in: Proc. 15th Int. IEEE Conf. on Intelligent Transportation Systems (ITSC 2012), Anchorage, pp. 1339-1344, IEEE 2012

Traffic Situation Assessment by Recognizing Interrelated Road Users

Matthias Platho, Horst-Michael Groß and Julian Eggert

Abstract— **With the trend to highly automated driving, future driver assistance systems are required to correctly assess even complex traffic situations and to predict their progress. As soon as other road users are present the number of possible situations becomes infinite, rendering their assessment based on learned situation types impossible. In this paper we propose to break the situation down into sets of interrelated entities by estimating for each road user the entities that affect its behavior most. The decomposition offers numerous advantages: Attention can be focused on relevant entities only and predictions can be performed with a smaller set of considered entities. As the high variability among situations requires a large amount of data for learning and testing, we implemented a simulation environment that gives access to the causes for the behavior of each road user. In a simulated intersection scenario we show that we can reliably infer the affecting entities for each road user only utilizing features that can be obtained by common sensors.**

I. INTRODUCTION

Correctly assessing the situation in which a car is currently involved in is an important requirement for advanced driver assistance systems. Especially when driving in an urban environment there is a wide variety of entities that need to be considered, like cars, traffic lights or pedestrians. For the interpretation of the current situation both the constellation and the dynamic state of these entities has to be taken into account. The high complexity of situations makes their assessment both challenging and computationally expensive which is exacerbated when also the prediction of future behavior of other road users is targeted.

In recent years there has been an increasing interest in the field of situation recognition and behavior prediction in traffic scenarios. A possible discerning property for the approaches is whether they aim at interpreting a situation as a whole or specialize on detecting a certain behavior. The goal to interpret a situation as a whole was pursued e.g. in [1] and [2]. In [1] case-based reasoning is employed in order to store a basic set of situations in memory and match new situations to it. The approach offers the ability to continuously extend the memory with encountered situations, but struggles with maintaining stability in the face of the unlimited number of possible situations. The authors of [2] focus on an intersection scenario, in which they use description logic for reasoning about the relations between cars. While they succeed to infer the relations and possible conflicts between road users, they report a high computation effort for the reasoning process.

A larger number of publications deal with approaches that specialize on detecting particular maneuvers or behaviors ([3], [4], [5], [6], [7], [8], [9]). In this field probabilistic modeling is widely used, ranging from particle filters to Object-Oriented Bayesian Networks. The approaches target either at predicting a single behavior ([3], [6], [9]), additionally recommending one [4] or recognizing one out of multiple maneuvers ([7], [5], [8]). In [3] an approach for recognizing and predicting situations involving two vehicles at an intersection is presented. Using particle filters possible conflicts can be predicted up to two seconds in advance, but the approach is limited to two vehicles only. In the work of [6] an upcoming overtake scenario is predicted up to one second in advance. This is accomplished by using Dynamic Bayesian Networks that are learned and tested in a simulator environment. In [4] the opposite goal is pursued: Using a Bayesian Network the current situation is assessed for judging the possibility to perform a lane change.

The authors of [7] use Hidden Markov Models for recognizing one of multiple maneuvers, namely "passing", "aborted passing" and "following". One of the drawbacks of their approach is its sensitivity to the way the situation is temporally segmented and normalized. Recognizing an even higher number of driving maneuvers is the goal of the work presented in [5]. Using Object-Oriented Bayesian Networks the relative movement and position between two vehicles is categorized into 27 maneuvers like "cut-in" or "cut-out". The approach is tailored to highway scenarios, which is also the focus of [8]. In that work behavior recognition is combined with trajectory prediction for driving modes like "free ride" or "following" and overtake maneuvers.

