

HD Map Generation from Noisy Multi-Route Vehicle Fleet Data on Highways with Expectation Maximization

Fabian Immel¹, Richard Fehler¹, Mohammad M. Ghanaat², Florian Ries²,
Martin Haueis² and Christoph Stiller³

Abstract— High Definition (HD) maps are necessary for many applications of automated driving (AD), but their manual creation and maintenance is very costly. Vehicle fleet data from series production vehicles can be used to automatically generate HD maps, but the data is often incomplete and noisy. We propose a system for the generation of HD maps from vehicle fleet data, which is tolerant to missing or misclassified detections and can handle drives with multiple routes, generating a single complete map, model-free and without prior reference lines. Using randomly selected drives as pivot drives, a step-wise lateral sampling of detections is performed. These sampled points are then clustered and aligned using Expectation Maximization (EM), estimating a lateral offset for each drive to compensate localization errors. The clustered points are replaced with the maxima of their probability density function (PDF) and connected to form polylines using a modified rectangular linear assignment algorithm. The data from vehicles on varying routes is then fused into a hierarchical singular map graph. The proposed approach achieves an average accuracy below 0.5 meters compared to a hand annotated ground truth map, as well as correctly resolving lane splits and merges, proving the feasibility of the use of vehicle fleet data for the generation of highway HD maps.

Index Terms— HD Map Generation, Vehicle Fleet Data, Automated Driving (AD), Expectation Maximization (EM).

I. INTRODUCTION

High definition (HD) maps are an integral part of many applications of automated driving [1]. Compared to standard definition (SD) maps, HD maps contain more detailed and accurate information about a road, such as detailed and centimeter-level accurate road boundaries and lane markings.

A large obstacle in the widespread adoption of automated driving is the cumbersome and costly generation and updating of HD maps, as normally special vehicles with a variety of sensors, such as accurate GNSS, LiDAR and cameras are used. Creating and maintaining a HD map for a large highway network solely with specialized mapping vehicles requires considerable effort and expenses.

Due to the widespread proliferation of driver assistance systems in current series cars, a large body of lane marking and road boundary data is available. Generating HD maps of a highway network from this data can enable a variety of systems and drastically reduce the costs of maintaining

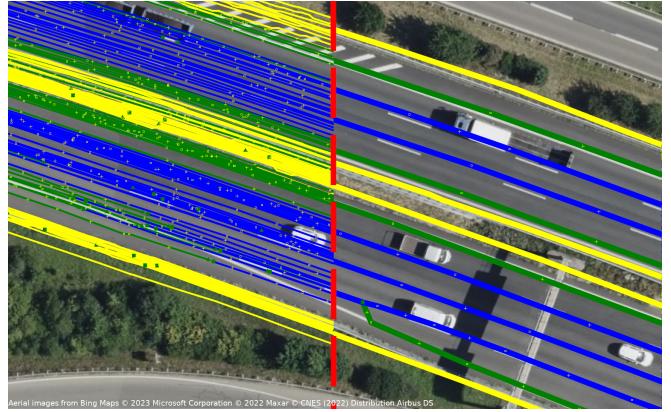


Fig. 1: Result of the mapping (right) on a German highway overlaid with the input data (left). Displayed are the road barriers (yellow), dashed lines (blue) and solid lines (green).

a HD map, as updates can be generated automatically from new fleet data.

However, processing this vehicle fleet data comes with several issues and requires aggregation and fusion of the individual detections. Furthermore, the accuracy and completeness of the detected features is not on the same level as data from specialized mapping vehicles. Inaccurate localization from series-level GNSS systems and missing detections of road features, possibly during large parts of a drive, complicate this aggregation and fusion.

In this paper we propose and evaluate a framework for a HD mapping pipeline, with an example of its results shown in Fig. 1. It is capable of ingesting vehicle fleet data consisting of incomplete, partially misclassified and inaccurate detected lane markings and road boundaries and generating an HD map of the physical road features as connected polylines as a result.

Figure 2 shows an overview of the proposed map generation process. The input data, connected lane marking detections of individual drives (Sec. III) are processed along randomly selected pivot drives by sampling detected points along a lateral intersection line (Sec. IV-A). Sampled points along the ego-road are clustered using k-means and aligned by estimating a localization offset for each drive, being iteratively refined using EM (Sec. IV-B). The aggregated points are replaced with the maxima of their PDF (Sec. IV-C) and connected to form a polyline (Sec. V). For connecting the points of consecutive sampling steps, a modified version of the Hungarian algorithm is used to find the optimal con-

¹FZI Research Center for Information Technology, Karlsruhe, Germany
{immel, fehler}@fzi.de

