DARPA Urban Challenge

Team AnnieWAY

Technical System Description

Soeren Kammel[1], Benjamin Pitzer[2], Stefan Vacek[3], Joachim Schroeder[4], Christian Frese[5], Moritz Werling[6], Matthias Goebl[7]

06/01/07

[1] kammel@mrt.uka.de
[2] pitzer@mrt.uka.de
[3] vacek@ira.uka.de
[4] joachim.schroeder@ira.uka.de
[5] frese@ies.uni-karlsruhe.de
[6] moritz.werling@iai.fzk.de
[7] goebl@lpr.e-technik.tu-muenchen.de

Urban Challenge
Technical System Description – Team AnnieWAY

The members of Team AnnieWAY have developed an autonomous vehicle capable of driving through urban scenarios. The vehicle of Team AnnieWAY is based on a VW Passat that has a drive-by-wire system, sensors like stereo vision and lidar, and a two level computing system: A main computer for sensor signal processing, situation assessment and behavior generation and a second computer dedicated to low-level control. A human-like understanding of the surrounding traffic scene is a key element to fulfill the requirements of the Urban Challenge. Therefore, the sensor data is transformed into an abstract and holistic scene representation. Probabilistic reasoning algorithms are applied subsequently which yield the recommended behavior for the given situation and traffic regulations.

1. Introduction

Autonomous driving requires substantiated knowledge in different domains like engineering, computer or cognitive sciences: A robust control that keeps the vehicle in a stable condition under all circumstances is mandatory. Onboard sensor technology must be able to consistently perceive the environment. Faulty hardware must be detected and omnipresent measurement uncertainties as well as contradictory sensor data have to be handled correctly. Finally, the car has to conclude which behavior is appropriate in a given traffic situation and has to execute this behavior strategy in real-time. The DARPA Urban Challenge connects interdisciplinary areas of research with relevance to science, industry and community that may hardly be overestimated.

1.2 Team Composition and Background

On January 1st 2006, the German Research Foundation (DFG) started a Collaborative Research Centre on Cognitive Automobiles. The Collaborative Research Centre is intended to last for twelve years. The University of Karlsruhe, the TU Munich, the Fraunhofer Gesellschaft (IITB in Karlsruhe) as well as the Universität der Bundeswehr Munich are working together in this Collaborative Research Center.

The project’s overall theme is the systematic and interdisciplinary research of machine cognition in the context of mobile systems as the basis of machine action and the development of a scientific theory of machine cognition. Its verifiability will exemplarily be proven in the way that the behavior of automobiles in traffic will be perceived, interpreted and even automatically generated. A cognitive automobile will be capable of individual as well as cooperative perception and interaction. A theory of machine cognition will make it feasible to propagate measurement uncertainty and symbolic vagueness throughout the complete cognition chain to give measures of certainty in order to quantify the trust in a specific generated behavior. Cognition includes perception, deduction and recognition and thereby supplies automobiles with completely novel abilities. Cognitive automobiles are able to sense themselves and their surroundings, as well as accumulate and organize knowledge.

Team AnnieWAY is a spin-off of the Collaborative Research Center on Cognitive Automobiles. Current team members are professionals in the fields of image processing, 3d perception, knowledge representation, reasoning, real time system design, driver assistance systems and autonomous driving. Some of the team members were part of the "Desert Buckeyes" and developed the 3d vision system for the Intelligent Offroad Navigator (ION) that traveled successfully 29 miles through the desert during the Grand Challenge 05.

2. System Design

2.1 Hardware Architecture

AnnieWAY is a modified 2006 VW Passat Variant B4 (Figure 1). The Passat is selected for its ability to be easily updated for drive-by-wire use by the manufacturer. The 1,731 liters luggage compartment offers enough space to host all electronic systems. The vehicle with all modification is 78 inches 1.991m) wide, 194 inches (4.924m) long, and 76 inches (1.935m) tall. The weight is about 4,000 lbs (1,800 kg).

Figure 1: AnnieWAY VW Passat Variant B4.

Figure 2: AnnieWAY driving autonomously.

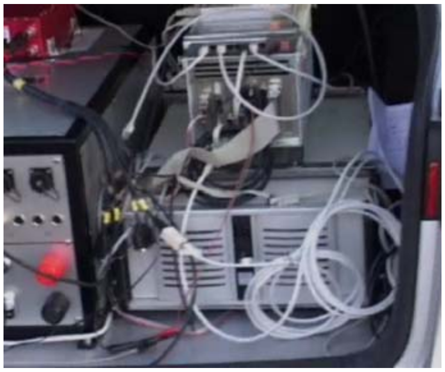

Figure 3: AnnieWAY's computers.

AnnieWAY's brain (Figure 3 bottom) is an AMD dual-core Opteron multiprocessor PC system which features excellent processing capacity to power consumption ratio. It delivers a computing power comparable to a small cluster, yet offers low latencies and high bandwidth for module interprocess communication. All sensors connect directly to the main computer that offers enough processing capacity to run almost all software components. Only for low-level control algorithms we dedicate a dSpace AutoBox (s. Figure 3 top, Figure 4) to drive the vehicles actuators in order to meet our safety requirements. Both computer systems communicate over a 1 Gbit/s Ethernet network. The drive by wire system as well as the car odometry is interfaced via Controller Area Network (CAN) bus. The (D)GPS/INS system (Figure 4) which is used for precise localization connects directly to the low-level control computer (AutoBox). The chosen hardware architecture is supported by a real-time-capable software architecture which will be described in chapter 2.2.

