Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Improved Tracking and Behavior Detection by Using Environment Knowledge from Online Map Databases in Bayesian-Filtering

Andreas Alin, Martin Butz, Jannik Fritsch

2013

Preprint:

This is an accepted article published in IEEE Intelligent Transportation Systems Conference (ITSC). The final authenticated version is available online at: https://doi.org/[DOI not available]

Improved Tracking and Behavior Anticipation by Combining Street Map Information with Bayesian-Filtering

Andreas Alin, Jannik Fritsch and Martin V. Butz

Abstract— **Estimating and tracking the positions of other vehicles in the environment is important for advanced driver assistant systems (ADAS) and even more so for autonomous driving vehicles. For example, evasive strategies or warnings need accurate and reliable information about the positions and movement directions of the observed traffic participants. Although sensor systems are constantly improving, their data will never be noise-free nor fully reliable, especially in harder weather conditions. Thus, the noisy sensory data should be maximally utilized by pre-processing and information fusion techniques. For this we use a augmented version of our spatial object tracking technique that improves Bayesian-based tracking of other vehicles by incorporating environment information about the street ahead. The algorithm applies attractor-based adjustments of the probabilistic forward predictions in a Bayesian grid filter. In this paper we show that context information – such as lane positions gained from online databases similar to open street map (OSM) – can be effectively be used to flexibly activate the attractors in a real-world setting. Besides the improvements in tracking other vehicles, the resulting algorithm can detect medium-time-scale driving behavior like turning, straight driving and overtaking. The behavior is detected by using a new plausibility estimate: Different behavior alternatives of the tracked vehicle are compared probabilistically with the sensor measurement, considering all possible vehicle positions. Thus, risk levels can be inferred considering alternative behaviors. We evaluate the algorithm in a simulated crossing scenario and with real-world intersection data. The results show that the attractor approach can significantly improve the overall performance of the tracking system and can also be used for better inference of the behavior of the observed vehicle.**

I. Introduction

Advanced driving assistant systems rely on precise information of their surrounding and assumptions about the behavior of vehicles for optimal decision making and control. If the assumptions are chosen correctly they can compensate for faulty sensory data. But those assumptions can also have drawbacks. Incorrect or imprecise lane knowledge can lead to unsuitable behavior anticipations. Moreover certain behaviors of car drivers may remain undetected if the belief of the system in its own anticipation is to high. To effectively combine context information with kinematic knowledge and sensor information a probabilistic framework is needed. Bayesian filters such as Kalman filters or particle filters fuse information sources effectively. However they do not take context information into account. Using a context information in the prediction, the information loss occurring from the previous to the current time step is reduced. In [1] we proposed a new approach to create more accurate predictive movement models by utilizing state-dependent context information. The approach is incorporating context information into the system by deriving attractors, which specify potential target locations for the observed traffic participants.

In this paper we show a way how this approach can be transferred into real-world scenarios. Obtaining accurate map data and self-localization in this map is the crucial point to derive the attractors. Self-localization is a very active field in automotive and robotics research. The position of the ego-vehicle can be received by visual odometry [2], IMU and Kalman-Filter Tracking (as was for example used to create the KITTI benchmark data [3]). Here we focus on map data retrieving and attractor generation. Moreover we improve an automatized attractor algorithm, by utilizing splines, minimizing the acceleration in x and y direction to generate reasonable trajectories. To evaluate the modifications and extensions, we track the position and detect the behavior of observed vehicles in different real-world intersection scenes. Moreover, a simulation of an intersection scene is used to cross-validate behavior detection.

The paper is organized as follows: Related work is reviewed in Section II. Next, an overview of the proposed approach is given in Section III-A. A short repetition of the basic concept of Bayesian filtering with context fusion is given in Section III-B. The basic concept of processing map data, deriving attractors, and incorporating attractors in Bayesian filtering is introduced in Section III-D Bayesian filtering with context fusion and the algorithmic approach for defining attractors and influencing the movement model is introduced in Section III-B. Results demonstrating the real-world capabilities of the approach and a cross-evaluation of the general behavior detection capabilities are presented in Section IV. The paper ends with a summary and conclusions.

II. Related Work

Various papers use bottom-up, sensory-driven approaches to improve their movement models of other vehicles. Barth and Franke [4], [5], for example, use a traditional tracking approach and introduce an additional estimation of the yaw rate by measuring the position from tracked optical flow vectors. In [5] they added the estimation of the yaw rate change, further improving the estimations in intersection

A. Alin is with the Department of Computer Science, Cognitive Modeling, University of Tuebingen, Tuebingen, Germany andreas.alin@uni-tuebingen.de

J. Fritsch is with the Honda Research Institute, Offenbach am Main, Germany Jannik.Fritsch@honda-ri.de

M. V. Butz is with the Department of Computer Science, Cognitive Modeling, University of Tuebingen, Tuebingen, Germany martin.butz@uni-tuebingen.de

scenarios with potential occlusions. The approach is very well-suited to handle intersection scenarios in real-world situations. However, drawbacks arise due to the more error-prone indirect derivation of the yaw rate estimations. Moreover, curves on the road are usually formed like clothoids, where radii continuously change so that the yaw rate is not sufficient for an accurate prediction of future motion.

