

3D Perception and Planning for Self-Driving and Cooperative Automobiles

Christoph Stiller and Julius Ziegler

Abstract—This presentation focusses on key technologies for automobiles that perceive *a priori* unknown environment and automatically navigate through everyday traffic. Methods for 3D Machine perception based on lidar and video sensors are outlined. Beyond classical metrology, the recognition and basic understanding of situations must be accomplished for automated trajectory planning in urban traffic. We discuss how to represent and acquire metric, symbolic and conceptual knowledge from video and lidar data of a vehicle. A hardware and software architecture tailored to this knowledge structure for an autonomous vehicle is proposed. Emphasis is laid on methods for situation recognition employing geometrical and topological reasoning and Markov Logic Networks. A quality measure for trajectories is imposed that considers safety, efficiency, and comfort. We adopt a flat input parameterization to plan trajectories that optimize the imposed quality measure. Results from the autonomous vehicle AnnieWAY that recently won the Grand Cooperative Driving Challenge are shown in real world urban and platooning scenarios.

I. INTRODUCTION

Autonomous Vehicles that perceive their environment, communicate with each other, understand the current traffic situation and may by themselves or cooperatively with others plan and conduct appropriate driving trajectories are an intense field of international research. This contribution outlines the concept and architecture of the ‘Cognitive Automobile AnnieWAY’ that has successfully participated in international competitions such as the 2005 Grand and the 2007 Urban Challenge, and recently won the 2011 Grand Cooperative Driving Challenge [1], [2], [3], [4]. The vehicle constitutes an experimental basis for automated machine behaviour [5], [6]. Within a few years, large improvements in traffic safety is expected from such technologies [7].

A major goal of the scientific research is to advance knowledge acquisition and representation as a basis for automated decisions. As illustrated in Figure 1, driving - whether by a human or by a cognitive machine - involves knowledge representation in various forms. *Metric knowledge*, such as the lane geometry and the position or velocity of other traffic participants is required to keep the vehicle on the lane at a safe distance to others. *Symbolic knowledge*, e.g. classifying lanes as either ‘vehicle lane forward’, ‘vehicle lane rearward’, ‘bicycle lane’, ‘walkway’, etc. is needed to conform with basic rules. Finally, conceptual knowledge, e.g. specifying a relationship between other traffic participants allows to anticipate the expected evolution of the scene to drive foresightedly.

C. Stiller and J. Ziegler are with Institut für Mess- und Regelungstechnik, KIT - Karlsruher Institut für Technologie, 76131 Karlsruhe, Germany
stiller, ziegler@kit.edu

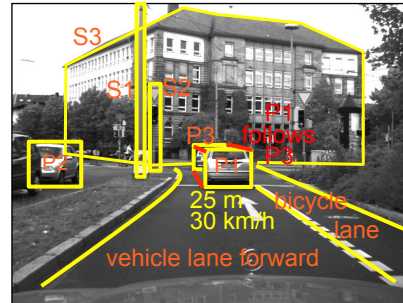


Fig. 1: Metric (yellow), symbolic (orange), and conceptual (red) knowledge for cognitive automobiles

II. ANNIEWAY SYSTEM OVERVIEW

A. AnnieWAY Hardware Architecture

Embodiment is widely considered a crucial element in cognitive systems research. To assess and validate theoretical findings we have adopted the unified hardware and software framework of the Karlsruhe-Munich *collaborate research center ‘cognitive automobiles’* [8], [9]. Based on the architecture depicted in Figure 2, meanwhile some ten experimental cognitive automobiles were set up [10], [6], [11]. To ensure real-

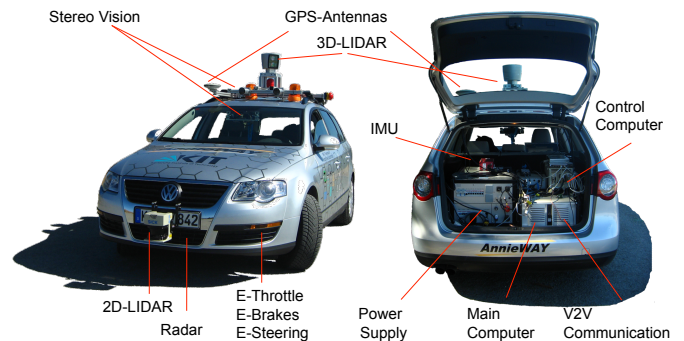


Fig. 2: Hardware setup for the cooperative cognitive automobile AnnieWAY.

time capabilities, vehicle control is performed on a dedicated dSpace AutoBox which directly communicates with the actuators over the vehicle CAN. All other perception and planning modules as well as sensor data acquisition are performed by a single multicore multiprocessor computer system which delivers sufficient computing power to host all processes providing low latencies and high bandwidth for inter-process communication.

