

EXTRACTING LANE GEOMETRY AND TOPOLOGY INFORMATION FROM VEHICLE FLEET TRAJECTORIES IN COMPLEX URBAN SCENARIOS USING A REVERSIBLE JUMP MCMC METHOD

O. Roeth^{a,*}, D. Zaum^b, C. Brenner^c

^a Corporate Research, Robert Bosch GmbH Hildesheim, Germany - oliver.roeth@de.bosch.com

^b Chassis Systems Control, Robert Bosch GmbH Hildesheim, Germany - daniel.zaum@de.bosch.com

^c Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Germany - claus.brenner@ikg.uni-hannover.de

KEY WORDS: Lane Accurate Map Construction, Trajectories, GPS, DGPS, IMU, Road Network, Reversible-Jump Markov chain Monte Carlo

ABSTRACT:

Highly automated driving (HAD) requires maps not only of high spatial precision but also of yet unprecedented actuality. Traditionally small highly specialized fleets of measurement vehicles are used to generate such maps. Nevertheless, for achieving city-wide or even nation-wide coverage, automated map update mechanisms based on very large vehicle fleet data gain importance since highly frequent measurements are only to be obtained using such an approach. Furthermore, the processing of imprecise mass data in contrast to few dedicated highly accurate measurements calls for a high degree of automation.

We present a method for the generation of lane-accurate road network maps from vehicle trajectory data (GPS or better). Our approach therefore allows for exploiting today's connected vehicle fleets for the generation of HAD maps. The presented algorithm is based on elementary building blocks which guarantees useful lane models and uses a Reversible Jump Markov chain Monte Carlo method to explore the models parameters in order to reconstruct the one most likely emitting the input data. The approach is applied to a challenging urban real-world scenario of different trajectory accuracy levels and is evaluated against a LIDAR-based ground truth map.

1. INTRODUCTION

Technologies for autonomous vehicles (AV) and advanced driver assistance systems (ADAS) are topics which are intensively investigated by many automotive OEMs and suppliers. While a lot of upcoming function of AV and ADAS are heavily related to a high accurate map (HAM) the challenges of map generation become one of the main fields of interest but remain partially unsolved so far. As a promising technique, crowd-sourcing methods are in the limelight of many works concerned with fully automated map construction.

This paper presents a new approach for extending a street accurate road network map to a lane accurate one using trajectories of GPS-monitored vehicle fleets. *Reversible Jump Markov chain Monte Carlo* and simulated annealing are used to explore the valid dimensions and configurations of a model representing the lane graph to reach the one which most probably emitted the input data.

The paper is organized as follows: Section 2 outlines the state of the art of lane accurate road network generation. Section 3 introduces the mathematical basics needed in our approach. Section 4 presents the basic models which are used by our subsequently described lane accurate map generation algorithm. Section 5 details the input data, namely the trajectory fleet measurement data in three accuracy levels and a LIDAR based ground truth map of an exemplary scenario. Finally, the results of our approach evaluated against the ground truth map are presented. Section 6 concludes the paper and presents possible future work.

*Corresponding author

2. STATE OF THE ART

Since GPS is publicly available, various approaches for the fully automated derivation of road network graphs from vehicle fleet trajectories have been presented. This section gives an overview of the state of the art of road and lane accurate map construction in 2.1 and 2.2 respectively.

2.1 Road Accurate Map Construction

Since our approach of lane accurate map construction presented in Section 4 is inseparable from the topic of road accurate map construction we will first present the state of the art with respect to the latter. To overcome the great diversity of approaches Ahmed et al. (2015) introduced a categorization in three classes: *A) intersection linking*, *B) incremental track insertion* and *C) point clustering*. Algorithms of category A firstly transfer detected curves and intersections into nodes in the graph and, secondly, analyze the trajectories to identify the connections between nodes. This technique is used with remarkable results by Karagiorgou and Pfoser (2012). Incremental track insertion algorithms insert the trajectories to an empty map iteratively and update the resulting graph at each iteration by using map matching procedures. Finally, point clustering approaches as used by Biagioni and Eriksson (2012) transform the input data into a point cloud to subsequently cluster them for example into a density based discretization image to identify the original graph.

2.2 Lane Accurate Map Construction

The categorization from Section 2.1 applies for lane accurate map construction. Bruntrup et al. (2005) provide a generic approach as a representative of the incremental track insertion algorithm. Their system uses AI techniques for inferring the geometries and

a cluster algorithm to generate the road map. As an intersection linking algorithm Schroedl et al. (2004) present an approach which firstly divides the input in roads and intersections in order to infer the roads and subsequently the lanes centerlines for each portion.

3. MATHEMATICAL BASICS

In this section the mathematical basics of our approach are summarized. Section 3.1 comprises the basic idea of *Markov chain Monte Carlo* (MCMC) methods which constitutes the principle of the subsequently presented *Reversible Jump MCMC* methods (Section 3.2).

3.1 Markov chain Monte Carlo Methods

Monte Carlo methods are statistical algorithms which can be used e.g. for the computation of large hierarchical models by generating representative random samples from the investigated function or numerical approximations of integrals.

While there are different approaches to sample from low - dimensional spaces like *Importance Sampling* (Hastings, 1970), *Markov chain Monte Carlo* methods enable sampling from arbitrarily complex distributions out of high - dimensional spaces. The main idea of MCMC methods is to define a transition kernel T of a chain which has the investigated function π as its stationary distribution which can then be used to generate a sample.

