Lane-Level Map Matching based on HMM

Citation for the original published paper (version of record):
Hansson, A., Korsberg, E., Maghsood, R. et al (2021). Lane-Level Map Matching based on HMM. IEEE Transactions on Intelligent Vehicles, 6(3): 430-439.
http://dx.doi.org/10.1109/TIV.2020.3035329

(article starts on next page)

Lane-Level Map Matching based on HMM

Anders Hansson, Ellen Korsberg, Roza Maghsood, Eliza Nordén, and Selpi

***Abstract*—Lane-level map matching is essential for autonomous driving. In this paper, we propose a Hidden Markov Model (HMM) for matching a trajectory of noisy GPS measurements to the road lanes in which the vehicle records its positions. To our knowledge, this is the first time that HMM is used for lane-level map matching. Apart from GPS values, the model is further assisted by yaw rate data (converted to a lane change indicator signal) and visual cues in the form of the left and right lane marking types (dashed, solid, etc.). Having defined expressions for the HMM emission and transition probabilities, we evaluate our model to demonstrate that it achieves 95.1% recall and 3.3% median path length error for motorway trajectories.**

***Index Terms*—map matching, lane-level map matching, road networks, hidden Markov model, Viterbi algorithm**

I. Introduction

MAP matching, or matching a noisy GPS trajectory to roads in a map, is an important problem in autonomous transportation. The reader is referred to Chao et al. for a recent survey on map matching algorithms [1]. In 2009, Newson and Krumm proposed a Hidden Markov Model (HMM) for the *road-level* map matching problem [2]. In their work, roads are represented as a graph of nodes and edges with nodes at intersections, dead ends, and road name changes. Edges represent road segments between the nodes, and they could be directional to indicate one-way roads. Each node has an associated longitude/latitude to indicate its location, and each edge has a polyline of longitude/latitude pairs to represent its geometry. The states of the HMM are the individual road segments, and the state measurements are the noisy GPS measurements. The most probable sequence of visited road segments is then inferred by the Viterbi algorithm [3]. Luo et al. evaluated HMM-based road-level map matching on four sets of GPS data and demonstrated that the matching accuracy exceeds 99.9% [4]. Inspired by the robustness and success of the Newson/Krumm framework when applied to road-level map matching, in this paper, we employ HMMs to the *lane-level* map matching problem.

Section II gives a brief background to lane-level map matching, which is a relatively new problem in the intelligent vehicles domain. Section III presents our algorithm in detail, while Section IV describes our input data. To assess the performance of our lane-level map matching algorithm, we suggest various performance metrics, which are discussed in Section V. Finally, results are presented in Section VI, before the paper is concluded in Section VII.

A. Hansson is with Zenuity, Lindholmspiren 2, Göteborg, Sweden, 41756 (e-mail: anders.hansson@zenuity.com).

E. Korsberg was with the Department of Computer Science and Engineering, Chalmers University of Technology, SE-412 96 Göteborg, Sweden (e-mail: korsberg.ellen@gmail.com).

R. Maghsood is with Zenuity, Lindholmspiren 2, Göteborg, Sweden, 41756 (e-mail: roza.maghsood@zenuity.com).

E. Nordén was with the Department of Engineering Physics, Chalmers University of Technology, SE-412 96 Göteborg, Sweden (e-mail: eliza.norden@gmail.com).

Selpi is with the Division of Vehicle Safety, Department of Mechanics and Maritime Sciences, Chalmers University of Technology, SE-412 96 Göteborg, Sweden (e-mail: selpi@chalmers.se).

II. Background

Many different types of sensor measurements can be used for lane-level map matching, including

- GPS measurements: Longitude and latitude.
- Inertial measurement unit (IMU) data: Heading, speed, and yaw rate.
- Vision data: Type and geometry of the left and right lane markings, respectively. The geometry can be modeled as a polynomial or a clothoid (i.e., is a parametric curve whose curvature changes linearly with its curve length).
- Radar measurements: Radar reflections can be post-processed to identify various landmarks (such as traffic signs and guard-rails), e.g., one can first filter out stationary objects by analyzing the Doppler shift, and subsequently cluster spatially close echoes. Assuming that one can correctly associate clusters of radar echoes with landmarks in the HD map, such additional information should improve the accuracy of lane-level map matching.

Prior work has also used laser scanners [5] or high-precision GPS [6] to attack the lane-level map matching problem, but those types of sensors are still too costly for production vehicles. However, Rabe et al. demonstrated that it is possible to achieve error rates below 0.2% using GPS, IMU, vision, and radar [7]; They solved the lane-level map matching problem through least-squares optimization. Furthermore, [8] reported that road markings information could be exploited together with Simultaneous Localization and Mapping (SLAM) algorithm to reach lane-level map matching accuracy.

The idea of using state-space models for map-matching is not novel. Besides the work by Newson and Krumm that addressed road-level map matching [2], Li et al. used a particle filter to match GPS measurements to lanes in an HD map [9]. However, visual information was not included in [2], [9]. In a later work, Li et al. added visual information [10].

Specifically for HMM, it has been used by [2] for continuous road-level map matching, by [11] for identifying junction at road-level, and by [12] for mobile phone positioning. However, as far as we are aware, this is the first time that HMM is used for continuous lane-level map matching, with application relying on lane-level accuracy.

III. Model description

This section presents the model in detail, first recapitulating some general information about HMMs.

A. Hidden Markov models

HMMs are probabilistic models that can be used in signal processing, such as speech recognition and financial modeling [13]. The model can be represented by a bivariate Markov process $\{X_k, Y_k\}_{k=0}^{\infty}$. It assumes that the observation sequence $\{Y_k\}_{k=0}^{\infty}$ is generated by hidden process $\{X_k\}_{k=0}^{\infty}$ which is not possible to measure. Y_k is an observed sequence of independent random variables such that the conditional distribution of Y_k only depends on X_k.

The HMMs consist of three main parameters: initial state distribution, transition probabilities, and emission probabilities.

1) Initial state probabilities: The probability that the process starts in a state i is

$$\pi_i = P(X_0 = i),\ i = 1, 2, ..., N, \tag{1}$$

where $\sum_{i=1}^{N} \pi_i = 1$ and N is the number of states in the HMM.

To characterize an HMM, we need to estimate the model parameters. In the following sections, we first represent HMM for lane-level map-matching algorithm, and then we describe how to estimate the model parameters.

2) Emission probabilities: An emission probability is the conditional distribution of Y_k given X_k which represents the probability distribution of observation in each state,

$$p(y_k|x_k) = f_{Y_k}(y_k|X_k = i; \theta),\ i = 1, 2, ..., N, \tag{2}$$

where θ denotes the set of parameters of observation distribution.

3) Transition probabilities: The probability of transition from state i to state j at time k is

$$q(i, j) = P(X_{k+1} = j|X_k = i),\ i, j = 1, 2, ..., N, \tag{3}$$

where $\sum_{i=1}^{N} \sum_{j=1}^{N} q(i, j) = 1$.