B. AnnieWAY Software Architecture

The hardware is complemented with a real-time capable software architecture as depicted in Figure 3. The framework has been proposed, implemented, and made publicly available by [8], [12]. Its central element is a real-time database for information exchange. The various driving and perception tasks run in separate processes that communicate via the database and share a centralized view on all available information at every time. The framework supports parallel operation of processes at variable update rates and ascertains hard real-time performance where needed.

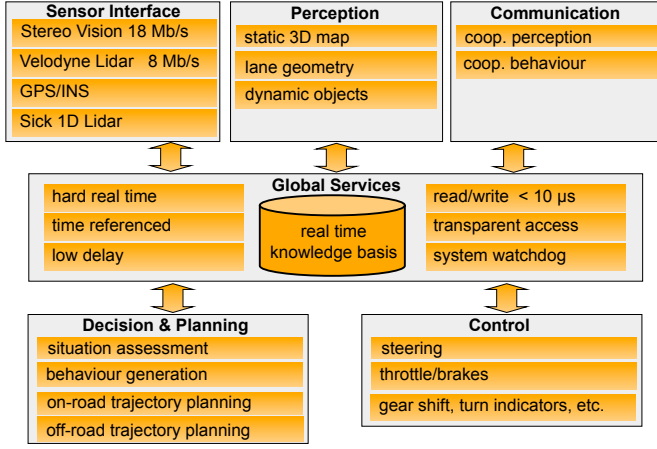


Fig. 3: Software setup for the cooperative cognitive automobile AnnieWAY.

III. SITUATION RECOGNITION

A. Simple geometric and topological reasoning

In this section, we will assume that a representation of the road network is available. This representation has to contain the geometry of single lanes as well as a topology, *i.e.* their interconnectedness within the network. Formally, this representation is a special *geometric graph*, *i.e.* a graph whose edges describe a distinctive road geometry, expressed by a planar curve. Such a representation was available during

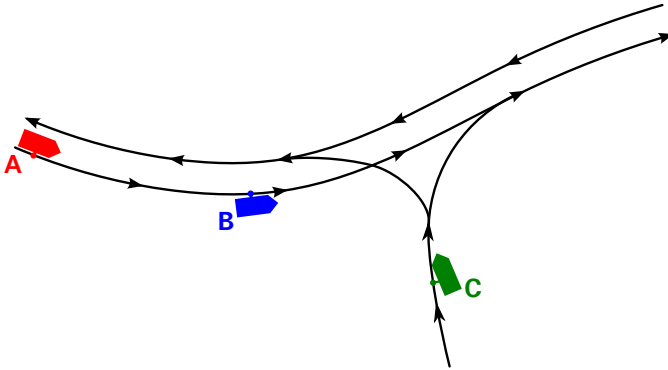


Fig. 4: Geometric graph for road representation and situation recognition.

the Urban Challenge in form of a so called *road network definition file* (RNDF). As has been shown in [13], such a representation can also be derived from vision cues using formal logic reasoning. Figure 4 shows an example for such a graph. The depicted situation is that of a one-way road forming a T-type-junction towards a road which allows two-way traffic.

Other road users are embedded into the graph using purely geometric reasoning. They are assigned to that edge in the graph which best explains their position and heading. A simple, orientation-aware point-to-curve distance function can be used for this task. Figure 4, depicts three vehicles and their association to edges in the graph.

The graph provides a rich description that readily allows to determine roles of and relations among other road users. From Figure 4, *e.g.*, the relations “A follows B” and “B must yield to C” can be derived *ad-hoc*.

B. Markov Logic Networks

Markov Logic Networks (MLNs) refer to a class of probabilistic logical models combining first-order predicate logics with Markov random fields [14]. An MLN is defined through a set of formulas $\{F_1, \dots, F_n\}$ in first-order predicate logics on a random field with random variables $\mathbf{X} = (X_1, \dots, X_q)$ and a set of scalar weights $\{w_1, \dots, w_n\}$ such that one weight is attributed to each formula.

The joint distribution of the random field is then defined by a Gibbs distribution

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \exp \left(\sum_{k=1}^n w_k F_k(\mathbf{x}) \right), \quad (1)$$

where $\mathbf{x} = (x_1, \dots, x_q)$ denotes a realization of the random field \mathbf{X} , and Z is a normalizing constant. The logical formulas F_k are instantiated by the realizations \mathbf{x} rendering each formula either *true* or *false*. Typically, each formula will depend on a small subset of variables in \mathbf{x} only that forms a clique of the Gibbs distribution.