Other approaches are top-down oriented. These approaches have behavior-based models and test the sensor measurement for compliance with the respective models.

Some approaches try to find out if driver behavior is compliant and non compliant to the expectation; for example, if a red light is violated. [6] is classifying the behavior at intersections with traffic lights. They propose two different approaches: one approach learns compliant and non compliant behavior by classifying the sensor measurement with a support vector machines and feeding the classification output into a Bayesian filter, which models behavior likelihoods as internal states. The second approach uses a forward model directly on the sensor measurements and compares the output with a hidden Markov model filter (HMM). We are using kinematic modeling instead, so that we compare different Bayesian filters' inherent internal position and velocity state distributions.

Another sophisticated system working with models is [7]. It detects dangerous situations by comparing intention and expectation of a driver by Bayesian inference. The intention is given by V2V communication. They also model the internal state of the vehicle with position and velocity distributions and use behavior models given by map annotations. In our approach we compare the fit of the measurement to the internal state given a certain model assumption instead of comparing intention and expectation. Therefore V2V communication is not needed but could be used as an optional feature to increase the prior assumption that a certain behavior will be executed. As a further distinction to this and many other approaches we use an attractor function approach to create the behavior models instead of giving a single exemplary path. This should on the long run pay out when using lane data from online databases.

III. System Overview

In this work we use the Bayesian histogram filter introduced in [8]. A short abstract of the original system is given in Section III-A. In this work we show that this approach can be used to anticipate behavior in intersection scenarios. To do so, we introduce a plausibility measure to the Bayesian histogram filter. The resulting behavior detection capabilities are robust-enough to be applied in real-world intersection scenes (Section III-B). In Section III-D we also illustrate how this real-world context data can be obtained.

A. Bayesian filter with context fusion

In order to track a traffic participant, we use the Bayesian histogram filter approach, which was introduced in [8], where it was shown that this filter approach can outperform simple Kalman filtering in non-linear motion scenarios. Note that the presented idea of fusing context information into the filtering can be used with particle filters and in a very limited way even with Kalman filters. All Bayesian filters have the mathematical model in common, but they differ in the way the state of the observed vehicle is represented [9].

A short introduction to the used grid filter is given here. The grid filter estimates the state x of vehicles in front of the ego-vehicle from an ego-vehicle centered perspective. A grid is equally distributed over this area, consisting of n grid nodes. Each grid node represents a rectangular Voronoi area (e.g. $0.25m \cdot 0.25m$) with the node as its center. A certain node i contains a probability $P(x_i)$ that the vehicle is currently within the area of the node. The node additionally saves the velocity $||\mathbf{v}||$ and direction $\omega(\mathbf{v})$ a vehicle at the nodes position (l_x, l_y) should have. This representation allows multi-modal probability distributions over the observed vehicle's state $x = (pos, ||\mathbf{v}||, \omega(\mathbf{v}))$.

The velocity and direction knowledge is needed to predict the position change of the observed vehicle from one time step to the next using the vehicles kinematics. This defines the motion model $P(x_{t+1}|x_t)$ which projects the probability mass $P(x_{i,t})$ into the surrounding nodes using the velocity and direction estimates. The velocity and direction estimates can be adapted by setting the yaw rates and accelerations by an attractor function based on the context.

In [1] we introduced an attractor algorithm, which estimates the most likely driving trajectory of the monitored vehicle, given its current state estimate $(\mathbf{l}_i, ||\mathbf{v}_i||, \omega(\mathbf{v}_i))$ and the current traffic context c_k. In the meantime we improved the attractor algorithm further by making the trajectory splines acceleration minimized in x and y direction. To put it in a nutshell the attractor algorithm generates trajectories starting at each grid node and which end point lies in the lane center in front of the estimate.

From the trajectory estimate, yaw rates and accelerations are derived. In particular, we specified an attractor function AF, which generates an attractor location $\mathbf{l}_i^A$ as well as a heading direction $\omega(\mathbf{v})_i^A$ and velocity $||\mathbf{v}_i^A||$ given a state estimate in a particular grid node i and the current context c of the road and surrounding traffic, that is:

$$\begin{pmatrix} \omega(\mathbf{v}_i^A) \\ ||\mathbf{v}_i^A|| \\ \mathbf{l}_i^A \end{pmatrix} = AF(||\mathbf{v}_i||, \omega(\mathbf{v}_i), \mathbf{l}_i, c_k) \tag{1}$$

The prediction model is then altered by the behavior B_k (cf. (2) and Fig. 2), which is modeled by the attractor function. This means that each yaw rate and acceleration (in each node) in the prediction function $P(x_{t+1}|B_k, x_t)$ is set by the trajectories depending on B_k. Therefore, in a intersection scenario several prediction models can be created by generating alternative attractor functions following certain lanes, turning, or driving straight ahead. The prediction model, which fits to the (unknown) actual behavior will have the best tracking results. If you knew the actual position and velocity of the observed vehicle, it would be an easy task to determine the actual behavior of the vehicle. But because the position as well the behavior is a hidden variable, more

elaborate techniques are required. The technique used in this work is explained in the next two subsections.

