PARTICLE FILTER MAP MATCHING AND TRAJECTORY PREDICTION USING A SPLINE BASED INTERSECTION MODEL

Florian Kuhnt, Ralf Kohlhaas, Rüdiger Jordan, Thomas Gußner, Thomas Gumpp, Thomas Schamm and J. Marius Zöllner
Parts of this work have been developed within the scope of the German research initiative UR:BAN. These are funded under the support code 19 S 12008 I by the German Federal Ministry of Economic Affairs and Energy on the basis of a decision by the German Bundestag. The authors are responsible for the content of this publication.
INHALTSVERZEICHNIS

1 Introduction .................................................................................................................................. 4

2 Related Work .......................................................................................................................... 4

3 Definitions .................................................................................................................................. 5

4 Concept ....................................................................................................................................... 5

5 Localization .................................................................................................................................. 6
  5.1 Precision .................................................................................................................................. 6
  5.2 Map Matching .......................................................................................................................... 7

6 Parametric Intersection Model ................................................................................................. 8
  6.1 Intersection Model ................................................................................................................ 8
  6.2 Corridor modeling .................................................................................................................. 8
  6.3 Low-detail map data .............................................................................................................. 9
  6.4 Model Extraction .................................................................................................................... 9

7 Prediction ..................................................................................................................................... 9
  7.1 Vehicle Corridor Matching ................................................................................................... 9
  7.2 Prediction ................................................................................................................................ 9

8 Evaluation ................................................................................................................................... 10
  8.1 Map Matching ....................................................................................................................... 10
  8.2 Prediction ................................................................................................................................ 11

9 Conclusions ............................................................................................................................... 12
Abstract
For Advanced Driver Assistance Systems and Autonomous Driving it is of major advantage to know future trajectories of traffic participants. These are influenced by many factors in the environment. One important factor is the geometry of the intersection a vehicle is approaching.

In this work, we describe how we can extract a spline based intersection model from low detail map data like OpenStreetMap that can be adjusted over time. A particle filter based map matching algorithm is used to localize the ego vehicle relative to the intersection model. Additionally, objects detected from the ego vehicle's sensors are matched onto the intersection model in order to predict the future trajectories of the ego vehicle and other traffic participants using the intersection model.

1 Introduction
One of the big challenges in current work towards autonomous driving in urban environments is creating an understanding of the infrastructure that allows to precisely predict the traffic participants’ future movements.

In this work we present an approach that uses low cost sensors (GPS, Odometry) and low detail map data (OpenStreetMap) to create a precise corridor-based model of intersections that can be used to predict the ego and other vehicles’ trajectories. The challenge is that map data and a predefined modeling of intersections does not actually fit reality. Thus, we need a method to update intersection layouts using sensor data. This can take place while approaching an intersection but also after passing the same intersection several times.

In a first step we focus on observed trajectories of the ego vehicle and observed trajectories of other vehicles. This information is observed by the ego vehicle since we do not want to rely on intersection surveillance infrastructure. The localization of the ego vehicle is very essential since it also affects the localization of the detected objects.

If the ego vehicle and the detected objects are correctly localized, the intersection model can be simultaneously adjusted to the given circumstances and vehicles can be predicted using the intersection model that was improved by previous vehicle observations. This way we can also model typical vehicle movements (tracks) that do not fit the actual lane markings. This can include curve cutting and popular lane change regions.

In this work we present the basic approach and first evaluations on a set of intersections using low-cost GPS and odometry combined with Ibeo Laserscanner object data. For ground truth reference we use an OXTS RT3003 GPS.

This work is structured as follows: Section 2 takes a look at related work. Section 3 clarifies some definitions before Section 4 introduces the concept. Section 5, 6 and 7 describe the concept in detail. Finally Section 8 evaluates the concept and Section 9 sums up the achievements.

2 Related Work
There are many different approaches to predict vehicle trajectories. An approach predicting oncoming vehicles’ trajectories using a stereo vision sensor with optical flow vectors is shown in [2]. The work of [10] uses Gaussian mixture models to predict the ego vehicle’s
trajectory. Both exclude infrastructure information necessary for long term prediction.