B. HMM representation for lane-level map-matching

1) Hidden states: In the HD map, the roads are segmented into shorter lane groups, see Figure 1. Each of these groups contains one or multiple parallel lane segments, which are chosen as the HMM states, collectively denoted $\mathbf{X}$. A state i always represents a lane. If a lane splits up into two new lanes, the resulting lanes will be represented by two states. A state is constituted by the following variables,

- width,
- speed limit,
- heading, and
- left and right lane markings.

The states are assigned a geospatial ID, thus assuring the uniqueness property of an individual state.

2) Observations: The main observation is the noisy vehicle positions provided by GPS at 1 Hz frequency. In addition, a combination of odometry and vision data are used to compute the emission probabilities. Therefore, the multivariate observation sequence $\mathbf{Y}$ constitutes the following variables,

- longitude/latitude location measurements,
- speed,

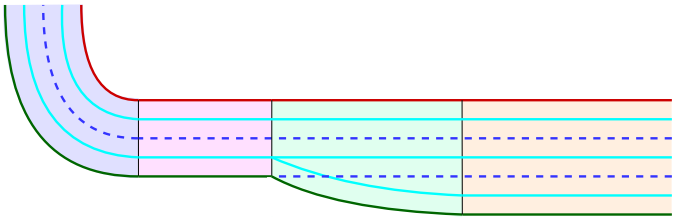


Fig. 1: Lane segments collected in different lane groups. The four colors, blue, pink, green and orange, show the lane groups containing two, two, three and three lanes respectively. The red solid line is the left lane marking of the one-way road, the green solid line is the right marking of the road and the blue dashed lane markings separate the individual lanes.

- heading,
- left and right lane markings,
- left and right lane change indications (received as a signal from the vehicle).

3) Emission probabilities: The emission probability $p(y_k|x_k)$ reflects how likely it is to observe GPS measurement y_k within the candidate lane $x_k \in \mathbf{X}$. To limit the number of lane candidates at each time, only lane segments within a 10 meters range from the observation are considered. While the GPS offset could reach beyond this limit, this number is a trade-off. A larger value would make the algorithm more tolerant to noisy GPS measurements but slower in execution time, while a smaller value more frequently would reject vehicle trajectories too far from the road network but would be faster in execution.

It is intuitive that the likelihood of observing y_k in lane x_k decreases with increasing distance between the GPS observation and the lane. To achieve this, we can refine the emission probabilities by introducing some penalty factors. As mentioned above, additional vehicle sensors and map data such as speed, heading and lane marker types, can be used to define penalty factors. The following paragraphs explicitly show the computation of emission probabilities by factor penalties.

a) GPS observation probability: According to Goh et al. [14], the probability of observing y_k being in lane x_k can be modeled as,

$$f_{Y_k}(y_k|x_k) = \frac{1}{w_{x_k}} \int_{-0.5w_{x_k}}^{0.5w_{x_k}} \frac{1}{\sqrt{2\pi\sigma_{\text{GPS}}^2}} e^{\frac{-(l-d)^2}{2\sigma_{\text{GPS}}^2}} dl, \tag{4}$$

where w_{x_k} is the lane width, d is the perpendicular distance between the GPS coordinate and the lane centerline, and σ_{GPS} is the standard deviation of the GPS error. As it has been shown to be a reasonable approximation by Newson and Krumm [2], the GPS error is assumed to have a Gaussian distribution with $\sigma_{\text{GPS}} = 4.07$. The integrand is integrated over variable l, which takes values in $[-0.5w_{x_k}, 0.5w_{x_k}]$.

b) Speed penalty: Goh et al. [14] proposed a factor penalizing for speed based on the hypothesis that drivers do not tend to largely exceed road speed limits. The penalty is given by,

$$S(x_k, y_k) = \frac{v_{x_k}}{v_{x_k} + \max(0, v_{y_k} - v_{x_k})}, \tag{5}$$

where v_{x_k} is the speed limit on lane x_k, and v_{y_k} is the speed registered for observation y_k.

When there are parallel roads located close to each other with different speed limits, this could diminish the number of possible candidate lanes [14].

c) Heading penalty: A vehicle will generally maintain about the same heading as the road it traverses. When there are different classes of closely located roads, e.g. urban areas or overpasses, candidate road lanes could be filtered by considering the vehicle heading compared to the lane headings. A heading penalty can thus be introduced according to,

$$H(x_k, y_k) = \begin{cases} 0 & \text{if } |h_{y_k} - h_{x_k}| \geq 90°, \\ \frac{h_{x_k}}{h_{x_k} + |h_{y_k} - h_{x_k}|} & \text{if } |h_{y_k} - h_{x_k}| \geq 20°, \\ 1 & \text{otherwise,} \end{cases} \tag{6}$$

where h_{y_k} is the heading registered for observation y_k, and h_{x_k} is the heading of the lane x_k. The heading for an observation is the direction in which the vehicle front is pointing, while the heading of a lane is the travel direction. If the heading difference is larger than $90°$, it means that the vehicle is either driving perpendicular to the lane or in the opposite direction. In such scenarios, it is not very likely that the vehicle is actually driving on the considered lane, hence the imposed penalty. When studying observation data corresponding to lane changes for the experimentation data set, it was established that the vehicle heading and the lane heading can differ up to $20°$ during a lane change. Therefore, no penalty is applied when the heading difference $|h_{y_k} - h_{x_k}|$ is smaller than $20°$.

d) Marker type penalty: The vision system equipped in the vehicle is able to detect lane markings on the road with three levels of confidence; $0, 1$ and 2, where 0 corresponds to the lowest confidence and 2 the highest confidence. For lane-level map matching algorithms, this information is essential since it can facilitate lane distinctions. As shown in Figure 2, a vehicle traverses a two-lane road where the marker types are not the same for the left- and rightmost lanes. The rightmost lane, which the vehicle traverses, has a dashed left marker and a solid right marker, while the leftmost lane has the same markers but in reversed order. In the example shown in Figure 2, if the vehicle vision sensor correctly reports the type of markers, there is no ambiguity regarding which lane the vehicle is in since there are only two lanes and they have different marker types.

For each observation y_k, there are two identified marker types for left and right lanes. We define two variables $p_{\textit{left equal}}$ and $p_{\textit{right equal}}$ for maintaining the probabilities of detecting the true marker types given the marker types of candidate lane x_k,

$$p_{\textit{left equal}}(x_k, y_k) = \\ = P\left(\textit{left marker type}_{x_k} = \textit{left marker type}_{y_k} \,\middle|\, \textit{left marker type}_{y_k}\right), \tag{7}$$

$$p_{\textit{right equal}}(x_k, y_k) = \\ = P\left(\textit{right marker type}_{x_k} = \textit{right marker type}_{y_k} \,\middle|\, \textit{right marker type}_{y_k}\right), \tag{8}$$

where $\textit{left marker type}_{x_k}$ and $\textit{right marker type}_{x_k}$ are the left and right marker types of the lane, and $\textit{left marker type}_{y_k}$ and $\textit{right marker type}_{y_k}$ are the left and right types registered by the vehicle. The probabilities in Eqs. (7) and (8) are estimated for the different marker types using a tuning data set.