A general MCMC method is the *Metropolis-Hastings-Algorithm* (MH) (Hastings, 1970)(Metropolis et al., 1953). It generates a possible new state θ' of the Markov chain from a proposal distribution $q(\cdot|\theta)$ and decides whether or not to accept it. This procedure is hereafter referred to as *MH step*. If the *detailed balance condition*

$$T(\theta'|\theta)\pi(\theta) = T(\theta|\theta')\pi(\theta')$$

is satisfied, a stationary distribution exists and if the generated chain stays irreducible and aperiodic (Meyn and Tweedie, 1993) the chain converges to this distribution. The probability of accepting the new state is defined as

$$\theta^{(t+1)} = \begin{cases} \theta', & \text{if } w \leq r(\theta^{(t)}, \theta') \\ \theta^{(t)}, & \text{else} \end{cases}$$

$$r(\theta, \theta') = \min \left\{ 1, \frac{\pi(\theta')q(\theta|\theta')}{\pi(\theta)q(\theta'|\theta)} \right\} \quad (1)$$

$$w \sim \text{Uniform}[0, 1].$$

Since the convergence requirements are fulfilled, π is the stationary function of the transition kernel $T(\theta^{(t+1)}|\theta^{(t)})$.

3.2 Reversible Jump Markov chain Monte Carlo Methods

MCMC methods are limited to problems of fixed dimension. In order to enable dimension changes of θ , Green introduced the *Reversible Jump MCMC* (rjMCMC) methodology in (Green, 1995). This method extends the general MCMC methods by transitions in the kernel T between different state spaces.

Let S_k be the k -th state space and π_k the target distribution defined on the k -th space. The combined state space of the disjunct spaces is defined as

$$S = \cup_{k=1}^K (\{k\} \times S_k)$$

where the state $\theta \in S_k$ from the k -th space is written as (k, θ) . The transition kernel on S within each subspace is defined as

$$T((k', \theta')|(k, \theta)) = \begin{cases} T_k & \text{if } k = k' \text{ (Section 3.1)} \\ 0 & \text{else} \end{cases}$$

Additionally to the transitions within a space S_k , inter-space transitions from S_i to S_j can be implemented by defining a joint space S' where the spaces are complemented by additional components so as to have the same dimension:

$$S' = (\{i\} \times S_i \times U_i) \cup (\{j\} \times S_j \times U_j)$$

$$U_k = \mathbb{R}^k,$$

$$\dim(S_i) + \dim(U_i) = \dim(S_j) + \dim(U_j), \forall i, j \in \mathbb{N}_0$$

Then, a bijective function $\tau : S_i \times U_i \rightarrow S_j \times U_j$ is defined which relates two elements $\Theta_i = (i, \theta_i, u_i)$ and $\Theta_j = (j, \theta_j, u_j)$ by

$$\tau(\theta_i, u_i) = (\theta_j, u_j).$$

The needed complementing components U are sampled from $u_i \sim q_i(\cdot|\theta_i)$. If such a transition is chosen, u_i is drawn and the transition $(\theta_j, u_j) = \tau(\theta_i, u_i)$ to the new state $(k_j, (\theta_j, u_j))$ is evaluated and accepted with probability

$$(k^{(t+1)}, \theta^{(t+1)}) = \begin{cases} (k_j, \theta_j) & \text{if } w \leq b((k_i, \theta_i), (k_j, \theta_j)) \\ (k_i, \theta_i) & \text{else} \end{cases}$$

$$w \sim \text{Uniform}[0, 1].$$

The detailed balance condition must still be satisfied for $b(\cdot, \cdot)$ to ensure that π becomes the Markov chains stationary distribution. This requires the following consideration:

$$\Phi(\Theta_i, \Theta_j) = p_i \cdot \pi_i(\theta_i) \cdot q_i(u_i|\theta_i)$$

$$\Phi(\Theta_j, \Theta_i) = p_j \cdot \pi_j(\theta_j) \cdot q_j(u_j|\theta_j) \cdot |J_{\tau|i}| \quad (2)$$

$$\text{with } J_{\tau|i} = \frac{\partial \tau(\theta_i, u_i)}{\partial(\theta_i, u_i)}$$

are the probabilities of switching between the space S_i and S_j and p_i/p_j are those of choosing this move. The probabilities are defined on the same space

$$\mathbb{R}^{\dim(S_i) + \dim(U_i)} = \mathbb{R}^{\dim(S_j) + \dim(U_j)}$$

but in different variables. Thus, according to the transformation theorem in (2) the determinant of the Jacobian of τ is multiplied to scale the equations right.

According to (1) the acceptance probability of the transition is

$$b((k_i, \theta_i), (k_j, \theta_j)) = \min \left\{ 1, \frac{p_j \cdot \pi_j(\theta_j) \cdot q_j(u_j|\theta_j)}{p_i \cdot \pi_i(\theta_i) \cdot q_i(u_i|\theta_i)} |J_{\tau|i}| \right\}.$$

4. APPROACH

In this section, we present our approach of lane accurate map construction. The algorithm described in this subsection infers a lane accurate road network from vehicle trajectories and a road accurate map which can be derived by using one of the algorithms mentioned in Section 2. Firstly, Section 4.1 states the models

used to describe a lane accurate road and, subsequently, the construction algorithm is explained in detail in Section 4.2.

4.1 Model

The models described in the following are able to map a general traffic situation on lane level. To ensure that the models stay valid and realistic they are subjected to restrictions which are imposed by using a set of primitives and assembly rules. The overall model is split into two different types of submodels, namely streets and crossings which can be separated in the roadmap graph.

4.1.1 Street Model A street model is represented by a list of nodes of the underlying road map (see Fig. 1 $\{S_1, S_2, S_3\}$). Any point along the road can be described by a parametrization value $v \in [0.0, 1.0]$ as it can be seen in Fig. 1. Based on the parametrization it is possible to define *blocks* on the road as it is exemplarily shown in Fig. 1 with a gray box which extends from the relative value 0.6 to 0.8. Thus, the absolute extension in meters depends on the length of the road. A general block can then be specified using the following parameters:

- Number of lanes
- Width of each lane
- Type of each lane marking (e.g. dashed, solid,...)
- Middle gap separating the opposite directions
- Curvature

In our work two different types of blocks are used. Firstly, a *street block* describes a section with constant information regarding the number of lanes as it is shown in Fig. 2a and Fig. 2b. Secondly, a *connector block* extends the properties of a general block by a connection permutation which makes up the difference between two consecutive street blocks as it can be seen in Fig. 2c. Fig. 2d shows a composition of a street block with three lanes which merges into a street block with two lanes via a connector block. Finally, each block can be transformed into a graph representation, e.g. Fig. 2e shows the lane graph of Fig. 2d.