²Mercedes-Benz AG, Research & Development, Sindelfingen, Germany
{mohammad.m.ghanaat, florian.ries, martin.haueis}@mercedes-benz.com

³Institute of Measurement and Control Systems, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany stiller@kit.edu

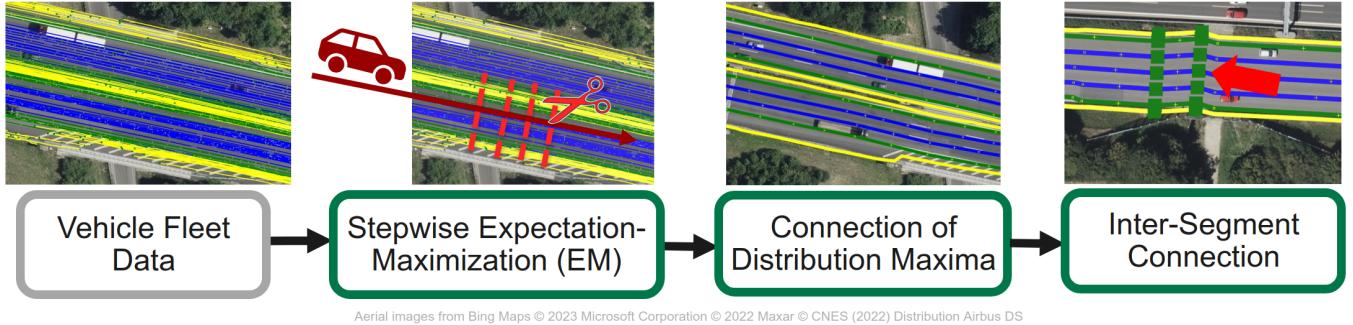


Fig. 2: Flowchart with an overview of the map generation process. The input data is described in Sec. III, the stepwise EM in Sec. IV, the connection of distribution maxima in Sec. V and the inter-segment connection in Sec. VI. The red arrow indicates the connection points of the inter-segment connection.

nection between all PDF maxima of the two sets. The thus modified connection algorithm is able to handle lane splits and merges. Pivot drives visiting the same road segment along their route are skipped, reducing the computation time.

The data of different connected segments, e.g. from vehicles on varying routes, is then integrated into a hierarchical singular map graph (Sec. VI).

The main contributions of this work can be summarized as follows:

- A novel end-to-end HD map generation pipeline capable of handling vehicle fleet data with incomplete, partially misclassified and inaccurate road feature detections.
- A model-free approach for the aggregation and fusion of detected lane markings and road boundaries, without prior reference lines.
- A graph-based method for the connection of fused detections capable of lane splits and merges and integration of multiple connected segments of different routes into a hierarchical singular map graph.

II. RELATED WORK

The automated generation of maps from crowdsourced or fleet data is a topic with longstanding research interest.

A large body of work focuses on the generation of SD maps and road networks, using driven trajectories as input data. Many approaches can be grouped into three categories [2]: k-Means based [3], [4], Kernel Density Estimation (KDE) based [5], [6] and trace merge based [7], [8]. One commonality of all these approaches is that they only focus on basic road geometries similar to SD maps, since data about physical road features is not available.

With the recent increase in advanced driver assistance systems in vehicles, both the interest in HD maps and the availability of fleet data on physical road features has increased. This lead to a variety of research utilizing multi-journey data for the generation of HD maps. Vision methods such as [9], [10] assume access to the recorded video data of the vehicles. [9] uses single camera images for visual SLAM to generate 3D boundaries of road feature elements. [10] describes an system to detect road features in single drives and then fuse these drives in a bundle adjustment pipeline.

Conceptually closer to the approach presented in this paper are [11]–[13], as they use already processed road boundary and lane marking detections as their input. [11] corrects detected lane borders by grouping and aggregating detected left road boundaries and then fuses the corrected detections with k-means clustering. [12] uses unconnected road feature point measurements and drive trajectories to formulate a graph SLAM and aggregate the connections. [13] also uses a graph slam, but fuses the individual detections using a road model that is selected with a beam-style search.

Common to the described systems for already processed detections is the focus on simple highway sections with mostly unchanging lanes. Other requirements are the use of external reference lines [12], always available detections of certain features [11] or only a single link sequence [13]. Important issues not tackled yet in literature are also data sets with multiple routes and the connection of generated patches into a complete map.

III. INPUT DATA

The algorithm takes as input a set of drives from a vehicle fleet, containing the drive trajectories and the associated detected road features. Importantly, the drives may belong to different routes in a highway network and can have sparse or partially misclassified road feature detections as well as only series-grade localization with the respective large errors. Detected points are assumed to be already connected to polylines if they belong to the same road feature, however these polylines do not need to be complete and may have gaps in their connections as well as misclassifications. For preprocessing, detection polylines with a length below 3 meters are removed, as these are very likely misdetections and the number of points in remaining polylines is reduced using the Ramer-Douglas-Peucker algorithm.