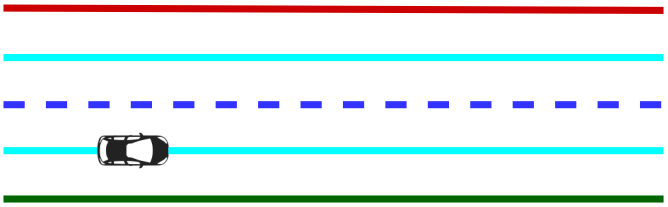


Fig. 2: The green and red lines represent the right and left markers at the edge of the one-way road. The blue dashed line represents a marker separating two lanes heading in the same direction, and the cyan-colored lines mark the centerlines of the two lanes. The vision system in the vehicle, traversing in the right lane, will report a dashed left marker and solid right marker. This excludes the possibility for the vehicle to be in the left lane.

Using Eqs. (7) and (8) and incorporating the confidence registered for every observation yield the following expressions,

$$\begin{aligned} &\textit{left penalty}(x_k, y_k, c) \\ &= (1 - p_{\textit{left equal}}(x_k, y_k)) \times \left(1 - c \times \frac{\textit{left confidence}_{y_k}}{\textit{max confidence}}\right) \\ &+ p_{\textit{left equal}}(x_k, y_k) \times \left(1 + c \times \frac{\textit{left confidence}_{y_k}}{\textit{max confidence}}\right), \end{aligned} \tag{9}$$

$$\begin{aligned} &\textit{right penalty}(x_k, y_k, c) \\ &= (1 - p_{\textit{right equal}}(x_k, y_k)) \times \left(1 - c \times \frac{\textit{right confidence}_{y_k}}{\textit{max confidence}}\right) \\ &+ p_{\textit{right equal}}(x_k, y_k) \times \left(1 + c \times \frac{\textit{right confidence}_{y_k}}{\textit{max confidence}}\right). \end{aligned} \tag{10}$$

The confidence levels $\textit{left confidence}_{y_k}$ and $\textit{right confidence}_{y_k}$ are reported by the vision sensor equipped in the vehicle, and *max confidence* represents the maximum value of the confidence levels.

c is a scale parameter which was selected by evaluating the performance of the algorithm for different values of $c \in [0, 3]$. It was found that the algorithm performs best when $c = 1$ in the tuning data set used in this work. These penalties take values in the interval $(1-c, 1+c)$, and clearly, the upper limit of the interval is larger than 1 for $c > 0$. As such, it cannot be considered to be a true penalty, but rather a penalty/bonus factor.

When the camera confidently reports that the left or right lane marker type is not the same as the types for some lane segment, the penalties will have a low value. Correspondingly, when it confidently reports that the marker types are equal, the penalty values are higher. Again, this is not a true penalty, and higher penalty values do not mean that confident camera sightings are punished. It rather means that the corresponding probabilities are up-scaled by the higher penalty, which then might be better referred to as a bonus factor.

The total lane marker type penalty is then obtained as the average of Eqs. (9) and (10),

$$M(x_k, y_k) = \frac{\textit{left penalty}(x_k, y_k, c) + \textit{right penalty}(x_k, y_k, c)}{2}. \tag{11}$$

e) Final emission probability: Having the observation probability defined according to Eq. (4) and the penalties in Eqs. (5), (6) and (11), the emission probability is given by the product of all these factors. Thus, the probability of observing y_k on lane x_k is calculated as

$$p(y_k|x_k) = f_{Y_k}(y_k|x_k) \times S(x_k, y_k) \times H(x_k, y_k) \times M(x_k, y_k). \tag{12}$$

Since $M(x_k, y_k)$ takes values in $(0, 2)$, the final emission probability risks resulting in a value larger than 1. This is dealt with by truncating the emission probability at 1.

4) Transition probabilities: The transition probability $p(x_{k+1}|x_k)$, according to Eq. (3), describes the likelihood of moving from lane x_k to lane x_{k+1} at time k.

In this paper, we model the transition probabilities by considering the connectivity of the road networks in HD map. This means that every lane will have a transition to lanes in a local neighborhood. As when deriving the emission probabilities, the time efficiency of the algorithm is improved by limiting the number of lane candidates at each time. By this, only lanes within a 10 meter range from the observation at time k and the observation at time $k + 1$, respectively, are considered.

In order to estimate the transition probabilities, we have considered different connectivity levels for the lanes. The level of connectivity concerns how closely the lanes are connected. A lane x_k and all the lanes located laterally next to it, are denoted by $\mathbf{X}^0$, which are represented in purple in Figure 3. Lanes located next to the lanes which follows directly after lanes in $\mathbf{X}^0$ are denoted by $\mathbf{X}^1$, colored green in Figure 3. Continuing in the same manner, lanes located at depth d from a lane, are denoted by $\mathbf{X}^d$, colored yellow in Figure 3, where depth indicates the number of intermediate lanes between lane x_k and lanes in $\mathbf{X}^d$.

We assume that a vehicle can move from its current lane to lanes in $\mathbf{X}^0, \ldots, \mathbf{X}^d$ with probabilities $p_0 \ldots, p_d$, respectively. Further, we assume that the probabilities of moving to lanes that are farther away (i.e., lanes at increasing depth) is decreasing, therefore the probabilities are assigned such that $p_0 > p_1 > \ldots > p_d$.

By this reason, the probabilities of transitioning from $x_k \in \mathbf{X}$ to lanes $x_{k+1} \in \mathbf{X}^d$ can be modeled as,

$$p(x_{k+1}|x_k) = \frac{D - d}{D}, \tag{13}$$

where D is the maximum depth. However note that the transition probability will be 0 for $d = D$, so the maximum depth for which the transition probabilities are larger than 0 is actually $D - 1$.