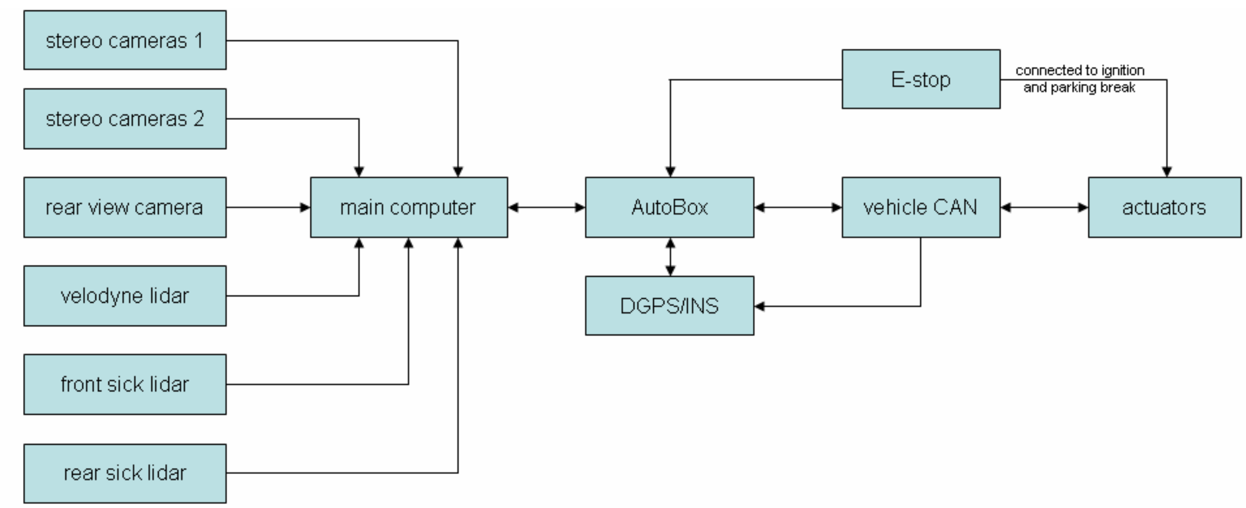

Figure 4: Overview of the hardware architecture**.**

2.1.1 Vision

Inspired by the human visual system, an active vision system usually consists of two or more cameras that can adjust gaze direction to currently important areas of the scene. Clearly, such a system can be useful in many traffic situations: E.g. when approaching a street crossing, active camera heads can extend the field of view to almost 180˚. Other capabilities of active vision systems are e.g. smooth following of objects or saccadic vision. In addition to the active system, the car of Team AnnieWAY has a second fixed stereo system to obtain valid depth information during saccades of the active system. Stereo rigs are sensors that on one hand provide rich texture information from vision that is crucial to many object classification tasks. On the other hand, they offer instantaneous depth measurements with good resolution in the near field which is imperative when reliable and immediate decisions are required. Figure 5 shows an example for a fast stereo-based ground (left) and obstacle (right) detection and classification.

Stereo cameras require a precise calibration which is usually done offline. Stereo vision is then performed using the pre-computed rotation axes and the commanded motor angles. However,

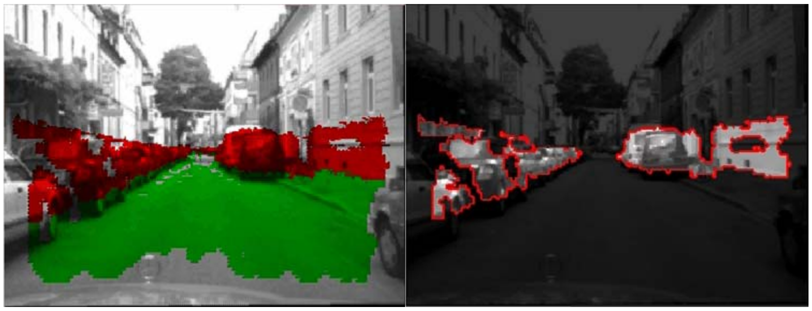

Figure 5: Stereo-based groundplane and obstacle detection.

such an approach entails very high demands on the mechanical setup to guarantee stability of the stereo calibration over longer time periods. Therefore, we use a stereo self-calibration algorithm that determines all extrinsic camera parameters of the active stereo rig up to scale. The algorithm does neither require precise offline calibration of the rotation axes nor a highly accurate mechanical setup. The self-calibration is implemented as a recursive algorithm, namely a robust Iterated Extended Kalman Filter (IEKF), to allow a continuous update of the camera parameters without having to store all previous images. Our approach combines the epipolar constraint on *spatial* correspondences between the left and right image and bundle adjustment over time that requires *spatio-temporal* tracking of an object point over several frames.
Another important task of the vision system is the detection of lane markers (s. Figure 6 for an example). Detected lane markers are also tracked with an EKF. To ensure that the reasoning system can be provided with this important information, a rear view camera supports the front cameras in case the vehicle is driving towards a bright light source like the sun.

Figure 6: Lane marker detection.

2.1.2 Long Distance Lidar

Since Lidar units produce their own light, low light conditions have no effect on this kind of sensor. The Velodyne HDL-64E rotating laser scanner (www.velodyne.com/lidar/index.html) is one of the latest scanning lidars. It is used in our vehicle to construct— together with the stereo cameras —a three-dimensional model of the environment. The lidar uses two groups of eight fixed lasers to cover a 26.5-degree vertical field of view and two 32-pixel Avalanche photodiode arrays for receiving reflected laser light. The lasers and diodes are mounted on a spinning platform that rotates at a rate of 600 rpm.

The HDL-64E provides a 360 degree field of view around the vehicle producing more than 1 million points per second at an angular resolution of 0.09 degrees horizontally and a distance resolution of 5cm with distances up to 70m. The result is a dense, highly accurate range scan representation of almost the entire scene surrounding the vehicle.

2.1.3 (D)GPS/INS

For precise localization we use the OXTS RT3003 Inertial and GPS Navigation System which is an advanced six-axis inertial navigation system that incorporates a Novatel L1/L2 RTK GPS receiver for position and a second GPS Receiver for accurate Heading measurements. Odometry is taken directly from the AnnieWAY's CAN bus. The RT3003 delivers better than 0.02m positioning under dynamic conditions using differential corrections and 0.1° heading using a 2m separation between the GPS antennas.

The RT3003 Inertial and GPS Navigation System includes three angular rate sensors (gyros), three servo-grade accelerometers, the GPS receiver and all the required processing in one very compact box. The RT3003 works as a standalone, autonomous unit and requires no user input before it starts operating.

2.1.4 Parking Lidars

Two additional Sick LMS 291 lidar units are mounted horizontally on the front and rear bumper to support the HDL-64E sensor and stereo cameras during parking maneuvers (they were added after the video).

2.1.5 Actuators

To actually control the car actuator for steering, brake, throttle and gear shifting are used. The steering actuator consists of a small electric motor attached to the steering column while the brakes are controlled by an additional brake booster.