The cited works show that there are various approaches for situation assessment and behavior recognition that differ in their emphasis of either the descriptive or the predictive aspect in their method. The predictive methods assume that the relevant entities that need to be incorporated - e.g. cars possibly causing a conflict - are known or can be extracted easily from the context. This holds true if the scenario is artificially limited to a certain number of entities or if it is at least partially constrained, like in a highway scenario. However, for driving in inner-city the complexity of situations is dramatically increased. One common solution to over boarding complexity is to divide the problem into parts that can be managed more easily. A possible application to the problem of situation assessment is to group entities that depend on each other via a cause-effect relationship. For a given road user this would mean to determine all

M. Platho and H-M. Groß are with Department of Neuroinformatics, Technical University of Ilmenau, D-98684 Ilmenau, Germany {matthias.platho, hm.gross}@tu-ilmenau.de

J. Eggert is with Honda Research Institute Europe GmbH, D-63073 Offenbach am Main, Germany jeggert@honda-ri.de

the other entities that have a significant influence on its behavior. We propose to decompose complex situations into (overlapping) sets each consisting of a road user and the entities affecting it. The decomposition does not only help to deal with complex situations but can also be used as a preprocessing step for further computations: It allows for limiting the entities subject to prediction to the relevant ones and provides a way to guide attention.

In this paper we show the feasibility of our approach and how such interrelated sets can be specified using Bayesian Networks. In order to obtain the required data for a quantitative evaluation we made use of a traffic simulator, for which we developed an agent model that controls the vehicles. This step became necessary as in standard traffic simulators the information about which entity triggered a road user's behavior can hardly be extracted, while our implementation was designed explicitly for this purpose. In Section II we detail on the implementation of the agent model, while in Section III we explain our method for modeling sets of interrelated entities. In Section IV we present the results of our evaluation and in Section V we give an outlook on extensions and future work.

II. DRIVER AGENT DESIGN

In order to develop and evaluate methods for situation recognition in traffic scenes a significant amount of data is needed. While it is possible to record videos of busy streets and intersections and label them using tracking methods and manual refinement, this process is both time-consuming and cumbersome. Further drawbacks are the need for repeating this procedure whenever a different street topology should be tested and the missing guarantee to record exactly the situations one is interested in. As long as the focus of research is more in the direction of developing the basic methodology rather than an exact parameterization for a specific real-world application, the use of a traffic simulator lends itself for data generation.

In the field of traffic engineering traffic simulators are important tools for designing roadways in a safe and efficient manner. That is why there are a multitude of commercially available traffic simulators like VISSIM, PARAMICS and AIMSUN, that either work on a macroscopic or microscopic level. Macroscopic traffic simulators describe traffic in an averaged way using *flow* and *density*. Microscopic traffic simulators (MTS) in turn simulate each entity (e.g. cars, trucks, pedestrians) individually and thus provide data for individual road user behavior. Most of the commercially available MTS claim to model all of their simulated types of entities as sophisticated agents [10], and that these models have been successfully evaluated with real traffic. Unfortunately, in many cases these simulators provide only limited access to the entity level for extracting the causes for the current behavior of an agent. As this information is crucial for our approach, we have implemented our own agent model that allows for the desired introspection. The implementation uses standard models for vehicle behavior. In the following we will detail on the implementation, as it is the basis for all the data generated for learning and testing.

Vehicles - or more precisely: their drivers - are modeled as autonomous agents. This means that they actively perceive their environment, base their behavior planning on the information obtained and finally act according to their collected information. Also in situations where cooperation between vehicles is needed (e.g. at an unsignalized intersection) each agent acts independently without using a central coordinating instance.

The agent possesses the capabilities for avoiding collisions, keeping a safe distance (*car-following*) to the nearest leading vehicle in the same lane, adhering to traffic lights and crossing intersections while obeying right of way. In order to choose the appropriate behavior it senses obstacles, traffic lights, vehicles and intersections in a limited sensor range and determines its driving mode accordingly. If the agent detects no relevant entities in its range it accelerates to the permitted speed. It is possible that a vehicle encounters a situation where multiple behaviors could be applied but their required actions differ significantly, for example a distant traffic light turns red (low deceleration) while the leading vehicle stops abruptly (high deceleration). In such cases the agent acts conservatively choosing the behavior that requires the highest deceleration.

