2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Monocular Localization in HD Maps by Combining Semantic Segmentation and Distance Transform

Jan-Hendrik Pauls*[1], Kürsat Petek*[1], Fabian Poggenhans[2], and Christoph Stiller[1,2]

Abstract— **Easy, yet robust long-term localization is still an open topic in research. Existing approaches require either dense maps, expensive sensors, specialized map features or proprietary detectors.**

We propose using semantic segmentation on a monocular camera to localize directly in a HD map as used for automated driving. This combines lightweight, yet powerful HD maps with the simplicity of monocular vision and the flexibility of neural networks.

The major challenges arising from this combination are data association and robustness against misdetections. Association is solved efficiently by applying distance transform on binary per-class images. This provides not only a fast lookup table for a smooth gradient as needed for pose-graph optimization, but also dynamic association by default.

A sliding-window pose graph optimization combines single image detections with vehicle odometry, smoothing results and helping overcome even misclassifications in consecutive frames.

Evaluation against a highly accurate 6D visual localization shows that our approach can achieve accuracy levels as required for automated driving, being one of the most lightweight and flexible methods to do so.

I. Introduction

Almost all approaches for automated driving still heavily rely on maps. In order to use map information, a vehicle needs to determine its position within the map. This task is called localization and requires precision up to few centimeters in lateral direction. For research and as reference for evaluation, expensive RTK-GNSS systems are coupled with high-precision IMUs. While this achieves the goal in terms of accuracy, these solutions do not scale well from an economic point of view. Also, their performance degrades in GNSS-denied areas like tunnels.

As an alternative, landmark-based localization has been established and recent publications were able to achieve comparable accuracy [1], [2]. As localization is supposed to be independent of daytime and weather, but also robust against structural changes, landmarks have to be chosen appropriately. Furthermore, localization is supposed to work with more than one kind of sensor to have a redundant and portable solution. Finally, landmarks need to be compact in memory. This requires landmarks that are sparse, rarely changing and robustly detectable with multiple sensor modalities.

*Authors with equal contribution to this work.
[1]Institute of Measurement and Control Systems, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. {pauls, stiller}@kit.edu
[2]Intelligent Systems and Production Engineering, FZI Research Center for Information Technology, Karlsruhe, Germany. {poggenhans, stiller}@fzi.de

A. HD Maps

For automated driving, HD maps [3] that contain information required for enhanced perception or prediction of other traffic participants have become the de facto standard.

The idea to directly use HD map elements, such as lanes, curbs, lane markings, traffic lights or traffic signs, for localization, is not new [4], [5]. They are not only sparse and more robust against changes than image descriptors that change with seasonality or weather conditions, they are also needed for other driving functions, anyways. Thus, they do not need extra memory or maintenance. Moreover, it also allows to share the maps with a humanly verifiable meaning across vehicles, fleets or even manufacturers. Additionally, it eliminates the need to properly align localization layer and other map elements [6].

The disadvantage that HD map elements are not dense or rich enough to provide sufficient localization accuracy will be gone with the increasing demand on HD map features.

B. Semantic Segmentation

For detecting HD map elements, previous approaches [1], [4] used proprietary detections from camera or lidar sensors. While this is valid, those detectors are rarely made publicly available. Approaches like [7] are an exception, but poles are not necessary for automated driving functions other than localization, thus, representing additional landmarks.

In this paper, we propose using convolutional neural networks (CNNs) trained for semantic segmentation tasks as an alternative. Not only are they widely available for research use [8], even with pretrained weights, we also show an easily reproducible approach to solve the association problem that stands between landmark detection and pose optimization.

Bounding box [9]–[11] or instance-level detections could be an alternative to pixel-wise segmentation. However, lanes, curbs and solid lane markings often to not fit well to such a spatially restricted representation.

C. Contribution and Outline

This paper contains three contributions. First, we propose to use CNNs that are trained for semantic segmentation in order to overcome spatial limitations of previously often used bounding box detectors. This allows to detect all kinds of map elements that are typically contained in sparse, shareable, sensor-independent HD maps for automated driving.

The second contribution is the use of distance transform in order to solve the association problem for dense semantic information on a pixel level. Additionally, this makes association inherently dynamic.

