3D Hough transform to determine the optimal transformation between the *RMB* and *MLS*-based keypoint sets, by a voting algorithm.

First, for limiting the parameter space, we allow maximum offsets of $\pm 60^\circ$ for the yaw rotation $(\Delta\theta)$ and ± 12 meters for planar translation (Δx and Δy) to tackle with the GPS inaccuracies. As required by the Hough schema, we discretize the transformation space between the minimal and maximal allowed values of each parameter, using 0.4 meters for the translation components and 0.5° degrees for rotation. This setup enables both reasonably accurate resolution and quick computation. Next, we allocate a three-dimensional array $A[\Delta x, \Delta y, \theta]$ with zero initial values to summarize the votes of the possible parameter triplets.

During the voting process, we search for possible keypoint correspondences between all *pillar-like* object pairs $(o_{\text{RMB}}, o_{\text{MLS}}) \in \mathcal{O}_{\text{RMB}} \times \mathcal{O}_{\text{MLS}}$. For a given keypoint couple k_{RMB}, k_{MLS} we add a vote for all possible $\mathbf{T}_{\Delta x,\Delta y,\Delta\theta}$ transforms, which map k_{RMB} to k_{MLS}. More specifically, we iterate over all the discrete $\Delta\theta$ values, and for each $\Delta\theta'$ we rotate k_{i3D} by $\Delta\theta'$ first, and calculate the corresponding translation vector $[\Delta x', \Delta y']^T$ as follows:

$$\begin{bmatrix}\Delta x'\\ \Delta y'\end{bmatrix} = k_{\text{MLS}} - \begin{bmatrix}\cos\Delta\theta' & -\sin\Delta\theta'\\ \sin\Delta\theta' & \cos\Delta\theta'\end{bmatrix} k_{\text{i3D}}$$

Next, we vote for the calculated $\mathbf{T}_{\Delta x',\Delta y',\theta'}$ transform so that we increase the $A[\Delta x', \Delta y', \theta']$ element of the accumulator array by one. After iterating through the whole parameter space, the optimal $T_{\Delta x^*,\Delta y^*,\theta^*}$ transform can be extracted as follows:

$$(\Delta x^*, \Delta y^*, \Delta\theta^*) = \underset{\Delta x,\Delta y,\Delta\theta}{\operatorname{argmax}}\ A[\Delta x, \Delta y, \Delta\theta]$$

Finally, we make an acceptance decision of calculated transform based on a minimum number of votes:

$$\mathcal{A}(\mathbf{T}) = \text{true if and only if } A[\Delta x^*, \Delta y^*, \Delta\theta^*] > t$$

We experimentally set $t = 5$. In case of an accepted transform, the estimated pose of the vehicle can be calculated as follows:

$$x^* = x_{\text{GPS}} + \Delta x^* \tag{1}$$
$$y^* = y_{\text{GPS}} + \Delta y^* \tag{2}$$
$$\theta^* = \theta_{\text{GPS}} + \Delta\theta^* \tag{3}$$

C. Pose tracking

Although we experienced that the above pose estimation method works robustly even in sparse scenes, covering only a few (5-10, depending on the scene characteristics) pillar-like objects, its accuracy is limited in scenarios without a sufficient number of matchable landmark object pairs. To overcome this problem, we track the estimated pose parameters by a constant velocity (CV) model based Kalman filter [19], whose true state vector is defined as follows:

$$\mathbf{x_t} = \begin{pmatrix} x_t & y_t & \theta_t & v_{xt} & v_{yt} & w_{\theta t} \end{pmatrix}^T,$$

where x_t, y_t and θ_t are the planar position and yaw orientation and v_{xt}, v_{yt} and $w_{\theta t}$ are the velocities of the vehicles, respectively. As the CV model assumes permanent velocity within a short observation period, the model's dynamics can be considered as follows:

$$\mathbf{x_{tk}} = \mathbf{\Phi} \mathbf{x_{tk-1}} + \mathbf{w_k},$$

where $\mathbf{x_{tk}}$ denotes the true state at time kT, T is the sampling interval determined by the applied RMB Lidar sensor's spin rate ($T \approx 66.67$ ms used), $\mathbf{w_k}$ is the process noise, and $\mathbf{\Phi}$ is the transition matrix from kT to $(k+1)T$, which is defined as:

$$\mathbf{\Phi} = \begin{pmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Next, we integrate the Lidar-perceived planar position and yaw orientation values (Sec. III-B) into this model dynamics, so that after each *accepted* transformation ($\mathcal{A}(\mathbf{T}) =$ true), we use the estimated poses (x^*,y^*,θ^*) defined by eq. (1)-(3) as new measurements as follows:

$$\mathbf{z_k} = \mathbf{H} \mathbf{x_{tk}} + \mathbf{v_k},$$

where $\mathbf{z_k} = (x^*, y^*, \theta^*)$ denotes the measurement vector, $\mathbf{H}$ denotes the measurement matrix, and $\mathbf{v_k}$ is the measurement noise. As our model is position-only-measured, $\mathbf{H}$ is:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Finally, we sequentially predict and estimate the state vectors based on the previous state values and measurements via the Kalman filter equations:

$$\mathbf{\tilde{x}_k} = \mathbf{\Phi} \mathbf{\hat{x}_{k-1}} \tag{4}$$

$$\mathbf{\hat{x}_k} = \mathbf{\tilde{x}_k} + \mathbf{K_k}(\mathbf{z_k} - \mathbf{H}\mathbf{\tilde{x}_k}) \tag{5}$$

where $\mathbf{\tilde{x}_k}$ and $\mathbf{\hat{x}_k}$ are the predicted, respectively estimated state vectors by the Kalman filter, while $\mathbf{K_k}$ denotes the Kalman gain that minimizes the errors in the estimated positions and velocities. If the Lidar-based estimation of the pose transformation is *not accepted* (i.e. $\mathcal{A}(\mathbf{T}) =$ false in Sec. III-B), we only execute the prediction step (eq. (4)), while the state vector re-estimation (eq. (5)) is skipped.

IV. Evaluation

We evaluated the proposed pose tracking technique on challenging scenarios from inner-city areas on real measurement sequences, in a pathway of around 0.6 km. The RMB Lidar data was collected by a Velodyne HDL-64E sensor with 15 Hz spin rate, while the MLS reference map was generated using multiple road measurements of a Riegl VMX-450 laser scanner. During quantitative evaluation, we compared the results of the proposed model to available Ground Truth (GT) information, generated through manually aligning the RMB Lidar frames to the global MLS point clouds. As evaluation metrics, we calculated the mean absolute error (MAE) for each estimated pose parameter: position errors along the x-axis (d_x) and y-axis (d_y), and the yaw orientation error (d_θ). For two-dimensional position error, we also calculated the average Euclidean distance (D) between the estimated and GT planar positions (x,y) of the vehicle. For comparative experiments, we developed a baseline, only GPS-based Kalman filter and also implemented a Lidar-based frame-wise pose estimation method [10] without tracking, besides the proposed method. The overall numerical results are summarized in Table I.

TABLE I
Quantitative summary of the pose parameter errors

Error metric Method	d_x [m]	d_y [m]	D [m]	d_θ [°]
Raw GPS	3.4214	3.2428	4.9288	29.768
GPS-only Kalman filter	1.5184	0.7691	1.8004	2.1962
Frame-wise pose estimation	0.5852	0.5628	0.9029	1.6131
Proposed method	**0.4850**	**0.4349**	**0.7304**	**1.0592**

Fig. 4 displays a sequence of error rates calculated for 180 consecutive time frames from a test scenario, covering a driven path of approximately 300 meters. During this drive, frames containing large moving objects (tram, bus) with significant occlusions were recorded, while the trajectory of the ego vehicle was turning to left of around 20 degrees. In the pose estimation step, 20% of the calculated transforms were

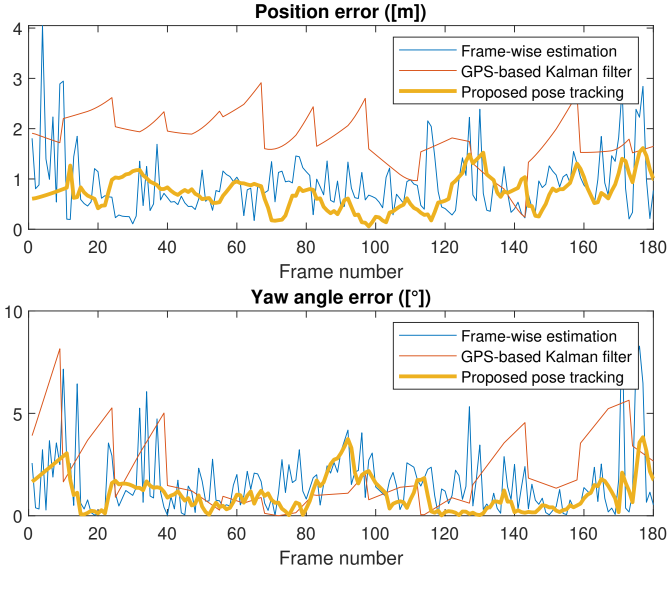

Fig. 4. Position (D) and orientation (d_θ) error of the tracked poses versus Ground Truth information, demonstrating the superiority of the proposed method.