Table I shows a simple example for an MLN with two generic formulas. The first formula is applied to each vehicle O_i while the second formula is applied to each pair of vehicle and lane (O_i, R_j) detected in the scene. For a specific scene

	w_i	F_i
1	1.4	$\forall o \text{ hasDirection}(o, \text{Same}) \Rightarrow \text{car}(o)$
2	0.6	$\forall o \forall r \text{ on}(o, r) \wedge \text{road}(r) \wedge \text{hasSpeed}(o, \text{Low}) \Rightarrow \text{car}(o)$

TABLE I: Formulas and weights specifying an MLN

with, *e.g.* two vehicles $\{O1, O2\}$ and one lane $\{R1\}$, one is left with the Markov random field shown in the graph of Fig. 5. This simple example supports the classification of cars through context information [15]. The formulas of an MLN can thus be considered as probabilistic rules with the weights quantifying our degree of belief in these rules. The Gibbs distribution (1) models world configurations as most probable the more they conform with rules that possess large weights.

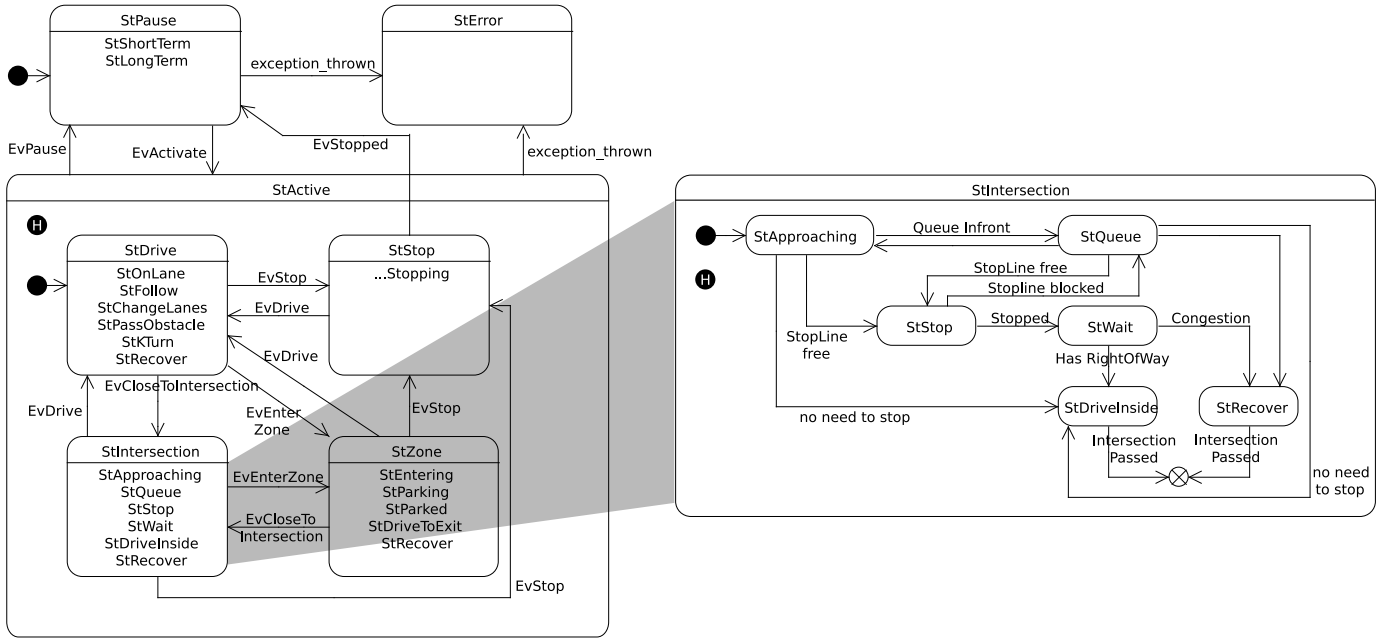


Fig. 6: AnnieWAY's hierarchical state automaton.

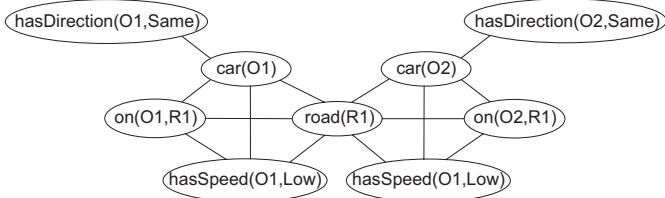


Fig. 5: Graphical representation of the Markov Logic Network defined through the generic formulas and weights from Table I and a scene with two vehicles $\{O1, O2\}$ and one lane $\{R1\}$.