The authors of [12] and [5] concentrate on predicting semantic behaviours of vehicles using Hidden Markov Models that were learned from recorded trajectories. These semantic decisions can be used to restrict trajectory predictions onto single lanes. This approach requires learning new trajectories for each intersection. Another approach is to describe the relations and causalities between traffic participants (and infrastructure) by using bayesian networks, in terms of Dynamic Bayesian Networks [1], Object Oriented Bayesian Network [4] or a template system called configurations [8]. These approaches result in a good and human-readable understanding of the actual situation. They usually need a well prepared model of the situation including precise localization and map data.

The authors of [11] construct a lane-based map from recorded trajectories without using any preexisting map data. They can detect streets, lanes and intersections and thus it’s possible to predict future lanes by matching the current position onto a lane. This approach needs enough observations before an initial map can be created.

Map-based long term motion prediction is done in [7] using a manually prepared precise lane-based map that uses optimized geometric Hermite curves. The prediction of vehicle movements is done by matching the vehicles onto the lanes and predicting the future trajectory by using a Kalman filter based approach. This is already the closest approach to ours. Since our focus is on using low-detail map data, creating the precise map online and updating this map using sensor data, our approach uses a different map model. Additionally, we show an approach for vehicle trajectory prediction that benefits from this intersection model.

3 Definitions

We define the following elements:

- **Corridor**: The corridor is the part of the street that is meant to be used for driving in one direction. A simple two way road has two corridors that do not overlap. On intersections there can be different corridors for driving straight and turning right or left that can also overlap. This is especially useful for semantic interpretations like *The vehicle will turn into the left street.*

- **Lane**: The lane is bordered by lane markings or curbs. A corridor can consist of multiple lanes. For instance two lanes turning right are described by one corridor.

- **Track**: A track is the actual center line that a vehicle drove or will drive. It has no time information.

- **Trajectory**: A trajectory is a track annotated with orientation and time. It’s not only considering the pose of the vehicle but also the time at that pose. This way it allows deriving the velocity, acceleration and yaw rate.

- **Arm**: Every road that connects to an intersection is called arm (of that intersection). Arms can be bidirectional or one way.

- **Door**: Every arm induces one or two doors (one or two-way). These are the points where the road changes into the intersection and the road corridor splits up into the intersection corridors. They are called *entrance doors* (entering the intersection) and *exit doors* (leaving the intersection).

4 Concept

The main goal is to achieve a prediction of the ego vehicle’s and other vehicles’ trajectories. Since in intersections the trajectories are strongly depending on the intersection layout, we focus on constructing a suitable intersection model.

In Fig. 1 the concept is described. The process shown on the left localizes the ego vehicle on a low detail map and extracts the intersection model using predefined rules. The ego vehicle and observed vehicles are then matched to the extracted intersection model. Finally, the prediction can be done for all matched vehicles. Additionally, the matched vehicles can be used to correct and refine the intersection model. The correction itself will be the focus of future work.

A precise prediction is only possible if the underlying model can be learned and improved over time. This leads to following requirements on the intersection model: It can be constructed from low-detail map data,
holds all necessary information for prediction, and is potentially able to learn from observations.
To achieve this we propose a parametric, hierarchical intersection model (Fig. 2). The hierarchy allows to adjust parameters on a higher level and influence all underlying structures. As an example an arm’s pose can be adjusted influencing all doors on that arm which in turn update the corridors’ geometry. More details on the intersection model itself are given in Sec. 6.

In the first step we cannot rely much on the intersection model since the derived intersection does not exactly fit real intersections. Therefore, the matching of the ego vehicle is done before entering the intersection using a particle filter based approach (Sec. 5). Prediction is done using a particle filter that simulates a vehicle motion model and evaluates the particles using the intersection model (Sec. 7).

Details on the actual challenges will be described in the following sections.

5 Localization

For being able to predict the ego vehicle’s future trajectory a good localization with respect to the map (i.e., the intersection model coordinate system) is required.
We only use common vehicle sensors such as a GPS sensor and the vehicles odometry data from the CAN interface (velocity, yaw rate, steering angle). Thus there are some challenges mainly due to the given accuracy which we address in the following.

5.1 Precision

GPS precision: GPS can be disturbed by many environmental influences, most important: atmosphere, trees and buildings. There are different methods to improve the information by temporal fusion; nevertheless, lane-precise positioning would not be possible by GPS only. The high precision GPS we use as reference combines a differential GPS with additional acceleration sensors to calculate the actual motion of the vehicle. Depending on the environment this results in an accuracy of better than 2 cm.