The car-following behavior is based on the linear model proposed by Helly [11], which has also been used in the MTS SITRA-B+ [12]. This behavior requires that the leading vehicle is less than the agent's specific reaction range for car-following ρ_f ahead. The mathematical formulation (as adapted from [13]) for each car is:

$$a(t) = C_1 \Delta v(t-T) + C_2 \Delta x(t-T) - D(t) \qquad (1)$$

$$D(t) = \alpha + \beta v(t-T) + \gamma a(t-T) \qquad (2)$$

where

$a(t)$	is the acceleration of the regarded vehicle at time t
$D(t)$	is the desired following distance to the nearest leading vehicle in front of it
v	is the speed of the vehicle
Δx	is the relative distance between the regarded vehicle and the leading vehicle
Δv	is the relative speed between the regarded vehicle and the leading vehicle
T	is the driver reaction time
$\alpha, \beta, \gamma, C_1, C_2$	are (vehicle-specific) calibration constants

The merits of this model are its reported good fit to observed data [13] and the direct real-world correspondence of its calibration constants. For example, the constant α corresponds to the desired minimum distance to the leading car and β is the factor for the desired velocity-dependent distance.

When approaching a red traffic light the agent performs a steady deceleration that brings the car to a stop near the traffic light's stopping line. For triggering this braking maneuver the traffic light has to be in the agent's perception range ρ_t. The strength of the deceleration is computed as follows:

$$a = -\frac{1}{2}\frac{v^2}{\Delta x} \text{ if } \Delta x < \rho_t \tag{3}$$

where

a	is the proposed deceleration
v	is the current speed of the vehicle
Δx	is the relative distance between the regarded vehicle and the nearest stopping line on its lane and direction

The behavior for crossing an intersection is the agent's most complex one. The vehicle's speed has to be lowered during approaching when the intersection is currently blocked or will be blocked shortly. The agent has to consider whether it is driving on a major or minor road and which lanes it will occupy during crossing. Based on this information a prediction of the behavior of other approaching vehicles has to be performed for identifying possible conflicts. The goal of all these considerations is to detect the next safe gap at which the intersection is free, allowing the vehicle to cross the intersection without interfering with other road users. The agent judges a gap as sufficient if the time for crossing plus a safety margin ϵ_s is smaller than the time the intersection is estimated to be vacant.

As mentioned above all agents are consistently implemented as individual, autonomous entities. Advantages of this approach are for example that it matches the way it works in the real world and that it enables a certain degree of emergence, such that the overall behavior of all vehicles differs from the sum of individual behaviors. The variability of the situations created by the agents is further increased by the fact that the parameters of the behaviors are varied between each agent.

The agents are deployed in a simulation environment that provides the roadways, switching of traffic lights and recording ability. During a simulation the dynamic states of all entities and the behavioral states of all agents are recorded. In addition, the information *why* an agent chose its behavior is also recorded. This ability was, as mentioned earlier, the main motivation for implementing an own agent model.

III. RECOGNITION OF RELATED ENTITIES

Our approach aims at decomposing a traffic situation into sets of related entities in order to obtain an understanding which road user is affected by what. This understanding can also be seen as a preprocessing step for attention control or prediction processes. When knowing that the leading vehicle's behavior is mainly determined by a car on a crossing major road then attention can be focused on this car. The same holds true for prediction processes: a prediction module can limit the considered entities to the relevant ones and thus save computational resources.

For specifying prototypical combinations of a reference entity and its affecting entities we use a representation that we term *configuration*. A configuration is defined by its participating entities along with the information how the entities have to be related to indicate the configuration's presence. The relations can specify the dynamic state of an individual entity (*unary* relations) or relative states between multiple entities (*k-ary* relations). Examples for unary relations are the state of a traffic light or the current acceleration of a car; examples for k-ary (e.g. 2-ary) relations are the relative distance or velocity between vehicles. When the relations of a reference entity and another entity match that of a certain configuration, the reference entity is said to be in that configuration. This is equivalent to the information that the reference entity is affected by the other entities. In this paper configurations consist of only a single affecting entity, but this is not a limitation of the method.

The configurations we are aiming to recognize are the ones leading to interesting behavior. We consider driving at or accelerating to a road's permitted speed as normal behavior, while considering significantly slowing down or stopping as interesting behavior. Deceleration is of special interest because it is both safety-relevant and can usually be explained from the situation (e.g. blocking obstacle, red traffic light, crossing tram).

The three configurations used in this paper were handcrafted according to these considerations and with regards to the entities simulated by the framework described in Section II. In order to limit the need for expert knowledge to a minimum, learning configurations from data, whether obtained from simulations or taken from the real world, is a topic of ongoing research.