IV. BEHAVIOUR GENERATION

Building on the information provided by the situation recognition module, the behavioural layer makes decisions on actions which need to be carried out in the current situation. Actions are communicated downstream to the trajectory generation stage in the form of center and boundary lines for the driving corridor, or as hard constraints which are imposed onto the generated trajectories (like forcing a stop at a stop line, or obeying a speed limit). Some simplistic tasks, like flashing an indicator, are passed on to the vehicle hardware directly. All these actions are generated using a state automaton which is organised in a hierarchical fashion. The possibility to describe state automata hierarchically has first been described by David Harel in [16] (*Harel state charts*). Figure 6 shows the state automaton which has been used on board ANNIEWAY during the Urban Challenge. Descriptors of states and events are prefixed by St...and Ev..., respectively. Substates are, for the most part, displayed in short form, *e.g.* the state StDrive contains sub states StOnLane, StFollow, StChangeLane *etc.* The principle of hierarchal organisation is illustrated by an exemplary “zoom” into the state StIntersection, which

shows the detailed structure of the relation of StIntersection's sub states. For a detailed treatment of state charts and their graphical notation *cf.* [16]. For a more detailed discussion of the specific use on board ANNIEWAY we refer interested readers to [17].

V. TRAJECTORY PLANNING

After the situation has been recognized and an appropriate behaviour has been identified a specific trajectory is planned. The planning concept described in the sequel belongs to the class of state lattice planers which has been adapted for on road driving in the presence of moving obstacles. A more complete description of the methodology can be found in [18].

A. Spatiotemporal state lattices

Static state lattices result from appropriate sampling of the continuous configuration space and are known as efficient representations for path planning in static environments [19], [20]. Spatiotemporal augment the configuration space of a standard state lattice with time into a single manifold, followed by discretization. To illustrate this concept, we will first consider the simplistic case of a one dimensional spatial configuration space.

Consider a vehicle traveling with varying velocity in \mathbb{R} . Its state is described by its distance from the origin, l and time t . In the spirit of the state lattice approach, we constrain the state space to an equidistantly sampled subset of \mathbb{R}^2 with sampling interval $\Delta l, \Delta t$.

Figure 7 depicts a spatiotemporal state lattice over the described workspace. The figure sketches state transitions for piecewise constant, positive velocities and C^2 continuous paths achieved by quintic polynomials, respectively. Quintic

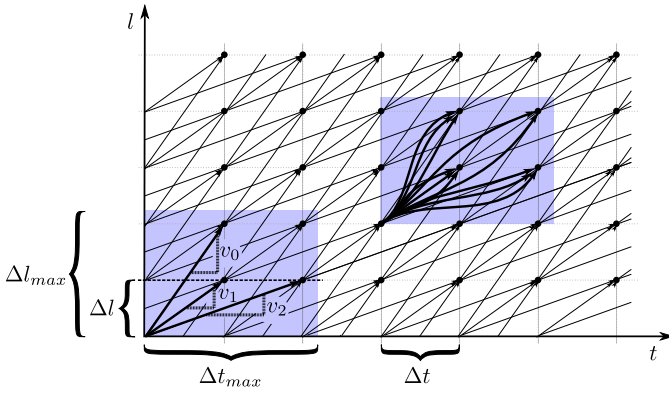


Fig. 7: A spatiotemporal state lattice over a one dimensional workspace. The lower left shaded area depicts a control set for paths in C^0 while the upper right one depicts one designed for higher order continuity, consisting of quintic polynomials.

polynomials are attractive for planning dynamic driving manoeuvres, because they minimize squared jerk [21] and allow for fast computation of their coefficients for given boundary conditions. Closed form expressions exist to describe the integral of squared jerk and for maximum speed, acceleration and speed along the trajectory [22]. Quintic splines have been used for automotive motion planning before [23], albeit only to describe kinematic paths without time parametrization.

B. Motion planning using spatiotemporal state lattices

In order to account for moving obstacles their future positions are predicted. Obstacles can then readily be transferred to the space-time manifold, as shown in Figure 8. The shaded area is occupied by a small object that moves with velocity $\frac{1}{2} \frac{\Delta l}{\Delta t}$. A trajectory is found within the spatiotemporal lattice that does not collide with the obstacle.

To deal with obstacles efficiently, we create a mapping between a discrete space-time obstacle map and the set of all edges in the graph. This can be done in the offline graph generation phase. Then, edges blocked by obstacles can be invalidated quickly by a single run over the obstacle map. This method scales well with the number of obstacles maintaining an almost constant overall processing time.

Edge costs consider the integral of the squared jerk of their geometric representations, as opposed to simply considering arc length. This improves safety, controllability and driving comfort.

Graph-based motion planning algorithms usually employ shortest path algorithms that maintain vertices visited in a partially ordered data structure. Algorithms belonging to this class include A* search, as well as Stentz' D* [24] and focused D* [25]. Spatiotemporal lattices belong to the class of directed acyclic graphs (DAG). Hence, sorting vertices by time yields a topological ordering in advance, and vertices can be just processed in this order. The resulting algorithm is linear in the number of vertices n , as opposed to Dijkstra's general scheme which is in $\mathcal{O}(n \log n)$.

