2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Recognition of Situation Classes at Road Intersections

Eugen Käfer, Christoph Hermes, Christian Wöhler, Helge Ritter and Franz Kummert

Abstract— **The recognition and prediction of situations is an indispensable skill of future driver assistance systems. This study focuses on the recognition of situations involving two vehicles at intersections. For each vehicle, a set of possible future motion trajectories is estimated and rated based on a motion database for a time interval of 2–4 seconds ahead. Realistic situations are generated by a pairwise combination of these individual motion trajectories and classified according to nine categories with a polynomial classifier. In the proposed framework, situations are penalised for which the time to collision significantly exceeds the typical human reaction time. The correspondingly favoured situations are combined by a probabilistic framework, resulting in a more reliable situation recognition and collision detection than obtained based on independent motion hypotheses. The proposed method is evaluated on a real-world differential GPS data set acquired during a test drive of 10 km, including three road intersections. Our method is typically able to recognise the situation correctly about 1–2 seconds before the distance to the intersection centre becomes minimal.**

I. Introduction

The recognition of the situation into which a vehicle is currently involved is an important information for an advanced driver assistance system (ADAS). The goal of such systems is to increase the passenger comfort and safety by supporting the driver with environmental information such as the current and expected future behaviour of traffic participants and obstacles. Of particular interest are cooperative situations, where an ADAS not only estimates and predicts the motion states of the traffic participants but also considers the possible interaction behaviour between them. Especially, manoeuvres at intersections with traffic lights involving oncoming vehicles may represent a potential hazard for the own vehicle e.g. due to distractions of other traffic participants. A classification of the current interaction can help to inform the driver about the situation "anticipated" by the vehicle. It may even be able to correct erroneous individual motion predictions and thus reduce false alarms.

To our knowledge, the topic of recognition of situation classes in the context of cooperative vehicles has not been studied extensively yet. Huang et al. [1] apply Dynamic Belief Networks (DBNs) to make inferences about symbolic traffic events such as lane changes or stalled vehicles. In the case of an occurrence of multiple objects, each vehicle is assigned its own DBN. Vehicle motion trajectories are extracted from image sequences using a Kalman filter. Oliver et al. [2] recognise the driver behaviour with a coupled Hidden Markov Model (CHMM). Although only a one-way influence from the surrounding environment to the driver is assumed, an extension to mutual object interactions is possible with a CHMM [3], [4].

Determining a certain situation class for a dynamic scenario is closely related to the field of symbolic gesture recognition. Nickel and Stiefelhagen [5] divide a hand gesture into three atomic states (begin, hold, end) and use a Hidden Markov Model (HMM) for the recognition of pointing gestures. A particle filter framework for gesture recognition is proposed by Black and Jepson [6], who model gestures as temporal trajectories of the velocities of the tracked hands.

In an ADAS, road safety is an important research issue. This requires reliable methods for predicting the behaviour of traffic participants considering their mutual relations. Broadhurst et al. [7] plan motion trajectories using a Monte Carlo sampling method. They also infer possible future motion states of other traffic participants. An extension is proposed by Eidehall and Petersson [8], who determine the threat level of a traffic scene. Althoff et al. [9] predict potentially hazardous situations based on stochastically reachable sets of traffic participants by defining discrete actions, e.g. acceleration or breaking. They also take into account multiple traffic participants. Batz et al. [10] recognise dangerous situations in a traffic scenario within a cooperative group of vehicles in a simulated environment. They use Extended and Unscented Kalman Filters to obtain vehicle trajectories and predict the vehicle motion for a prediction horizon of 1 s. For prediction, they also include the shape of the road and the positions of obstacles and other vehicles.

Large et al. [11] cluster a set of trajectories in order to retrieve typical motion patterns. These patterns are used for motion planning and prediction of robots and vehicles. Johnson and Hogg [12] represent a set of pedestrian trajectories using a neural network combined with vector quantisation. The authors suggest an event recognition method using probability densities which are determined by the distribution of the prototype vectors, but they do not evaluate the system. Hu et al. [13] apply a hierarchical clustering algorithm to retrieve typical motion patterns from a given set of trajectories. Each motion pattern is represented by a chain of Gaussian distributions which are used for statistical anomaly detection and behaviour prediction. Croitoru et al. [14] present a non-iterative 3D trajectory matching framework based on shape signatures. Their approach is invariant to translation, rotation, and scale. They apply their system to 3D trajectories for

E. Käfer and C. Wöhler are with Daimler AG Group Research, Böblingen / Ulm, Germany, {Eugen.Kaefer, Christian.Woehler}@daimler.com

C. Hermes, H. Ritter, F. Kummert and C. Wöhler are with the Faculty of Technology, Bielefeld University, Bielefeld, Germany, {chermes, helge, franz, cwoehler}@techfak.uni-bielefeld.de

which the beginning and the end are known.