The resulting detection polylines are then inserted into an axis-aligned bounding box tree [14], a data structure similar to kd-trees for use with shapes instead of single points. We take advantage of this data structure to efficiently find detection polylines during IV-A.

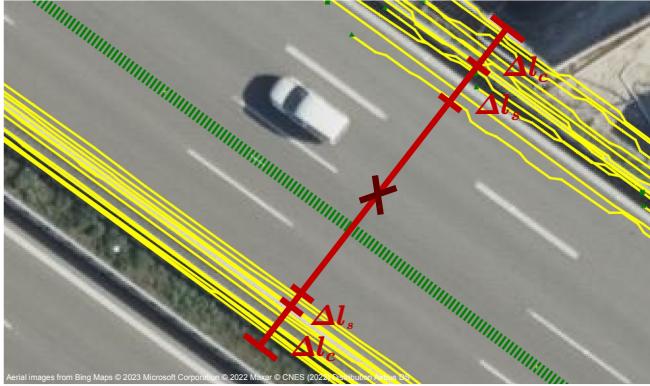


Fig. 3: Visualization of the refined cut line calculation. The trajectory of the pivot drive can be seen in green, the filtered road boundary detections in yellow and the cut line in red. The line width is computed from the distance to the closest road boundary detections plus a margin Δl_s depending on the detection spread and a constant margin Δl_c . The recomputed cut line midpoint is marked by the cross.

IV. STEPWISE EXPECTATION MAXIMIZATION

A. Step-wise Lateral Sampling of Detection Points

To avoid the requirement of reference lines that provide the basic road shape, the processing of detections is performed along pivot drives. A pivot drive is randomly selected from the set of drives until all drives are processed and detections are sampled along its trajectory with small fixed-size steps in driving direction (e.g. 2 m). Starting with the trajectory point as the midpoint, a lateral cut line segment is formed, whose intersection points with the detection polylines are sampled. The purpose of the pivot drive is only to provide a very rough reference for the course of a road and to enable the lateral sampling of detections. Sampled detections from drives from the opposite road direction are removed based on the heading information of the respective trajectories.

The goal of the lateral sampling is to obtain a set of all detections that belong to the ego-road of the pivot drive. The road width is however a priori unknown and also cannot be inferred from the incomplete pivot drive detections. Therefore the initial cut line segment has a fixed length of several times the average highway road width to ensure that all associated detections are included. Since drives of various different routes are expected to be part of the data, refinement of the initial cut line segment and therefore filtering of the sampled detection points is necessary, to ensure that only detections belonging to the ego-road are included.

This refinement is performed with the help of the sampled detections, a visualization of which is shown in Fig. 3. A new cut line is calculated from the first road boundaries encountered from the original midpoint in both lateral directions, with the start and end point being extended by Δl_s depending on the spread of the first road boundary detections encountered. This provides a first estimate of the ego-road width, additionally including a constant margin Δl_c so noisy detections are still included. Length and midpoint

of the new cut line segment are smoothed using a moving average filter with decreasing weights for older values and a regular moving average filter, respectively, to provide a road width estimate even in sections with missing road boundary detections. The detections are then again sampled using the new cut line.

B. Lateral Offset Estimation with Expectation Maximization

Fig. 4a shows a distribution of sampled detection points before the lateral correction. The large spread of detections needs to be corrected for sensible aggregation of the measurements. We therefore try to minimize the alignment error between the different drives, assuming drive localization errors as the main source of misalignment rather than individual measurement noise. Referring back to this assumption, we estimate one lateral offset for each drive as correction for the localization. This means that all detection points belonging to a single drive can only be moved in the same direction and with the same offset.

As the co-referencing of detections, meaning the assignment of detections to the same road feature, is interdependent with the lateral offset estimation, we use an iterative EM approach. First, the detections are separated by type and clustered using the k-means algorithm, using the silhouette score as the selection criterion for the number of clusters. We note