Depending on the output of the lane change indicator, the probabilities of transitioning into left, right or succeeding lanes are increased. A schematic view of the left, right and succeeding neighborhoods is shown in Figure 4. The probabilistic increase is defined by,

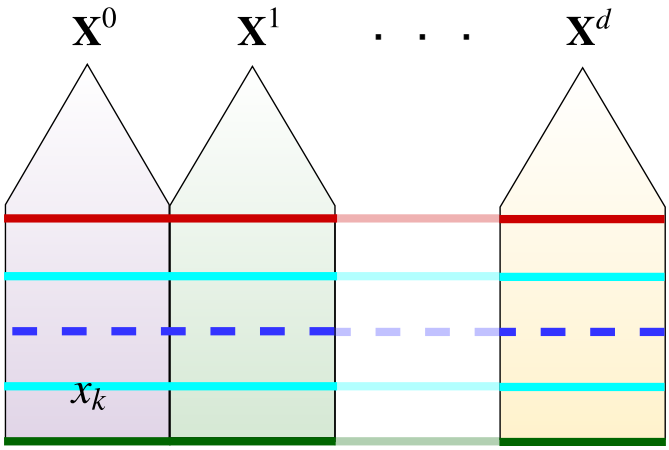


Fig. 3: The lanes which lane x_k is directly or indirectly connected to can be grouped into sets $\mathbf{X}^0, \mathbf{X}^1, \ldots, \mathbf{X}^d$, where $0, 1, \ldots, d$ reflect on which level the lanes are connected. The green and red lines represent the right and left markers at the edge of the road. The blue dashed line represents a marker separating two lanes heading in the same direction, and the cyan-colored lines mark the centerlines of the lanes.

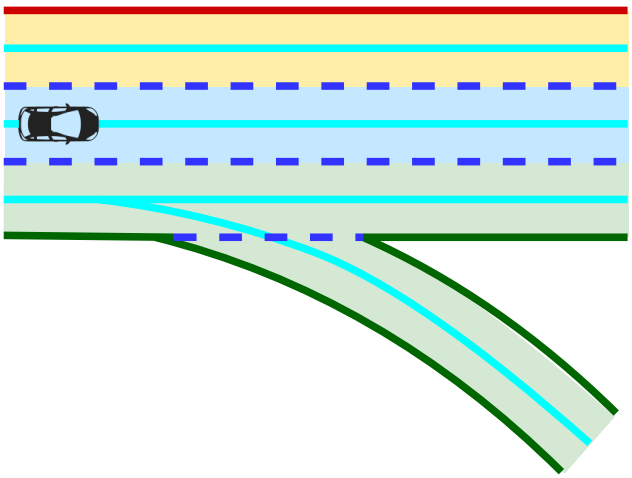


Fig. 4: Left (yellow), right (green) and succeeding (blue) lane neighborhoods by color from the car's point of view.

$$p(x_{k+1}|x_k) = p(x_{k+1}|x_k) + 0.5, \quad \text{for all } x_{k+1} : p(x_{k+1}|x_k) > 0, \tag{14}$$

if x_{k+1} is a left/right neighbor or successor to x_k, and a left/right or no lane change has been detected. The term 0.5 is the increase of transition probability, which was chosen by evaluating the performance of the algorithm for different probability increases in the range $(0, 1]$. The transitions are then normalized to true probabilities (such that their sum add up to one).

If, for instance, the map has a high level of degradation and is missing lanes, the connectivity-based transition model proposed above will not capture all true transitions. In cases like these, it may be desirable to instead model the transitions

by other spatial transition models. These have the benefit of not making any assumptions regarding the traversability of the roads.

One such method is to sample points in one lane x_k, and predict their position in the next time step by extrapolating them into the spatial neighborhood of x_k. These extrapolations can be decided by sampling speed and headings from some distributions based on the current speed limit and lane heading. The transition probability is then taken as the relative frequency of extrapolating from lane x_k into lane x_{k+1}.

C. Finding the most likely lane trajectory

The Viterbi algorithm (VA) is employed to find the most likely sequence of traversed lanes [3]. The time complexity of VA is $\mathcal{O}(KN^2)$, where K is the number of observations and N the number of states. For the purpose of lane-level map matching, K is the number of time steps when vehicle observations were registered, and N the number of lanes.

We will make two adaptations to the standard implementation of the Viterbi algorithm, as described below.

1) Adaptation 1: Handling HMM breaks: If the map is not connected, there may not be any possible transition from the states at time k to the states at time $k+1$, which would break the HMM chain. Such a break could also occur if the GPS error is large enough so that no lane geometry is retrieved when querying the HD map for close geometries. Similarly to Newson and Krumm, we handle breaks by matching to an additional state, which we label "No Lane," and then we let the VA operate on each separate chain. The results are subsequently concatenated into a complete path.

2) Adaptation 2: Online decisions: In offline implementations, the backtracing operation of VA is not run until the forward recursion has finished (at the end of the data block). This leads to a decision delay that grows linearly in the block length, which is not acceptable in online applications of lane-level map matching, such as ADAS and AD. Therefore, similarly to Goh et al. we will use an online version of the VA, in which a sliding window limits the decision delay [14]: If all survivor edges at a given time index point back to the same state, a so-called *convergence point*, the backtracing operation can commence from the convergence point without loss of optimality. This technique is called *variable sliding window*, which reflects that the decision window grows for each new observation and shrinks from behind as soon as a convergence point is encountered. Still, until a convergence point appears, the sliding window keeps growing, leading to an ever longer decision delay. Thus, in order to support a limited decision delay, we can restrict the window size and make (possibly sub-optimal) decisions at a certain lag from the current time index.

IV. Data

A. Map

The HD map is provided by TomTom and covers a ring road (or beltway) around the city of Gothenburg, Sweden. This map has information about HD-specific objects such as lanes, lane markings, barriers, traffic signs and vertical poles. More specifically, it has information about width, speed limit and geometry of individual lanes, type of lane markings, e.g. dashed, solid etc., as well as their geometries, speed restrictions specified on traffic signs and so on. While there is lots of information contained in the map, only a subset of it is utilized in our proposed map matching algorithm — namely information concerning attributes of lanes and markings.

To access the HD map data efficiently, we first build an R-tree of the HD map lane geometries and pack it using the sort-tile-recursive (STR) algorithm [15], [16]. This allows us to perform spatial queries on lanes in an efficient manner.

B. Sensor

The vehicle sensor data is sampled through Volvo's Drive Me project [17]. In total the driving data used for the experiments in this paper corresponds to 63 hours of driving collected in 2017. This corresponds to a set of drives lasting between 1-12 minutes, covering 10-10000 meters. This data is divided into three sets, one meant for tuning parameters in the model, an experimentation set and a test set to evaluate the algorithm. Besides an inertial navigation system, the test vehicles were equipped with a lane change indicator and a forward looking camera, capable of detecting and classifying lane markings. GPS samples are recorded at 1 Hz, while the vision system reports lane marking information at a frequency of 40 Hz. For simplicity, we have decided not to assist the GPS by dead-reckoning with the help of IMU data. Consequently, we have excluded tunnels in our analysis. It should however be noted, that IMU data is used to infer e.g. the vehicle heading.

1) Ground truth: In order to later evaluate the performance of the proposed algorithm, it is necessary to derive the ground truth. Through real time kinematics (RTK), an additional set of GPS coordinates were provided. With centimeter-level precision, the RTK coordinates can be assumed to be the true vehicle positions. As such, the ground truth states are identified as the lane segments in which the RTK points lie. In the case of multiple valid lanes, as for overpasses, filtering is performed on the lane and vehicular heading. If the RTK observation is not contained by any known lane, the lack of match is noted.