In this study we propose a framework for recognition of intersection situations involving two oncoming vehicles. The basis for this framework is a computationally efficient implementation of the long-term motion prediction method presented in [15], which is applied to each vehicle separately. A database of previously acquired vehicle motion patterns (trajectories) is used for a long-term prediction ($\sim$2 s) for each vehicle independently (cf. Section II). Subsequently, the interaction behaviour of the traffic participants is taken into account by a situation classifier (cf. Section V). The proposed framework is evaluated on a real-world differential GPS data set (cf. Section VI). The results are summarised in Section VII.

II. Motion Representation and Motion Database

We represent the motion patterns of vehicles by trajectories, which are defined as ordered tuples $X = ((\mathbf{x}_1, t_1), ..., (\mathbf{x}_N, t_N))$ combining states $\mathbf{x}_i$ with a time stamp t_i. Therefore, each trajectory element $\mathbf{x}_i$ describes the current state of the tracked object over time. This representation does not depend on a specific sensor type, and parts of $\mathbf{x}_i$ may originate from different types of sensors. When applied to vehicle motion, the basic information is the object position in the 2D plane and the orientation angle relative to the ego-vehicle, here termed yaw angle. Furthermore, we found it useful to additionally encode the temporal derivatives, i.e. the velocity and the yaw rate, in the trajectory.

Humans are able to learn motion patterns and to predict the behaviour of traffic participants fairly accurately over time when the history of the moving objects is known. Our basic approach in this study is to adopt this capability of learning motion patterns by building a motion database consisting of observed trajectories. A combination of these reference trajectories from multiple vehicles leads to an early recognition of the current situation.

III. Motion Comparison

Our approach requires an efficient matching technique for trajectories in order to infer a prediction of the motion states. The Longest Common Subsequence (LCS) metric [16] on trajectories has been shown to be an adequate metric, but is not capable of comparing trajectories under different rotation and translation conditions. These properties are necessary because the similarity of motion patterns does not depend on the initial position and orientation. We thus adopt the LCS metric and introduce the property of rotational invariance using the method by Horn [17], which finds the optimal orthogonal transformation to superimpose two point sets based on quaternions. This optimisation method is computationally more efficient than the approach proposed in [15].

A. Longest Common Subsequence (LCS)

The LCS metric was introduced in the field of string matching and yields the length of the longest common substring contained in two strings. To apply this technique to trajectories, a similarity matching function between two states (points) $\mathbf{a}_i$ and $\mathbf{b}_j$ from the given trajectory points $(\mathbf{a}_i, t_{a,i}) \in A$ and $(\mathbf{b}_j, t_{b,j}) \in B$ has to be defined. In the context of string matching algorithms, the similarity between two single characters is defined according to their equality. For the trajectory points $\mathbf{a}_i$ and $\mathbf{b}_j$ we define a fixed decision boundary $\boldsymbol{\varepsilon}$ with values in each dimension d and apply a linear function in the range $[0, \varepsilon^{(d)}]$ to obtain the distance between $\mathbf{a}_i$ and $\mathbf{b}_j$, where $L^1(\cdot)$ denotes the L1 norm (Manhattan distance):

$$\mathrm{dist}(\mathbf{a}_i, \mathbf{b}_j) = \begin{cases} 0 \textbf{ if } \exists d \in D\colon L^1(a_i^{(d)}, b_j^{(d)}) > \varepsilon^{(d)} \\ \frac{1}{D}\sum_{d=1}^{D}\left(1 - \frac{L^1(a_i^{(d)}, b_j^{(d)})}{\varepsilon^{(d)}}\right) \textbf{ otherwise} \end{cases} \tag{1}$$

The sizes of the trajectories A and B are denoted by N_A and N_B, respectively, corresponding to the number of motion states they comprise, and the sequence $[(\mathbf{a}_1, t_{a,1}), \ldots, (\mathbf{a}_{N_A-1}, t_{a,N_A-1})]$ by $\mathrm{head}(A)$. We then define the LCS on trajectories as follows:

$$\mathrm{LCS}(A, B) = \begin{cases} 0 \textbf{ if } N_A = 0 \ \wedge \ N_B = 0 \\ \mathrm{LCS}(\mathrm{head}(A), \mathrm{head}(B)) + \mathrm{dist}(\mathbf{a}_{N_A}, \mathbf{b}_{N_B}) \\ \qquad \textbf{if } \mathrm{dist}(\mathbf{a}_{N_A}, \mathbf{b}_{N_B}) \neq 0 \\ \max\{\mathrm{LCS}(\mathrm{head}(A), B), \mathrm{LCS}(A, \mathrm{head}(B))\} \\ \qquad \textbf{otherwise} \end{cases} \tag{2}$$