$$\mathbf{o} = (o_1, o_2, \dots, o_m) \quad (1)$$

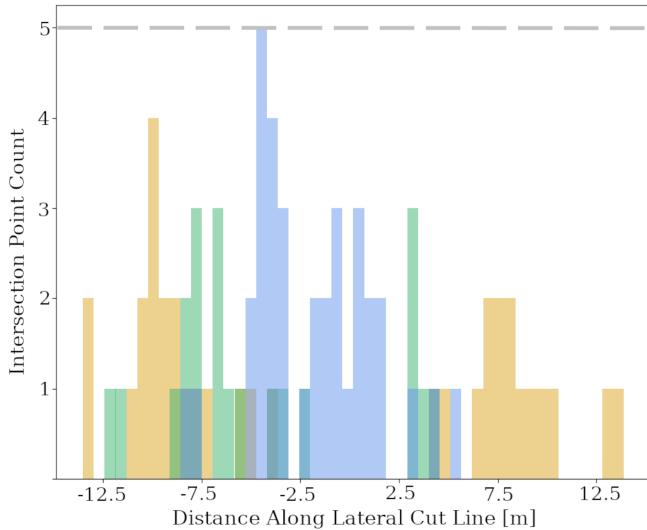
as the vector of m drive offsets, with one offset assigned per drive and all offsets parametrized using the cut line vector. The lateral offset for each drive is then estimated by minimizing the cost function given below:

$$\mathbf{o}_{\text{opt}} = \arg \min_{\mathbf{o}} \sum_{i=1}^n \text{Var}(c_i(\mathbf{o})) + \sum_{i=1}^n f(c_i(\mathbf{o})) \quad (2)$$

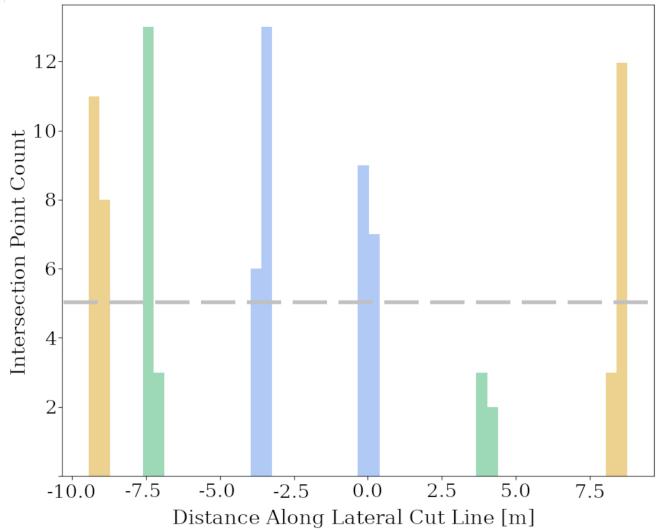
$$f(c_i) = e^{-s \cdot d_{min}^2} \quad (3)$$

Where $\text{Var}(c_i(\mathbf{o}))$ denotes the variance of cluster c_i corrected with \mathbf{o} . $f(c_i)$ describes a penalty function that increases for small distances to the nearest cluster of the same type d_{min} and can be adjusted depending on the expected lane width with the parameter s . The described optimization problem can be solved with any non-linear least squares algorithm, e.g. Levenberg-Marquardt. Finally, the detections are re-clustered using the new offsets and the process is repeated for a fixed number of iterations.

Fig. 4b displays the result of the EM optimization on the points in Fig. 4a and shows how overlapping detection clusters with large spread can be aligned and separated, also exemplifying localization errors as the main source of misalignment. Optimization results are scored with their silhouette score, and results below a threshold of 0.67 are rejected and the respective sampling step is skipped to prevent wrongly aggregated points from being used in later steps of the pipeline. Additionally, possible drift of the



(a) Before lateral correction



(b) After lateral correction

Fig. 4: Histogram of sampled detection points before and after EM-based lateral correction. Road boundary samples are shown in yellow, dashed lane marking samples in blue, solid lane marking samples in green.

optimization results compared to the actual road shape is corrected by aligning the road midpoint post-optimization with the one pre-optimization.

C. Aggregation of Clustered Detection Points

After sec. IV-B, we have a set of clustered and aligned detection points. To aggregate the measurements of a single cluster, we estimate its PDF using Kernel Density Estimation. The KDE is estimated using a Gaussian kernel with a bandwidth of 6 times the clusters standard deviation. The detection points belonging to the same cluster are then replaced by the maximum of the estimated PDF. If there are misclassifications of road feature detections present, estimated PDFs of different clusters may overlap each other, which can be used to filter out these misclassifications. If two PDFs overlap area exceeds a certain threshold, the cluster with the lower number of detection points is removed.

The calculated maximum points of the PDFs now each represent the aggregated detections of a single road feature. They are saved for every drive whose detections were included in the optimization, with the traveled distance along the trajectory of the respective drive as the key. With this information, points on drives that were already processed during another pivot drive do not need to be calculated again, saving large amounts of computation time. The whole process described in this section is repeated for every drive in the input data.