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

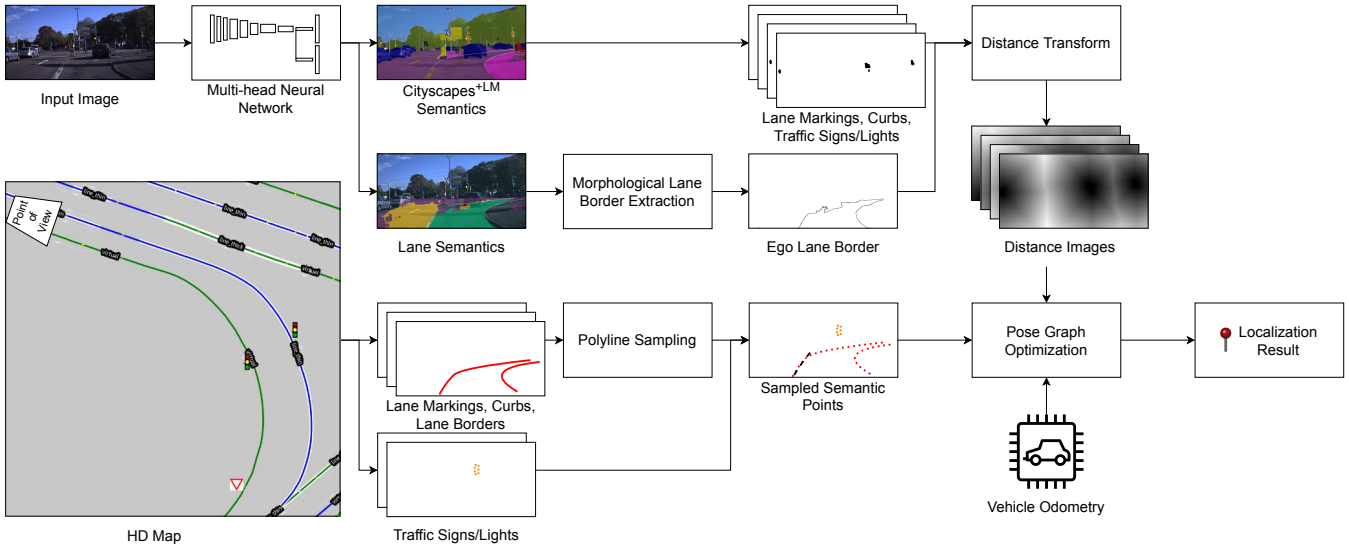


Fig. 1: Overview of our approach. Camera images are fed into a multi-head, real-time capable neural network that predicts enhanced Cityscapes^{+LM} and lane semantics, the latter being post-processed morphologically (Section III). Distance transform is applied on each relevant semantic slice of the output tensor (Section IV). Semantically corresponding elements of the HD map are gated and polylines are sampled to obtain semantic point landmarks in 3D (Section V). Finally, combining this semantic localization with vehicle odometry in a pose graph optimization (Section VI) yields a smooth and robust localization result.

Finally, we show how to compose this semantic information with a standard vehicle odometry into a robust pose graph that overcomes the weaknesses of semantic localization using single images and enables accurate 6D localization in lightweight HD maps with widely available hardware and software.

The next section puts our contribution in the context of related work. Section III explains the output of the CNN used for detection and how it is post-processed. In Section IV, association using distance transform is described. The resulting pose optimization problem for a single image is formulated in Section V, the subsequent pose graph and its optimization in Section VI. The final two sections contain evaluation, conclusion and future work.

II. Related Work

Various approaches proposed localization in HD maps using proprietary or handcrafted detectors for road features [4], [12], [13], such as lane markings or curbs, or traffic signs [14], often involving error-prone inverse perspective mapping in case of flat landmarks. While the results performed comparable, the detectors are hard to reproduce or limited to only a single class of semantic landmarks, like traffic signs.

Other approaches used additional landmarks like poles [2], [7], [15], [16] or facades of buildings [2] or simply assumed some kind of detector [1]. This is valid, but means extra memory and maintenance compared to an HD map with only lanes and regulatory elements.

The third approach is to use deep learning-based approaches to detect semantic landmarks. [9]–[11] proposed bounding box detectors in 2D and 3D, respectively, to map and recognize objects. This, however, does only work for spatially bounded objects. Sharing the idea of using a distance transform with our approach, but with a custom, learned framework and a dense output, lane-level accurate localization in 2D was proposed by [5]. While the customization can easily adapt to new data, it is not as available as simpler neural networks and similar performance in 3D has to be proven.

Closest to our approach is the idea to extract and store semantically labeled points and curves in 3D and use them for localization [17]. However, their error functions are not as generic as a simple distance transform. Also, they use 3D points and curves that only partially resemble actual objects and, thus, are neither as ubiquitous nor as shareable across sensor modalities or vehicles as HD maps are.

III. Semantic Segmentation and Post-processing

For map feature detection, we use a modified ResNet-38 CNN that has multiple detection heads [18].

One head predicts enhanced Cityscapes [19] classes, hereafter called Cityscapes^{+LM}. Enhanced means that we extended the Cityscapes dataset by an additional lane markings class, containing all kinds of lane markings.

The second head predicts lanes. This head works particularly well for the ego lane. Here, we use the approach by [20].