Since a block has its curvature as a variable property, the models are not described by linear interpolations but by cubic Hermite splines (Catmull and Rom, 1974) to get a smooth and realistic representation of a road. The required values are the start- and the endpoint of the block and the gradients in these points are derived from the road accurate graph. The length of the gradient vectors influences the form of the block.

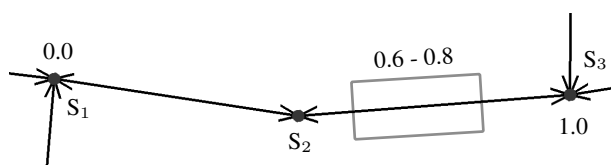


Figure 1. An exemplary street model referenced to $\{S_1, S_2, S_3\}$ of the road accurate map and an exemplary block spanning from the relative value 0.6 to 0.8. The filled bullets mark nodes and the arrows mark edges of the road accurate graph.

4.1.2 Crossing Model A crossing model consists of an accessible area (asphalt) and connection lanes. It is defined by a centroid point equivalent to the intersection node(s) of the road accurate graph. Each attached road is connected by a cross section with an incoming and outgoing part and a parameter describing the distance of the cross section to the centroid point as it

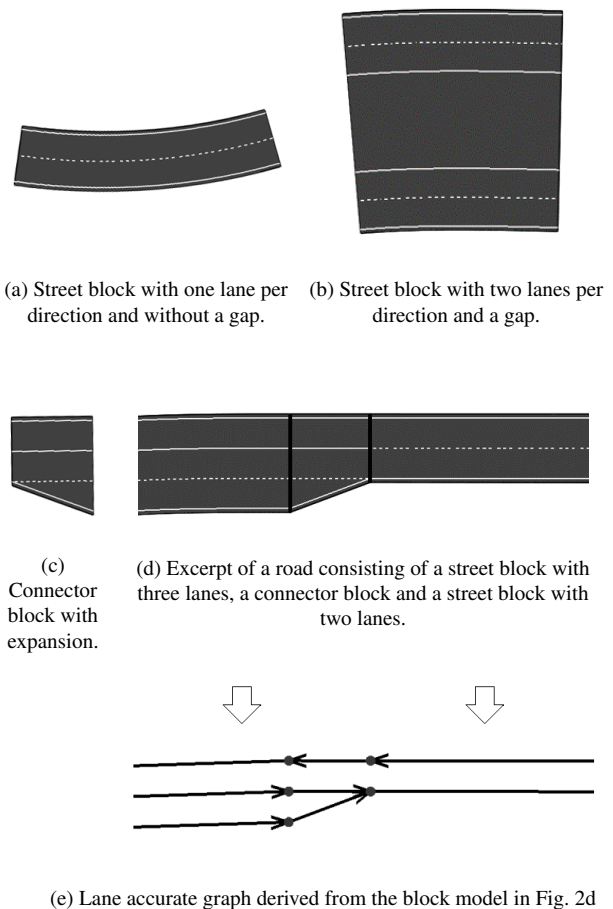


Figure 2. Examples of specified block types.

is schematically shown in Fig. 3a. The connected road model must end or start with a connector block, so it can make up the difference in number of lanes if there is one. Each input can be connected to any output by a connection lane which is again described by cubic Hermite splines as it exemplarily shown in Fig. 3c. Finally, an intersection can be transformed to a lane accurate graph, e.g. Fig. 3b shows the lane graph of Fig. 3c.

4.2 Map Construction

The main idea of the algorithm is to minimize a score function Φ (Eq. 3) which assesses the deviation of the lane graph G from the trajectories T where the minimization is performed using a MCMC simulation. Since the deviation is measured on lane level, a matching of the trajectories to the lane graph is necessary. This is done by creating a Hidden Markov Model (HMM) with the graph edges as hidden states and the trajectory points as emitted observations which is solved by using the Viterbi Algorithm as it is done in (Newson and Krumm, 2009). Using this, each model ξ from the set Ξ of all crossings and streets can be evaluated against the model's lane graph G_ξ .

$$\Phi = \sum_{\xi \in \Xi} \Phi_\xi = \sum_{\xi \in \Xi} \sum_{t \in T_\xi} \int \|G_\xi(t(\lambda)) - t(\lambda)\|^2 d\lambda \quad (3)$$

$$G_\xi(x) = \arg \min_{p \in G_\xi} \|p - x\|$$

where $\|\cdot\|^2$ is a metric which uses the euclidean distance and the difference in the driving direction angle. T_ξ denotes the trajectory-