2) Parameter estimation: The lane marker type penalty is associated with the two parameters in Eqs. (7) and (8) that describe the likelihood of the vision system outputting correct lane markings. Given the ground truth lane markings, these probabilities can be measured as the ratio of correctly identified markings in the data set. These are not assumed to be independent of the underlying lane marking type. Nor are they assumed to be independent of the reported confidence levels. The probabilities are thus estimated for all lane markings and all vision confidences.

V. Performance metrics

We will now discuss various ways of assessing the proposed algorithm.

A. Classification of matched lanes

By comparing the estimated path with the true path, given by the VA and ground truth, respectively, we can assess the

accuracy of our proposed lane-level map matching model. Estimated lanes that correctly match ground truth are classified as *true positives* (TP). Estimated lanes that are not present in ground truth are called *false positives* (FP), while lanes in ground truth that are not present in the estimated path are referred to as *false negatives* (FN). Formally, given an estimated path $(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_K)$ and the true path $(x_1, x_2, \ldots, x_K)$, we define the three classes

$$\mathrm{TP} = \{\hat{x}_i : \quad \hat{x}_i = x_i\}, \tag{15}$$

$$\mathrm{FP} = \{\hat{x}_i : \quad \hat{x}_i \neq x_i\}, \tag{16}$$

$$\mathrm{FN} = \{x_i : \quad \hat{x}_i \neq x_i\}. \tag{17}$$

B. Recall

Having defined classification sets (15)–(17), the recall of an estimated path can be evaluated as

$$\text{recall} = \frac{|\,\mathrm{TP}\,|}{|\,\mathrm{TP}\,| + |\,\mathrm{FN}\,|}, \tag{18}$$

where $|\,\mathrm{TP}\,|$ and $|\,\mathrm{FN}\,|$ are the number of true positives and false negatives, respectively.

C. Path length error

Inspired by the loss function proposed by Newson and Krumm, we also compute the so-called *path length error* (PLE) [2]. Similarly to their metric, the PLE is the fraction of the length of the incorrect route, but at lane-level resolution. For an ideal map matcher, the PLE value is zero. Its formal expression is

$$\mathrm{PLE} = \frac{\sum\limits_{\tilde{x}_i \in \mathrm{FP}} \mathrm{length}(\tilde{x}_i) + \sum\limits_{\tilde{x}_i \in \mathrm{FN}} \mathrm{length}(\tilde{x}_i)}{\sum\limits_{\tilde{x}_i \in \mathrm{TP}} \mathrm{length}(\tilde{x}_i) + \sum\limits_{\tilde{x}_i \in \mathrm{FN}} \mathrm{length}(\tilde{x}_i)}, \tag{19}$$

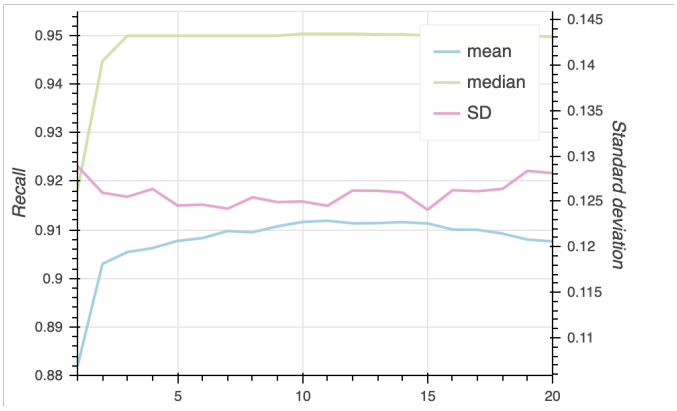
where $\mathrm{length}(\tilde{x}_i)$ is the length of lane segment $\tilde{x}_i$. Lane segment $\tilde{x}_i$ differs from the original lane x_i in that it has been segmented based on the GPS locations of the matched observations.

VI. Results

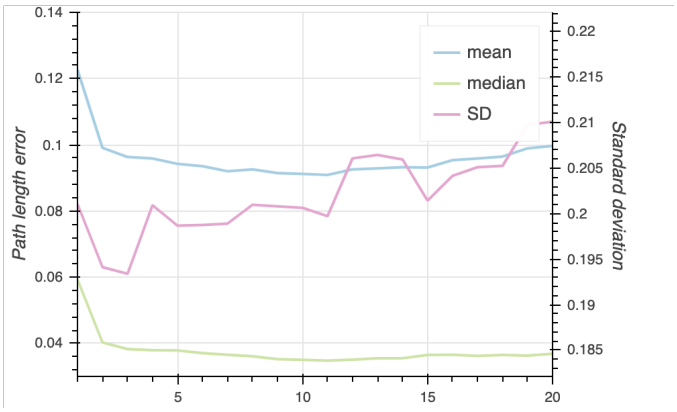
Let us now investigate the performance of the proposed model. Note that the proposed algorithm has the repeatability property, meaning that the results we present here can be reproduced by using the same test data set and having the same model setup.

A. Tuning neighbor search depth

As described in Section III-B4, transition probabilities vary depending on the maximum depth considered. Generally, a smaller depth means that fewer transitions are considered, while larger depth means more transitions. It is not intuitively obvious which depth yields optimal result in terms of the performance metrics presented in Section V. A small depth truncates the candidate space and may lead to cases where a transition is entirely missed. On the other hand, a large depth may amplify the errors in the underlying assumptions, leading to highly unlikely transitions. Therefore, an experiment was done to choose which depth to use.



(a) Recall



(b) Path length error

Fig. 5: Performance of algorithm using connectivity-based transition probabilities with different depths.

The performance of the model when using transition probabilities with different depths ranging from 1 to 20 was measured. The performance in terms of recall and PLE using various depths can be seen in Figure 5. Looking at the recall, its mean reaches 0.9119 at best, which corresponds to the peak of the blue curve in the figure at depth 11. The median improves with increasing depth from 1 to 3, whereon it remains at 0.95 for all succeeding depths. The standard deviation remains within a tight interval for all depths and only has local fluctuations.

Considering the PLE, given in Figure 5b, its mean reaches 0.0909 as the lowest at depth 11. Similarly to the median recall, the median PLE decreases for the initial depths and then remains quite stable around 0.035. The standard deviation clearly has an increasing trend with larger depth.

From these results, it can be decided that the algorithm yields the best result for transition probabilities yielded with search depth 11. Note that this is the optimal search depth for the map used in this work. When using another map which might have another approach to split lanes into segments, this number has to be re-estimated.

TABLE I: Results on the test set. ”Std” stands for standard deviation.