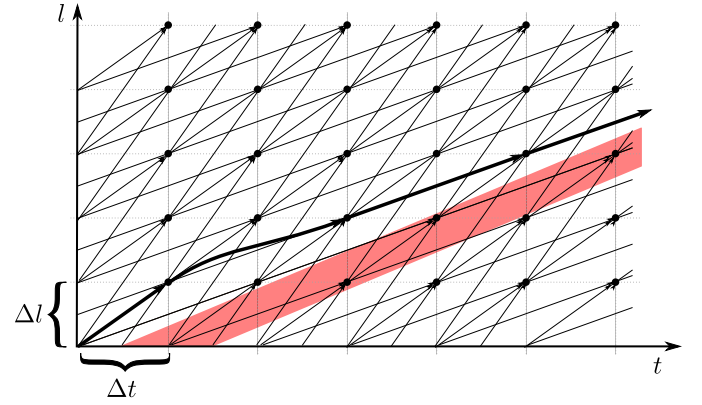


Fig. 8: Planning with a moving obstacle in the space-time manifold. The shaded area is covered by a moving object. A trajectory is shown that is composed of elements of the control set. Shortest paths can be found by relaxing vertices from left to right.

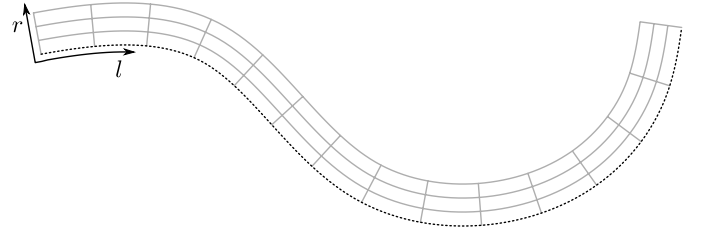


Fig. 9: Reparametrisation of the Cartesian plane. The dotted line indicates the original run of the road, (X, Y) . The grey structure illustrates the discrete reparametrization in l and r .

C. Lane-adapted reparametrization

The principle of spatiotemporal state lattices developed in the preceding sections generalizes naturally to two dimensions. Doing this naively, however, produces dimensionality problems due to the required dense sampling of the state space. Note that, in comparison with [20] the dimensionality of the sampling space for the state lattice rises from 3 (2D position and orientation, in [19], curvature is considered additionally) to 7 (2D position, 2D velocity, 2D acceleration and time), due to moving from a kinematic to a higher order dynamic model and the incorporation of time. With dimensionality rising, coverage of the configuration space requires an exponentially growing number of samples. Hence, an efficient way of sampling the configuration space is needed that is adapted to the special case of navigating on a road whose run is known *a priori*, e.g. from digital map data.

Given a continuous, piecewise twice differentiable, arc length s parametrized representation $(X(s), Y(s))$ of the course of the road, we define the following reparametrization (l, r) of the 2D workspace, where (x, y) denote Cartesian coordinates, $l(t)$ is the distance travelled along the road, and $r(t)$ is the lateral offset towards the road centre:

$$x(t) = X(l) - rY'(l) \quad (2)$$

$$y(t) = Y(l) + rX'(l). \quad (3)$$

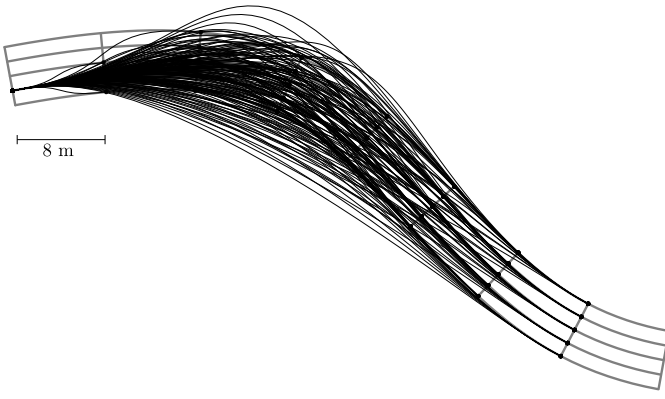


Fig. 10: State transitions on the transformed grid. The successors of one vertex are shown in black.

This is a base change towards a local orthogonal coordinate system that has its abscissa aligned with the road for any l . It defines a two dimensional manifold as depicted in Figure 9. As described earlier, differential boundary conditions of up to second order are required for edge generation. We therefore need to transform them through equations (2) and (3): Given \dot{l} , \dot{r} , \ddot{l} and \ddot{r} , by application of the chain rule we obtain

$$\dot{x} = \dot{l}X'(l) - \dot{r}Y(l) - r\dot{l}Y'(l) \quad (4)$$

$$\dot{y} = \dot{l}Y'(l) + \dot{r}X(l) + r\dot{l}X'(l) \quad (5)$$

and

$$\ddot{x} = \ddot{l}X' + \ddot{l}^2X'' - \ddot{r}Y - (2\dot{r}\dot{l} + r\ddot{l})Y' - \dot{r}\dot{l}^2Y'' \quad (6)$$

$$\ddot{y} = \ddot{l}Y' + \ddot{l}^2Y'' - \ddot{r}X - (2\dot{r}\dot{l} + r\ddot{l})X' - \dot{r}\dot{l}^2X'' \quad (7)$$