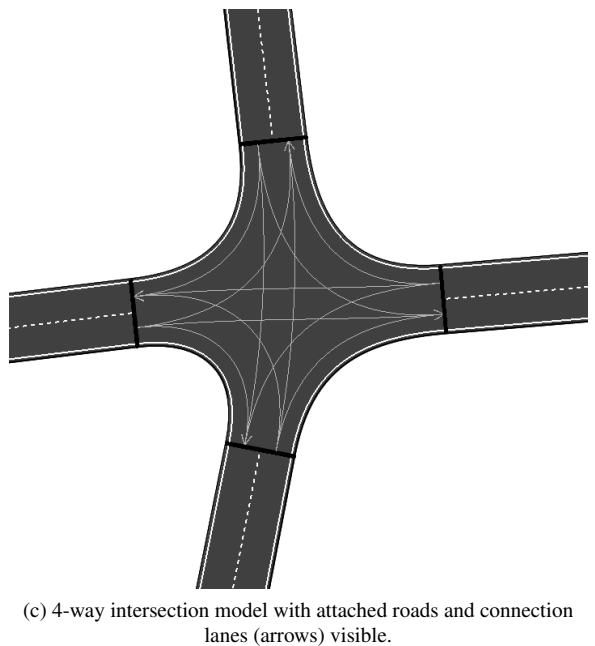
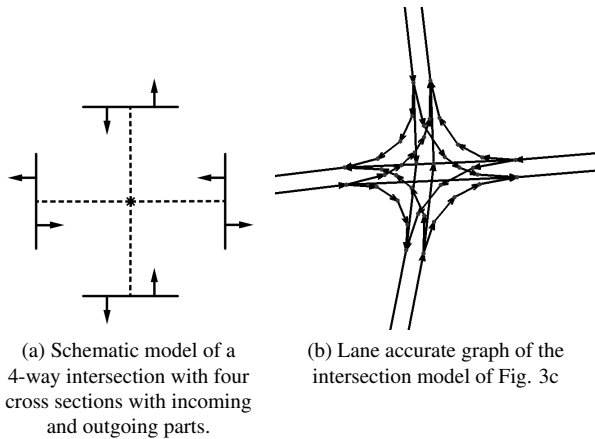


Figure 3. Intersection model

ries passing the model ξ . Since a trajectory is not a continuous function but a list of m points the integral can be approximated by a sum which results in

$$\hat{\Phi} = \sum_{\xi \in \Xi} \hat{\Phi}_{\xi} = \sum_{\xi \in \Xi} \sum_{t \in T_{\xi}} \sum_{n=m_{\xi,s}}^{m_{\xi,e}} \|G_{\xi}(t_n) - t_n\|^2 \quad (4)$$

where $m_{\xi} \leq m$ points are matched to the regarded model ξ and $m_{\xi,s}$ denotes the first and $m_{\xi,e}$ the last of them.

To formulate this as an RJMCMC problem which can be solved by using the Metropolis-Hastings-Algorithm, (4) must be turned into a posterior probability distribution $\pi(\cdot|\cdot)$ of the model given the data. Therefore, the approximated evaluation function (4) is replaced by a function based on the probability that t_n is recorded from the matched edge of the graph or not.

$$\pi = \prod_{\xi \in \Xi} \prod_{t \in T_{\xi}} \prod_{n=m_{\xi,s}}^{m_{\xi,e}} \Upsilon(G_{\xi}(t_n) - t_n)$$

As it is common practice the logarithmic values are summed up.

$$\pi = \sum_{\xi \in \Xi} \sum_{t \in T_{\xi}} \sum_{n=m_{\xi,s}}^{m_{\xi,e}} \log(\Upsilon(G_{\xi}(t_n) - t_n))$$

where $\Upsilon(\cdot)$ is a probability formulation of the above mentioned evaluation criteria depending on the used GNSS system. Additionally, the posterior probability distribution is modified by using a *simulated annealing* approach:

$$\pi(\vec{x}) \rightarrow \pi^{1/T_j}(\vec{x})$$

This modification is motivated by the approach of (Andrieu et al., 2000) and causes a reduction in the acceptance probability of low evaluated proposals in the MH step proportional to the number of iterations. This forces the Markov chain towards the global maximum of π .

Fig. 4 gives an overview of the iterative process. It states that firstly a model (block or crossing) is selected. Secondly, an operation is chosen independently from the current state based on a uniform distribution and, finally, the MH step is performed by the decision on acceptance or rejection. There are three levels of operations, two of them are related to block models. *Block-level* operations affect the internal properties of a block like the number of lanes while *Streetlevel* operations affect the external properties like the parametrization borders on the street. Regarding the probability z_o for choosing an operation $0 \leq z_o \leq 1$ and $\sum z_o = 1$ applies. The *add lane*, *remove lane*, *split block* and *merge block* operations are increasing or decreasing the dimension of the model θ by 1 while the other operations only vary the existing parameters.

The overall algorithm can be summarized into the following steps:

1. Create a road accurate network from the input trajectory data
2. Transfer the network to parameterized models and initialize all blocks and intersections
3. Each model: Iteration i
 - Sample $w \sim \text{Unif}[0, 1]$
 - Select an operation depending on w (Fig. 4)
 - Calculate the acceptance probability according to the operation and perform a MH step
 - Accept or reject the step. If the step is rejected, the operation is reverted
4. Update the cooling function
5. $i \leftarrow i + 1$ and go to 3.

In the following, the operations are considered in detail and in terms of unification hereafter a model ξ is in the context of RJMCMC referred to as a state θ (see Section 3.1 and 3.2).

Block level operations The operation *adjust gap* varies the distance between the opposite driving directions and *adjust width* varies the width of a lane. Both of them are not changing the model dimension and thus can be realized by performing an MH step according to (1) with a random walk (uniform distribution) as proposal function q .