Note that the whole network is optimized to run on our measurement vehicle. For offline use or to easily reproduce our results, the significantly slower Seamseg approach [21] would be an interesting alternative. This is due to their use of the Mapillary Vistas dataset [22] which contains HD map-related classes such as road surface, lane markings, curbs, traffic signs or traffic lights.

The semantic prediction output of a CNN is one class $c(\mathbf{u}) \in \mathbb{C}$ for each pixel $\mathbf{u} = (u, v)$ with image coordinates u and v. Usually, $c(\mathbf{u})$ is determined as the class which has the highest activation $a_k(\mathbf{u})$:

$$c(\mathbf{u}) = \underset{k=1,\dots,|\mathbb{C}|}{\arg\max}(a_k(\mathbf{u}))$$

Using this one-hot encoding, the predicted output can be seen as a $M \times N \times \mathbb{C}$ tensor T where exactly one entry along the $\mathbb{C}$ dimension is 1 while all others are 0.

For localization, only those classes which correspond to map elements matter. In our case, these are curbs (C), lane borders (LB), lane markings (LM), traffic lights (TL) and traffic signs (TS), defining the set of classes of interest $\mathbb{C}_I \subset \mathbb{C}$. Slicing T to only those classes defines the tensor of interest, T_I.

Our network cannot directly detect lane borders, but for localization, they are far more informative than the lane area. Thus, we subtract the eroded LB slice, T^-_{LB}, from the dilated LB slice, T^+_{LB}, to morphologically extract the borders of the ego lane: $T_{\text{LB}} = T^+_{\text{LB}} - T^-_{\text{LB}}$.

IV. Distance Transform

To solve the association problem between dense image pixels and sparse, vectorial map elements, we propose applying distance transform on a per-class image level. This spreads the information of the few pixels belonging to the detections across the whole image, later allowing to use this information for optimization.

Each slice of the tensor of interest T_I corresponding to a class c can also be seen as a binary image B_c as required for a distance transform.

$$B_c(\mathbf{u}) = \begin{cases} 1 & \text{if } c(\mathbf{u}) = c \\ 0 & \text{else} \end{cases}$$

Distance transform can be used to transform B_c into a distance image D_c of equal dimensions, but continuous pixel values.

$$D_c(\mathbf{u}) = \begin{cases} 0 & \text{if } B_c(\mathbf{u}) = 1 \\ \min\limits_{\tilde{\mathbf{u}}: B_c(\tilde{\mathbf{u}})=1} \|\mathbf{u} - \tilde{\mathbf{u}}\| & \text{else} \end{cases}$$

Using OpenCV [23] as software and L_2 as norm, a distance image D_c for each class of interest can be created efficiently.

V. Semantic Localization

Next, map features of each class $c \in \mathbb{C}_I$ are projected into the corresponding distance images D_c using an initial pose.

As HD map format, we use Lanelet2 [3] (see the HD map in Figure 1 for an example). In a Lanelet2 map, there is a physical element layer which can be annotated with types. Physical elements are either points or linestrings/polylines made up by a sequence of points, all in 3D. A left and a right polyline serve as borders of a so-called lanelet, a short piece of a lane that has contiguous semantics of both borders.

For TL/TS landmarks, we use the four or eight points defining their shape as landmarks. Lane boundaries contain only very sparse points. For lane markings or boundaries, we sample landmarks as points along the polyline using sampling distance $d_s = 0.05\,\text{m}$, converting a sparse, vectorial HD map into denser, categorical point landmarks $\mathbf{l} \in \mathbb{R}^3$ with an additional semantic class $c(\mathbf{l})$. The same approach can be used for arbitrary shapes and the sampling distance offers a trade-off between accuracy and optimization speed.

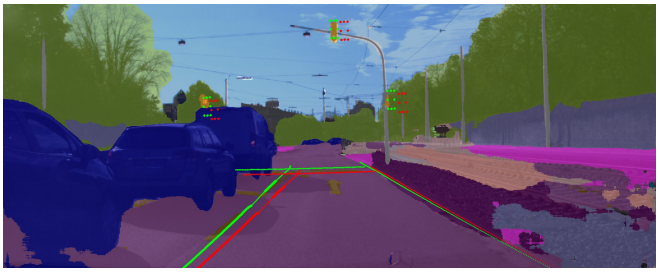


Fig. 2: Example of the semantic single image localization. Map elements of different types are reprojected into the image based on an erroneous initial pose (red). By applying our optimization on a single image, new reprojections are obtained (green).