The distance between two trajectories A and B can then be obtained by

$$\mathrm{dist}_{\mathrm{LCS}}(A, B) = 1 - \frac{\mathrm{LCS}(A, B)}{\min\{N_A, N_B\}} \tag{3}$$

with $\mathrm{dist}_{\mathrm{LCS}}(A, B) \in [0, 1]$.

B. Quaternion-based Rotationally Invariant LCS (QRLCS)

The LCS can be computed with the dynamic programming (DP) algorithm in $O(n^2)$ time using tables where partial (optimal) results of the algorithm are stored. Each partial LCS distance in the DP table is optimal for the given (sub-) trajectories. The optimal, partial translation and rotation is obtained by regarding the (sub-) trajectories as two point sets in which the point-to-point assignments are given by the DP table. Hence, the rotation angle and both the mean values for each set are computed incrementally.

The best translation of a trajectory A to a trajectory B can be obtained using the mean values $\boldsymbol{\mu}_a$ and $\boldsymbol{\mu}_b$ of the two point sets in the xy plane. T denotes the number of assignments:

$$\boldsymbol{\mu}_{a,T} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{a}_t = \frac{T-1}{T}\boldsymbol{\mu}_{a,T-1} + \frac{1}{T}\mathbf{a}_T \tag{4}$$

For the optimal rotation, Horn [17] provides a closed-form solution for the least-squares problem of absolute orientation with quaternions. A quaternion can be considered as a complex number with three different imaginary parts or the combination of a scalar with a 3D Cartesian vector, $\dot{\mathbf{q}} = q_0 + iq_x + jq_y + kq_z \equiv [q_0, \mathbf{q}]$. It can be used as a rotation operator on a three-dimensional vector $\mathbf{x}$ as follows,

where vectors are treated as quaternions with zero scalar component:

$$[0, \mathbf{x}^R] = \dot{\mathbf{q}}[0, \mathbf{x}]\dot{\mathbf{q}}^{-1}. \quad (5)$$

A rotation quaternion can be constructed when the rotation angle θ and the three-dimensional rotation vector $\mathbf{n}$ are known:

$$\dot{\mathbf{q}} = \sqrt{\frac{1+\cos\theta}{2}} + \frac{\sin\theta}{\sqrt{2(1+\cos\theta)}}(in_x + jn_y + kn_z) \quad (6)$$

Because of the reasonable flat-world assumption the rotation vector can be set to the fixed value $\mathbf{n} = [0, 0, 1]$. For the rotation in a plane, Horn [17] suggests an efficient method where the rotation vector from two given point sets can be constructed using implicit trigonometric values:

$$\sin\theta = \frac{S}{\sqrt{S^2 + C^2}}, \quad \cos\theta = \frac{C}{\sqrt{S^2 + C^2}}. \quad (7)$$

By rotating the point sets (trajectories) up to an assignment T around their mean centres $\boldsymbol{\mu}_{a,T}$ and $\boldsymbol{\mu}_{b,T}$, the auxiliary variables S and C become

$$S_T = \left\langle \left(\sum_{t=1}^{T} (\mathbf{a}_t \times \mathbf{b}_t) - T(\boldsymbol{\mu}_{a,T} \times \boldsymbol{\mu}_{b,T}) \right), \mathbf{n} \right\rangle \quad (8)$$

$$C_T = \sum_{t=1}^{T} \langle \mathbf{a}_t, \mathbf{b}_t \rangle - T \left\langle \boldsymbol{\mu}_{a,T}, \boldsymbol{\mu}_{b,T} \right\rangle, \quad (9)$$

where $\langle \cdot, \cdot \rangle$ and $\times$ denote the dot product and the cross product, respectively. As a result, only the incremental parts $\boldsymbol{\mu}_{a,T}$ and $\boldsymbol{\mu}_{b,T}$ (Eq. (4)), $\sum_{t=1}^{T}(\mathbf{a}_t \times \mathbf{b}_t)$ (from Eq. (8)) and $\sum_{t=1}^{T} \langle \mathbf{a}_t, \mathbf{b}_t \rangle$ (from Eq. (9)) have to be stored at each assignment test in the DP table to guarantee the best rotation and translation.