2.1.6 E-Stop System

The E-Stop system consists of two 3 components: a wireless transmitter with the buttons to switch between run/pause mode and activate the emergency stop, a receiver in the car and an interface box which connects the receiver to the vehicle. Wireless transmitter and receiver are provided by the team during the site visit and are provided by DARPA during NQE and the final. The interface box allows to exchange the team/DARPA provided components easily. It is directly connected to the ignition and the parking break to perform an emergency stop without computer interaction. To switch between run and pause mode it also has a connection to the low-level control computer. The team provided transmitter and receiver are Satelline-3ASd EPIC DB data modems.

2.2 Software Architecture

The core components of the vehicle are the perception of the environment, an interpretation of the situation in order to select the appropriate behavior, a behavior network providing constraints for path planning and a component for control of the vehicle.

Figure 7 depicts a block diagram of the information flow in our autonomous automobile. Spatial information from the lidar and the stereo cameras are combined to a static 3D map of the environment. Moving objects have to be detected and treated differently. Dynamic objects are also include traffic participants that are able to move but have zero velocity at the moment. To detect *moving* objects, an optical flow approach employed. The results of this approach are associated with the tracked and clustered objects detected by the spatial sensors. To detect *possibly moving* objects (which are standing still right now), a simple form of probabilistic

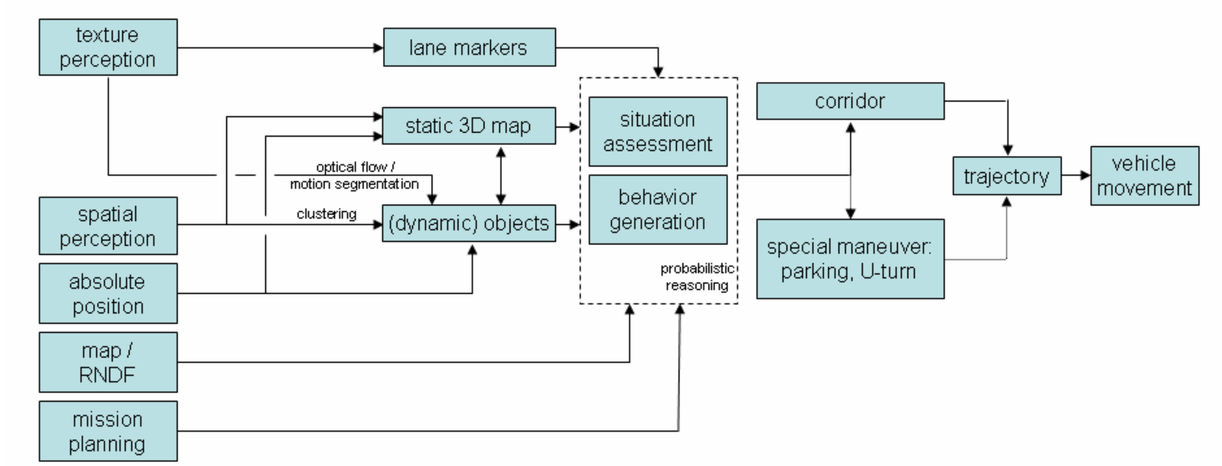

Figure 7: Flow of information in the autonomous car. The behavior generation module creates a driving corridor *or* signals a special maneuver to the trajectory planning system.

reasoning is used: if an object has the size of a car, the additional characteristics of a shadow beneath and a certain amount of symmetry and is located on a detected lane, it is considered to be probably moving. Lane markers are detected by the vision system. Together with the map/RNDF, the absolute position obtained from the (D)GPS/INS system and the mission plan, this information serves as input for the situation assessment and the subsequent behavior generation. Most of the time, the behavior will result in a driveable corridor. If a road is blocked or the car has to be parked, special maneuvers are signaled to the trajectory planning module.

2.2.1 Environmental Mapping


Figure 8: Areal image of environment.

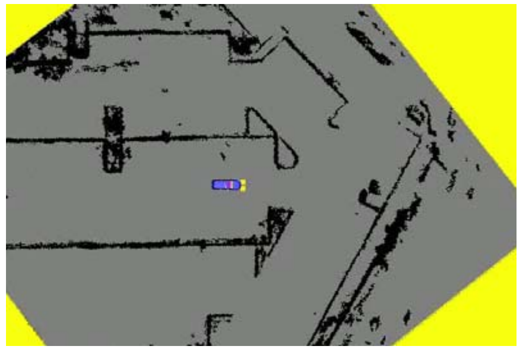
Figure 9: Map created by AnnieWAY.

To avoid obstacles on the road and on unstructured environments like parking lot an autonomous vehicle must be capable of accurately and robustly detecting obstacles at a sufficient range. AnnieWAY uses a 2.5D elevation grid where each cell stores information on the flatness, drivability and elevation. The grid map is asynchronously constructed by the lidar sensors. As the vehicle moves, the map is shifted. This limits the map to only contain all cells within a fixed margin around the vehicle.

Figure 8 shows the aerial image of a parking lot which is mapped by our mapping algorithm. The final map is displayed in Figure 9. Black areas are cells classified as obstacles whereas grey areas are drivable and yellow corresponds to unknown area.

2.2.2 Object Tracking

For an autonomous vehicle in urban environments, detecting and tracking other vehicles in real time is very important to capture and estimate the dynamics of these environments. In our vehicle we use a system that consists of two parts:

1) Segmentation of point cloud/image data into objects
2) Tracking of objects in subsequent sensor frames

The segmentation is done asynchronously for the different sensors types, at rates that vary from 10Hz to 75Hz. E.g. the point clouds from the lidar sensor are analyzed for clusters of high densities. Every time a cluster is found a new object is initialized.
We use a linear Kalman filters to both predict the next location of a moving object, and to extract useful information from the tracking process which cannot be measured directly with our sensors like velocities or accelerations. The Kalman filter is a recursive method which incorporates information as soon as it is available. Therefore the state of the vehicle's environment can be updated right away instead of waiting until information is accumulated. Recursive methods also require less computing time which is crucial in a real time system. By using Kalman filters for object tracking we have to make the following assumptions:

- Sensors and tracked objects do not move instantly
- Relative position between sensor and scene changes incrementally
- The motion of sensors and objects can be modeled with a linear model

The assumptions are obviously simplified regarding the complexity of real world environments but they hold as long as the update frequency is small enough.