Odometry precision: Odometry precision is dependent on many factors. Tyre pressure and side wind are just a few examples. Typical information is the vehicle’s velocity, yaw rate and steering angle. Each value can have an offset that can be calibrated and a variance that changes over time and can only be modeled as an uncertainty in the map matching algorithm. Yaw rate and steering angle can both be used to calculate the vehicle’s angular movement but each has its drawbacks.

Map precision: Low detail map data like from OpenStreetMap is usually recorded by simple GPS tools with precision of around 1 meter. Streets are usually not
modeled lane based. This results in cases where the centerline of the street is actually located on the real lane that was driven while recording the track. So map failures can be either an offset or completely wrong geometries. When using a high precision GPS sensor we can reduce the sensor uncertainties so that the difference to the matched position only results from map precision (Fig. 3).

5.2 Map Matching

For our target application we need a good longitudinal and lateral localization of the vehicle in front of intersections that allows matching the vehicle to corridors. While the lateral position can be matched quite easily the longitudinal position has to be evolved over time. In [6] a particle filter based map matching using odometry and GPS data is introduced. The method tracks many pose hypotheses, updates them using odometry data and rates the hypotheses using the GPS position and a so called zone map derived from a road map. The zone map mainly separates the map into zones in which specific assumptions are made on how fast a vehicle can drive in this area (Highway ≠ Street ≠ Building).

As a more appropriate approach we use an algorithm rating the vehicle's pose according to the closest street instead of using zone maps. We use a rating function that respects the streets possible driving directions and the side on which the vehicle usually drives. This way we get the best lateral score at the center line of the right corridor (in a right-side driving environment). The longitudinal position is evolved over time mainly due to curves and turns. For example after a 90° turn the new longitudinal precision is approximately the precision of the former lateral position.

Here it is important to decide when to trust the map and when not. If a road is not straight a wrong longitudinal localization also affects the lateral localization.
This can be handled by a map confidence value that is higher if the road is straight.
To be able to also use curved roads for localization the map confidence value has to be increased. Therefore, a precise representation of how vehicles usually drive specific turns on specific roads is necessary. Using observations from previous drives will help modeling this into the map. Finally SLAM-based approaches can be applied.
Tests have shown that using our approach in typical inner city situations results in sufficient localization (Fig. 7).

6 Parametric Intersection Model

6.1 Intersection Model
The basis of the intersection model consists of the arms’ position and orientation given in a local cartesian intersection coordinate system. At the arms doors are located according to the street's information about driving directions and number of lanes (Fig. 5). Each entrance door is connected to each exit door using a spline based corridor. Occupancy probability distributions are annotated onto the corridor as described below. These distributions describe how vehicles usually drive this corridor and can also model lanes and other typical paths through the corridor like curve cutting and popular lane change regions.

6.2 Corridor modeling
The corridors hold the most important information when interpreting the intersection layout. Lanes are secondary and only give an occupancy probability distribution inside the corridor boundaries. Therefore, we model the corridors as two-dimensional objects and annotate the lane information as occupancy probability distributions onto these corridors. When a corridor is moved also its lane information is moved.
For the corridors we choose a representation that allows easy manipulation and utilizes given position and orientation of discrete measurement points (i.e. map data). The goal is to set the position and deviation of the end points and adjust the curvature between these points with as little parameters as possible. A quadratic B-Spline representation consisting of five control points directly derivated from the intersection’s arms’ positions is used in [3]. Three of the five points represent the actual corridor. To increase flexibility we extend this representation to a typical bezier spline representation using the Cubic Bezier Curve consisting of four Bezier points:

\[(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\]

We convert these representation and describe the corridor by using the start and end point’s position \((x, y)\), their orientation \(\phi\) and the weight \(w\) towards the other spline end:

\[(x_1, y_1, \phi_1, w_1), (x_4, y_4, \phi_4, w_4)\]

with

\[\phi_1 = \arctan \left(\frac{y_2 - y_1}{x_2 - x_1}\right)\]
\[\phi_4 = \arctan \left(\frac{y_4 - y_3}{x_4 - x_3}\right)\]
\[w_1 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}\]
\[w_4 = \sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2}\]

Each point and its deviation (i.e. orientation) on the spline can be calculated. Occupancy probability distributions are annotated onto each spline according to an adjustable sampling rate along the spline. These distributions are onedimensional occupancy grids lying orthogonal to the splines direction (i.e. deviation). This
way we can easily model lane distributions and other typical behaviors inside the corridor model.