Next, using a camera model and an initial pose $\boldsymbol{p}_0 \in \text{SE}(3)$, all landmarks within or close to the (initial) field of view are projected into the distance images D_c. This allows, already for a single image, to optimize the camera pose $\boldsymbol{p} \in \text{SE}(3)$ by minimizing the distances of all landmarks $\mathbf{l}$. This is expressed by the cost function $J(\boldsymbol{p})$.

$$J(\boldsymbol{p}) = \sum_{\mathbf{l}} \rho\Big(\big(D_{c(\mathbf{l})}(\boldsymbol{p}^{-1}\mathbf{l})\big)^2\Big)$$

The idea is that a pose is optimal if all landmarks lie perfectly in image patches of the corresponding semantic class. For non-optimal cases, each landmark has to move to the next suitable image patch. This information can be extracted by interpolating the distance images to derive a smooth gradient.

Hence, the interpolated distance image can be used as a fast lookup table that is computed only once and used in each optimization step. Another advantage is that this lookup makes the association between landmark and image patch dynamic, i.e. it can change in each optimization step without extra work.

Bi-cubic interpolation is done using Ceres [24]. Finally, also using Ceres, the the sparse optimization problem can be solved by Cholesky factorization [25].

For outlier rejection, we apply Tukey's biweight loss as robust loss function ρ with variable width. As some map elements, like traffic signs and traffic lights, are typically detected from farther away than lane markings or lanes boundaries, the width depends on the semantic class and ranges from 80 to 120 pixels.

For distinct locations, like when approaching intersections, this optimization based on a single image already leads to a unique pose. In many places, however, the pose is only laterally constrained given the very sparse map information and the limited range of view.

VI. Pose Graph Optimization

Semantic segmentation – in particular for real-time capable neural networks – is hardly perfect. Significant areas are misclassified, making the single-image problem to converge to a solution that seems acceptable for the semantic labeling, but does not fit well to the actual image.

Additionally, long, straight roads without traffic signs, intersections or other landmarks with information for longitudinal localization make the localization problem underconstrained: Mainly longitudinal position, but to some degree also height, roll and pitch angles are not well-observable or well-determined.

To solve both problems and provide a stable, smooth trajectory, we couple single-image localization with vehicle odometry (VO) in a sliding-window pose graph optimization. As VO, we obtain high-frequency measurements of longitudinal velocity and yaw rates. Both information is available on most vehicle interfaces.

We combine VO and semantic image (SI) measurements in a pose graph by creating a pose for each of the most recent 20 images. VO is integrated between two SI poses to obtain a motion estimate between consecutive frames.

For VO integration, the vehicle is abstracted as a point mass, which can move in longitudinal direction with constant velocity v and rotate around the up axis with constant yaw rate ω. Thus, the vehicle movement is modeled using a modified constant turn rate and velocity (CTRV) model [26]. The non-linear update of the partial 2D pose $\tilde{\boldsymbol{p}} = (x, y, \theta)$ in the vehicle frame is given as follows.

$$\Delta\tilde{\boldsymbol{p}} = \begin{cases} \begin{pmatrix} \frac{v}{\omega}\sin(\omega\Delta T) \\ \frac{v}{\omega}(1-\cos(\omega\Delta T)) \\ \omega\Delta T \end{pmatrix} & \text{if } \omega \geq \omega_{min} \\ \begin{pmatrix} v\Delta T \\ 0 \\ \omega\Delta T \end{pmatrix} & \text{if } \omega < \omega_{min} \end{cases}$$

ΔT denotes the temporal difference between two consecutive frames and ω_{min} allows to avoid singularities.

This partial pose update is then transformed to the camera frame and complemented with a weak regularization on height, pitch and roll angle as a simplified 6D motion model.

VII. Evaluation

The approach is evaluated on two urban scenarios in the cities of Karlsruhe (scenario 1) and Ludwigsburg (scenario 2). As even our reference-grade GNSS/IMU solution tends to degrade in urban environments, we instead use a multi-camera visual localization system [27], [28] that uses DIRD image features [29], vehicle odometry and GNSS. Additional pole landmarks are used for georeferencing [6] for scenario 1 where the map is supplied by a third-party company. This solution provides smooth, high-precision localization in 6D with only few centimeters errors, but at map sizes of multiple gigabytes.

In case of scenario 1, the map was supplied in Lanelet2 format by a third-party map supplier, not containing curbs. For scenario 2, the map was created semi-automatically in a process similar to the one described in [13], containing curbs, but no traffic lights. Also, in both scenarios, only a few traffic signs are contained in the map.

A. Qualitative Analysis

By reprojecting map elements into the original images, the localization result can be judged qualitatively.

Since not only for the pose estimation algorithm, but also for humans, localization results are hard to judge on straight roads, we refer to the quantitative section to judge them and mainly consider intersections for qualitative analysis.

In Figure 3a, you can see that the localization result is accurate despite the lane prediction being confused by the dashed line for the crossing bike lane, an artifact rarely occurring in Cityscapes.