IV. Probabilistic Search in Motion Database

For motion prediction retrieval, it is possible to compare each trajectory in the motion database with an observed history using the QRLCS motion comparison method described in Section III. However, this is not very efficient when the database becomes large, e.g. because of adding new observed motions at runtime. Hence, we replaced this greedy search by a probabilistic search framework adapted from Sidenbladh et al. [18], where the time needed for the search only depends on a specified number of sampling points.

The value $p(\boldsymbol{\phi}_T | M_{1:t})$ denotes the probability that a future vehicle state $\boldsymbol{\phi}_T$ occurs, given a motion history $M_{1:t}$ up to the current time step t:

$$p(\boldsymbol{\phi}_T | M_{1:t}) = p(\boldsymbol{\phi}_T | \Psi_t)\, p(\Psi_t | M_{1:t}) \quad (10)$$

where $p(\boldsymbol{\phi}_T | \Psi_t)$ is the probability of observing a future state $\boldsymbol{\phi}_T$, given the current state Ψ_t, and is determined from the motion database. The future state $\boldsymbol{\phi}_T$ can also be regarded as a series of future poses, i.e. a future trajectory. The current state Ψ_t represents a sequence of trajectory points including the position and the current time t and its history over a given travelled distance d.

Applying the Bayes Rule on the remaining distribution $p(\Psi_t | M_{1:t})$ results in an estimation of the current state based on the current measurement and the previous states according to

$$p(\boldsymbol{\phi}_T | M_{1:t}) = \eta\, p(M_{1:t} | \Psi_t) \int p(\Psi_t | \Psi_{t-1})\, p(\Psi_{t-1} | M_{1:t-1})\, d\Psi_{t-1} \quad (11)$$

with η as a normalisation constant. This distribution is represented by a set of samples or particles $\{\Psi_t^{(s)}\}_S$, which are propagated in time using a particle filter [6]. Therefore, each particle $\Psi_t^{(s)}$ represents a sub-trajectory for the current state. According to [18], it is sufficient to sample the particles from the distribution $p(\Psi_t | \Psi_{t-1})$ in a motion database. The method constructs a binary tree over the motion database and performs the prediction step by a probabilistic search of the particles in the tree.

The distribution $p(M_{1:t} | \Psi_t)$ represents the likelihood that the measurement trajectory $M_{1:t}$ can be observed when the model trajectory is given. In the context of particle filters, this value corresponds to the weight of a particle and is constructed as the product of individual normalised contributions as follows. The first part is the QRLCS value describing the distance between the observed history and a particle, the second and third one are penalty terms for differences in velocity and yaw angle, respectively, and the last term takes into account the road topology and penalises future trajectories leading into the edge of the road.

Each particle is a representation of the assumed current object state with an assigned likelihood and yields a future trajectory by a lookup in the reference trajectory contained in the motion database.

V. Situation Classification

A situation depends on the temporal behaviour of one or more traffic participants usually acting dependently on each other. The motion prediction method described in section IV yields several future trajectories, i.e. prospective vehicle movements, for each traffic participant. The method also provides a likelihood $p(\boldsymbol{\phi}_T | M_{1:t})$ for each future trajectory $\boldsymbol{\phi}_T$ based on the previously observed motion history $M_{1:t}$ according to Eq. (10).

For simplification, it is assumed in the following sections that two vehicles A and B approach each other from opposite directions of an intersection in the form of two crossing roads ("x-intersection", cf. Fig. 3(a)). The previously computed probability distribution $p(\boldsymbol{\phi}_T | M_{1:t})$ for each vehicle does not depend on the movements of other vehicles, i.e. possible future trajectories of other vehicles were not taken into account in Section III for the prediction of a specific vehicle. In the context of the particle filter, at every time step the motion predictions yields for both vehicles a set of trajectories $X_A^{(1)}, ..., X_A^{(N)}$ and $X_B^{(1)}, ..., X_B^{(M)}$ with associated likelihoods $P_A^{(1)}, ..., P_A^{(N)}$ and $P_B^{(1)}, ..., P_B^{(M)}$. A trajectory pair $(X_A^{(n)}, X_B^{(m)})$ represents one possible situation with the

occurrence probability $P^{(nm)} = P_A^{(n)}\, P_B^{(m)}$ if the vehicle movements are regarded as independent.

Furthermore, it is assumed that the vehicles are able to turn right (R), turn left (L), or drive straight on (G), which results in a set Ω of possible situation classes

$$\Omega = \{LL, LG, LR, RL, RG, RR, GL, GG, GR\}, \quad (12)$$

where in each pair the ith character denotes the motion of the ith vehicle.