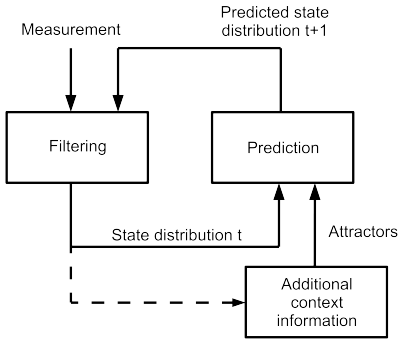


Fig. 1. A schematic view of the proposed Bayesian filter with context fusion algorithm

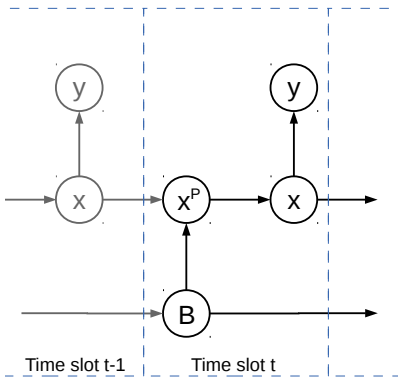


Fig. 2. The Bayesian filter in Bayesian network HMM notation: The behavior state B influences the trajectory of the observed vehicle and therefore the predicted state x_t^P depends on the former state x_{t-1} and the behavior B_t. The sensor output y_t depends on the current state x_t. The separation of x_t and x^P is artificial but will be useful in later equations.

B. Behavior Detection by Plausibility

The estimated position after the prediction depends on the used prediction model and the estimated position of the former time step. Equation (2) shows the probability that the observed vehicle is at a certain position in the next time step $t+1$, assuming that a certain behavior model B_k is right. It is denoted as predicted state x^P.

$$P(x_{t+1}^P|B_k) = P(x_{t+1}|B_k, x_t)P(x) \quad (2)$$

$$P(x_{t+1}^P) = \sum_k P(x_{t+1}|B_k)P(B_k|x_t) \quad (3)$$

The first product in (2) is the prediction model itself, incorporating lane information by the attractor function via B_k. $P(x)$ is the prior state assumption, given the previous prediction and the measurement. $P(B_k|x_t)$ is the probability that the prediction model using B_k is correct and it can be used like in (3) to sum the different distributions weighted to an overall distributions.

$P(B_k|x_t)$ answers the question "How high is the probability, that a certain prediction model from a set of prediction models is right, given the vehicle state?" or in other words "will the observed vehicle turn left or drive straight ahead?". In this section, we do not want to use this to sum up the distribution but to give a probabilistic answer to the question itself. To derive the answer, we derive a plausibility measure $Pl_{B_k} \simeq P(B_k|x_t)$.

Various plausibility measures have been used in the robotics literature, but no state-of-the-art has emerged, yet. The easiest way is to measure the distance between measurement and expectation value of the predicted state. Other more sophisticated approaches compare the probability overlap of both functions. For example the Kullback-Leibler divergence, the scalar product, or a shape-independent scalar product has been used [10][11]. The right measure has to be chosen dependent on the intended application. For our task, which is the behavior detection of other vehicles, we apply the shape independent scalar product. The benefit of this measure in our application is that the output is independent from the form of the sensory noise, which is not constant but declines with shrinking distance between us and the observed object.

$$Z_{d,B_k} = \sum_i^N P(\hat{x_{t+1}}_i|y+d)P(\hat{x_{t+1}}_i|B_k, x_t) \quad (4)$$

$$Z_{B_k} = max_d Z_d \quad (5)$$

$$Pl_{B_k} = \frac{Z_{d=0}}{Z_{B_k}} \quad (6)$$

Equation (4) is a convolution of the predicted state anticipating behavior B_k (second factor) and the sensor model (first factor) shifted by the position vector d. The convolution is executed over the whole state space $\hat{x_{t+1}}_i$ (where the hat just indicates that the real state despite of the estimated state is used). In (5) the maximum over all possible shifts is calculated, which is used for normalization in (6). The intuitive output of the calculation is the convolution (or overlap) of the sensor distribution and the predicted state distribution normalized by the highest possible overlap an optimal fitting sensor measurement could produce. For example, if the sensor creates the highest possible overlap with the predicted state, the sensor fits best with the predicted state, therefore receiving $Pl = 1$. Despite that, if the sensor model shifted by an optimal d would fit better than the sensor itself, Pl receives a smaller value by the normalization term in (6).

C. Estimating Plausibility by Filtering Over the Observed Plausibility

To estimate plausibilities, the basic idea is to compare fits between the sensor measurements and the predicted behavior B_k from the set of behavior models $\{B_k|k \in 1..K\}$. The plausibility measure itself, however, is not appropriate to make this comparison. This is because a direct comparison of the probability state distribution after the prediction step with the noisy sensor measurement will result in a noisy observed plausibility measure. In order to derive the actual plausibility, we track the observed plausibility measure over time with a hidden Markov Model (HMM) (cf. Fig. 3).