Both, Figures 3a and 3b show that the optimum is often already reached when map elements only lie on the border of a detection. This is to the distance images which cannot distinguish if a map element lies on the border or well within an areal or extended detection. To overcome this effect, signed distance transform could help to generate a slight gradient even within a detection.

The intersection depicted in Figure 3c is a negative example. In the frames before, tram rails were often misclassified as lane markings, leading to an obviously wrong height and pitch estimate. As the only recommendable solution would be to add a class for tram rails, we stopped evaluating our approach just after intersection I_3 and more than $4\,\mathrm{km}$ of our $5\,\mathrm{km}$ urban track.

Figures 3d and 3e show that well-detected lane markings and curbs lead to good pose estimates even without traffic signs or traffic lights.

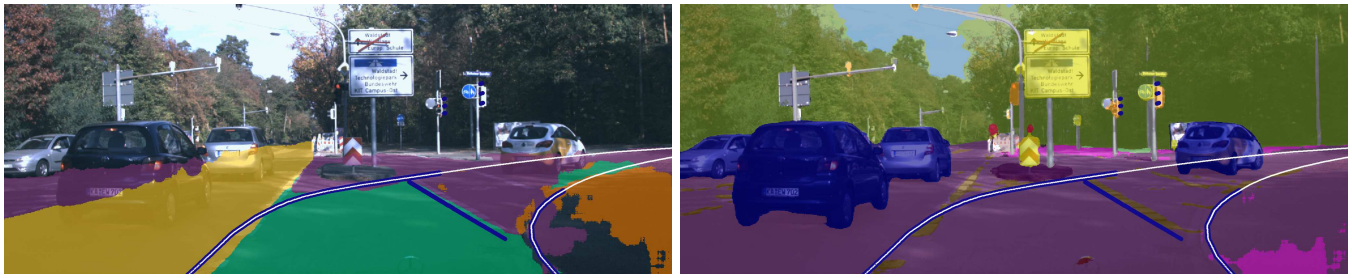
B. Quantitative Analysis

Using the poses of our reference system, we can also evaluate the performance of our approach quantitatively. Hereby, the reference system serves as ground truth. Note that for scenario 1, due to a non-perfect georeferencing, this might not always be optimal. For scenario 2, the reference poses are almost perfect.

TABLE I: Mean absolute errors (MAE) and root mean squared errors (RMSE) in meters or degrees relative to the 6D visual localization reference.

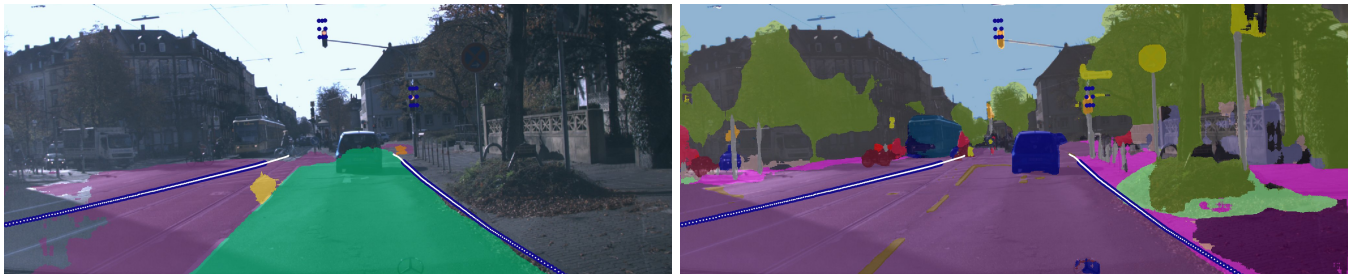
		lon	lat	up	roll	pitch	yaw
Scenario 1	MAE	0.70	0.19	0.06	1.19	0.28	0.45
	RMSE	0.86	0.23	0.08	1.53	0.37	0.58
Scenario 2	MAE	0.73	0.07	0.06	0.77	0.28	0.28
	RMSE	0.90	0.11	0.07	1.00	0.38	0.56

Table I and Figure 4 show that the lateral error and yaw angle is accurate up to the demand of automated driving by providing far more than lane-level accuracy. Especially in scenario 2, which is free from possible georeferencing errors, only after one intersection, there is any significant lateral error. Typically, error spikes occur after turns or intersections,