We now restrict parameters $l, r, \dot{l}, \dot{r}, \ddot{l}$ and \ddot{r} to a discrete, grid like set (the vertices of the search graph) and transform them through equations (2) - (7). The resulting $x, y, \dot{x}, \dot{y}, \ddot{x}$ and \ddot{y} , together with discrete values for time t , are used as boundary values to calculate quintic polynomial trajectories as described in section V-A. To assert dynamic and kinematic feasibility, a respective edge is only added to the graph, if velocity, acceleration and jerk stay within bounds defined in advance. In the effort to further reduce the number of vertices, some ad hoc reductions can be applied to the sets of discrete parameters: r is constrained to an interval so as to restrict all vertices of the lattice to be within the bounds of the road. We set $\dot{r} = 0$ and constrain \dot{l} to be positive, since we wish the vehicle to make progress along the road, while crosswise motion is to be avoided. Second derivatives \ddot{l} and \ddot{r} of the untransformed coordinates are set to zero at the grid points.

Figure 10 gives an impression of the graph we used for our experiments by displaying successor edges of a single vertex. The outdegree of vertices is approximately 200.

VI. EXPERIMENTAL RESULTS

Figure 11 shows an exemplary result of the proposed trajectory planning method. The scenario selected for this example is that of merging into running traffic at a T-junction. As can be seen, the proposed method yields a trajectory that is

smooth in the sense of minimum mean squared jerk and safe in the sense of entering the gap at a safe distance to all other vehicles with a velocity of the gap itself. The proposed method inherently selects the optimum gap to cut in. The planner is also able to find trajectories under more complex traffic conditions.

Figure 12 shows AnnieWAY performing in the Urban Challenge in 2007, where AnnieWAY was one of only two non-US Vehicles that successfully entered the finals and belonged to the the few vehicles that drove collision-free through a mock up suburban environment. The graph based representation of the road network can be seen. In the top right of each frame, the sequence of states the hierarchical state automaton traverses is displayed.

The *Grand Cooperative Driving Challenge 2011* (GCDC) was the first international competition to implement cooperative driving in a realistic, heterogenous scenario. It was organized by the *Netherlands Organisation for Applied Scientific Research* (TNO) on a highway near Helmond. Vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication in the ITS band at 5.8 GHz following the IEEE 802.11p standard built the basis for cooperation between vehicles [26]. The information broadcasted by each vehicle included vehicle length and width, latitude and longitude of position, heading and yaw rate, and velocity and acceleration. The main challenge for participating teams was to develop longitudinal control strategies that allowed autonomous and cooperative vehicle platooning manoeuvres without knowing the algorithms and technical equipment of other vehicles in the platoon. Control strategies had to cope with standard manoeuvres, like platoon merging as well as with unexpected behavior of other vehicles, such as, e.g., varying data quality, or sudden failure of communication. Vehicles were repeatedly and randomly teamed up for heats in two platoons. Both platoons were led by the same vehicle that induced challenging braking and acceleration manoeuvres to the competitors. Figure 13 shows one of these heats of the GCDC, illustrating the large variety of vehicles and technical solutions in the competition. Our vehicle AnnieWAY is the silver vehicle directly in front of the truck. AnnieWAY won the *Grand Cooperative Driving Challenge 2011* tightly followed by the team from Halmstedt. In contrast to previous challenges, the GCDC not only assessed individual driving of each vehicle but also considered its impact on other traffic participants, i.e. the criteria included platoon velocity, length, and damping of acceleration/deceleration cycles [27]. For more details we refer interested readers to the IEEE Trans. ITS special issue on the GCDC including team AnnieWAYs contribution [4].

VII. CONCLUSIONS

AnnieWAY is an experimental autonomous vehicle that perceives *a priori* unknown environments, recognizes situations, plans appropriate trajectories, and controls its actuators to follow these. It acquires metric, symbolic and conceptual