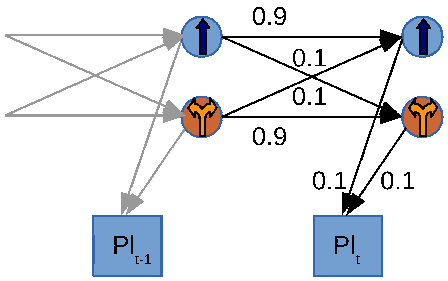


Fig. 3. The Plausibility $\hat{Pl}$ that a certain behavior model is true given the plausibility measurement Pl. The plausibility measure is an observable variable. The plausibility of a certain behavior model is a hidden variable in a HMM, since the ego-observer does not know the intention of the observed objects driver. The constants are set by hand and reflect that vehicles keep their behavior constant with a high probability.

The HMM improves the model estimation $P(B_k)$ by filtering over time, assuming that the behavior observed in the last time step adds information to the knowledge in the current time step. This is a valid assumption since an observed vehicle with the behavior "turning" has a higher probability to stay in the "turning" state than changing to another state from one time step to the next. Doing so, we can introduce the behavior in the last time step $B_{b_{t-1}}$ as a conditional variable. This leads to a first order HMM and the probability that model k is right is given by:

$$\begin{aligned} P(B_k) :=& P(B_{k_t}|x_t^P, Y) \simeq P(B_{k_t}|x_t^P, x_t) \\ =& \sum_{b \in B} P(B_{k_t}|x^P(B_{b_{t-1}}), x_t) \cdot P(B_{b_{t-1}}) \end{aligned} \quad (7)$$

$P(B_{k_t})$ is the new observed probability that model B_k is right given the predicted internal state x_t^P and the new observation Y. This approximates the probability that the model B_k is right given the predicted internal state and the internal state after sensor fusion x_t. This is given by the sum over the behavior transition functions $P(B_{k_t}|x^P(B_{b_{t-1}}), x_t)$, which output how the behavior from the last time step $B_{b_{t-1}}$ to the current time. $P(B_{b_{t-1}})$ is the prior model assumption. In the beginning it is initialized with a prior value B_{b_0}, which can be set equally distributed or by a prior assumption. For example, risky situations can be given a higher prior, in which case the system will believe in the risky behavior until strong evidence against it is available. Moreover a threshold value θ can be introduced as hysteresis to avoid oscillation between various models during ambiguous moments. The assumption switch that a model $k = 2$ is correct instead of model $k = 1$ is done when $P(B_{k=2}) > P(B_{k=1} + \theta)$.

D. Open Street Map Data as Source for Context Information

Now lane information has to be gathered in order to generate the behavior alternatives. There are two different ways how to gain lane information. One way is to use on-board sensors, such as cameras. The advantage is that there are no localization errors of the own vehicle in the global coordinates, but the disadvantage is, that there is no way to look behind vehicles in the view. This is critical, since we have to know the street in front of the other vehicles movement direction in order to predict its driving. Using a pure on-board sensor approach would lead to the limitation that only the behavior of oncoming vehicles can be tracked. Vehicles that are driving ahead typically occlude the driving space ahead of them so that no lane information may be available (cf. Fig. 15). Alternatively, global lane positions may be used. This information can be gathered by global map databases. Since the information is saved globally, the occlusion drawback does not apply. However, the map data is only available in a global reference frame and must be translated into the moving local ego-vehicle reference frame (cf. Fig. 5). Position offsets and heading errors can lead to inaccurate lane positions in the local reference system, which may lead to inaccurate attractor positions and thus useless movement models.

Thus, we suggest to combine both sources of information. Figure 4 shows the overall process. The vehicle's sensors return relative positions of the observed vehicle and the position by GPS/IMU. The lanes in vicinity of the observed vehicles position are looked up in an online database like OSM. Using the heading and the ego-vehicle position the lanes are converted into the vehicle's local coordinate system. While the relative observed vehicle position is directly given to the grid filter as measurement input, the lane information needs further fine positioning. There are multiple reasons for that: first the localization given by GPS and IMU is not accurate enough. Second, the map data may not be accurate, so that the true street position deviates from the street position in the map. Third, the lanes may not be directly derived from the database. If, for example, OSM is used for the context information, only a rather coarse road graph can be derived. While this may not cause a problem in rural areas, large intersections in urban environments are much more complex. Therefore the lane fine positioning module needs additional information from on-board sensors or from other databases. On-board sensors may be cameras, which may detect how the lane proceeds. No matter which data sources are used, the chosen trajectory in intersections depends on many factors. For example, drives of vehicles that are turning left may choose a different trajectory dependent on if oncoming traffic may have the right-of-way or if all left-turning vehicles have currently the right-of-way, for example, due to a left-turn signal. In the former alternative, they will probably choose a wider turn. Furthermore, lane markings in the intersection area will influence the choice of the turn, and other factors may be of influence. The result of the lane fine positioning module delivers the local lane information on which the attractor function adjusts the movement models.