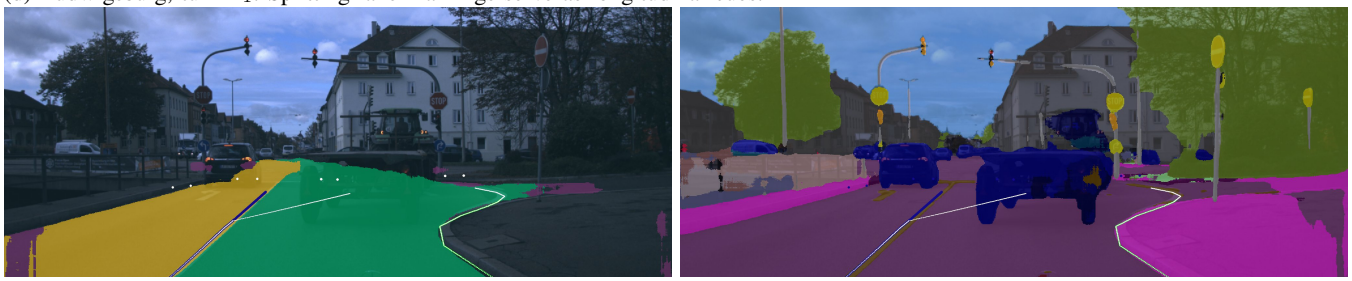
(a) Karlsruhe, intersection I_2. Note that the traffic lights belong to the road coming from the right while the traffic lights for the road ahead are not included in the map.

(b) Karlsruhe, intersection I_3. Only the traffic lights belonging to the turning lane are mapped, the right of which is ignored due to gating.



(c) Karlsruhe, intersection I_4. The tram rails on the left were often misclassified as lane markings in the previous frames, leading to an unrealistic height and pitch estimate.

(d) Ludwigsburg, turn T_1. Splitting lane markings serve as longitudinal cues.



(e) Ludwigsburg, intersection I_2. Note that traffic lights are not contained in the map.

Fig. 3: Qualitative results. The original, cropped images are overlaid with lane (left) and Cityscapes^{+LM}(right) segmentation results as well as with map elements. Best viewed in color with digital zoom. Cityscapes^{+LM}segmentation follows the original Cityscapes colors but also uses yellow for lane markings. Lane segmentation encodes lanes as follows: ego lane = green, left adjacent = yellow, right adjacent = orange, other lanes = violet. For map elements, lane borders = white, curbs = green, lane markings = blue and traffic lights = white/blue.