6.3 Low-detail map data
OpenStreetMap has the potential to deliver a very precise and up to date map. However, one drawback is the varying precision. Furthermore, the interpretation of semantic attributes can differ from region to region. Especially intersections are modeled in at least three different types of precision (Fig. 6). The first type models the arms and the fact that these arms meet in one intersection. The second type models the arms and corridors of the arms but not the different kinds of transition corridors inside the intersection. The third type is the most precise where also each transition corridor is modeled as a separate OpenStreetMap way. These are usually only modeled for large intersections and also their positioning is often questionable.

We propose an extraction algorithm that only uses the most reliable information of the intersection: Arm position, orientation, possible driving directions and number of lanes.

6.4 Model Extraction
To achieve a more precise intersection layout we use additional information on how intersections are usually meant to be built. These can be retrieved from road planning organizations. They especially discuss how to construct intersections so that specific vehicles can drive these intersections with ease [9]. Important parameters are the minimum curve radius and the lane width.

At first, the intersection of interest is selected in the map. This can be done using the localization explained in Sec. 5.

The arms of the intersection are then used as line segments. Doors can be derived on these arms using the information about driving directions and number of lanes from the map. The only value that is not derivable is the position of the door along the arms direction. This influences the beginning of the corridors and therefore the possible curvatures the corridor can adopt. Depending on the arm segment this can also influence the orientation of the arm itself. We use an incremental algorithm that fits corridors respecting the minimum curve radius into several possible arm configurations and chooses the smallest configuration that respects all constraints.

If the valid corridor configuration is found the initial occupancy probability distributions are created according to entrance and exit doors’ lane configurations. Derived from low-detail map data with additional intersection model information we obtain a suitable initial intersection layout that can be refined in future steps.

7 Prediction

7.1 Vehicle Corridor Matching
On an n-way intersection we get a total of \( n \times (n - 1) \) corridors. A common four-way intersection has twelve corridors. To reduce complexity we run a fast matching of vehicles onto their most relevant corridors. The matching is done comparing position and orientation of the vehicle with the foot-point of the position on the selected corridor. The best corridors are selected as relevant and further algorithms are run only on this subset. At a four-way intersection this usually reduces the twelve corridors to one to three relevant corridors per vehicle.

7.2 Prediction
Our goal is a prediction that also considers drivability aspects and corner cutting.

Prediction is done using an additional particle filter and can be seen as simulating a lot of possible driving maneuver hypotheses.

Every particle (= hypothesis) represents the current vehicle state consisting of the pose (position and orientation), the longitudinal velocity and acceleration, and the yaw rate and yaw acceleration. Additionally, every particle belongs to one corresponding corridor. When initializing the particles they are uniformly distributed onto the relevant corridors since every corridor should have the same initial probability to be currently driven by the vehicle. The vehicle state parameters are set.
according to current sensor data and variance. Position and orientation variance can directly be derived from the map matching algorithm (5). Velocity, acceleration, yaw rate and yaw acceleration are randomized to model the uncertainty about the vehicle’s current maneuver. The motion model of the particle filter follows a basic forward projection of the vehicle state assuming constant longitudinal and yaw acceleration.

The sensor model of the particle filter uses the corridor information. The closest occupancy probability distribution of the corresponding corridor (section 6.2) is evaluated to rate the particle. In each step the vehicle hypotheses are resampled so that hypotheses with a high rating get many new instances. The new instances are jiggled to model driving changes. These influence the trajectories in the next motion step.

A given time horizon is predicted. Afterwards, the particles can be sampled into a 3D occupancy grid consisting of 2D grids along a third time axis. Since every particle belongs to a corridor the final distribution over corridors can be used to derive the most probable driving corridor.

8 Evaluation

8.1 Map Matching

For evaluating the map matching we compare the matched trajectory of an approximately five kilometre long drive to the map itself and to the trajectory of the high precision GPS (HPGPS). Fig. 7 shows the resulting trajectories of raw GPS, matched GPS and HPGPS. To compare the matching results to the original raw GPS we built the difference of each to the HPGPS. Typically, the error of the matched position is less than five metres. This is still quite a lot and can be explained by map precision and the fact that the vehicle is not always driving on a lane modeled in the map. The latter

Fig. 7. A series of a map matching scene. The raw GPS trajectory (blue), the matched trajectory (green) and the ground truth OXTS RT3003 track (gray) are drawn onto the OpenStreetMap map data that is used for map matching. After a turn, the former longitudinal variance can be reduced due to the lane information. Afterwards the trajectory follows the map.