An exemplary decomposition of a schematic situation is shown in Figure 1. It depicts a typical traffic situation at a signalized T-junction. The red car is about to turn in the upper left arm of the intersection, but has to yield to the approaching blue car and therefore stops. This case can be described by a configuration that is tagged "Stopped by intersection", where the red car is the affected entity (reference entity) and the blue car is the affecting entity. Not only road users participate in configurations, but also stationary elements like traffic lights or crosswalks. For example the white car has stopped because the traffic light on its lane is red, thus creating a "Stopped by Red Traffic Light"-configuration that uses a traffic light entity. Another important aspect is that a road user can be both affecting as well as affected entity at the same time (but in different instances of configurations). In the presented example the green vehicle has to slow down because the red car blocks

in: Proc. 15th Int. IEEE Conf. on Intelligent Transportation Systems (ITSC 2012), Anchorage, pp. 1339-1344, IEEE 2012

its path, making the red car thereby additionally to its role as reference entity in the intersection configuration an affecting entity in a “Stopped by Leading Vehicle”-configuration. It is also possible that a vehicle is the reference entity in two configurations at once: A vehicle waiting behind the white car would be blocked by both the traffic light and its leading car. Note that in the following we abbreviate “entity X is the reference entity in configuration Y” by “entity X is in configuration Y” for better readability.

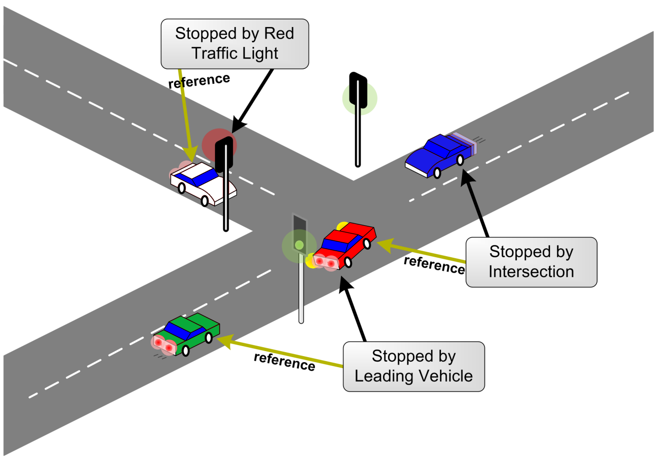

Fig. 1. A traffic situation with various present configurations.

A configuration can be most naturally represented by a graph, a simplified example is shown in Figure 2. It specifies a “Stopped by leading car“ configuration where two cars are driving one after another and the leading car’s velocity forces the following one to slow down or even stop. The root of a configuration yields its label; entities are represented as rectangular nodes while relations are denoted by rounded nodes. In the example the reference entity is the car that is forced to slow down as it is blocked by the leading car. The presence of the configuration is indicated by the fact that the following car is either stopped or decelerating, represented by the nodes *Velocity* and *Acceleration*, respectively. Further indicators are that the leading car is slower (*Relative Velocity*) and that it is within a certain range (*Distance*) to the reference car. Note that the actual values of the relations are not shown in the Figure for simplicity.

For learning and recognizing configurations we use Bayesian Networks [14]. A Bayesian Network is a directed acyclic graph that represents a set of random variables. Random variables are represented by nodes while edges denote probabilistic dependencies between them. We chose Bayesian Networks as it is a well-researched method for modeling probabilistic processes and allows for considering sensor noise and classifier confidence. Bayesian Networks furthermore provide an intuitive representation of probabilistic dependencies and have the ability to cope with incomplete evidence. Additionally, configurations as those shown in Figure 2 can be mapped to Bayesian Networks without significant changes.

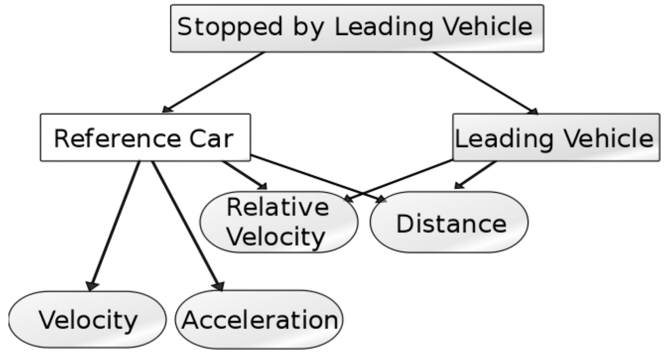

Fig. 2. A schematic representation of the configuration “Stopped by leading car”. It contains both unary relations (Velocity, Acceleration) as well as binary relations (Relative Velocity, Distance).