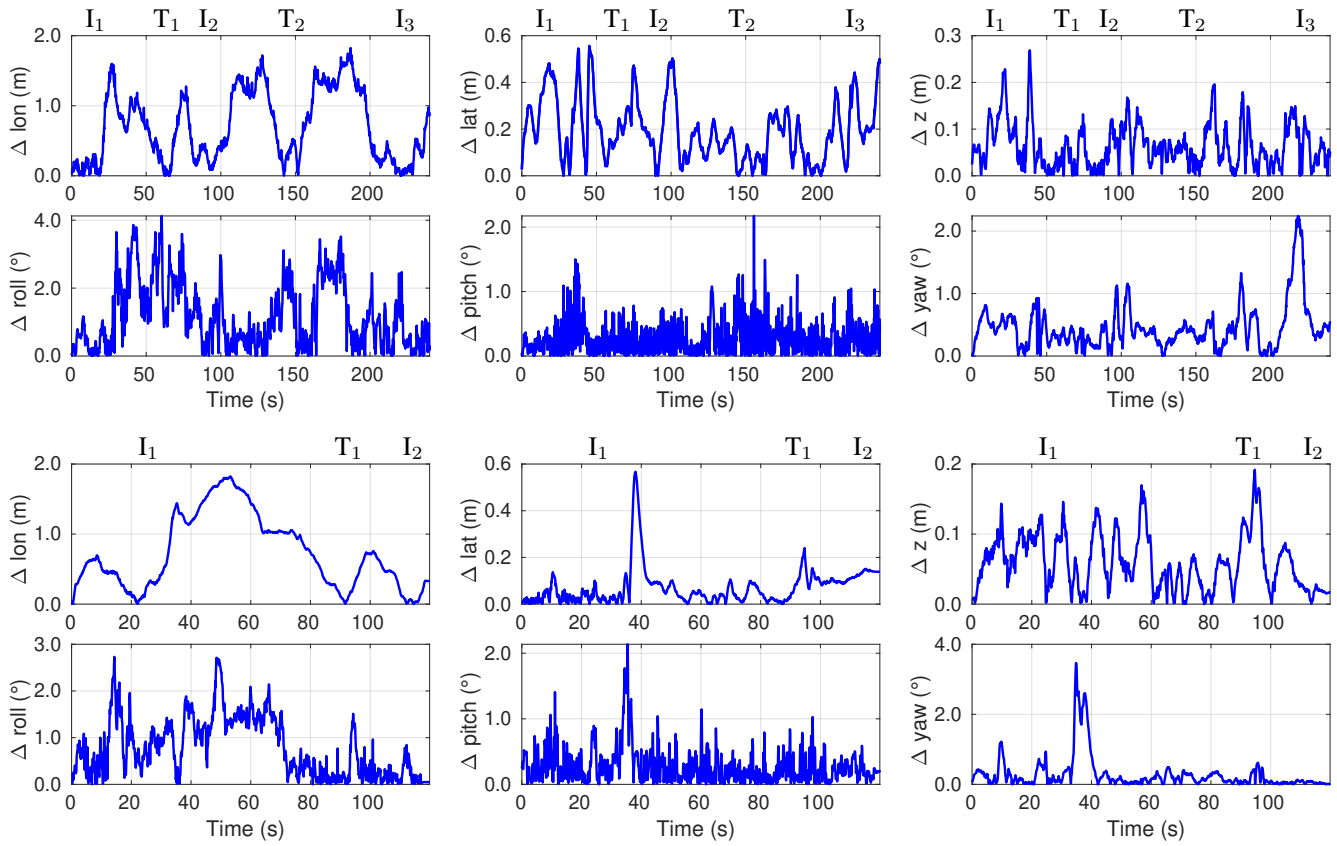


Fig. 4: 6D errors relative to the reference poses for scenario 1 / Karlsruhe (upper six plots) and scenario 2 / Ludwigsburg (lower six plots). T and I denote turns and intersections with longitudinal cues, respectively.

leading to the assumption that our motion model is not perfectly parametrized or too simple.

Overall, the results in lateral direction and yaw angle are in the same range as pole-based localization approaches [1], [7], methods using other additional landmarks not necessarily contained in a typical HD map [2] or methods using proprietary detectors [13].

Longitudinal accuracy cannot keep up to this during long straight sections, but always comes down to comparable level (few centimeters) at turns or intersections, thus, where it actually matters. Figure 3d together with the longitudinal error plot for scenario 2 in Figure 4 shows that only splitting lane markings can serve as longitudinal cues, too. However, this also gives the impression that more traffic signs could improve performance.

Interestingly, even roll and pitch angle as well as the height can be estimated well. While these degrees of freedom might not be necessary for motion planning, other tasks like traffic light classification can benefit from the map using an accurate 6D localization. Additionally, this means that, with better detections and additional cues, our system could also serve as reference in 6D.

VIII. Conclusion

We proposed to use widely available neural networks trained for semantic segmentation as detection front end, overcoming the spatial limitations of previously used bounding box approaches. This enables detecting elements contained in lightweight HD maps for automated driving, such as lanes, lane markings or traffic lights/signs. Combined with vehicle odometry, our approach requires only a monocular camera as hardware and still provides pixel-accurate localization in 6D.

Association is done implicitly by applying a distance transform on binary images of each semantic class of interest. A gradient can be interpolated and used by projecting map elements into the distance images. Advantageously, this leads to an inherently dynamic association between landmarks and detections.

HD map elements are transformed into 3D points, thus, making the approach adaptable to arbitrary shapes of landmarks. To iron out misclassifications and to provide a smooth localization result, we combined multiple consecutive images with vehicle odometry in a sliding window pose graph optimization.

The approach only failed completely in one section where tram rails are consistently misclassified as lane markings. Otherwise, results provide similar accuracy as more expensive or more complex approaches in lateral direction and yaw angle. In longitudinal direction, this is only the case when there are enough cues. However, around intersections and turns where longitudinal accuracy is actually necessary,

usually enough cues are present.

In future work we will include a more advanced motion model as well as GNSS information. Storing all relevant traffic signs or even single lane markings would yield additional information in longitudinal direction, helping to overcome the longitudinal drift on straight roads.

Improving or learning intermediate steps, like [5], could boost accuracy even further. Finally, better, but slower, neural networks, like [21], are an obvious way to improve results as landmarks can be detected more accurately, from greater distance and with fewer misclassifications.

ACKNOWLEDGMENTS

We would like to thank Niels Ole Salscheider and Annika Meyer for their work and great support regarding the CNN as well as Haohao Hu for providing the 6D reference poses.

REFERENCES

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

[2] J. Kümmerle, M. Sons, F. Poggenhans, T. Kühner, M. Lauer, and C. Stiller, "Accurate and Efficient Self-Localization on Roads using Basic Geometric Primitives", in *2019 Int. Conf. Robotics and Automation (ICRA)*, May 2019, pp. 5965–5971.

[3] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, *et al.*, "Lanelet2: A high-definition map framework for the future of automated driving", in *2018 21st Int. Conf. Intelligent Transportation Syst. (ITSC)*, Nov. 2018, pp. 1672–1679.

[4] M. Schreiber, C. Knöppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps", in *2013 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2013, pp. 449–454.