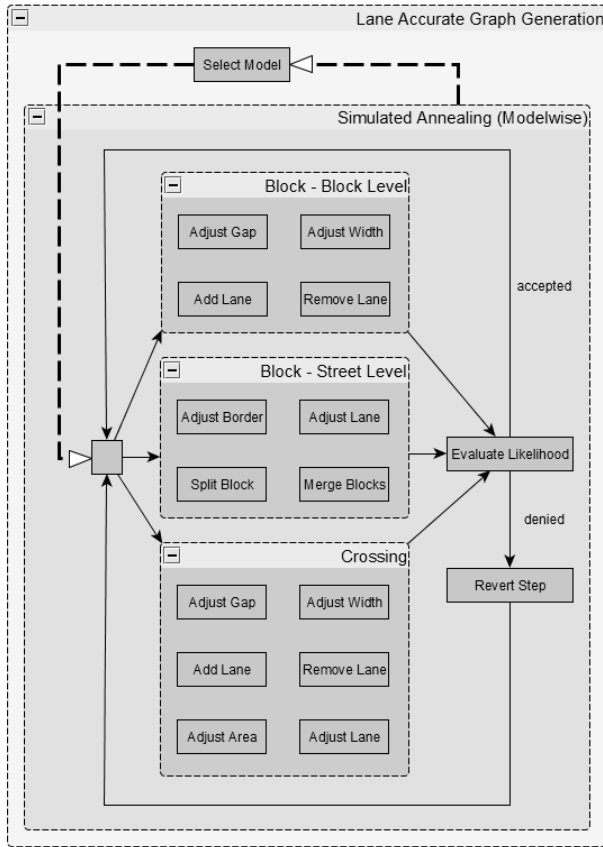


Figure 4. Abstract procedure of the optimization. In each iteration firstly the model and secondly the operation is selected.

The operation *add lane* can add a new lane to a block on the right or left side. Thus, the state is complemented by an additional lane width. The initial lane width is sampled from a normal distribution which complies with the *directive of urban traffic with public bus service* in accordance with (Baier et al., 2006) and applies to our urban scenario.

The dimension transition function in accordance to Section 3.2 is defined as

$$\tau_{b,add}(\theta, u) = (\theta, u)$$

$$u \sim N(3.25, 0.5)$$

and the add lane acceptance ratio is

$$b_{b,add}((k_i, \theta_i), (k_j, \theta_j)) = \min\{1, B\}$$

$$B = \frac{z_{ij} \cdot \pi_j(\theta_j)}{z_{ji} \cdot \pi_i(\theta_i)} \cdot \frac{|J_{\tau}|_i}{\sqrt{2\pi\sigma_{b,add}^2}} \cdot \exp\left(-\frac{(u - \mu_{b,add})^2}{2\sigma_{b,add}^2}\right)$$

In this case, the Jacobian matrix is 1. Similarly, a lane can be removed the inverse way, therefore the acceptance ratio is

$$b_{b,rem}(\cdot|\cdot) = \min\{1, B^{-1}\}$$

where u is replaced by the width of the removed lane.

Street level operations The operation *adjust border* varies the parametrization borders of the block on the street level. Thus, the block can be extended or shortened longitudinally. The operation *adjust lane* varies the length of the direction vectors of the cubic Hermite splines representing the centerline of a block. This way

the curvature of the street course can be influenced and therefore a block can be adapted to a curve with arbitrary radius. The operation *split block* can divide a block into two individual blocks. This step generates a new block model with the same properties as the original block, thus no special acceptance ratio but a slight modification to the interpretation of the posterior distribution is needed. Normally, the evaluation of the new and the old state are compared in the MH step. However, in this case, the evaluation of the original block is compared with the average value of the two resulting blocks. The other way around, the operation *merge block* can unite two adjacent blocks into one and the MH step deals with the average evaluation of the merged blocks on the one hand and with the evaluation of the resulting block on the other. All operations can be realized with a random walk proposal function.

Crossing operations The crossing operations are equal to the block- and streetlevel operations. Additionally, *adjust area* change the parameters defining the spatial extension of the crossing. Thus, this operation changes the surface of the intersection and can be realized by a random walk proposal function.

5. RESULTS

In this section our algorithm is applied to a dataset in three different accuracy levels, which are described in Section 5.1. In Section 5.2 the results are evaluated against a LIDAR based lane accurate network map (Section 5.1.2).

5.1 Input Data

5.1.1 Trajectory Data To assess the quality of the results of our algorithm we recorded ego trajectory data. The test vehicle was equipped with an Applanix POS LV¹ system comprising a 220 channel GNSS receiver (GPS-17 component), a set of Trimble 540 AP antennas, an *inertial measurement unit* (IMU-42 component) and a *distance measurement instrument* (DMI). Additionally, we made use of the SAPOS² services, a satellite reference service which can be used to increase the position accuracy. This hard- and software setting enabled us to record the dataset in three different accuracy levels. The first level comprises data recorded with the GPS system only. In the second level, these data are merged with the IMU und DMI data. Finally, the second level data are merged with post processing data from SAPOS to reach a high accuracy level. The dataset maps an inner city scenario comprising two intersections and seven roads with high traffic volume as it can be seen in Fig. 5a. Per level it consists of 54 trips with an overall length of about 32km running at an average speed of about 30km/h and is recorded in 1Hz. The datasets reach an average position error of 1.65 m, 1.19 m and 0.20 m (data from Applanix system) in the three accuracy levels, respectively.

5.1.2 Ground Truth Data For the evaluation, we use a high accurate LIDAR based map³ which fulfills particularly high requirements in accuracy. The used laser scanner has a measurement error of about 2cm which results in combination with a differential GPS system in an absolute geo-referenced position error of 10cm. The map is generated semi-automatically, with a considerable amount of human post-processing, out of the point

¹POS LV 220 V5 System (<http://www.applanix.com/>)