V. CONNECTION OF DISTRIBUTION MAXIMA

The difficult problem of connecting arbitrary point sets to form semantically useful polylines can be massively simplified with the contextual information of the lateral sampling along the associated pivot drive and thus the ego-road. Since we know that the order of point sets follows the trajectory of the drive, we can infer that the aggregated points

of succeeding point sets must also be connected. The hard problem of finding polylines can therefore be separated into finding the optimal connections between succeeding point sets, which is a rectangular linear assignment problem for bipartite graphs. This problem can be solved efficiently using the algorithm described in [15], using the distance between the points of opposite sets as the edge weights.

However, the algorithm does not allow for the assignment of leftover nodes, which can occur at lane splits and merges, where the number of points is not equal on both sides of the bipartite graph. To solve this problem, we use a modified version of the algorithm, where the leftover nodes are assigned to their closest neighbor on the other side of the bipartite graph. This approximates the topological connections commonly encountered for road markings.

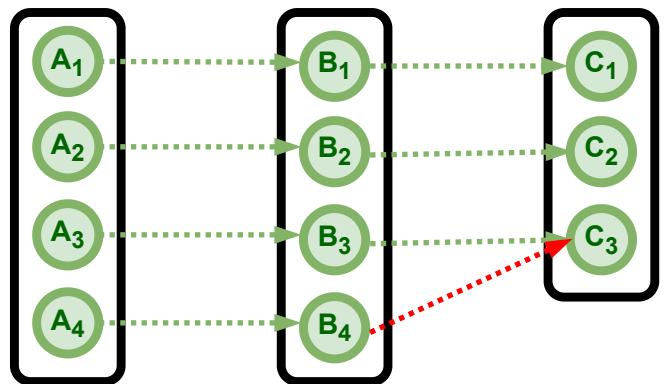


Fig. 5: Example of assignments of the used modified rectangular assignment algorithm [15]. The leftover node B_4 is assigned to its closest bipartite neighbor C_3 .

Figure 5 shows an example of the modified algorithm, where the unmodified algorithm produces the assignment

shown by the green arrows, with node B_4 left unassigned. The modified algorithm assigns leftover nodes to their closest neighbor on the other side of the bipartite graph, assigning node B_4 to node C_3 , as shown by the red arrow.

This procedure is performed sequentially for each point set, filtering out implausible optimization results via an average connection angle threshold per bipartite graph as well as preventing connections of detections of a different type to road borders. As described in Sec. IV-B, failure of the optimization may lead to patches with no aggregated detections, which can be bridged by connecting to the next valid point set, if the distance is below a given threshold.

To receive polylines out of the calculated graph edges, a set of open and closed polylines is maintained. Polylines are opened for the starting points of a drive. They are closed and a new polyline is opened in the following conditions:

- The connection edge connects two detection points of a different type, e.g. a dashed line and a solid line.
- The connection is made after the regular rectangular assignment algorithm (leftover node, as in B_4 to node C_3 in Fig. 5)
- The distance between the two detection point sets exceeds a given threshold. This can happen when the optimization fails to find a solution for a long road segment.
- We reach a section that was processed by a different pivot drive.

VI. INTER-SEGMENT CONNECTION

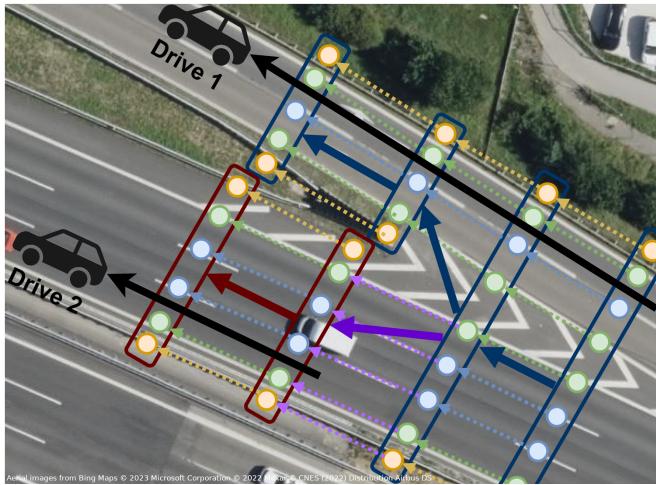


Fig. 6: Visualization of the hierarchical graph structure used for stitching of map segments. The high-level topology graph (blue/red) contains information about the road topology, while the low-level road feature graph (colored nodes) contains the actual road feature polylines. Nodes in both graphs contain references to each other. Two segments (blue and red) from different drives can be stitched together as shown with the purple edges.

Sec. V results in individually connected segments for every drive. These segments are stitched together into a complete

map using a hierarchical graph structure comprised of a high-level topology graph and a low-level road feature graph. This structure is created for every segment during Sec. V as well as for the final complete map graph.