[5] W.-C. Ma, I. Tartavull, I. A. Bârsan, S. Wang, M. Bai, G. Mattyus, *et al.*, "Exploiting Sparse Semantic HD Maps for Self-Driving Vehicle Localization", in *2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Syst. (IROS)*, Nov. 2019, pp. 5304–5311.

[6] H. Hu, M. Sons, and C. Stiller, "Accurate Global Trajectory Alignment using Poles and Road Markings", in *2019 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2019, pp. 1186–1191.

[7] A. Schaefer, D. Buscher, J. Vertens, L. Luft, and W. Burgard, "Long-Term Urban Vehicle Localization Using Pole Landmarks Extracted from 3-D Lidar Scans", in *2019 European Conf. on Mobile Robots (ECMR)*, Sep. 2019, pp. 1–7.

[8] A. Milioto and C. Stachniss, "Bonnet: An Open-Source Training and Deployment Framework for Semantic Segmentation in Robotics using CNNs", in *2019 Int. Conf. on Robotics and Automation (ICRA)*, May 2019, pp. 7094–7100.

[9] B. Mu, S.-Y. Liu, L. Paull, J. Leonard, and J. P. How, "SLAM with objects using a nonparametric pose graph", in *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Syst. (IROS)*, Oct. 2016, pp. 4602–4609.

[10] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual Quadrics From Object Detections as Landmarks in Object-Oriented SLAM", *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 1–8, Jan. 2019.

[11] S. Yang and S. Scherer, "Cubeslam: Monocular 3-D Object SLAM", *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 925–938, Aug. 2019.

[12] J. Jeong, Y. Cho, and A. Kim, "Road-SLAM : Road marking based SLAM with lane-level accuracy", in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 1736–1473.

[13] F. Poggenhans, N. O. Salscheider, and C. Stiller, "Precise Localization in High-Definition Road Maps for Urban Regions", in *2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Syst. (IROS)*, Oct. 2018, pp. 2167–2174.

[14] A. Welzel, P. Reisdorf, and G. Wanielik, "Improving Urban Vehicle Localization with Traffic Sign Recognition", in *2015 IEEE 18th Int. Conf. on Intelligent Transportation Syst. (ITSC)*, Sep. 2015, pp. 2728–2732.

[15] R. Spangenberg, D. Goehring, and R. Rojas, "Pole-based localization for autonomous vehicles in urban scenarios", in *2016 IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, Oct. 2016, pp. 2161–2166.

[16] M. Sefati, M. Daum, B. Sondermann, K. D. Kreisköther, and A. Kampker, "Improving vehicle localization using semantic and pole-like landmarks", in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 13–19.

[17] C. Toft, C. Olsson, and F. Kahl, "Long-Term 3D Localization and Pose from Semantic Labellings", in *2017 IEEE Int. Conf. on Computer Vision Workshops (ICCVW)*, Oct. 2017, pp. 650–659.

[18] N. O. Salscheider, "Simultaneous object detection and semantic segmentation", in *9th International Conference on Pattern Recognition Applications and Methods*, Feb. 2020, pp. 555–561.

[19] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, *et al.*, "The Cityscapes Dataset for Semantic Urban Scene Understanding", in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 3213–3223.

[20] A. Meyer, N. O. Salscheider, P. F. Orzechowski, and C. Stiller, "Deep Semantic Lane Segmentation for Mapless Driving", in *2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Syst. (IROS)*, Oct. 2018, pp. 869–875.

[21] L. Porzi, S. R. Bulo, A. Colovic, and P. Kontschieder, "Seamless scene segmentation", in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 8277–8286.

[22] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder, "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes", in *2017 IEEE Int. Conf. on Computer Vision (ICCV)*, Oct. 2017, pp. 5000–5009.

[23] G. Bradski, "The opencv library", *Dr. Dobb's Journal of Software Tools*, 2000.

[24] S. Agarwal, K. Mierle, *et al.*, *Ceres solver*, `http://ceres-solver.org`.

[25] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, "Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate", en, *ACM Transactions on Mathematical Software*, vol. 35, no. 3, pp. 1–14, Oct. 2008.

[26] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking", in *2008 11th Int. Conf. on Information Fusion*, Jun. 2008, pp. 1–6.

[27] M. Sons, H. Lategahn, C. G. Keller, and C. Stiller, "Multi trajectory pose adjustment for life-long mapping", in *2015 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2015, pp. 901–906.

[28] M. Sons and C. Stiller, "Efficient Multi-Drive Map Optimization towards Life-long Localization using Surround View", in *2018 21st Int. Conf. Intelligent Transportation Syst. (ITSC)*, Nov. 2018, pp. 2671–2677.

[29] H. Lategahn and C. Stiller, "Vision-Only Localization", *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 1246–1257, Jun. 2014.