The lane fine positioning or vise versa the localization of the ego-vehicle are big ongoing research fields of their own [12]. Since we intend to show that such information can be used to derive movement models, we annotated the lane information on Google Earth satellite data for our real world

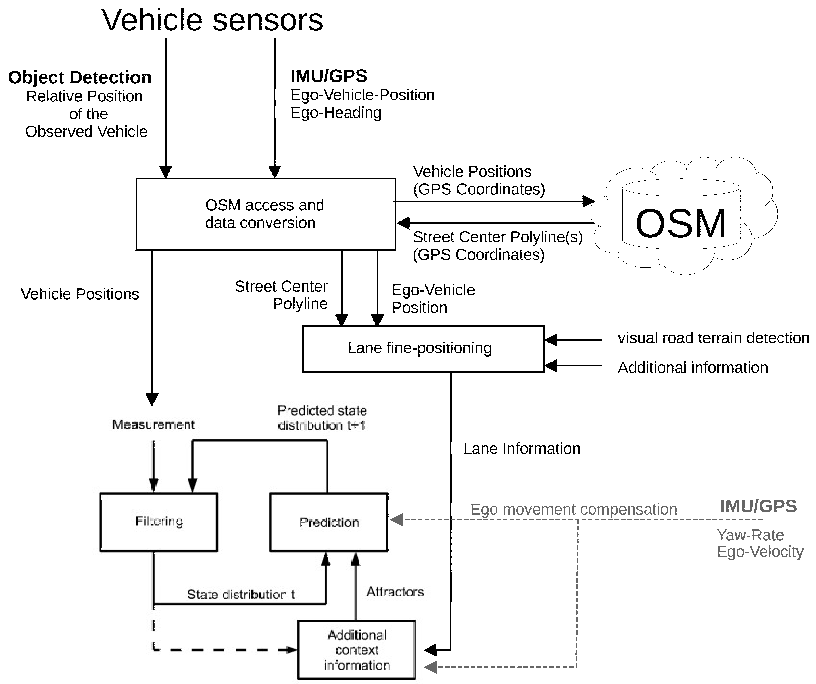


Fig. 4. A schematic view of the proposed data processing for open-street-map (OSM) data. Moreover it outlines the ego-movement compensation, that adapts the attractors and the prediction function. Thereby the non-inertial circular moved ego-reference-system is taken into account.

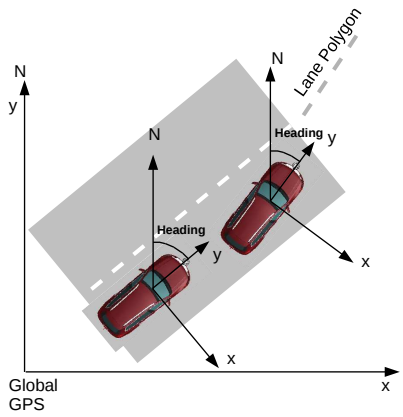


Fig. 5. Heading (yaw) and position of the ego-vehicle is needed in order to translate the global context data into the moved local ego-vehicle reference system.

scenarios 2 and 3.

IV. Experimental Evaluations

We have tested the approach with a simulated carmaker intersection scenario (scenario 1) in order to detect false positive and false negative detections by adding artificial sensor noise in 10 runs. The advantage of a simulation is that we can run the simulation several times with different sensor noise while using the equivalent, simulated series of vehicle positions in each run. Also, it is possible to split the trajectories into different behaviors starting from exactly the same conditions.

In scenarios 2 and 3 we tested real-world scenes with the parameters adjusted in the simulation. The data was obtained by a test-drive with a car using an camera object detection algorithm – GPS and IMU on-board. The lanes were annotated using data from Google earth for the reasons stated above. Parameter settings: The (virtual) vehicle detection sensor and CAN-data is incorporated in a 10 Hz rate. The street information is incorporated once in the beginning and rotated and translated into the local coordinate system using the vehicle position given by IMU and GPS in 10 Hz steps. The Bayesian grid filter uses a node distance of 0.25 m and covers an area of 50 m x 110 m leading to 44000 rectangular Voronoi areas. The initial position estimate distribution is set by the first measurement. The velocity is initialized by the position difference between the measurement and the grid node position.

A. Cross-validation by simulation

The simulation allows to test the algorithm for false positive and false negative classifications. The intersection scenario seen in Fig. 6 evaluates the algorithm with an oncoming vehicle (yellow). The vehicle has two behavior alternatives. It can follow the straight lane or turn left. The simulation allows to drive both alternatives with exactly the same path. Both vehicles drive with 30 km/h and no velocity reduction takes place before entering the curve. This makes it impossible for the algorithm to use the absolute velocity as an easy criteria for behavior detection. The modeled sensors

of the red ego-vehicle detect the position of the other vehicle with a quite strong sensor noise. This would in practice also lead to difficulties using the velocity as criteria since the derivative of a noisy time series is even noisier than the time series itself. The reader should also recall that even the noisy position given alone - without tracking - could be used to estimate the behavior of the observed vehicle, but the high noise level would lead to a very late detection. E.g. when the vehicle is sensed on the other lane that may indicate a turn of the vehicle, but the high sensor noise is a more probable explanation for that sensing. Without Bayesian tracking, non-model-based methods like a moving averaging has to be done. But non-model-based methods implicate a serious temporal delay for the behavior detection. Such high delays are unwanted in the ADAS domain. With this simulation runs we want to show that our system copes with high sensory noise. False positives and true positives can be prevented by the right parameter set. Therefore when the a priori behavior assumption was right, the behavior of no run should flip to the false model assumption during the scene (avoiding false positives). And when the a priori behavior assumption was wrong, the behavior of all models should switch to the right model assumption in time (avoiding false negatives).