Each predicted pair of motion trajectories $(X_A^{(n)}, X_B^{(m)})$ yields a so-called multiple participant trajectory (MPT) $T^{(nm)}$ which consists of difference components of the yaw angle and velocity (two dimensions) of two vehicles. A Chebyshev decomposition [19] is applied to these features, which results in coefficient vectors $\mathbf{T}_c^{(nm)}$. In order to infer the current situation class, a polynomial classifier [20] is trained on a labelled set of MPTs, such that it returns a probability $Q_k(\mathbf{T}_c^{(nm)})$, $k = 1, \ldots, K$, for each of the K situation classes.

The overall probability $W(k)$ for situation class k is given by

$$W(k) = \eta_w \sum_{n=1}^{N} \sum_{m=1}^{M} \rho^{(nm)}\, P^{(nm)}\, Q_k(\mathbf{T}_c^{(nm)}), \quad (13)$$

where η_w is a normalisation constant. In Eq. (13), we extend the occurrence probability $P^{(nm)}$ by an "interaction variable" $\rho^{(nm)}$. Assuming that the trajectories associated with $\mathbf{T}_c^{(nm)}$ are independent of each other, $\rho^{(nm)}$ becomes 1. Normally this is not the case because drivers tend to avoid potentially hazardous situations. The value of $\rho^{(nm)}$ is therefore chosen corresponding to the expected behaviour of the participating vehicles. As a simple "mental model" [21], we assume that the drivers have a strong tendency to avoid collisions only when the time-to-collision t_c is larger than their typical reaction time t_r, such that $\rho^{(nm)} \approx 1$ for $t_c^{(nm)} \ll t_r$ and $\rho^{(nm)} \ll 1$ for $t_c^{(nm)} \gg t_r$. We therefore set $\rho^{(nm)}$ according to the heuristically chosen relation

$$\rho^{(nm)} = (1 - \rho_{\min}) \exp\left(-\frac{1}{2}\left(\frac{t_c^{(nm)}}{t_r}\right)^2\right) + \rho_{\min} \quad (14)$$

where the parameter $\rho_{\min}$ denotes the minimum possible weight value. The time-to-collision $t_c^{(nm)}$ is determined based on the trajectory pair (nm). If no collision occurs, we have $\rho^{(nm)} = 1$. The reaction time t_r is typically chosen from the interval 0.5–1.0 s.

VI. Evaluation

For evaluation of the proposed method, we evaluate situations on different intersections generated by two vehicles approaching each other, respectively. Fig. 1 shows the data set used in this work. It consists of real-world differential GPS positions acquired during a test drive of 10 km. It includes three intersections with vehicles approaching from all possible directions, respectively. The GPS signals were tracked over time by a Kalman filter and yield estimates of the vehicle position, velocity, and orientation relative to a global coordinate system. The test vehicle with the differential GPS sensor had no further sensors for environment perception. In order to construct a situation at an intersection, we combined two separately recorded manoeuvres and aligned their timestamps manually such that the time $t = 0$ is defined as the moment in time when the last vehicle reaches its minimum distance to the intersection centre. Thus, the goal is to determine the earliest moment in time at which the situation class is recognised correctly. We assume that since the individual motion trajectories were acquired in real traffic, the correspondingly constructed situations are fairly realistic.

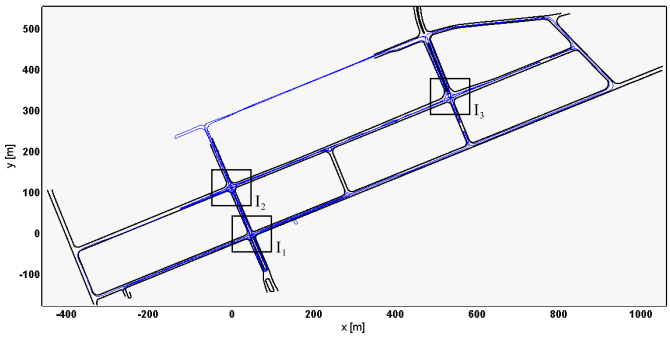


Fig. 1. Differential GPS data set obtained by a 10 km test drive. The three intersections are denoted by I_1, I_2 and I_3.

For motion prediction, the edges of the road are helpful for penalising unlikely predictions such as driving into a wall. Due to the sensor limitations of the test vehicle the edges were extracted from a map. The road surface is divided into quadratic grid cells, where each cell has a width of 0.24 m and is attributed by a flag determining the occupancy.