IV. EVALUATION

In order to verify the feasibility of our approach, we set up an intersection scenario in which we simulated traffic consisting of vehicles that were controlled by the agents described in Section II. The intersection consists of two crossing roads; one of them is a major road having two lanes in both directions (see Figure 3). The other one is a minor road with only one lane in each direction. The intersection is signalized by traffic lights. Cars approach the intersection from all incoming lanes, on average there are always about fifteen cars around the intersection.

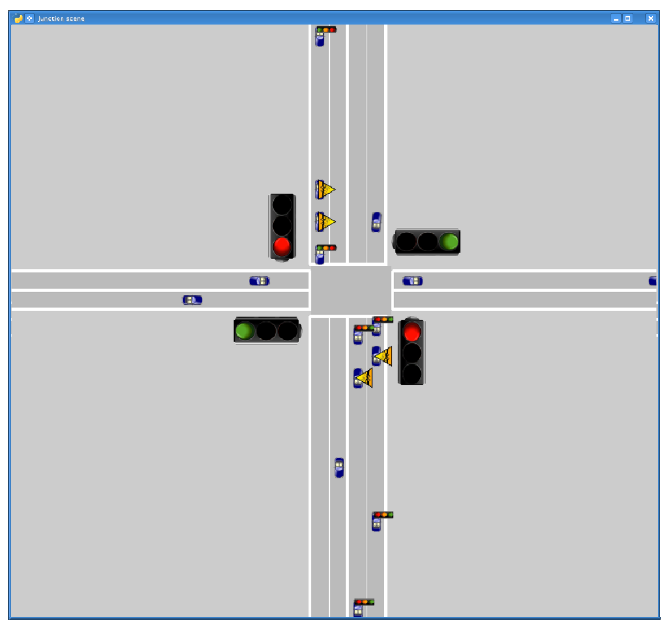

Fig. 3. Screenshot from the simulated intersection. The symbol above a car denotes the configuration it is currently in. Cars without symbols are accelerating or driving at constant speed.

The goal was to recognize for each car the configuration it is currently involved in. We used three basic configurations, namely “Stopped by red traffic light” (TL), “Stopped by leading car” (LC) and “Stopped by intersection” (IS). The last mentioned configuration is present when a car has to slow down or stop because it has to yield right of way to another road user. In case a car was currently in no configuration, this should be recognized as well.

Since the different configurations share several of their relation nodes, all three configurations were incorporated in a single Bayesian Network, that computes for a given car the probabilities for being in a configuration (see Figure 4). Assembling all configurations in a single network introduces additional conditional probabilities but in return it allows normalizing the beliefs in the configurations directly. Each configuration is modeled as Bayes classifier, where the class variable denotes the probability of being in that particular configuration. The feature variables are designed to match the configuration's relations. Each classifier possesses four feature nodes, two of which appear in all configurations, namely the *Velocity* and *Acceleration* of the reference car. These nodes are required by all configurations as they test whether a configuration is present at all, because it requires the reference car to be decelerating or stopped by definition (since we are only interested in these behaviors, see Section III). Continuous features were discretized into at most five intervals, after pre-tests had shown that the obtained speed-up in relation to continuous modeling outweighs the small loss in recognition performance.

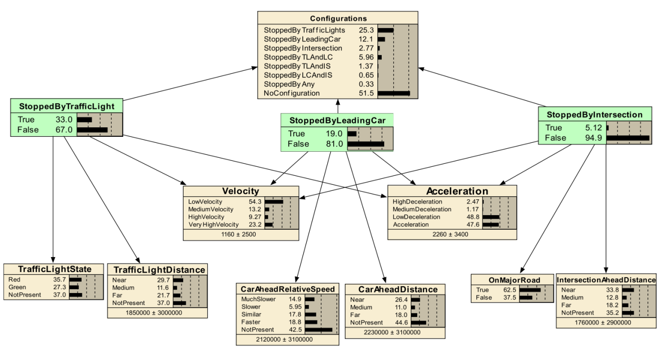

Fig. 4. The Bayesian Network that was used for the second mode of the evaluation. In the network for the first mode the "Configurations"-node has no states for combined configurations. The class nodes are colored green.