2.2.3 Mission Planning and Map Information Processing

A parser has been implemented that reads RNDF and MDF files provided by DARPA. The parser makes extensive use of modern software engineering approaches such as factory classes and the hash maps of the C++ Standard Template Library. All elements of the RNDF (lanes, checkpoints, exits, etc.) can be addressed directly by their name. This enables an easy conversion of the RNDF data to a graph-based map representation. RNDF waypoints form the vertices of the graph; lanes and exits are represented by graph edges. Information such as distance, lane markings, and speed limits from the MDF is attached to the respective edges. These information can be updated dynamically, e.g. to describe a road blockage.
The graph is input to the mission/route planner. It finds the optimal route from one checkpoint to another using the A* graph searching algorithm. This process is repeated for every pair of subsequent checkpoints in the MDF. The resulting route is written into the central real-time database together with all relevant information. Besides the obvious application of driving the shortest route, this enables other modules to make use of the map data, e.g. lane recognition may be improved by a priori knowledge on the existence and type of lane markings.
In addition to the information provided by the RNDF, the map also contains virtual turnoff lanes at intersections. A virtual lane is connects each exit waypoint with the corresponding entry waypoint. It is generated assuming standard intersection geometry. Of course, this information is uncertain and the lane recognition module has to verify whether the proposed virtual lane is drivable in reality. If deviations between map and reality are detected, the map can be updated dynamically, enabling a safer and faster rerun of the respective lane.
High-level behavior decisions of the car can be derived from the mission plan, together with the map data and the car’s current position obtained by the GPS/INS system. For example, a deceleration command is issued when approaching a stop waypoint. Other behaviors initiated by mission information encompass switching on/off the direction indicators, keeping to speed limits, and performing parking maneuvers.

2.2.4 Assessment of traffic situations

One of the major problems for driving autonomously in inner-city environments is the increased complexity of the traffic scenarios compared to simple environments such as expressways. This requires more sophisticated approaches for vehicle guidance. It is not sufficient to use the information about the course of the road directly for control. Rather than using a simple control strategy there is need for a higher-level component. This higher-level component consists of a situation interpretation and a decision making part. The situation interpretation is used to get a deeper understanding of the current situation and the output serves as a basis for the decision making process. The data acquired by various sensors is fused, converted into a more abstract symbolic representation and interpreted using prior knowledge. Prior knowledge can be divided into common knowledge such as traffic rules and knowledge derived from previous made experiences. Summarizing, the goals of the interpretation are:

- Extraction and identification of relevant data for behavior execution
- Selection of higher level behavior such as “turn off right”

The perception delivers data in terms of predefined objects such as lanes, junctions, vehicles and obstacles. These are stored in a central database and the situation interpretation converts this data into an abstraction in order to feed it into the reasoning process. The result of the interpretation consists of a selected behavior together with the relevant data for behavior execution. The behavior is then executed within a behavior network, which monitors the progress of the behavior.

Structure of the interpretation

The interpretation collects the quantitative data from the perception modules in order to obtain a holistic view of the scene. The data is transformed into an abstract representation and rules are applied which yield the recommended behavior together with the relevant data, i.e. the current lane, relevant vehicles and obstacles and the current right-of-way regulation.

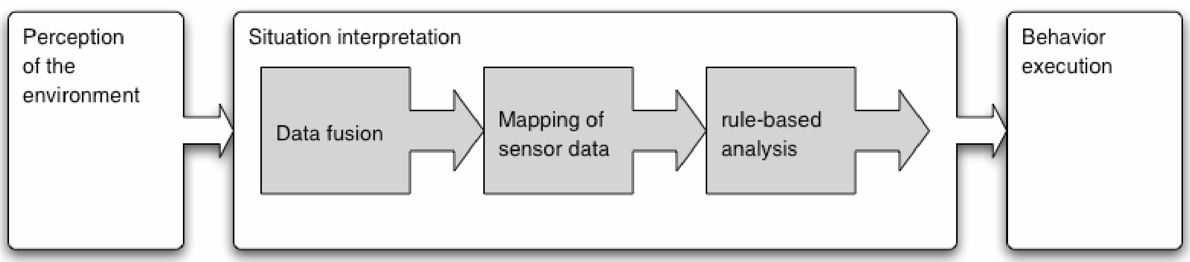

Figure 10: Structure of the situation interpretation as a three-step process.

As can be seen in **Fehler! Verweisquelle konnte nicht gefunden werden.**, the interpretation of the current situation is performed at three different steps. First, the data from the perception is collected and integrated into the existing scene representation. Since the data stems from different sensors with different temporal and spatial resolutions, the data has to be fused. In the second step, the quantitative sensor data is mapped onto a qualitative description in order to facilitate high-level reasoning. Part of this transformation is the evaluation of topological and spatial relations between objects such as `is_far(obj1, obj2)`. In the last step, a rule-based analysis is applied to deduce implicitly existing knowledge. Basically, this knowledge consists of the current right-of-way regulation.

Knowledge modeling and situation representation

For the definition of the vocabulary, which is used throughout the interpretation process, an ontology was set up, which guaranties a consistent, logical modeling of the knowledge. The ontology describes the concepts and relations needed for describing the scene. Reasoning mechanisms can be used to ensure consistency. OWL defines a most abstract concept `OWL:thing`, and the three main concepts