Fig. 6 shows a visualization of this structure. A node in the topology graph, shown by the red and blue rectangles, is created for each sampled step in Sec. IV-A and contains the road midpoint at its position as well as a reference to the aggregated detection points of its step. The topology graph can be seen as similar to a regular SD map, as it contains information about the high-level road topology. The low-level road feature graph (the small circles in Fig. 6) contains the actual road feature polylines, which are created by connecting the aggregated detection points of the topology graph nodes. To insert the results of a new drive into the complete graph, visualized by the purple arrows, the following steps are performed:

- 1) The drive topology graph is separated into its connected components. Breaks in the connection may for example occur in sections that have already been processed by another pivot drive.
- 2) The start and end nodes of a connected component are extracted and their nearest neighbor nodes in the complete topology graph are found using a kd-tree.
- 3) The possible connection to the neighbor node is validated with a distance and connection angle check.
- 4) If successful, the connection to the neighbor node is added to the complete topology graph and the low-level road feature polylines are also connected using the referenced low-level nodes in the topology nodes. The method for connecting the polylines is as described in Sec. V, with the addition that only nodes of the same road feature type are connected and that the unmodified version of the rectangular assignment algorithm in [15] is used.

The system proposed here allows for the creation of a map with entire highway networks, including exit and entry ramps, as well as updating an existing map with newly mapped sections.

VII. EVALUATION

To evaluate the performance of the proposed system, we use a dataset of vehicle fleet data from 134 drive snippets on a highway section near Stuttgart, Germany. The section consists of two opposite highway directions with a total length of 29.1 km, including highway exit and entry ramps.

A. Ground Truth Dataset

We evaluate the lateral error of the resulting map by comparing it to a hand-labeled ground truth dataset generated from satellite imagery. The ground-truth data is validated with multiple imagery sources, eliminating parts of the highway for evaluation where recent construction sites hinder accurate ground-truth annotation. The total length of annotated polylines is 204 km. The fleet data has a bias towards one direction, showing a higher number of drives in this direction as visible in the first 15 kilometers of fig. 7.

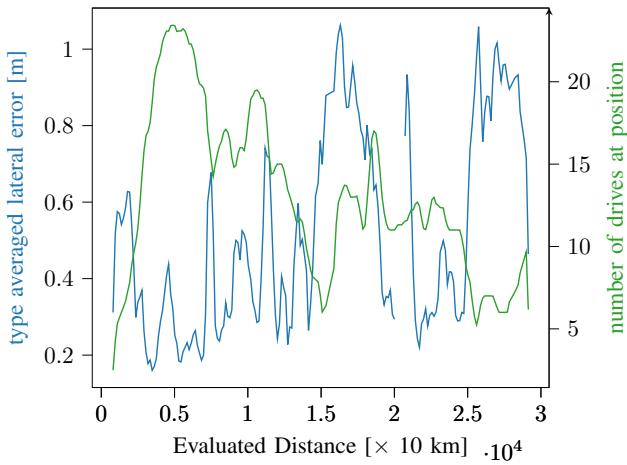


Fig. 7: Average lateral error over driving distance in relation to the number of drives traversing that location in the dataset. This data is sliding window low-pass filtered for clear visibility. Number of drives and total error correlate inversely.

The generated ground truth additionally provides a reference line and region-of-interest in comparison to the sampling of pivot drive trajectories and cut line refinement during the optimization step. Like in section IV, a lateral cut line is computed for every two meters along the reference line. This cut line is restricted by the region-of-interest borders and all intersecting polylines in the resulting map and ground truth data are retrieved. We associate an intersection point of the mapping result with the closest ground truth point and compute the distance between measurement and ground truth for each map feature type. The influencing factors on the resulting lateral distance error are investigated in the remainder of this section.

B. Evaluation Metrics

The evaluation metric is limited to the error in lateral driving direction, since longitudinal features are sparse in the measurement data. We evaluate the lateral optimization result as described in section IV-B. The lateral error consists of two parts, the lateral *offset* error resulting from an uncompensated localization error and the remaining *non-offset* error, which is less correlated to the number of drives. This effect can be observed in Fig. 9. The non-offset error is derived by first by computing an offset o applied to all mapped points simultaneously, that minimizes the total error compared to the ground truth position. The remaining error is denoted as *offset corrected error* and visualized in orange in Fig. 9 and Fig. 10, contrary to the blue total error. In a real-world application of the method, this ground truth based offset correction is not possible, thus the number of drives visiting the mapped position should be kept well above 18 drives.