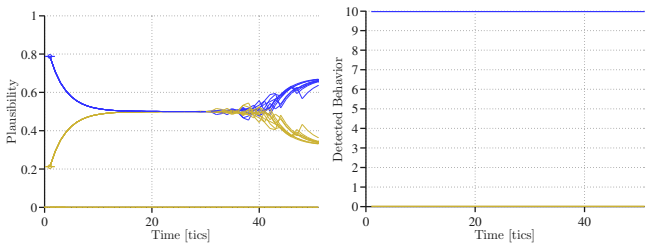


(a) (b)

Fig. 6. Intersection scenario 1. The first picture is identical in scenario 1a and 1b. (b) shows the turning action of scenario 1a in a distance of about 20-30 m.

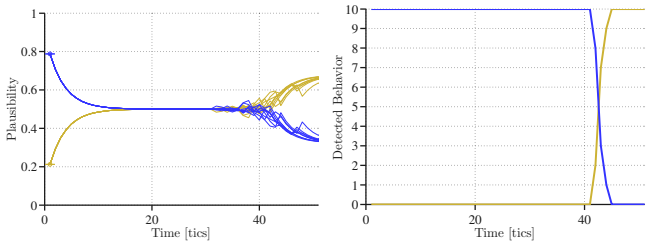
The result of 10 runs with the vehicle turning (Scenario 1a) is depicted in Fig. 8 and 7. The cross validation result of the algorithm when the vehicle drives straight (Scenario 1b) is shown in Fig. 10 and 9. In Fig. 7 the prior plausibility is set to turning. This is the most important setting incorporating risk reflections into the prior. It is a risky situation for the red car if yellow turns to the left unexpectedly, which driving straight ahead is without risk for both. In this setting the belief into the risky turning behavior stays active in all 10 runs, because no evidence contradicts the prior assumption (cf. Fig 7(b)). To test the unexpected detection capability, we ran the same scenario with the prior set to driving straight ahead. Thus, sufficient evidence against the prior needs to be accumulated (cf. Fig. 8(a)). All runs detected this change (Fig. 8(b)).

The simulation also enables us to test if the algorithm correctly detects that the vehicle is not turning. When the prior is set to turning behavior, Fig. 9 shows that the algorithm's belief appropriately changes from "turning" to "straight driving". Note that the horizontal position in Fig. 9(b) indicates how far from the yellow vehicle's lane



(a) (b)

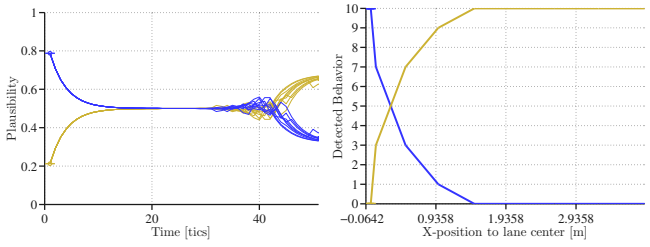
Fig. 7. Evaluation of scneario 1a. Blue is the prior assumption (Turning) (a) The plausibility over time. (b) The detected behavior of 10 different runs.



(a) (b)

Fig. 8. Evaluation of scneario 1a. Blue is the prior assumption (Driving Straight) (a) The plausibility over time. (b) The detected behavior of 10 different runs.

center this occurs (lane width is 3 m).

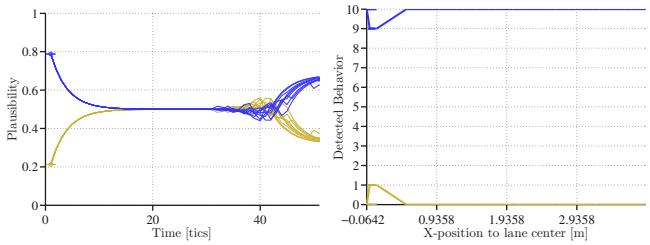


(a) (b)

Fig. 9. Evaluation of scneario 1b. Blue is the prior assumption (Turning) (a) The plausibility over time. (b) The detected behavior of 10 different runs. At x-position 1.5 m the street center line is crossed.

Figure 10 shows the compliant prior again. Assuming that the vehicle drives straight from the beginning, the belief should not change since the vehicle is indeed driving straight. But this time a false positive occurs due to the high sensor noise. This false positive vanishes if changing the threshold value from $\theta = .12$ to $\theta = .13$, but in this case the detection occurs slightly later. Thus, the threshold value allows to fine tune the trade-off between detection delay and detection accuracy.