- Object,
- Attribute,

- General condition

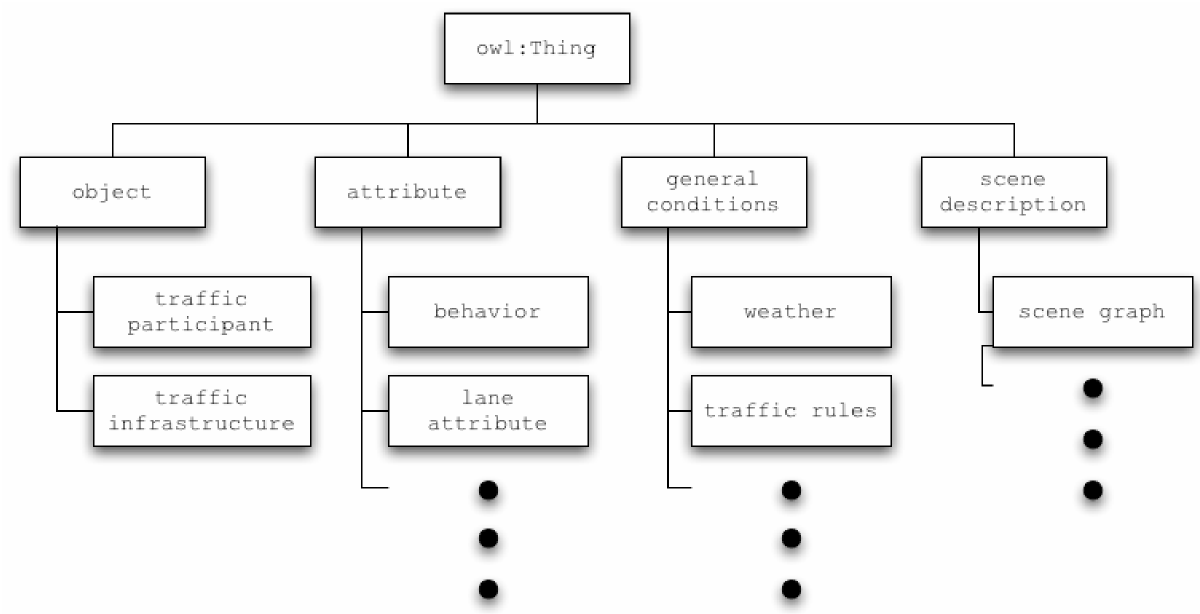

Figure 11: First level of defined ontology for describing traffic situations.

- and Scene description

are derived. The following Figure 11 shows the first levels of the resulting hierarchy of the ontology.

The concept `object` consists of only two subconcepts: traffic participants and traffic infrastructure. Traffic participants are further divided into cars, motorcycles, pedestrians and so on. The concept `traffic infrastructure` is further detailed into lanes, streets and intersections. A lane is defined by its geometry and additional attributes, which describe the effective traffic rules, relations to other lanes (such as left or right neighbor) and other traffic participants on that lane. The specializations of the concept `attribute` define concepts, which are used to describe other objects more precisely such as the current behavior of a traffic participant or the properties of a lane. The concept `general conditions` contains concepts to describe static background knowledge such as traffic rules. Concepts under `scene description` do not represent real world objects but can be seen as a collection concept. Furthermore, different roles are defined to describe relations between objects such as `is_on(Vehicle_A, Lane_B)`.
The internal representation of the current situation uses a so-called *scene graph*. It is a directed graph and the nodes represent instances of objects and attributes, respectively or relations. An edge exists between an object and an attribute to describe the object's attribute or between a relation and objects to describe the relation with the attached objects. Figure 12 shows on the left the general layout of a scene graph and a short example on the right. For clarity of presentation, attributes are directly annotated within an object and the relation `is_on` is written as a label of the connecting edge.

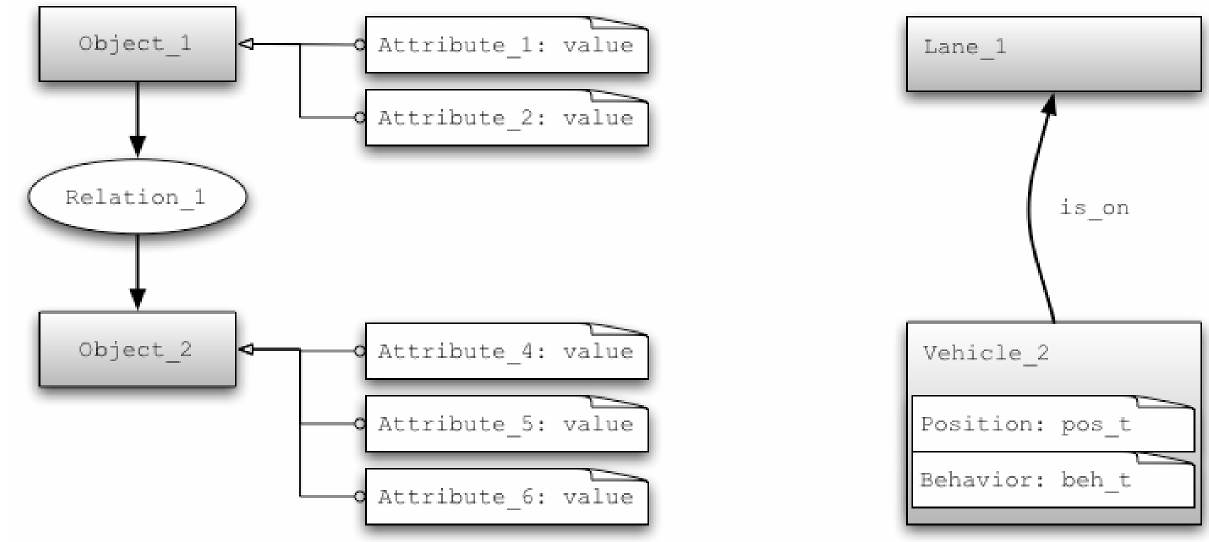

Figure 12: General layout of the scene graph (left) and short example on the right.

Interpretation of traffic situations

In each time step, the interpretation collects the data from the perception, integrates it into the scene graph, and interprets the situation.

Data fusion

The data fusion step takes the data from the perception and updates the scene graph. If a node already exists for a particular data type, the new data is fused with the existing information and the change is propagated through the graph in order to update the nodes that depend on this node. For newly acquired data types, e.g. a newly detected vehicle, a new node is created and inserted into the graph. Outdated nodes are eliminated if they don't receive an update for a specified amount of time.

Mapping of sensor data

The mapping of sensor data converts the qualitative data from the perception into the higher-level descriptions defined by the ontology. The continuous measurements are mapped onto discrete attributes and additional relations are evaluated. For example the relative geometric position of two vehicles is transformed into a symbolic attribute such as `in_front_of` or `left_of`. Additionally, vehicles are assigned to lanes, expressed with the relation `is_on(Vehicle_A, Lane_B)`.