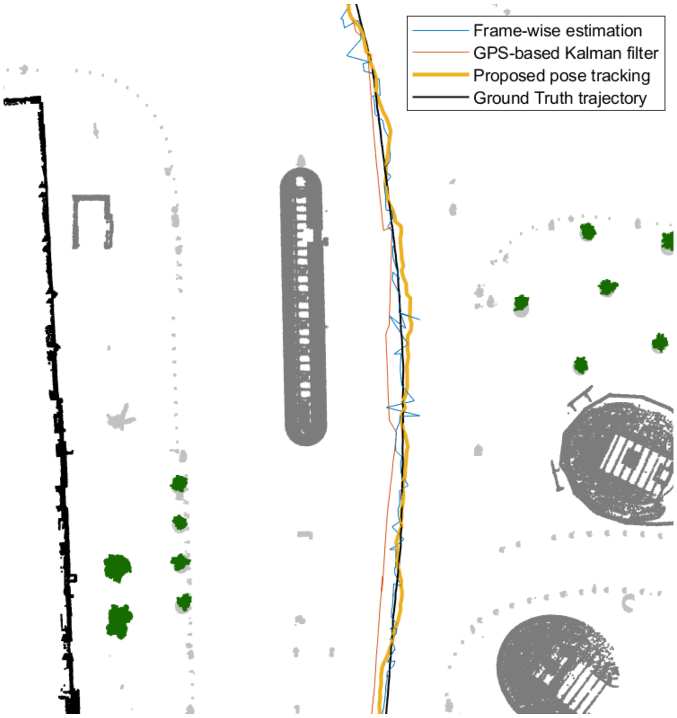

Fig. 5. Estimated trajectories of the different methods and the Ground Truth path.

dropped due to the lack of enough pairable objects. The planar trajectories estimated by the different methods can be also visually compared in Fig. 5. For the sake of demonstrating the proposed method's accuracy, a video is available at the following link[1].

From the results of Table I and Fig. 4, we can conclude that using the frame-wise Lidar-based pose estimation we can significantly improve the accuracy of the GPS-only positioning, reducing the average location error from around 5 meter to 1 meter. However, the value of the error is still strongly fluctuating frame-by-frame without considering the vehicle dynamics (see Fig. 4). The proposed joint method can largely overcome this artifact by efficiently integrating the Lidar-perceived pose information into the Kalman filter based dynamic model of the moving vehicle, achieving an average global position error of around 70 centimeters and orientation error around 1° in approximately only 40-50 milliseconds.

Regarding the computation time of the whole workflow, the proposed method is fully able to operate in real-time, as it runs with 20-25 Hz on a desktop computer. The presented result may also serve as a fast and accurate initial alignment for an ICP-based [7] point level registration algorithm (which can also run in real time with parallel implementation), that may decrease the location error to a few centimeters.

V. Conclusion

We proposed a robust, real-time method for estimating and tracking the 3DoF pose of a moving vehicle based on its Lidar and inaccurate GPS measurements against a prior MLS map. In our work, we efficiently integrated the Lidar-based pose information into a Kalman filter based dynamic vehicle model, and qualitatively and quantitatively demonstrated the proposed methods accuracy and usability in a large-scale urban scenario. In the future, we aim to extend the dynamic model and integrate visual odometry information [6] by incrementally registering the captured Lidar frames.

[1] Demonstration video: https://youtu.be/-7GijZzXMIA

References

[1] H. G. Seif and X. Hu, "Autonomous driving in the iCity—HD maps as a key challenge of the automotive industry," *Engineering*, vol. 2, no. 2, pp. 159–162, 2016.

[2] X. Chen, I. Vizzo, T. Läbe, J. Behley, and C. Stachniss, "Range image-based lidar localization for autonomous vehicles," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5802–5808, 2021.

[3] D. Wilbers, C. Merfels, and C. Stachniss, "Localization with sliding window factor graphs on third-party maps for automated driving," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5951–5957.

[4] H. Zheng, R. Wang, and S. Xu, "Recognizing street lighting poles from mobile lidar data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 1, pp. 407–420, 2017.

[5] D. Rozenberszki and A. Majdik, "Lol: Lidar-only odometry and localization in 3D point cloud maps," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4379–4385, 2020.

[6] J. Zhang and S. Singh, "LOAM: lidar odometry and mapping in real-time," in *Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014*, 2014.

[7] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, pp. 119–152, 2005.

[8] S. Scherer, J. Rehder, S. Achar, H. Cover, A. Chambers, S. Nuske, and S. Singh, "River mapping from a flying robot: state estimation, river detection, and obstacle mapping," *Autonomous Robots*, vol. 33, no. 1, pp. 189–214, Aug 2012.

[9] W. Burgard, O. Brock, and C. Stachniss, *Map-Based Precision Vehicle Localization in Urban Environments*, pp. 121–128, 2008.

[10] Ö Zováthi, B. Nagy, and Cs. Benedek, "Point cloud registration and change detection in urban environment using an onboard Lidar sensor and MLS reference data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 110, pp. 102767, 2022.

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

[12] B. Douillard, A. Quadros, P. Morton, J. P. Underwood, M. De Deuge, S. Hugosson, M. Hallström, and T. Bailey, "Scan segments matching for pairwise 3D alignment," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 3033–3040.

[13] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: 3D segment mapping using data-driven descriptors," in *Robotics: Science and Systems (RSS)*, 2018.

[14] N. K. Ratha, K. Karu, S. Chen, and A. K. Jain, "A real-time matching system for large fingerprint databases," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 799–813, Aug 1996.

[15] B. Nagy and Cs. 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] 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 2011, pp. 1–4.

[17] A. Börcs, B. Nagy, and Cs. Benedek, "Instant object detection in lidar point clouds," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 7, pp. 992–996, 2017.

[18] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3D point clouds," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5266–5272.

[19] K. Saho, *Kalman Filter for Moving Object Tracking: Performance Analysis and Filter Design*, 02 2018.