C. Evaluation Results

The lateral accuracy this method achieves in best case scenarios with a high number of traversing drives and simple

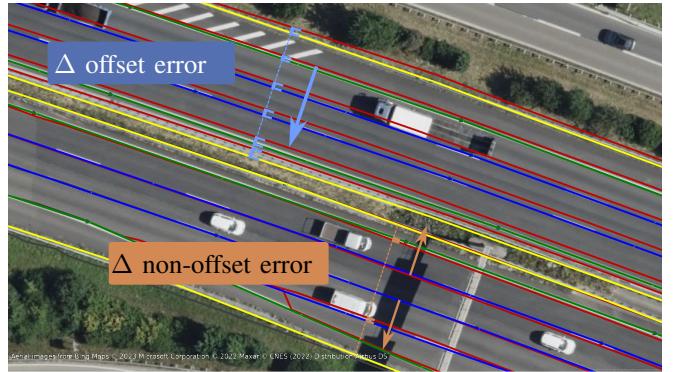


Fig. 8: Visualization of the evaluation metrics. The ground truth with road boundaries is shown in yellow, dashed lines in blue and solid lines in green. The mapped result is shown in red, without type color coding. The offset error, resulting from localization errors, is indicated by the blue arrow. As these errors are related to a small number of drives in some sections of the evaluated dataset, we also evaluate a offset corrected error. A sample of remaining uncorrectable non-offset error is visualized in orange.

highway structures results in *mean* total errors below 0.30 meters. Averaging all evaluated positions, including mapped exits and entry ramps and all map feature types results in a *mean* total error of 0.49 meters. The mean absolute offset error as described in Fig. 8 constitutes 0.41 meters. Correcting for this offset error, a resulting 0.27 *mean* error over all evaluated positions can be achieved in this theoretical setting. For an increasing number of drives the offset corrected error and actual error converge as seen in Fig. 9.

Since we have multiple routes crossing the evaluated highway section, we have a varying number of drives at different points along the reference line. Fig. 7 illustrates this by showing the number of drives in comparison with the lateral error along the evaluated distance. The relationship of the drive count at a mapped position with the lateral error is further investigated in Fig. 9. As the optimization threshold criterion for valid results is not achieved at all times, e.g. when passing some highway exits and ramps, the evaluation is only possible for around 90 percent of the ground truth reference line distance.

VIII. CONCLUSIONS

In this work, we proposed a system for a HD mapping pipeline for vehicle fleet data with processed sensor detections, making the approach scalable for large fleets. The system is tolerant to missing or misclassified detections and can handle drives with multiple routes, generating a single complete map, model-free and without prior reference lines.

The pipeline uses randomly selected drives as pivot drives, along which a step-wise lateral sampling of detections is performed. These sampled points are then clustered and aligned using k-means and EM, estimating a lateral offset for each drive to compensate localization errors. The aggregated points are replaced with the maxima of their PDF and connected to form a polyline using a modified rectangular

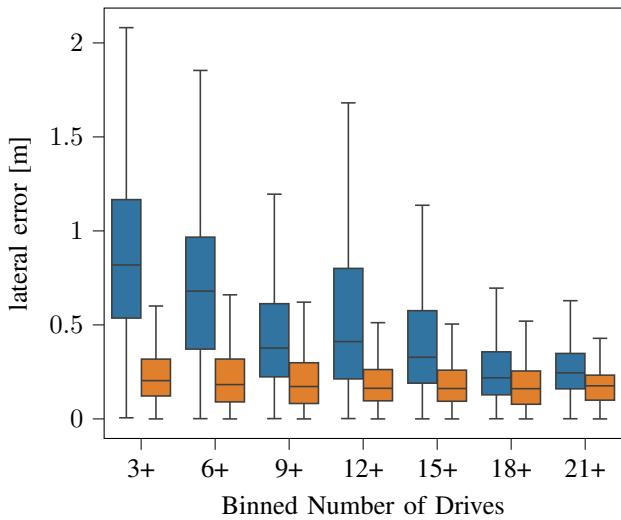


Fig. 9: Statistical box plot with lateral error of a position over the traversing number of drives at that position. The total lateral error is displayed in blue and the offset corrected error in orange. The error induced by localization offsets can be reduced with increasing number of drives. All box plots end whiskers at 1.5 times the interquartile range.

linear assignment algorithm. The data of different connected segments, e.g. from vehicles on varying routes, is then integrated into a hierarchical singular map graph.

The proposed approach achieves a high accuracy below 0.5 meters compared to a hand annotated ground truth map, as well as correctly resolving lane splits and merges, proving the feasibility of the use of series fleet data for the generation of highway HD maps. For future work, we would like to improve the reliability of the algorithm in complex scenarios and include other features such as traffic signs to correct misalignments in the longitudinal direction.

ACKNOWLEDGEMENTS

We would like to thank the Mercedes-Benz AG for providing the vehicle fleet data used in this work.