Rule-based analysis

Most of the relations can be updated in the mapping phase by simply evaluating the geometric relation between objects. But for some relations, an evaluation of more complex rules has to be applied. The premise holds all conditions that have to be met and the conclusion describes the relations that are valid. The following rules illustrate the processing of the right-of-way rule at a four-way stop:

```
IF   (is_a(traffic_participant, ?obj1) AND
      is_a(traffic_participant, ?obj2) AND
      is_a(lane, ?lane_1) AND
      is_a(lane, ?lane_2) AND
      belongs_to(?intersection, ?lane_1) AND
      belongs_to(?intersection, ?lane_2) AND
      is_on(?lane_1, ?obj1) AND
      is_on(?lane_2, ?obj2) AND
      arrived_before(?obj1, ?obj2)
     )

THEN (has_ROW(true, ?obj1) AND
      has_ROW(false, ?obj2)
     )
```

The reading of the rule is as follows: A question mark followed by a variable name is considered as a variable for which a valid object is searched. For example, `is_a(lane, ?lane_1)` searches for objects which are of type `lane`. Additionally, all relations, which contain the object `lane_1` must be valid, e.g. `belongs_to(?intersection, ?lane_1)` must be fulfilled. The first two lines search for two traffic participants. Then, two lanes are searched which belong to the same intersection. Furthermore, the extracted traffic participants need to be on these lanes. Finally, a check is performed, if `obj1` arrived earlier than `obj2`. If all these premises are met, it is deduced, that `obj1` has the right-of-way over `obj2`. Therefore, the relation `has_ROW` is set to true for `obj1`.

2.2.5 Behavior Generation and Execution

The goal of the behavior generation and execution is to find a suitable representation for individual behaviors, and to be able to perform safe driving maneuvers according to the given situation interpretation.

Behavior generation in human brains is a process which involves different regions of the brain, each one considering specific driving aspects. These regions interact with each other and are able to stimulate or suppress other areas. The executed behavior is a result of a fusion process and depends on the structure of the network and the motivations of individual behaviors.

The system architecture, called *Behavior Network*, pays special interest to the interaction between behaviors and is going to be used to control the cognitive automobile. This biologically-oriented method seems adequate to derive behaviors for driving and perception, which are often combinations of several sub-behaviors with different motivations (road-following, lane-keeping, speed control, collision avoidance). The attempt to control the vehicle with human-like behaviors has the advantage of good traceability of the executed maneuvers, as well as the use of humans as teachers to parameterize behaviors.

The behavior generation and execution is placed in-between the situation interpretation and the trajectory planning. Looking at the overall process, objects in the traffic scene like lane markings, obstacles or other participants are detected with vision systems and LIDAR, classified and the result will be handed to the situation interpretation. This module generates information about object relations and adds logic features, considering the own intention according to the navigation system.
The results of this process are being transferred to the behavior component, which is split into a decisive and an executive part. Strategic behaviors on the cognitive level know how to perform a driving maneuver and have to decide about motivation of underlying tactical behaviors, fitting best for the actual situation. Behaviors of tactical or reactive character are executed according to their included scheme and have only limited knowledge about the overall situation, but are directly supplied by the database with newest sensor information.

The lower set of reactive behaviors is generating input values for the control part. To perform cooperative maneuvers in case of emergency, an interface to a cooperative decision unit provides the ability to overwrite individual decisions.

Behavior Representation and Organization

Behavior networks were developed with the goal to obtain a modular and robust control architecture for walking machines. The idea was to combine a classical hierarchical control approach with conclusions of biology. It is known, that certain movements or impulses from humans or animals stimulate always the same neurons, and an activity of a certain region has influence on other areas. This leads to the idea of paying special interest to the interactions of behaviors, and to model them separately. As a result, a behavior unit with several connections for user and status data arose, as can be seen in Figure 13.

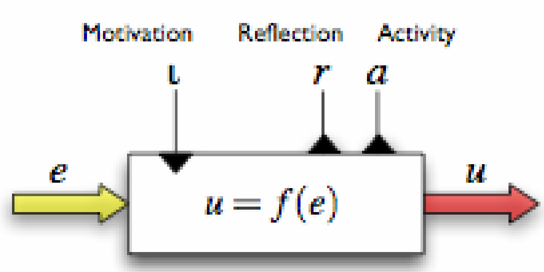

Figure 13: Single behavior module.

The transfer function *f* contains the basic character of the behavior. In a first step, the result *f(e)* is computed, and modified in a second step according to the motivation ι. Every behavior has defined an internal goal which it is trying to reach. The reflection *r* describes to which extent the actual state differs from the desired state. The rating is generated independently from the activity of the behavior. This aspect of *virtual sensors* is a very important issue within the architecture, and different to other behavior-based architectures.

Through the activity *a*, the behavior shows the actual effort in reaching the defined goal. This information can be used for weighted fusion of different behavior outputs. Standardization of status data connectors to [0;1] allows an abstraction from internal physical values, and qualifies the connections for coupling of behaviors.

Behavior Network

Three main categories of behavioral decision making were identified in the human decision process while driving a car: a reactive, tactical, and strategic category (see Figure 14). The reactive category consists of more or less unconsciously made actions, like keeping the car in the center of the lane or making an evasive action because of a child unexpectedly crossing the street. These actions can be categorized in two subcategories: Actions influencing the transverse dynamics (e.g. keeping the car in the lane) and actions influencing the longitudinal dynamics (e.g. reducing or increasing the speed) of the car.

Tactical actions are those made more consciously like stopping in front of a traffic light or overtaking another car. These are actions that a driver has to actively plan in short-term, but that are not directly connected to the bigger goal the driver wants to achieve. Actions falling in this category are strategic ones like following the street, taking the next turn to the right, or switching from a normal road to a highway.

The reactive layer generates output for the underlying control. It only has a limited view on the world and a basic set of actions to generate the driving corridor, which was chosen as interface to the control. This layer consists of behaviors for changing longitudinal and transversal dynamics. Their output is merged in fusion nodes for speed and the driving corridor. There are several safety behaviors that can override the decisions of all other behaviors to enforce the safety of the car.

Behaviors of the tactical layer are controlling the action primitives of the reaction layer to fulfill a special task. One could see a node of the tactical layer as a specialized agent that knows how to use the action primitives to solve a short-term-task. These agents act independently from each other. Cooperation is handled in parts in the fusion nodes between the reactive and tactical layer. They also get supervised by the active strategic behavior that can prevent hazardous situations by changing the motivation of the cooperating tactical behaviors.

Conflict Zones and Decision Making

Many objects can influence the driving decision. To reduce complexity, we redefine the term lane to also include virtual lanes defined by everything (besides moving obstacles) that can influence the driving decision. With this definition a pedestrian overpath is a (virtual) lane crossing the street because the car has to wait when there are pedestrians on it. Certain events or installations may be modeled as (virtual) lanes as well, so e.g. a traffic light or a stop sign, which crosses every lane of the street it is binding for.