The characteristic of the noise was set fix on a relatively high level for all runs (Gaussian white noise with $\sigma_x = 1.8m$ and $\sigma_y = 0.9m$). Higher noise values would lead to a higher false positives rate, lower values to less false positives. This errors can be avoided by adapting the θ value, where a higher theta value leads to a later detection but more accurate



(a) (b)

Fig. 10. Evaluation of scneario 1b. Blue is the prior assumption (Driving straight) (a) The plausibility over time. (b) The detected behavior of 10 different runs. At x-position 1.5 m the street center line is crossed.

behavior detection. Thereby the detection time is indirectly determined by the sensor noise, via the threshold value.

B. Real-world Intersection Scenarios

We used the same parameter set gained in the simulation to evaluate the algorithm in real-world intersection scenarios. A satellite image (Fig. 11) and an on-board camera view (Fig. 12) of Scenario 2 is shown. The observed vehicle and our ego-vehicle are leaving the road heading North via an exclusive left turning lane towards the east. We modeled two behavior alternatives. Turning left towards one of the two destination lanes or driving straight on the most-left straight driving lane. The other lanes are omitted since their prior would be near zero as a result of the fact that the vehicle was detected on the most-left lane. Setting the prior belief to "turning left" (Fig. 13(a)) leads to no change in the behavior belief. In Fig. 13(a) the driving straight ahead prior belief was quickly deemed incorrect. This occurs very early due to the lane difference. Note also that the plausibility estimates of both behaviors meet each other again after the turn. This is due to two reasons. First, when the sensor measurement deviates too much from the model assumption, the attractor has only a small influence. Second, in our runs we simply continued both attractors and did not account for their applicability; that is once the vehicle fully moved into the other road the attractor moving straight ahead should not be applied anymore.

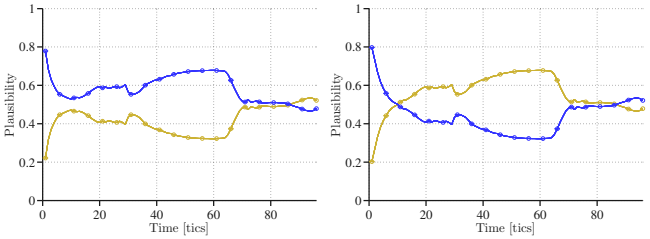
Scenario 3 is a rather unusually skewed intersection (cf. Fig. 14 and 15). The observed vehicle and the tracking vehicle are approaching from the south turning left. This time there is only one lane for all behavior alternatives. The road from the North is a one-way road, so that either a left turning behavior or right turning behavior can be expected. The evaluation (Fig. 16) shows that the detection works well. In comparison with Scenario 2, the plausibility graph is not as clear-cut, though. The main reason for this fact is that the left turning model assumes an insufficiently wide turn. Nevertheless the threshold value ($\theta = .12$) ensures that "left turning" was detected successfully.

V. Summary and Conclusion

In this paper we applied a spline-based attractor algorithm which derived its relevant parameters from annotated map



Fig. 11. In scenario 2 the ego vehicle is coming from the north and turning to the east. There is one exclusive left turning lane and two possible target lanes in the road Spessartring. (Map data by Google Images/GeoBasis-DE/BKG and AeroWest)

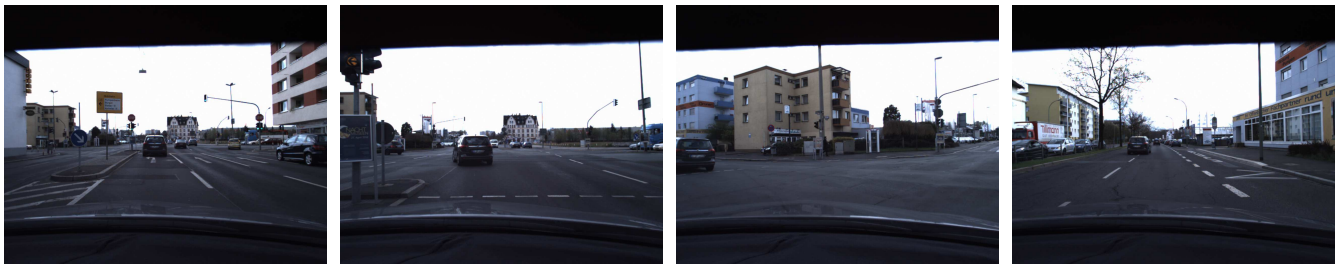


(a) (b)

Fig. 13. Scenario 2. The plausibility over time. Blue is the prior assumption. (a) Prior assumption is doing a turn. (b) Prior assumption is driving straight. In (a) turning was assumed all the time. In (b) turning was detected at time step 17 until the end.