The evaluation was performed in two different modes. For both modes we tried to identify for each car present in the scene its current configuration. The difference between these two modes was the way how cases were treated in which a car was in multiple configurations at once. In the first mode we tried to identify for each car only the configuration affecting the agent's behavior, while in the second mode we tried to recognize all current configurations. The difference between these two modes becomes clear, when remembering that a vehicle can be in multiple configurations at once. While only the most influential affects the agent's behavior, the others can theoretically still be recognized from their relations. We observed that two combinations of configurations - although possible in theory - did not occur in the data generated. These combinations are the joint appearance of "Stopped by leading car" and "Stopped by Intersection" (LC+IS) and the combination of all configurations at once (TL+LC+IS).

TABLE I

CONFUSION MATRIX FOR RECOGNIZING THE MOST RELEVANT CONFIGURATION FOR EACH CAR.

TL	LC	IS	None	← Measured/Actual ↓
30530	792	17	241	TL
293	**25481**	1185	0	LC
0	30	**1184**	0	IS
0	355	138	**81784**	None

Nevertheless these combinations were kept as distractors for the recognition evaluation.

The data generation for the evaluation was accomplished using the agents within our simulation environment, as introduced in Section II. The simulations recorded the dynamic state, the agent's behavior and the entities affecting it every 0.1 seconds. The simulation was run for about 20 minutes. For each car at each recorded time instance a case was generated, containing the measurements for all of the mentioned features, yielding a total of 142030 cases. These cases were then randomly distributed in ten partitions of equal size in order to perform a 10-fold cross validation. In each fold, the conditional probabilities of the Bayesian Network were learned using Expectation Maximization [15] on the training data. The reported recognition rates are the averaged results on all ten partitions. The presented confusion matrices were obtained by summing the confusion matrices of all partitions.

In the first mode, where only the most relevant configuration had to be recognized, a recognition accuracy of 97.9% was achieved. The corresponding confusion matrix is given in Table I. The confusion matrix has high values on the main diagonal showing a high accuracy for recognizing each of the configurations.

In the second mode, where all current configurations of a car had to be recognized, still a recognition accuracy of 94.2% was achieved. Table II depicts the corresponding confusion matrix. It shows that in 17581 cases a vehicle was both affected by a traffic light and a leading car (TL+LC) at the same time, of which 14558 were recognized correctly, yielding an accuracy of about 83%. The combination TL+IS occurred in only 140 cases, but was nevertheless correctly recognized 95 times.

Errors in the recognition are mainly a result of the low number of features used, which complicates recognizing ambiguous situations. Such a situation arises for example if two cars approach an intersection and the leading one starts braking to give way to a car on a major road. Then it can be hard to discern whether the following car brakes because of leading ("Stopped by leading car") or crossing car ("Stopped by Intersection").

In both runs configurations were reliably detected. The results obtained confirm the feasibility of our approach; however, the effect of noisy measurements has still to be investigated. As recognizing a configuration is equivalent to determining the affecting entities of a road user, the results show that in the given scenario a complex traffic situation was successfully decomposed into its related entities. While

TABLE II

CONFUSION MATRIX FOR RECOGNIZING ALL CONFIGURATIONS FOR EACH CAR.

TL	LC	IS	TL+LC	TL+IS	LC+IS	TL+LC+IS	None	← Measured/Actual ↓
26444	0	17	171	2405	0	27	241	TL
0	**9867**	1065	5	0	738	0	19	LC
0	17	**1194**	0	0	19	0	0	IS
1465	0	0	**14558**	205	0	1353	0	TL+LC
45	0	0	0	**95**	0	0	0	TL+IS
0	0	0	0	0	**0**	0	0	LC+IS
0	0	0	0	0	0	**0**	0	TL+LC+IS
0	187	123	65	0	5	19	**81681**	None

this provides a basis for understanding a situation, we are going to detail in the next section how we plan to build on this knowledge.