²<http://www.sapos.de/>

³generated by <http://www.3d-mapping.de/>

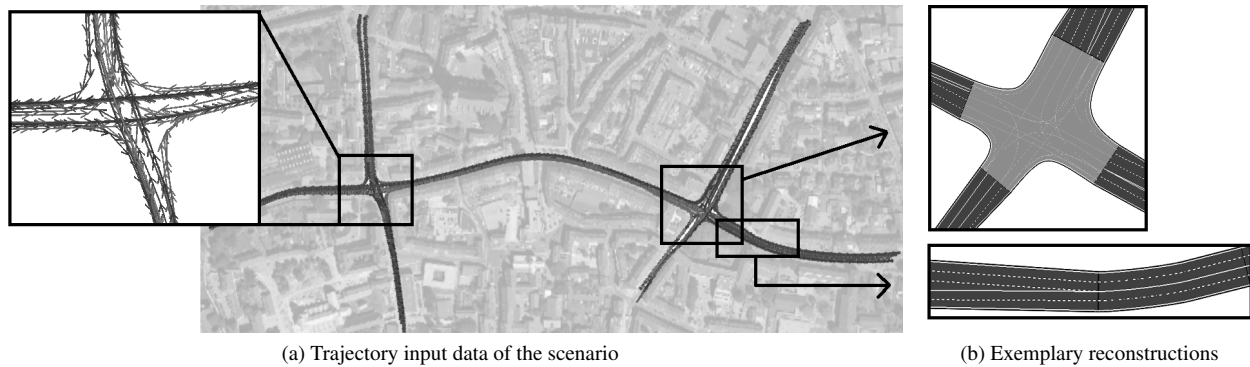


Figure 5. Raw trajectory input data of the scenario and exemplary reconstructions as quantitative results.

cloud of the laser scanner and results in an NDS⁴ associated map with lane accurate topological and geometrical information. The map comprises the centerlines of the lanes which implies their widths and is represented as a graph which implies the connectivity.

5.2 Evaluation

In this subsection, the results produced by our algorithm, based on level three input data, are qualitatively evaluated. To this end, we matched the reconstructed map to the ground truth map to associate the streets and crossings respectively. Based on this, on the one hand, the individual properties of the models are compared to state a quality measure at the model level and, on the other hand, the matching distance between the graphs is evaluated as a general measure. These criteria will be applied in two scales. Firstly, the mentioned measures will be applied to a single street to consider an exemplary comparison of the individual properties. Secondly, the whole scenario is taken into account to assess its overall quality. Furthermore, we applied two path-based comparison methods to evaluate the topology and geometry in a sufficient way.

Regarding a single street from the third stage dataset of a length of about 200 m, we analyzed the total error in terms of each parameter of the model. In Fig. 7 the street model is shown in a three dimensional context where the vertical dimension is the magnitude of the error. Fig. 7a shows the ground truth and Fig. 7b the reconstructed street. Fig. 6 presents the total matching error of the lane's centerlines of both graphs as a boxplot⁵ with a span from 0.03 m to 0.34 m, a mean of 0.2 m and a median of 0.15 m. In Fig. 7c the error regarding the number of lanes can be seen. Obviously, in the middle of the street the corresponding block is too short which results in a missing lane error. This error is difficult to avoid since the exact position of an upcoming lane can only be estimated from the trajectories. Overall this exemplary street model reconstructed 92% of the road correctly regarding the property of lane numbers. Fig. 7d shows the difference of the lane width of the associated lanes. It is striking that in the *right* section where a lane is missing the match of the lanes fails. In this case, the upcoming lane is matched to a remaining one which results in an error that is already detected (in the number of lane property). There is one more conspicuous error in the most left block which originates from a new lane in the ground truth map which is not upcoming but appearing. This characteristic cannot be derived

from the trajectories only, thus, the error cannot be avoided. On the remaining parts of the street, the matches are correct and the error of the reconstructed lane widths is 0.2 m in average. Finally, Fig. 7e shows the error concerning the gap between the opposite driving directions. In this example, the gap's shape is quite complex and can be estimated with an average error of 0.4 m.

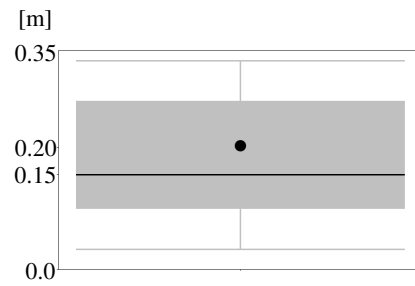


Figure 6. Matching error of the lane's centerlines [m].
 Min.: 0.03 m, Median: 0.15 m, Mean: 0.2 m, Max.: 0.34 m

Investigating a single street, the mentioned properties can be visualized as it is shown in Fig. 7. Regarding the whole scenario these properties must be quantified and therefore Fig. 8 shows the overall matching, lane width and middle gap error as boxplots. To create these result datasets, each street model is discretized in portions of 1 m length which are then evaluated. Fig. 8a states an overall matching error out of $[0; 1.75]$ m which implies, that there is at least one perfect match of 0 m and one worse match which is likely caused by a missing lane and thus is of the dimension of half a lane width. Fig. 8b presents the overall error of the lane width estimation in the range of $[-0.53; 0.75]$ m. Here, the errors caused by mismatches of upcoming lanes as it is occurred in Fig. 7d are filtered since these errors are already accounted for in the matching error. This criteria is evaluated in both directions, which means, that a negative value implies a too narrow reconstruction of the lane and the other way around respectively. Finally, Fig. 8c shows the overall error of the middle gap estimation out of $[-0.37; 0.68]$ m which is evaluated in both directions too. Overall, the quite low quantitative errors of the individual properties and additionally the visual comparison of Fig. 7a and 7b show the large potential of the method for ADAS applications, such as lane accurate route planning and navigation.