	Recall			PLE		
	Mean	Median	Std	Mean	Median	Std
Model	0.914	0.951	0.134	0.098	0.033	0.220
Benchmark	0.721	0.767	0.223	0.383	0.258	0.386

B. Final result

After fixating the search depth, the final model can be defined and run on the test set. The results are summarized in Table I and are the final results of the map matching algorithm proposed in this paper. Additionally, the table includes results obtained by using a benchmark map matcher, which will be described next.

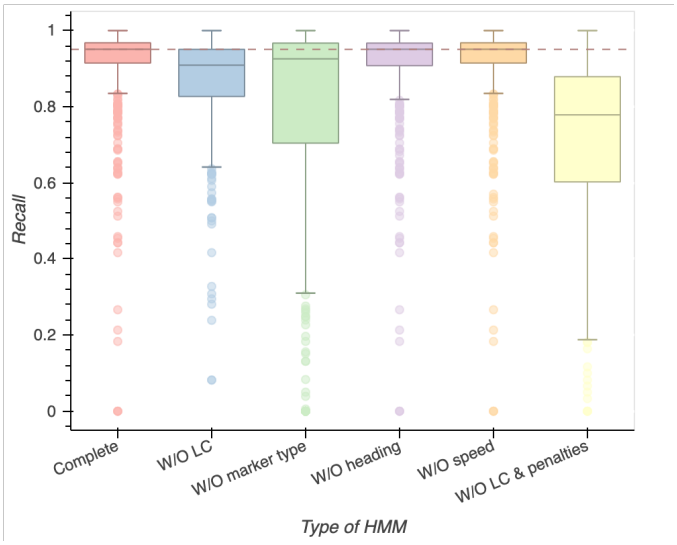
The benchmark algorithm (the so called naive match matcher) used here only considers GPS location measurements as input and matches them to their nearest lane within a 10 meter radius. The results of this model are also included in Table I.

C. Discussion

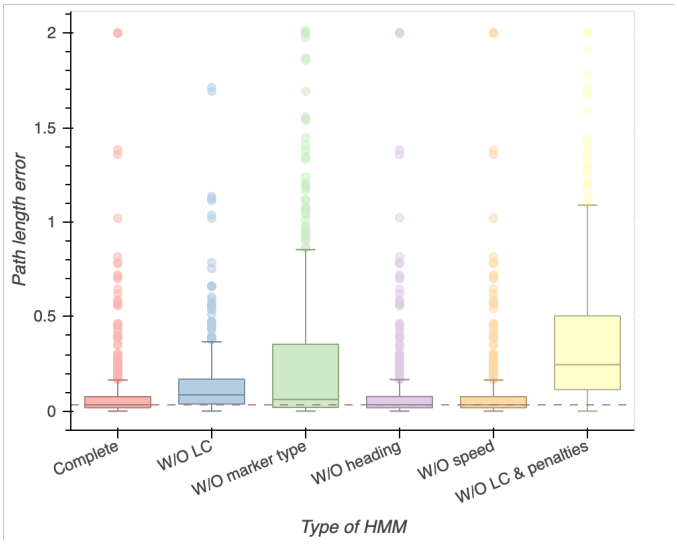
Comparing the results of the model against other algorithms aiming to solve lane-level map matching is not straightforward. Primarily, different performance metrics are used as well as different map and sensor data. For example, the lane-level map matching algorithm proposed by Rabe et al. [7] cannot be compared with the algorithm proposed here because they used RADAR data, which is excluded in this work. Adding redundant RADAR data to our model would arguably make our algorithm less dependent on the vision system and make the algorithm more competent at handling failing lane marker type detections.

The result of 95.1% median recall and 3.3% median PLE shown in Table I can only be compared to the benchmark map matcher reported in Section VI-B, for which all data and map conditions remain the same. From the performance gain in the quality measures, it can be concluded that our map matcher gives better results, as indicated by the increase of 18.4 percent units in median recall and 22.5 percent units decrease of median PLE.

1) Variable importances: The HMM observations were incorporated into the model in various ways. The proposed transition probabilities do not only use the current and next states as inputs, but also the value of the lane change variable. As such, the transitions are dynamic. Other variables, i.e. speed, heading and left, right lane markings, affect the map matching via penalties, based on assumptions about the driving characteristics.



(a) Recall



(b) Path length error

Fig. 6: Box plots of the performance on the test set (consisting of 1107 individual drives) after down-scaling the model. From left to right, the resulting models are; the complete model (red), a model without added lane change transitions (blue), without added lane marker type penalty (green), without heading penalty (purple), without speed penalty (orange), and a model without any added penalties or lane change transitions (yellow). The boxes are drawn from the first quartile (25th percentile) to the third quartile (75th percentile) and the horizontal lines in the boxes indicate the median (50th percentile) PLE. The whiskers show the minimum and maximum PLE noted for the various models among the 1107 drives. The dots located outside the whiskers are outliers. The dashed horizontal line shows the median score of the complete model.

Since not all variables may be readily available at all times, an investigation of their effect on the algorithm performance is conducted. Model down-scaling is performed by omitting each of the penalties and the lane change transitions. These simplified models are then evaluated on the test set. The

resulting box plots are shown in Figure 6.

One can see from Figure 6 that removing the lane change signals from the transition probabilities (i.e., this makes the probabilities of driving on the left, same, or right of the current lane are equally likely according to the transition probabilities even though the vehicle signals a lane change) yields the biggest decrease in performance of the algorithm. This is not surprising because the ability to capture vehicle movements between parallel lanes is essential for a lane-level map matching algorithm and that GPS measurements cannot be reliably used to detect lateral movement of the vehicle. From the same figure, one can also see that excluding the lane marker penalty from the model makes the median recall of the algorithm drops with a few percentage points. The drop is not as significant as when removing the lane change signals from the transition probabilities. This might be explained by the fact that the marker types are not always unique for parallel lanes and that the reported lane markings cannot always be trusted, even when the sensor is very confident. The same figure also shows that exclusion of the heading and speed penalties do not really affect the performance of the algorithm. This is understandable because parallel lanes are almost always have the same speed limits and lane headings. It can be concluded from Figure 6 that the lane changes and lane markings are important to the algorithm, while speed and headings have a negligible effect. However, if the map data used in this work would contain not only highways, but also areas of densely located roads having different headings and speed restrictions, one can assume that the algorithm would benefit from including both the speed and heading penalties.

2) Identified strengths/weaknesses: The ability of the algorithm to handle certain circumstances are discussed in this section.

a) GPS error: The GPS observation probability presented in Section III-B3 gives the probability distribution of the distance between a candidate lane and the perceived vehicle (i.e. GPS) location. As the GPS error gets larger, the observation probability of the true lane becomes smaller.