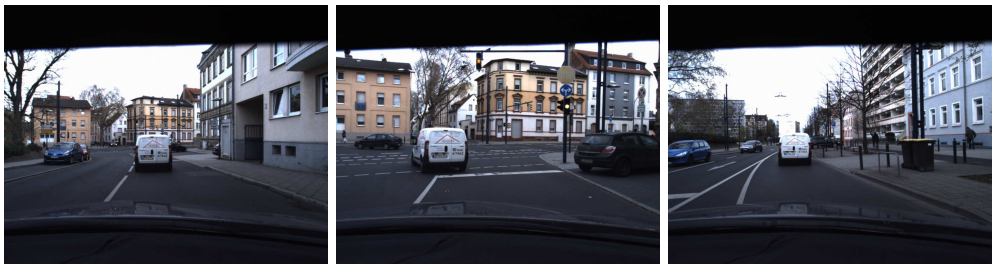


Fig. 14. Intersection scenario 3 shows a intersection scene with lanes for west-east traffic only. The ego vehicle follows the observed vehicle which approaches from south and turns left. Since the northern road is a one way road the vehicle can turn left or turn right. (Map data by Google Images/GeoBasis-DE/BKG and AeroWest)



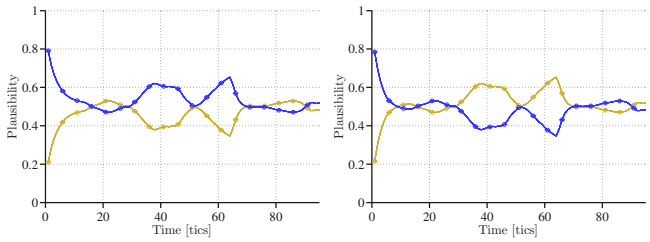
(a) (b) (c) (d)

Fig. 12. Intersection scenario 2 camera output



(a) (b) (c)

Fig. 15. Intersection scenario 3 camera output



(a) (b)

Fig. 16. Scenario 3. The plausibility over time. Blue is the prior assumption. (a) Prior assumption is doing a left turn. (b) Prior assumption is a right turn. In (a) left turning was assumed all the time. In (b) from time step 34 to the end was detected that a right turn is not the real behavior.

data. This attractor algorithm was used to create different behavior models which basically represent probabilistic vehicle trajectories. A plausibility measure was introduced to compare different behavior models. In the evaluation we have shown the challenges for the algorithm like inaccurate map data or ego positions and heading. The evaluation on simulation and real world data have shown that the combination of anticipatory and factual information allows to infer the behavior of other vehicles effectively. However we have also shown challenges to the algorithm: inaccurate map data and inexact ego-localizing can result in inaccurate attractor influences and thus in worse behavior classification. In the future new behavior models should be created and removed from on the fly dependent on the context. Also the trajectory adjustment based on additional information sources can be further improved in order to gain more robustness.

REFERENCES

[1] A. Alin, M. V. Butz, and J. Fritsch, "Incorporating environmental knowledge into bayesian filterung using attractor functions," in *IEEE Intelligent Vehicles Symposium (IV '12)*, Alcala de Henares, June 2012, pp. 476–481.

[2] C. Herdtweck and C. Curio, "Experts of probabilistic flow subspaces for robust monocular odometry in urban areas," in *Intelligent Vehicles Symposium*, 2012, pp. 661–667.

[3] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR)*, Providence, USA, June 2012.

[4] A. Barth and U. Franke, "Where Will the Oncoming Vehicle be the Next Second?" in *Proceedings of the IEEE Intelligent Vehicles Symposium*. Eindhoven: Springer, June 2008, pp. 1068–1073.

[5] ——, "Tracking oncoming and turning vehicles at intersections," in *Intelligent Transportation Systems, IEEE Conference on*, Madeira Island, Portugal, 2010, pp. 861–868.

[6] G. S. Aoude, V. R. Desaraju, L. H. Stephens, and J. P. How, "Behavior classification algorithms at intersections and validation using naturalistic data," in *Intelligent Vehicles Symposium*. IEEE, June 2011. [Online]. Available: http://acl.mit.edu/papers/IV11AoudeDesarajuLaurensHow.pdf

[7] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Risk Assessment at Road Intersections: Comparing Intention and Expectation," pp. 165–171, 2012. [Online]. Available: http://hal.inria.fr/hal-00743219

[8] A. Alin, M. V. Butz, and J. Fritsch, "Tracking moving vehicles using an advanced grid-based bayesian filter approach," *IEEE Intelligent Vehicles Symposium (IV)*, pp. 466–472, 2011.

[9] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, USA: MIT Press, 2006.

[10] C. Zhang and J. Eggert, "Tracking with multiple prediction models," in *ICANN (2)*, 2009, pp. 855–864.

[11] S. Ehrenfeld and M. V. Butz, "The modular modality frame model: continuous body state estimation and plausibility-weighted information fusion," *Biological cybernetics*, vol. 107, no. 1, pp. 61–82, 2013.

[12] R. Toledo-Moreo, D. Bétaille, and F. Peyret, "Lane-level integrity provision for navigation and map matching with gnss, dead reckoning, and enhanced maps," vol. 11, no. 1, 2010, pp. 100–112.