The parameters are set as follows and are not changed during the evaluation. The value of the decision boundary ε of the QRLCS distance metric (cf. Eq. (1)) is determined by the motion database and result in $\boldsymbol{\varepsilon} = [3.57, 3.57, 1.07, 0.05]$ for differences in x and y position, velocity, and yaw rate, respectively. The particle filter in the motion prediction uses $S = 100$ particles. Further parameters for the motion prediction are described in [15] and their values have not been changed for this evaluation. In the situation classifier method we use four Chebyshev coefficients, set the minimum possible weight value to $\rho_{min} = 0.1$ and the reaction time to $t_r = 1.0$ s.

Since the particle filter is a probabilistic approach, each run on the same trajectory will give slightly different predictions. However, in [15] it is shown that these deviations are reasonably small and the position standard deviation is typically below 0.5 m for a prediction horizon of 2.0 s.

In our approach, first the prediction is performed for each vehicle independently, then the MPTs were constructed and classified, leading to a discriminant value for each situation class. The influence of the interaction variable on the situation recognition is examined based on an example

TABLE I

CLASS-SPECIFIC PROBABILITIES $W(k)$ FOR THE SITUATION SHOWN IN FIG. 2 WITHOUT (W/O) AND WITH (W) INTERACTION.

	w/o	w		w/o	w		w/o	w
LL	0.04	0.06	RL	0.00	0.00	GL	0.17	0.24
LG	**0.41**	0.19	RG	0.19	**0.25**	GG	0.00	0.01
LR	0.01	0.00	RR	0.11	0.13	GR	0.07	0.12

in which one of the individual predictions is erroneous. Furthermore, we report a detailed analysis of the situation recognition results in our test scenario.

A. Influence of the Interaction Variable

In cases where the proposed prediction method leads to configurations implying a collision, the interaction variable defined in Eq. (14) becomes relevant. An example of an early RG situation is given in Fig. 2, where the red vehicle turns right and the oncoming blue vehicle drives straight on. Due to a false individual motion prediction obtained for the red vehicle (turn-left with a probability of 0.49) the situation is classified without the interaction variable as LG. Table I shows the class probabilities with (w) and without (w/o) using the interaction variable. Activating the interaction variable leads to a correct classification of the RG situation in this example.

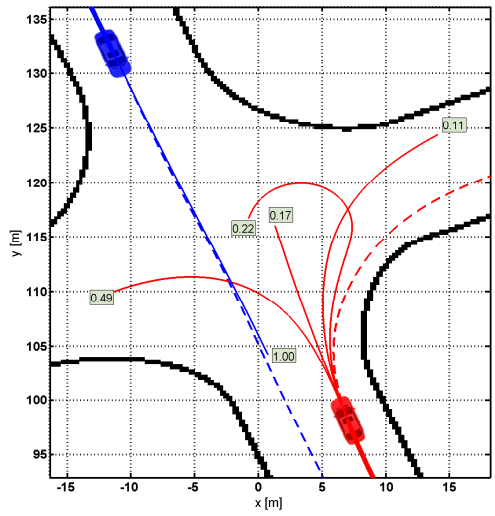


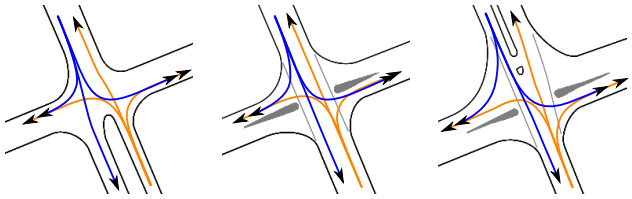
Fig. 2. False prediction for the red vehicle. Icons denote the current pose of each object. The dashed lines show the future positions (ground truth). The solid lines attributed with likelihood values represent the individual predictions. For the red vehicle, the (erroneous) turn-left prediction is the most likely hypothesis.

B. Prediction of the Situation Class

The proposed method recognises the class of a given situation. As a situation is a process in time, i.e. the vehicles do not only approach each other but also the intersection itself, at the beginning there is high uncertainty about the current situation and in the end the correct situation class can generally be determined with a low uncertainty. It is thus desired to recognise the correct situation class as early as possible.

For each situation class k an overall discriminant value $W(k)$ is estimated according to Eq. (13) by the polynomial classifier. We define the certainty γ_{k_r} of a classified situation class $k_r = \text{argmax}_{k\in\Omega} W(k)$ by its difference to the situation class with the second-largest discriminant value $k_{r2} = \text{argmax}_{k\in\Omega\setminus\{k_r\}} W(k)$ as $\gamma_{k_r} = W(k_r) - W(k_{r2})$. If γ_{k_r} exceeds a pre-defined threshold, the recognised situation class is accepted. In the following, the value of this threshold is set to 0.1.