REFERENCES

- [1] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, *et al.*, “Making bertha drive—an autonomous journey on a historic route,” *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014, ISSN: 1941-1197.
- [2] J. Biagioli and J. Eriksson, “Inferring road maps from global positioning system traces: Survey and comparative evaluation,” *Transportation Research Record*, vol. 2291, no. 1, pp. 61–71, 2012. eprint: <https://doi.org/10.3141/2291-08>.
- [3] S. Edelkamp and S. Schrödl, “Route planning and map inference with global positioning traces,” in *Computer Science in Perspective: Essays Dedicated to Thomas Ottmann*, R. Klein, H.-W. Six, and L. Wegner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 128–151, ISBN: 978-3-540-36477-1.
- [4] S. Schroedl, K. Wagstaff, S. Rogers, P. Langley, and C. Wilson, “Mining gps traces for map refinement,” *Data mining and knowledge Discovery*, vol. 9, pp. 59–87, 2004.
- [5] J. Davies, A. Beresford, and A. Hopper, “Scalable, distributed, real-time map generation,” *IEEE Pervasive Computing*, vol. 5, no. 4, pp. 47–54, Oct. 2006, ISSN: 1558-2590.
- [6] W. Shi, S. Shen, and Y. Liu, “Automatic generation of road network map from massive gps, vehicle trajectories,” in *2009 12th International IEEE Conference on Intelligent Transportation Systems*, Oct. 2009, pp. 1–6.
- [7] L. Cao and J. Krumm, “From gps traces to a routable road map,” in *17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2009)*, November 4–6, 2009, Seattle, WA, Nov. 2009, pp. 3–12.
- [8] B. Niehofer, R. Burda, C. Wietfeld, F. Bauer, and O. Lueert, “Gps community map generation for enhanced routing methods based on trace-collection by mobile phones,” in *2009 First International Conference on Advances in Satellite and Space Communications*, Jul. 2009, pp. 156–161.
- [9] M. Herb, T. Weiherer, N. Navab, and F. Tombari, “Crowdsourced semantic edge mapping for autonomous vehicles,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 7047–7053.
- [10] O. Dabeer, W. Ding, R. Gowaiker, S. K. Grzechnik, M. J. Lakshman, S. Lee, *et al.*, “An end-to-end system for crowdsourced 3d maps for autonomous vehicles: The mapping component,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 634–641.
- [11] K. Massow, B. Kwella, N. Pfeifer, F. Häusler, J. Pontow, I. Radusch, *et al.*, “Deriving HD maps for highly automated driving from vehicular probe data,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, ZSCC: 0000042 ISSN: 2153-0017, Nov. 2016, pp. 1745–1752.
- [12] C. Doer, M. Henzler, H. Messner, and G. F. Trommer, “HD map generation from vehicle fleet data for highly automated driving on highways,” in *2020 IEEE Intelligent Vehicles Symposium (IV)*, ZSCC: 0000022 ISSN: 2642-7214, Oct. 2020, pp. 2014–2020.
- [13] M. Liebner, D. Jain, J. Schauseil, D. Pannen, and A. Hackelöer, “Crowdsourced hd map patches based on road model inference and graph-based slam,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2019, pp. 1211–1218.
- [14] K. A. Hart and J. J. Rimoli, “Generation of statistically representative microstructures with direct grain geometry control,” *Computer Methods in Applied Mechanics and Engineering*, vol. 370, 2020, ISSN: 0045-7825.
- [15] R. M. Karp, “An algorithm to solve the $m \times n$ assignment problem in expected time $O(mn \log n)$,” *Networks*, vol. 10, no. 2, pp. 143–152, 1980. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/net.3230100205>.

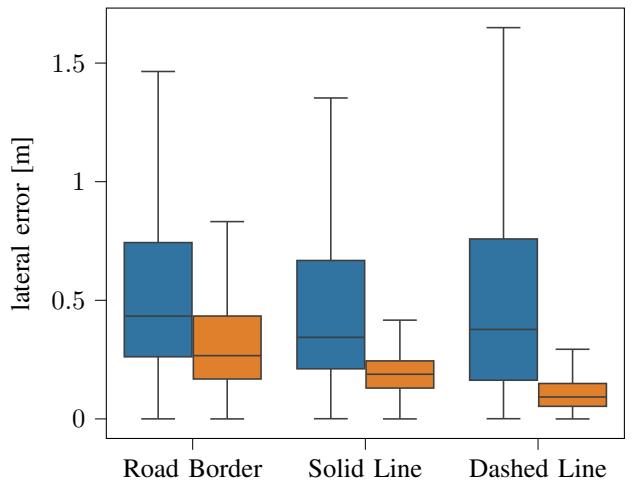


Fig. 10: Lateral error separated by feature type. The observed error strongly depends on the map feature type, mostly due to different accuracy of the available road border features. The total lateral error is shown in blue and the offset corrected error in orange. We do not differentiate the number of drives in this plot.