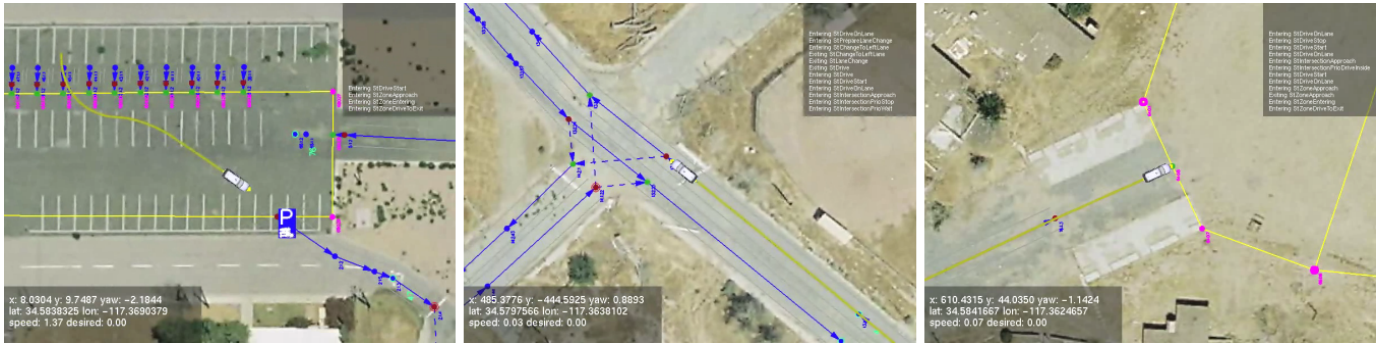


Fig. 12: AnnieWAY in the Urban Challenge of 2007

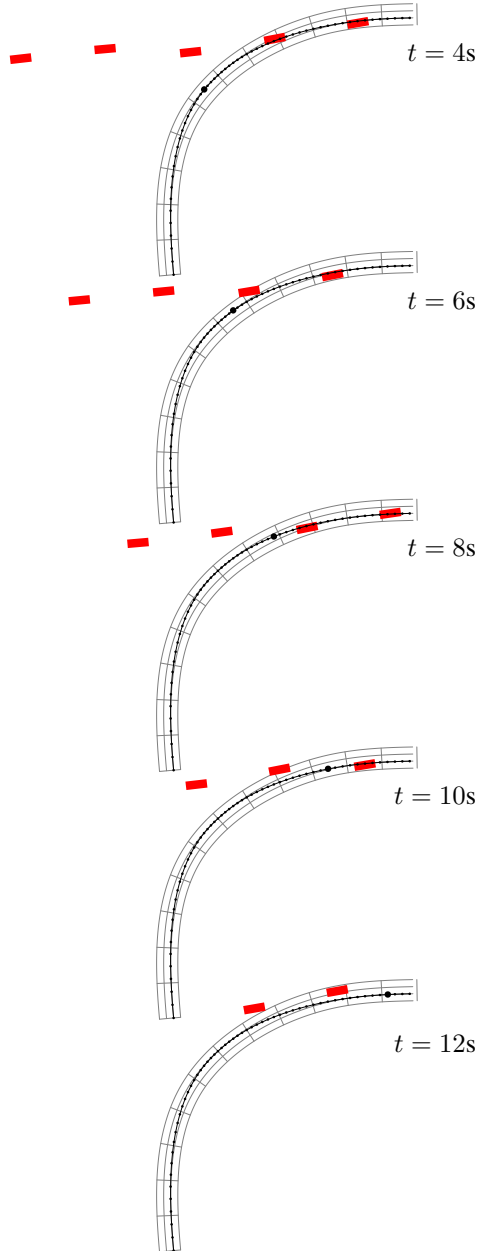


Fig. 11: Merging into traffic. The planner selects a comfortable and safe trajectory into a convoy of five vehicles traveling towards a T-junction.



Fig. 13: One heat of the GCD with six competing vehicles.

knowledge from video and lidar data and employs prior knowledge from maps. The current driving situation is recognized employing geometrical and topological reasoning. Furthermore, we experiment with Markov Logic Networks for automated inference of pair relations among objects. Spatiotemporal state lattices have been shown to allow for efficient trajectory planning without severe restrictions on optimality and generality. Moving vehicles are accounted for by disabling spatiotemporal nodes along their paths. Thus the method maintains its computational load irrespective of scene complexity. The quality criterion optimized during planning can be chosen with large freedom. Currently, we minimize mean squared jerk, but extended measures could account for safety, energy efficiency, or phsyckinetic comfort in the future. Results are shown from the autonomous vehicle AnnieWAY that is able to autonomously and cooperatively travel in urban and platooning scenarios.

REFERENCES

- [1] Ü. Özgüner, C. Stiller, and K. Redmill, "Systems for safety and autonomous behavior in cars: The DARPA Grand Challenge experience," *IEEE Proceedings*, vol. 95, no. 2, pp. 1–16, Feb. 2007.