At each test intersection I_i (cf. Fig. 3) one situation of each class was generated based on the individual trajectories. We have always taken into account the interaction variable given by Eq. (14). Table II displays the values of t_{rec} indicating the moment in time from which on the situation is correctly classified according to the previously defined γ_{k_r} criterion until the end of the situation at $t = 0$. The values of t_{rec} for a certain intersection I_i were obtained using the trajectories of the other two intersections $I_{j\neq i}$ as training data in a leave-one-out manner.



(a) I_1 (bottom) (b) I_2 (middle) (c) I_3 (right)

Fig. 3. Three test intersections from Fig. 1 with extracted road edges and possible manoeuvres for each vehicle.

The fact that all obtained t_{rec} values are negative indicates that all situations have been successfully recognised at the end. Some situation predictions are rather late, especially for intersection I_1. This observation can be explained by the fact that its geometry is rather different from those of intersections I_2 and I_3 used for training. Furthermore, the small values of t_{rec} for intersection I_1 are a result of ambiguous or temporal erroneous individual prediction. The median values of t_{rec}, however, are between -1.2 and -2.3 s, indicating a reasonably early correct prediction. This is especially true for the situation classes LL, LG, and GL, which are the most hazardous ones since collisions may occur as a result of the turn-left manoeuvre of one of the vehicles. For illustration, Fig. 4 shows the vehicles at time $t_{\text{rec}} = -1.92$ s, i.e. at the moment in which the situation is just recognised correctly, for a situation of class LL. It is apparent that both vehicles are just about to enter the intersection, and their orientations are still nearly parallel to the edges of the road, i.e. there are only weak indications of a turning manoeuvre so far. The temporal behaviour of the output discriminants in this example is shown in Fig. 5. The situation class LL is recognised in a stable manner at $t_{\text{rec}} = -1.92$ s.

VII. CONCLUSIONS AND FUTURE WORK

In this study we have described a method for the recognition of situations involving two vehicles at road intersections.

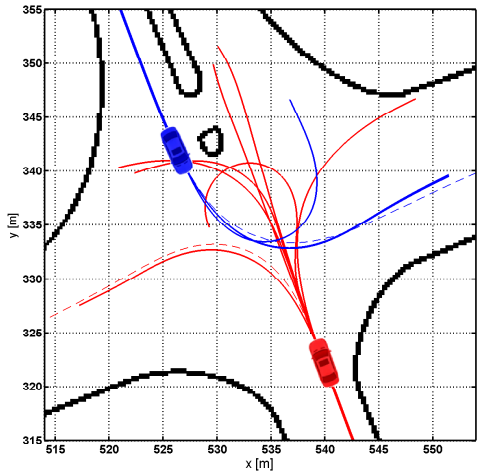


Fig. 4. Example of an early detection stage of a LL situation class at $t_{\mathrm{rec}} = -1.92$ s. The future position (ground truth) for a vehicle is represented by the dashed line, whereas the solid lines represent the individual predictions.

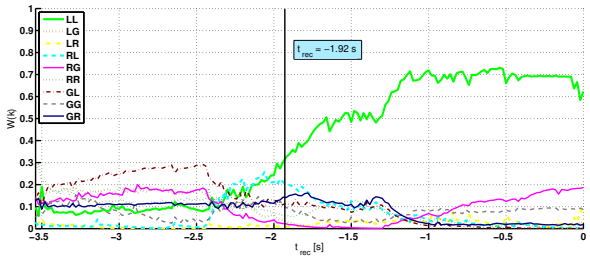


Fig. 5. Temporal behaviour of discriminant values of the LL situation class at intersection I_3. The time $t_{\mathrm{rec}} = -1.92$ s denotes the moment in which the situation is just recognised correctly.

For each vehicle, a possible set of future motion trajectories has been estimated and rated based on a trajectory database for a time interval of 2–4 seconds ahead. Realistic situations were generated by a pairwise combination of individual motion trajectories and classified according to nine categories with a polynomial classifier. In the proposed framework, situations are penalised for which the time to collision significantly exceeds the human reaction time. We have combined the correspondingly favoured situations using a probabilistic approach, resulting in a more reliable situation recognition and collision detection than obtained based on independent motion hypotheses. We have evaluated the method on a real-world differential GPS data set acquired during a test drive of 10 km, including three road intersections. We found that our method is typically able to recognise the situation correctly about 1–2 seconds before the distance between the vehicles becomes minimal. Situations involving more than two vehicles may be addressed by considering all possible pairs of vehicles or by a one-versus-all approach.