A *conflict zone* is modeled as the intersection region of the current driving lane and every other (virtual) lane. Conflict zones are the only regions to be considered for the behavior selection.

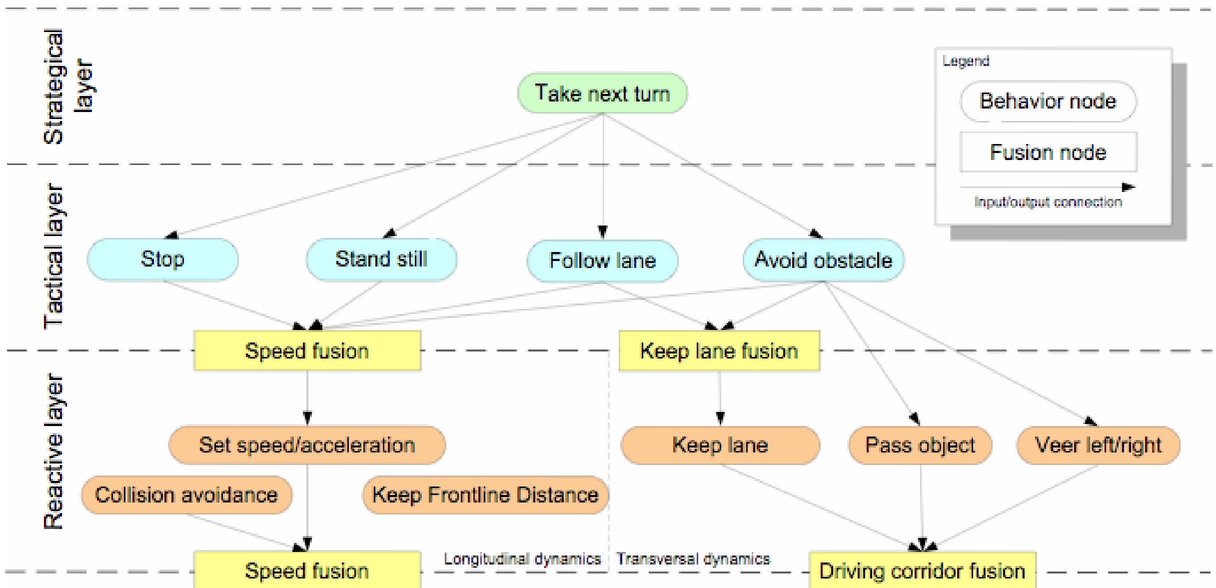

Figure 14: Behavior network with behaviors on several layers.

They are classified into the following three grades:

1. This conflict zone is currently not a conflict. The car can pass this zone unconditionally but the zone could become a conflict zone of a higher grade. E.g. a green traffic light is a conflict zone of grade 0.
2. This conflict zone might be a conflict, depending on other criteria. The behavior network has to analyze further information and has to decide whether the car can pass this zone with the desired tactical behavior or not. E.g. a yellow traffic light is a conflict zone of grade 1.
3. This conflict zone is a conflict and cannot be passed. E.g. a red traffic light is a conflict zone of grade 2.

Every conflict zone has objects associated with it. E.g. a pedestrian overpath with a traffic light would have the traffic light and all detected pedestrians associated as objects. By means of these objects, the conflict zone needs to be analyzed to find the reason for this (potential) conflict. Conflict zones with associated objects, (virtual) lanes, and their interconnections will be provided from the scene interpretation.

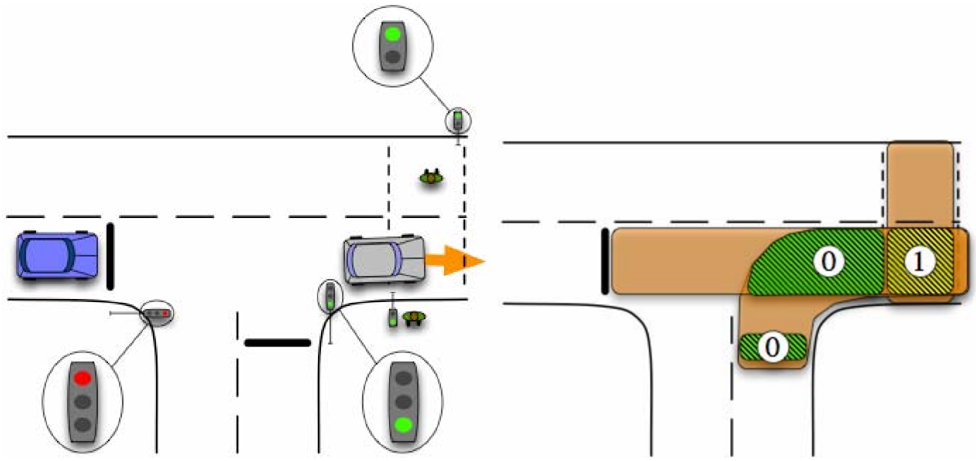

Figure 15: Potential conflict zones for manoeuvre *turn right into street.*

The decision making component, called *Scenario Monitor* serves as an interface between the strategical layer and the situation interpretation. All strategical behaviors consult the *Scenario Monitor* before they motivate tactical behaviors. The *Scenario Monitor* has a generic check algorithm for every tactical behavior. It knows the preconditions that have to be met to allow a motivation of the action, and it returns constrains for a safe execution of the action.

Normally the strategical behavior chooses the action that contributes the most to the desired goal. Another possible policy is to motivate a tactical behavior depending on its virtual sensors. E.g. the strategical behavior should consider to motivate *avoid obstacle* whenever the reflection of this behavior indicates its dissatisfaction. Before motivating the desired action, the strategic behavior needs to check the feasibility of it. If the action is not allowed, the strategical behavior has to repeat the selection process with a backup-action. This has to be done until a feasible action was found that the strategical behavior can motivate.
On every execution cycle of the strategic layer checks of the currently running tactical behavior have to be repeated.

Internally the *Scenario Monitor* has a set of check-primitives that can be applied on any conflict zone. A feasibility test for a tactical behavior consists of a list of check-primitives that are executed on every available conflict zone.