V. CONCLUSIONS AND FUTURE WORK

In this paper an approach for assessing complex traffic situations in urban environments has been presented. We propose to decompose a situation into sets, each consisting of a road user and its affecting entities. These sets termed *configurations* can be modeled as graphs, specifying the participating entities, their states and their relations that have to be matched to indicate the presence of the configuration. The graphical specification of a configuration can be directly mapped to a Bayesian Network that we employ for learning and recognizing configurations in traffic scenes. In order to generate sufficient amounts of data for learning and testing that also includes the information about which road user is affected by what we have implemented a driver agent for use in a traffic simulator. In a simulated complex intersection scenario we are able to recognize the configurations a vehicle is involved in with an accuracy of up to 97.9%. Recognizing the configurations of road users provides an elementary understanding of a situation, which we are targeting to exploit in future works, e.g. for predicting the evolution of a situation.

We are aware that there are classifiers that are simpler or more powerful than Bayesian Networks, but one reason for using them in this evaluation is that they can be directly created from a configuration specification. Furthermore, their properties play an important role in future extensions of our approach. In order to be more robust against sensor noise, measurements will be modeled probabilistically, considering uncertainties and expected noise. Additionally, we investigate how the recognition of configurations can be performed in an incremental manner, instead of testing each road user against each entity, which scales quadratically. For this purpose, we target that our system selects measurements of features based on their expected information gain, which can be directly computed in a probabilistic model. It is investigated how this can be used to discard improbable configurations early in the recognition process and save a significant amount of measurements.

REFERENCES

[1] S. Vacek, T. Gindele, J. M. Zöllner, and R. Dillmann, "Situation classification for cognitive automobiles using case-based reasoning," in *Proc. IEEE Intelligent Vehicles Symp*, 2007, pp. 704–709.

[2] M. Hülsen, J. M. Zöllner, and C. Weiss, "Traffic intersection situation description ontology for advanced driver assistance," in *Proc. IEEE Intelligent Vehicles Symp. (IV)*, 2011, pp. 993–999.

[3] E. Käfer, C. Hermes, C. Wöhler, H. Ritter, and F. Kummert, "Recognition of situation classes at road intersections," in *Proc. IEEE Int Robotics and Automation (ICRA) Conference*, 2010, pp. 3960–3965.

[4] R. Schubert, K. Schulze, and G. Wanielik, "Situation assessment for automatic lane-change maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 607–616, 2010.

[5] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, and W. Rosenstiel, "Object-oriented bayesian networks for detection of lane change maneuvers," in *Proc. IEEE Intelligent Vehicles Symp. (IV)*, 2011, pp. 673–678.

[6] I. Dagli, M. Brost, and G. Breuel, "Action recognition and prediction for driver assistance systems using dynamic belief networks," in *Proceedings of the NODe 2002 agent-related conference on Agent technologies, infrastructures, tools, and applications for E-services*, 2003, pp. 179–194.

[7] D. Meyer-Delius, C. Plagemann, and W. Burgard, "Probabilistic situation recognition for vehicular traffic scenarios," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA '09*, 2009, pp. 459–464.

[8] T. Gindele, S. Brechtel, and R. Dillmann, "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments," in *Proc. 13th Int Intelligent Transportation Systems (ITSC) IEEE Conf*, 2010, pp. 1625–1631.

[9] T. Schamm and J. M. Zöllner, "A model-based approach to probabilistic situation assessment for driver assistance systems," in *Proc. 14th Int Intelligent Transportation Systems (ITSC) IEEE Conf*, 2011.

[10] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (2nd Edition)*, 3rd ed. Prentice Hall, Dec. 2003.

[11] W. Helly, "Simulation of bottlenecks in single-lane traffic flow," *Theory of Traffic Flow*, pp. 207–238, 1961.

[12] J. Gabard and L. Breheret, "The sitra-b+ microscopic traffic simulation model - examples of use and future developments," ONERA/CERT, Tech. Rep., 1999.

[13] S. Panwai and H. Dia, "Comparative evaluation of microscopic car-following behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 3, pp. 314–325, 2005.

[14] J. Pearl, "Bayesian networks: A model of self-activated memory for evidential reasoning," in *Proceedings of the 7th Conference of the Cognitive Science Society, Irvine*, Aug. 1985, pp. 329–334.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, vol. 39, no. 1, pp. 1–38, 1977.