To investigate the structure of the generated lane accurate map, we apply two path-based evaluation methods. Firstly, we project all the input trajectories to both the reconstructed map and the ground truth, using again the Viterbi algorithm. The resulting associated paths are compared regarding their length difference and the Hausdorff distance which can be computed by

⁴Navigation Data Standard, <http://www.nds-association.org/>

⁵Data division: bottom whisker: 0%, gray box: 25%-75%, black line (median): 50%, black point: average value, upper whisker: 100%

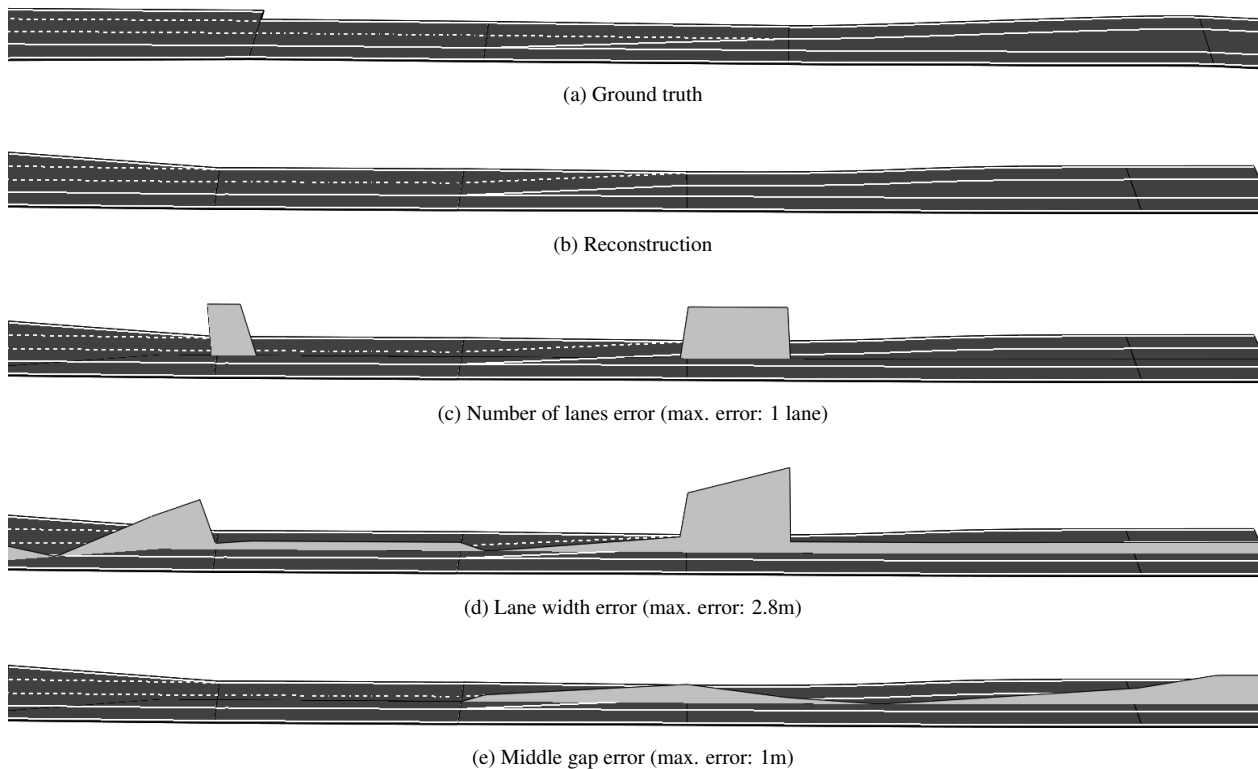
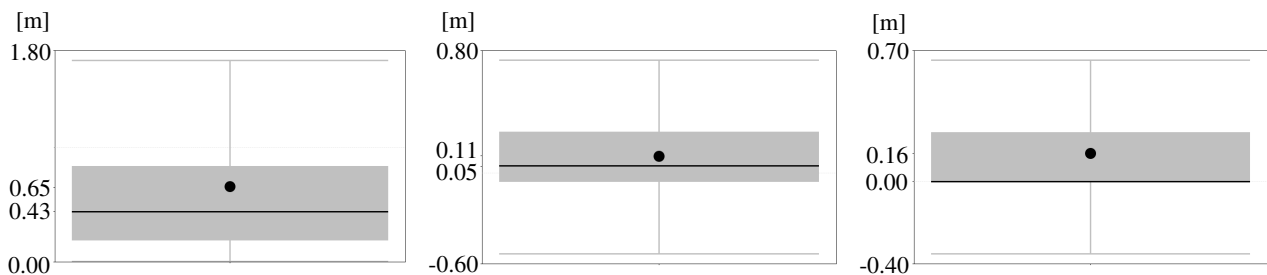


Figure 7. The ground truth street, the reconstructed street and the error, per model property



(a) Overall matching error [m]. Min.: 0.0 m, Median: 0.43 m, Mean: 0.65 m, Max.: 1.75 m (b) Overall lane width error [m]. Min.: -0.53 m, Median: 0.05 m, Mean: 0.11 m, Max.: 0.75 m (c) Overall middle gap error [m]. Min.: -0.37 m, Median: 0.0 m, Mean: 0.16 m, Max.: 0.68 m

Figure 8. Overall error

$$d_H(X, Y) = \max \left\{ \sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right\}.$$

The result of the length comparison is shown in Fig. 9a. This is a major measure of the topological similarity of the graphs. A topological error like a missing street lane or crossing connection will result in a different path, usually of different length. In comparison, geometrical errors will only have a minor impact. In Fig. 9b the Hausdorff distance of the paths can be seen which is a major measure of the geometrical similarity. Here, geometrical errors like lane widths, middle gap or position errors will result in an offset which increases the Hausdorff distance. Topological errors have an impact too, because a significant different path causes a great distance.

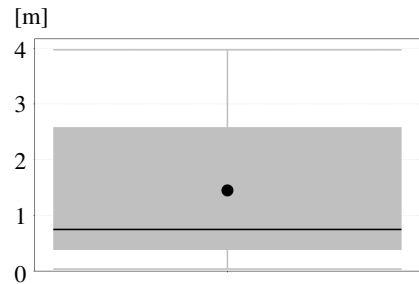
The presented results in Fig. 8 and Fig. 9 are generated based on the level three accuracy dataset. The corresponding results of the first dataset, comprising GPS data only, are summarized in Table 1. The evaluations of the second dataset, comprising GPS, IMU and DMI, are shown in Table 2.