Fig. 8. Top: Approaching a simple intersection at 5 km/h (left) and 25 km/h (right). Bottom: Derived probability for taking the left turn (blue) over different velocities.
happens more often in areas between intersections than close to intersections. In front of most intersections the difference reduces to less than two metres, what fulfills the goal of having a good localization before entering an intersection.

8.2 Prediction
Prediction is evaluated using a simple oneway-street intersection with two corridors (Fig. 8). With such an intersection the decision can be evaluated if a vehicle takes a turn or stays on the main road.

As a first example in Fig. 8 we simulate a vehicle located in front of an intersection with different velocities. With lower velocity hypotheses can be found that take the turn. With higher velocity no hypotheses can be matched onto the left turning corridor so all the hypotheses lie in the straight corridor. This is reflected in the accumulated corridor probability (Fig. 8 bottom).

A complete real life example is given in Fig. 9. We have chosen an intersection with an oncoming vehicle while the ego vehicle intends to take a left turn. The sensors in use are an USB GPS sensor, the vehicle’s odometry from a CAN interface and the object information from an Ibeo laser scanner. Object information can be replaced by any other object detection sensor.

Map matching is applied to the GPS and odometry information and the intersection model is extracted from OpenStreetMap. Ego vehicle and detected vehicles are matched onto the model and the prediction algorithm is applied.

The oncoming vehicle has a high velocity and is correctly predicted as driving straight over the intersection. For the ego vehicle the left turn corridor’s probability increases while braking. Once the oncoming vehicle has left the intersection the ego vehicle follows the predicted way.

![Fig. 9. Small 3-way intersection with prediction running on ego and other vehicles. Ego vehicle (green) approaches an intersection with an oncoming vehicle (blue). The actual measured vehicle positions are at the back of the small black arrows. In front of the arrows are the predicted hypotheses; behind the arrows the measured track history is shown.](image)
9 Conclusions

In this work, we have introduced a new approach on modeling an intersection in a parametrized way. The model meets the following requirements. Firstly, it allows easy extraction from low detail map data. Secondly, it has the potential to be adjustable by future algorithms. Finally, vehicle trajectories can be predicted using the intersection model.

The functionality of the necessary map matching using common GPS and odometry data has been evaluated. Additionally, a particle filter based prediction algorithm can use the intersection information to predict ego and other vehicles’ trajectories.

Since the model is prepared for updates using observations from environment sensors or former drives on the same intersection it has the potential to also solve mapping issues. Instead of expensive recording and preparation of map data, low detail map data can be used to extract a basic model that can evolve over time.

Future work will concentrate on algorithms for updating the model parameters using sensor data. The spline based modeling can also be extended to model the roads between intersections.
References


IMPRESSUM

Authors
Florian Kuhnt (kuhnt@fzi.de),
Ralf Kohlhaas (kohlhaas@fzi.de),
Thomas Gumpp (gumpp@fzi.de),
Thomas Schamm (schamm@fzi.de),
und J. Marius Zöllner (zoellner@fzi.de)

Rüdiger Jordan (ruediger.jordan@de.bosch.com)
und Thomas Gußner (thomas.gussner@de.bosch.com)

FZI Forschungszentrum Informatik
Haid-und-Neu-Str. 10–14
76131 Karlsruhe, Germany

Robert Bosch GmbH
Corporate Research and Advance Engineering, Driver Assistance Systems and Automated Driving (CR/AEV2)
70442 Stuttgart, Germany

Scientific Director responsible for the content
Prof. Dr.-Ing. J. Marius Zöllner

Design and Layout
Department Corporate Communications and Media, FZI

FZI Forschungszentrum Informatik
am Karlsruher Institut für Technologie
Haid-und-Neu-Str. 10–14
76131 Karlsruhe, Germany
www.fzi.de

Board of Executive Directors
Prof. Dr. Andreas Oberweis
Prof. Dr. Ralf Reussner
Dipl.-Wi.-Ing. Jan Wiesenberger
Prof. Dr.-Ing. J. Marius Zöllner

Head of Board of Trustees
Ministerialdirigent Günther Leßnerkraus

Release Date: 2nd of October, 2014