- [2] Sören Kammel, Julius Ziegler, Benjamin Pitzer, Moritz Werling, Tobias Gindele, Daniel Jagzent, Joachim Schröder, Michael Thuy, Matthias Goebel, Felix von Hundelshausen, Oliver Pink, Christian Frese, and Christoph Stiller, "Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 615 – 639, Sept. 2008.
- [3] F. v. Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche, "Driving with tentacles: Integral structures for sensing and motion," *Journal of Field Robotics*, vol. 25, no. 9, pp. 640 – 673, Sept. 2008.
- [4] Andreas Geiger, Martin Lauer, Frank Moosmann, Benjamin Ranft, Holger Rapp, Christoph Stiller, and Julius Ziegler, "Team annieway's entry to the grand cooperative driving challenge 2011," *IEEE Transactions on Intelligent Transportation Systems (in print)*, 2012.
- [5] C. Stiller, G. Färber, and S. Kammel, "Cooperative cognitive automobiles," in *Proc. IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, June 2007, pp. 215–220.
- [6] M. Thuy, M. Althoff, M. Buss, K. Diepold, J. Eberspächer, G. Färber, M. Goebel, B. HeiBing, S. Kraus, R. Nagel, Y. Nao, F. Obermeier, F. Puente León, F. Rattei, C. Wang, M. Schweitzer, and H.J. Wünsche, "Kognitive Automobile - Neue Konzepte und Ideen des Sonderforschungsbereiches/TR-28," in *3. Tagung Aktive Sicherheit durch Fahrerassistenz*, Garching bei München, 2008.
- [7] European Union, "E-Safety," http://www.ec.europa.eu/information_society/activities/esafety, November 2010.
- [8] Matthias Goebel and Georg Färber, "A real-time-capable hard- and software architecture for joint image and knowledge processing in cognitive automobiles," in *Proc. IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, June 2007, pp. 734–739.
- [9] Moritz Werling, Matthias Goebel, Oliver Pink, and Christoph Stiller, "A hardware and software framework for cognitive automobiles," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2008*, Eindhoven, Niederlande, 2008, pp. 1080 – 1085.
- [10] M. Goebel, M. Althoff, M. Buss, G. Färber, F. Hecker, B. HeiBing, S. Kraus, R. Nagel, F. Puente León, F. Rattei, M. Russ, M. Schweitzer, M. Thuy, C. Wang, and H.J. Wuensche, "Design and Capabilities of the Munich Cognitive Automobile," in *Proc. IEEE Intelligent Vehicles Symposium*, Eindhoven, the Netherlands, June 2008, pp. 1101–1107.
- [11] J. Schröder, T. Gindele, D. Jagszent, and R. Dillmann, "Path planning for cognitive vehicles," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Eindhoven, Holland, 2008, pp. 1119–1124.
- [12] Matthias Goebel, *Eine realzeitfähige Architektur zur Integration kognitiver Funktionen*, Dissertation, Technische Universität München, München, 2009.
- [13] Britta Hummel, Werner Thiemann, and Irina Lulcheva, "Description logic for vision-based intersection understanding," in *Proc. Cognitive Systems with Interactive Sensors (COGIS)*, 2007.
- [14] Matthew Richardson and Pedro Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.
- [15] C. Stiller, S.Kammel, I. Lulcheva, and J.Ziegler, "Probabilistic methods for environment perception of cognitive automobiles," *at - Automation*, vol. 11, pp. 568–574, 2008.
- [16] David Harel, "Statecharts: A visual formalism for complex systems," *Science of Computer Programming*, vol. 8, no. 3, pp. 231–274, June 1987.
- [17] Tobias Gindele, Daniel Jagzent, Ben Pitzer, and Rüdiger Dillmann, "Design of the planner of team annieway's autonomous vehicle used in the darpa urban challenge 2007," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Eindhoven, Die Niederlande, 2008.
- [18] Julius Ziegler and Christoph Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *International Conference on Intelligent Robots and Systems*, 2009.
- [19] M. Pitvorai and A. Kelly, "Efficient constrained path planning via search in state lattices," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, 2005.
- [20] Maxim Likhachev and Dave Ferguson, "Planning long dynamically-feasible maneuvers for autonomous vehicles," in *Proceedings of Robotics: Science and Systems IV*, 2008.
- [21] A. Takahashi, T. Hongo, Y. Ninomiya, and G. Sugimoto, "Local path planning and motion control for AGV in positioning," in *International Workshop on Intelligent Robots and Systems*, 1989.
- [22] R.L. Andersson, "Aggressive trajectory generator for a robot ping-pong player," *Control Systems Magazine*, vol. 9, pp. 15–21, 1989.
- [23] A. Piazzini, C.G. Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic G^2 -splines for the iterative steering of vision-based autonomous vehicles," *Transactions on Intelligent Transportation Systems*, vol. 3, pp. 27–36, 2002.
- [24] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *International Conference on Robotics and Automation*, 1994.
- [25] A. Stentz, "The focussed D* algorithm for real-time replanning," in *International Joint Conference on Artificial Intelligence*, 1995.
- [26] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments," in *IEEE Vehicular Technology Conference Spring 2008*, 2008, pp. 2036–2040.
- [27] Martin Lauer and Anton Gerrits, "Next steps for the grand cooperative driving challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 1, no. 4, pp. 24–32, 2010.