TABLE II

MOMENT OF CORRECT RECOGNITION t_{rec} FOR ALL SITUATION CLASSES.

Class	I_1 [s]	I_2 [s]	I_3 [s]	median [s]
LL	−2.16	−1.38	−1.92	−1.92
LG	−0.14	−1.36	−1.26	−1.26
LR	−0.47	−1.28	−1.54	−1.28
RL	−0.70	−1.60	−1.96	−1.60
RG	−0.72	−2.02	−0.38	−0.72
RR	−0.39	−2.00	−1.62	−1.62
GL	−0.52	−1.90	−2.32	−1.90
GG	−0.22	−2.26	−2.38	−2.26
GR	−0.44	−1.86	−1.70	−1.70

REFERENCES

[1] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell, and J. Weber, "Automatic symbolic traffic scene analysis using belief networks," in *Proc. Nat. Conf. Art. Intell.*, vol. 2, 1994, pp. 966–972.

[2] N. Oliver and A. P. Pentland, "Graphical models for driver behavior recognition in a smartcar," in *IEEE Proc. Intell. Veh. Symp.*, 2000, pp. 7–12.

[3] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden markov models for complex action recognition," in *IEEE Proc. Int. Conf. on Computer Vision and Pattern Recognition*, 1996.

[4] N. Oliver, B. Rosario, and A. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 831–843, 1999.

[5] K. Nickel and R. Stiefelhagen, "Real-time person tracking and pointing gesture recognition for human-robot interaction," in *Computer Vision in Human Computer Interaction, ECCV Workshop*, 2004, pp. 28–38.

[6] M. J. Black and A. D. Jepson, "A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions," in *Proc. Eur. Conf. Comp. Vis.* London, UK: Springer-Verlag, 1998, pp. 909–924.

[7] A. E. Broadhurst, S. Baker, and T. Kanade, "Monte carlo road safety reasoning," in *IEEE Proc. Intell. Veh. Symp.* IEEE, June 2005, pp. 319 – 324.

[8] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using monte carlo sampling," *IEEE Trans. Intell. Transport. Syst.*, vol. 9, no. 1, pp. 137–147, march 2008.

[9] M. Althoff, O. Stursberg, and M. Buss, "Stochastic reachable sets of interacting traffic participants," in *IEEE Proc. Intell. Veh. Symp.*, June 2008, pp. 1086–1092.

[10] T. Batz, K. Watson, and J. Beyerer, "Recognition of dangerous situations within a cooperative group of vehicles," in *IEEE Proc. Intell. Veh. Symp.*, June 2009, pp. 907–912.

[11] F. Large, D. A. Vasquez Govea, T. Fraichard, and C. Laugier, "Avoiding cars and pedestrians using velocity obstacles and motion prediction," in *IEEE Proc. Intell. Veh. Symp.*, 2004, pp. 375–379.

[12] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image and Vision Computing*, vol. 14, no. 8, pp. 609 – 615, 1996, 6th British Machine Vision Conference.

[13] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1450–1464, 2006.

[14] A. Croitoru, P. Agouris, and A. Stefanidis, "3d trajectory matching by pose normalization," in *Proc. of the 13th annual ACM intl. workshop on GIS*, 2005, pp. 153–162.

[15] C. Hermes, C. Wöhler, K. Schenk, and F. Kummert, "Long-term vehicle motion prediction," in *IEEE Proc. Intell. Veh. Symp.*, June 2009, pp. 652–657.

[16] M. Vlachos, G. Kollios, and D. Gunopulos, "Elastic translation invariant matching of trajectories," *Mach. Learn.*, vol. 58, no. 2-3, pp. 301–334, 2005.

[17] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Am. A*, vol. 4, pp. 629–642, 1987.

[18] H. Sidenbladh, M. J. Black, and L. Sigal, "Implicit probabilistic models of human motion for synthesis and tracking," in *Proc. Eur. Conf. Comp. Vis.* London, UK: Springer-Verlag, 2002, pp. 784–800.

[19] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge University Press, 1992, ch. 5.8 Chebyshev Approximation, pp. 190–194.

[20] J. Schürmann, *Pattern classification: a unified view of statistical and neural approaches*. NY, USA: John Wiley & Sons, Inc., 1996.

[21] P. N. Johnson-Laird, "Mental models," in *Foundations of Cognitive Science (Second Edition)*, M. I. Posner, Ed. Cambridge, MA: MIT Press, 1990, pp. 469–499.