Criteria	Min.	Median	Mean	Max.
Matching error	0.0	1.22	1.44	2.89
Lane width error	-0.57	0.07	0.17	0.83
Middle gap error	-1.17	0.0	0.34	1.96
Path length difference	0.44	3.72	4.89	13.30
Path Hausdorff distance	1.22	3.74	5.01	9.74

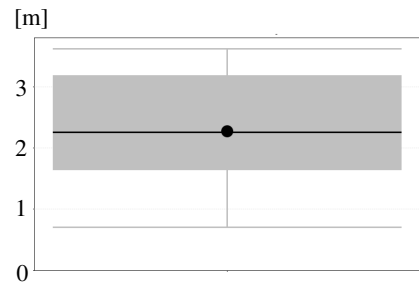
Table 1. Stage 1 (GPS) evaluation [m]

Criteria	Min.	Median	Mean	Max.
Matching error	0.0	1.01	1.18	2.71
Lane width error	-0.49	0.07	0.16	0.74
Middle gap error	-1.11	0.47	0.41	1.90
Path length difference	0.38	3.18	3.65	9.47
Path Hausdorff distance	0.83	3.53	4.65	7.97

Table 2. Stage 2 (GPS, IMU, DMI) evaluation [m]



(a) Length difference of associated paths in the reconstructed and the ground truth map [m].
 Min.: 0.03 m, Median: 0.74 m, Mean: 1.45 m,
 Max.: 3.97 m



(b) Hausdorff metric of associated paths in the reconstructed and the ground truth map [m].
 Min.: 0.70 m, Median: 2.25 m, Mean: 2.27 m,
 Max.: 3.62 m

Figure 9. Path-based evaluation results

Accuracy Level	Median error	Mean error
Stage 1	0.65%	0.85%
Stage 2	0.55%	0.63%
Stage 3	0.13%	0.25%

Table 3. Relative error of path length differences (absolute error in Table 1 & 2 and Fig. 9a)

Since the length difference error is a measure depending on the individual trajectory it can be transformed to a relative error by relating it to the trajectories length. The relative errors are shown in Table 3 which states a path length difference error of less than 1% for each stage.

Overall, our algorithm produces satisfying results regarding measures of the internal properties and of the structural characteristics. Some more results of a crossing and a street are shown in Fig. 5b.

6. CONCLUSION AND FUTURE WORK

In this paper a new approach for the derivation of lane accurate maps from vehicle fleet motion data is presented. Basing on publicly available road network graphs, lane models as parameterized blocks on the roads and intersections are initialized. The lane models are optimized and derived using a Reversible Jump Markov chain Monte Carlo approach to explore the parameter space of the model in order to find the best possible fit between the input data and the model. To evaluate the approach we recorded ego trajectory data of vehicles in three different accuracy levels with an overall average position error of 1.65 m, 1.19 m and 0.20 m respectively. We applied the algorithm to the input data and compared it to a LIDAR based ground truth map by evaluating the individual properties of the models and different

path-based comparison methods. Both the qualitative and quantitative analysis of the results states a large potential of the method for generating data for ADAS applications out of vehicle fleet sensor data. In the future, we plan to extend the options of the block models catalogue to be able to cover individual situations. Currently, a road consists of constant street blocks and variable connection blocks which make it difficult to fit e.g. a temporarily narrowed lane which could be handled by a special *narrowed street block*. Additionally, the algorithm will be extended to derive information about the type of road markings, e.g. dashed lines from mono camera data.

References

- Ahmed, M., Karagiorgou, S., Pfoser, D. and Wenk, C., 2015. A comparison and evaluation of map construction algorithms using vehicle tracking data. *GeoInformatica* 19(3), pp. 601–632.
- Andrieu, C., De Freitas, N. and Doucet, A., 2000. Reversible jump mcmc simulated annealing for neural networks. In: *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., pp. 11–18.
- Baier, R., Eilrich, W., Haller, W., Heinz, H., Lentz, D., Lerner, M., Maier, R., Manns, F., Müller-Ettler, M., Nikolaus, H. et al., 2006. Richtlinien für die anlage von stadtrassen. *Forschungsgesellschaft für Straßen- und Verkehrswesen*.
- Biagioni, J. and Eriksson, J., 2012. Map inference in the face of noise and disparity. In: *Proceedings of the 20th International Conference on Advances in Geographic Information Systems, SIGSPATIAL '12*, ACM, New York, NY, USA, pp. 79–88.
- Bruntrup, R., Edelkamp, S., Jabbar, S. and Scholz, B., 2005. Incremental map generation with gps traces. In: *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, pp. 574–579.
- Catmull, E. and Rom, R., 1974. A class of local interpolating splines. *Computer aided geometric design* 74, pp. 317–326.
- Green, P. J., 1995. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika* 82(4), pp. 711–732.
- Hastings, W. K., 1970. Monte carlo sampling methods using markov chains and their applications. *Biometrika* 57(1), pp. 97–109.
- Karagiorgou, S. and Pfoser, D., 2012. On vehicle tracking data-based road network generation. In: *Proceedings of the 20th International Conference on Advances in Geographic Information Systems, SIGSPATIAL '12*, ACM, pp. 89–98.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E., 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21(6), pp. 1087–1092.
- Meyn, S. P. and Tweedie, R. L., 1993. *Markov Chains and Stochastic Stability*. Springer-Verlag.
- Newson, P. and Krumm, J., 2009. Hidden Markov map matching through noise and sparseness. *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09* pp. 336–343.
- Schroedl, S., Wagstaff, K., Rogers, S., Langley, P. and Wilson, C., 2004. Mining gps traces for map refinement. *Data Mining and Knowledge Discovery* 9(1), pp. 59–87.