When the GPS error is within a few meters while all other vehicle sensors perform ideally, the algorithm generally has no difficulties to accurately identify the traversed lanes. For example, when the GPS position is less than eight meters off the true position, and the reported lane marker types can be assumed mostly correct, the model has recall above 80%. However, when the GPS error gets larger than 8 meters, the true lane becomes a highly unlikely candidate, even if the lane markers are correctly identified. In other words, the observation probability is more influential than the lane marker type penalty. The result is an incorrectly matched path.

On the other hand, when the GPS error is only moderately wrong (up to four meters) as in Figure 7, the algorithm succeeds at mapping to the correct lane. This implies that there is a limited tolerance for bad GPS input. The high scores on the test set indicate that the GPS error is normally contained within that tolerance.

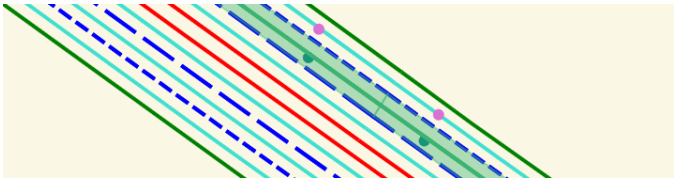


Fig. 7: An example where the correct lane-path (colored green) is found with the help of the lane marker type penalty, even though the GPS points are located in a different lane. The vehicle GPS measurements are shown in pink circles and the ground truth positions are represented by the blue circles. The green and red lines show the markings at the right and left road edges, respectively. The blue dashed lines are markers separating individual lanes and the cyan-colored lines mark the center of the lanes.

b) Uncertain vision detections: No lane marker type penalty, given in Section III-B3d, is enforced during drives with uncertain vision detections. In that case, only the GPS measurements are used to identify the traversed lanes. In combination with erroneous GPS location measurements, the path returned by the algorithm is likely incorrect. Inclusion of RADAR data as input data could reduce the dependence of the algorithm on the vision system, as shown by [7]. Such investigation could certainly be an interesting task for future work.

c) Lane change detections: Incorporation of the lane change indicator in the transition probabilities, as described in Section III-B4, helps our algorithm both when actual lane changes are detected as well as when no lane changes are detected. In the latter case, the information of no lane changes helps the algorithm to follow a straight path rather than change lanes according to noisy GPS locations.

Nonetheless, lane change transitions in the transition probabilities can in some situations cause undesired output. This may happen in situations where the GPS measurements are faulty and lane marker types are missing or reported incorrectly. Consider a situation where faulty lane matching was assigned in the beginning of a drive due to inaccurate location measurements. Although the location measurements become more accurate over time, the transition model still favors a straight path from the previous incorrectly matched lanes. This behavior originates from the enhanced transition model, which disfavors lateral movements unless the lane change detector signals a positive lane change. Combined with the lack of lane marker detections, this transition model can become too inflexible.

d) Ambiguous lane marker types: When there is more than one lane on a road where at least two lanes have the same marker types, information picked up by the vision system is not sufficient to uniquely determine which lane the vehicle is in. Consider a case (see Figure 8) when the vehicle correctly identifies dashed lane markers on both sides. However, the road contains three lanes, whereof two of them have dashed markers on both sides. According to the vision system, the vehicle could be in any of the two lanes with the same markers. If the GPS measurements would have low error and place the vehicle in the actual lane, ambiguous lane marker types would

not provide any problems. Now, the whole problem of map matching is motivated by the inability to derive paths solely from inaccurate GPS measurements. Thus, it is primarily interesting to consider how the algorithm performs where the location measurements are not perfect. In lack of additional sensors with spatial awareness, there is no obvious way to resolve the ambiguity.

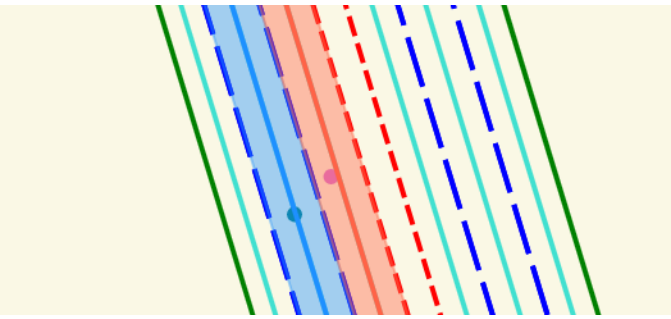


Fig. 8: An example where more than one lane have the same lane marker types (left and right lane markers for both coloured lanes are of a dashed type), making it difficult for the vision system to uniquely determine which lane the vehicle is in. The blue-colored lane corresponds to the ground truth vehicle lane path, while the red-colored lane shows the path retrieved by our proposed algorithm. The vehicle GPS measurements are shown in pink circles and the ground truth positions are represented by the blue circles. The green and red lines show the markings at the right and left road edges, respectively. The blue dashed lines are markers separating individual lanes and the cyan-colored lines mark the center of the lanes.

Inspired by [9], [10] who address ambiguity issues by letting the map matching algorithm output multiple lane hypotheses, a possible direction of future work is to investigate if our algorithm could be modified to output more than the most likely path (i.e., to also output the second, third, etc., probable paths). The correct path would then need to be verified from these hypotheses by other means (e.g., as in [10]).

D. Comparison

The outcome of our lane-level map matching algorithm can be directly compared to the results maintained from the road-level map matcher introduced in [2]. Figure 9 shows the result of map matching a sequence of vehicle coordinate observations to road segments in a map, colored purple. As the underlying map has HD-features, it shows the number of lanes on the various road segments, separated by lane markings. The road segment in the upper left corner of the figure has four parallel lanes, but the road-level map matcher is not able to identify which of the four lanes the vehicle is in — only that it is located somewhere in the area covered by all four lanes. Further down the road, the four lanes are split up into two road segments containing two lanes each. Still, the road-level map matcher can only say in which road segment the vehicle is, and not distinguish between the individual lanes.

The outcome of applying our lane-level map matching algorithm on the same instance can also be seen in Figure 9, where the matched lanes are colored orange. As can be seen, the vehicle coordinates are matched to individual lanes in the underlying map, providing a higher resolution of the traversed path compared to that returned by the road-level map matcher (indicated by the lanes in purple). Combining the result of both map matchers clearly show the enhanced resolution provided by our proposed algorithm.

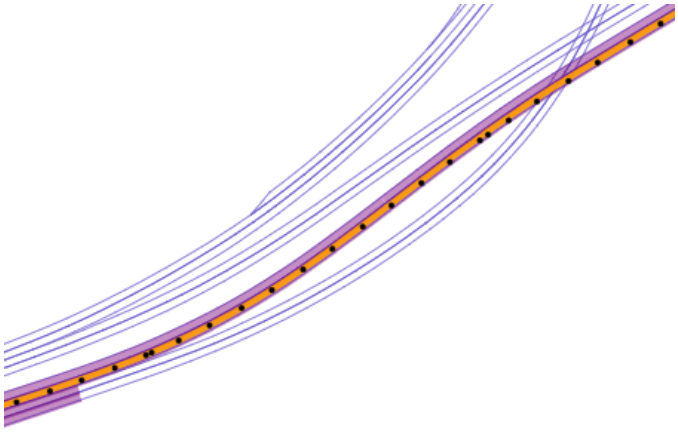


Fig. 9: Estimated road-level path (in purple) and lane-level path (in orange) given the true vehicle coordinates (black circles). The blue lines are lane markings in the underlying map.

E. Future work

Our algorithm has only been tested on motorway scenarios, so it should also be tested on in-city conditions. Also, the algorithm relies on information from a vision system. It would be interesting to study how other landmarks such as traffic signs and road barriers could make the algorithm more robust when information on lane marker types is not reliable.

VII. Conclusions

In this paper, we have investigated the usefulness of HMMs to identify the lanes traversed by a vehicle. Our implementation makes use of an HD map, in which the lane segments are the hidden states of the model. Apart from noisy GPS data, the proposed model relies on a lane-change signal as well as visual information about nearby lane markings.Speed and headings information were found to have a negligible effect on the algorithm. Our results showed that 95.1% median recall and 3.3% median path length error can be obtained for motorway scenarios. Future work will test the algorithm on in-city conditions, and include RADAR data and landmarks to increase information redundancy.

References

[1] P. Chao, Y. Xu, W. Hua, and X. Zhou, "A survey on map-matching algorithms," in *Proceedings of the 31st Australasian Database Conference (Lecture Notes in Computer Science: Databases Theory and Applications)*, R. Borovica-Gajic, J. Qi, and W. Wang, Eds. Cham, Switzerland: Springer Nature Switzerland AG, 2020, pp. 121–133.

[2] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Seattle, WA, 2009, pp. 336–343.

[3] G. David Forney, Jr., "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.

[4] A. Luo, S. Chen, and B. Xv, "Enhanced map-matching algorithm with a hidden markov model for mobile phone positioning," *ISPRS International Journal of Geo-Information*, vol. 6, pp. 327–343, 2017.

[5] R. Matthaei, G. Bagschik, and M. Maurer, "Map-relative localization in lane-level maps for adas and autonomous driving," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 49–55.

[6] Jie Du and Matthew J. Barth, "Next-generation automated vehicle location systems: Positioning at the lane level," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 48–57, March 2008.

[7] J. Rabe, M. Meinke, M. Necker, and C. Stiller, "Lane-level map-matching based on optimization," in *IEEE 19th International Conference on Intelligent Transportation Systems*, Rio de Janeiro, Brazil, 2016, pp. 1155–1160.

[8] J. Jeong, Y. Cho, and A. Kim, "Road-slam: Road marking based slam with lane-level accuracy," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA)*. IEEE, 2017, pp. 1736–1773.

[9] F. Li, P. Bonnifait, J. Ibanez-Guzman, and C. Zinoune, "Lane-level map-matching with integrity on high-definition maps," in *IEEE Intelligent Vehicles Symposium*, Los Angeles, CA, 2017, pp. 1176–1181.

[10] F. Li, P. Bonnifait, and J. Ibanez-Guzman, "Map-aided dead-reckoning with lane-level maps and integrity monitoring," *IEEE TRANSACTIONS ON INTELLIGENT VEHICLES*, vol. 3, no. 1, 2018.

[11] H. Qi, X. Di, and J. Li, "Map-matching algorithm based on the junction decision domain and the hidden markov model," *PLOS ONE*, 2019.

[12] A. Luo, S. Chen, and B. Xv, "Enhanced map-matching algorithm with a hidden markov model for mobile phone positioning," *ISPRS Int. J. Geo-Inf.*, vol. 6, no. 327, 2017.

[13] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. New York, NY: Springer, 2005.

[14] C. Goh, J. Dauwels, N. Mitrovic, M. T. Asif, A. Oran, and P. Jaillet, "Online map-matching based on hidden markov model for real-time traffic sensing applications," in *15th International IEEE Conference on Intelligent Transportation Systems*, Anchorage, AK, 2012, pp. 776–781.

[15] S. Leutenegger, M. Lopez, and J. Edgington, "Str: A simple and efficient algorithm for r-tree packing," in *Proceedings 13th International Conference on Data Engineering*, Birmingham, UK, 1997, pp. 497–506.

[16] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, 1984, pp. 47–57.

[17] V. Cars. Drive me. [Online]. Available: https://www.volvocars.com/intl/buy/explore/intellisafe/autonomous-driving/drive-me

Anders Hansson (S'98–M'03–SM'09) received the MSc degree from Chalmers University of Technology in 1996. In 1997 he was working with microlithography technology in Stockholm and Tokyo. He enrolled in a doctoral program in 1998, was the 2000/2001 Sweden-America Foundation Fellow at University of Southern California, before he was awarded a PhD degree from Chalmers in 2003. He continued to follow a nonlinear trajectory to Los Alamos National Laboratory, where he was promoted to Team Leader, conducting research on analysis and modeling of large-scale socio-technical systems. His scientific contributions were recognized with a Docent degree (habilitation) at Chalmers in 2008. That same year, he joined Ericsson as a System Manager. Since 2017 he has been working for Zenuity, developing machine learning algorithms to maintain, update, and create HD maps for autonomous driving.

Ellen Korsberg received an MSc degree in computer science from Chalmers University of Technology in Sweden in 2019. Her master's thesis was carried out at Zenuity, where she and her thesis partner implemented a lane-level map matcher based on Hidden Markov Models. She currently works as a consultant at Zenuity, on behalf of Ictech.

Roza Maghsood received her PhD in Mathematical Statistics in 2016 at the Chalmers University of Technology, with a thesis on hidden Markov models for detecting steering events and evaluating fatigue damage. She is currently a big data analyst at Zenuity company, working with map data processing concepts and tools.

Eliza Nordén received an MSc degree in complex adaptive systems from Chalmers University of Technology in Sweden in 2019. Her master's thesis was carried out at Zenuity, where she and her thesis partner implemented a lane-level map matcher based on Hidden Markov Models. She is currently working at Jeppesen through Sigma Young Talent.

Selpi received a PhD degree in computing from the Robert Gordon University in the UK in 2008, an MSc degree in bioinformatics from Chalmers University of Technology in Sweden in 2004, and a BSc degree in computer science from the University of Indonesia in 2000. She currently works at Chalmers University of Technology. Her current research interests include applications of machine learning and data science for transport and traffic safety domain, and understanding how mixed traffic, with vehicles with different driving styles and automation levels sharing the same roads, affects traffic safety and efficiency. Beside academic work, she has several years of experiences in software industry. Dr. Selpi is a member of the IEEE Intelligent Transportation Systems Society's technical committee on Naturalistic Driving Data Analytics. She has served as a reviewer and an associate editor for several IEEE conferences for years.