2.2.6 Path Planning

Once reactive behaviors are motivated from upper behaviors, their output needs to be combined before being handed over to the control part. A driving corridor was chosen as interface to the control, representing a free street and containing the combination of reactive driving outputs.

Extended potential field maps will serve to represent each behaviors output in form of risk values. In this approach, the potential map will not only be used to determine the shortest way around obstacles in the environment (as usually done in mobile robotics path planning), but will also contain the driving intention. A final map will result through maximum-fusion of the individual maps.

This ensures direct safety in the sense that correctly detected obstacles will be avoided and that the vehicle will be kept within the detected street.

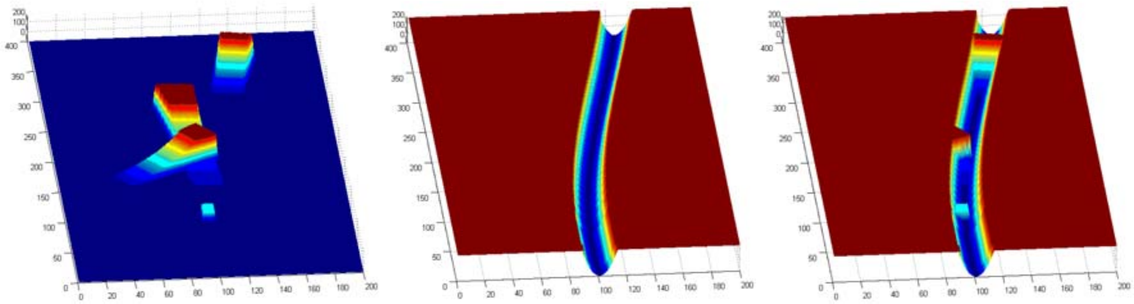

Figure 16: Fusion of multiple reactive behavior outputs: *Avoid obstacle* (left), *Keep lane* (center) and combination (right).

A horizontal cut of the risk map results in a driving corridor, which will be delivered to the control and represents the driving intention. Variation of the cutting plane allows to modify the accepted risk level, and to enlarge the scope in case of emergency.

2.2.7 Vehicle Control

Lateral Control

To the vehicle's steering column an electric motor was attached in order to realize steering motion. A cascade structure was chosen as steering control architecture. This architecture combines fast hardware controllers (PI-rotation-speed and current controller) with a P-position software controller that is implemented on a dSpace real-time hardware. A feed-forward term assures quick response to requested position changes of the steering wheel. In combination with the as static assumed inverse steering cinematic the lateral controller uses the steering angle of the front wheel as input to the plant.
For the lateral movement of the vehicle a software state space controller was chosen. This controller minimizes the distance and heading error of the car as it moves along the planned curve. A feed-forward term, that uses the curvature of the planned curve, strongly increases accuracy.

Longitudinal Control

Since the vehicle's longitudinal dynamic is nonlinear, a compensation algorithm similar to the steering controller was chosen, which converts a requested acceleration into acceleration pedal and brake pressure values. The compensation of the nonlinearity was again achieved by using the inverse of the static characteristic (engine and brake system). Velocity-dependant disturbances like wind drag and roll resistance of the wheels were measured in advance and accounted for in the control structure. For unpredictable disturbances, such as additional wind, an integrator was added. The acceleration controller is then utilized by the actual velocity or distance controller.

3. Tests

In-car testing has played a major role in the development of AnnieWAY car. The primary testing area was the Mackensen Military Base in Karlsruhe, Germany.
Notable tests include:

1. Numerous test runs to show vehicle safety.
2. 100 miles on our test track in Karlsruhe demonstrating basic navigation at velocities up to 30mph.
3. Extensive testing of basic traffic skills via simulation:
    - Intersection precedence
    - Vehicle following
    - Queuing
    - 4-way stop behavior
    - Dynamic replanning
    - Road following

4. Conclusion

With twenty-seven days remaining to the Urban Challenge Site Visit qualifier, Team AnnieWAY shows good performance and viable reliability in terms of vehicle safety and basic navigation tasks. Additionally planned tests include end-to-end race day simulations, 100 hours of urban driving without human intervention, difficult urban settings with multiple cars, and simulation of temporary system failures.
A still remaining key challenge is to proof reliable behavior of AnnieWAY in complex traffic situations. Almost all tests of navigation in traffic were conducted in a simulator so far. For some tests simplified assumptions e.g. constant velocities were enforced to show proof of concept of Team AnnieWAY's approach towards autonomous driving.

5. Acknowledgments

The authors gratefully acknowledge support of this work by the Deutsche Forschungsgemeinschaft (German Research Foundation) within the Transregional Collaborative Research Centre 28 'Cognitive Automobiles'.

6. References

[1] H.-H. Nagel, W. Enkelmann, and G. Struck, "FhG-co-driver: From map-guided automatic driving by machine vision to a cooperative driver support," Math. and Computer Modeling, vol. 22, pp. 101 –108, 1995.
[2] C. Thorpe, Vision and navigation - The Carnegie Mellon Navlab. Kluwer Academic Publishers, 1990.
[3] U. Ozguner, C. Stiller, and K. Redmill, "Systems for safety and autonomous behavior in cars: The DARPA Grand Challenge experience," IEEE Proceedings, Feb. 2007.

[4] J. Rasmussen, “Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models,” IEEE Trans. Systems, Man, and Cybernetics, vol. SMC-13, no. 3, pp. 257–266, Mai/Juni 1983.
[5] M. Goebl and G. Färber, “A Realtime-capable Hard- and Software Architecture for a Joint Image and Knowledge Processing in Cognitive Automobiles,” in Proc. IEEE International Conference on Intelligent Vehicles, to appear, 2007.
[6] T. Dang, C. Hoffmann, and C. Stiller, “Self-calibration for active automotive stereo vision,” in Proc. IEEE Intelligent Vehicles Symposium, Tokyo, Japan, June 2006, pp. 364 – 369.
[7] H. Ruser and F. Puente Leon, “Informationsfusion - eine Übersicht,” Technisches Messen, vol. 74, no. 3, 2007.
[8] J. Beyerer, F. Puente Leon, and K.-D. Sommer, Eds., Informationsfusion in der Mess- und Sensortechnik. Karlsruhe: Universitätsverlag Karlsruhe, 2006. [Online]. Available: [1]http://www.uvka.de/univerlag/volltexte/2006/159/
[9] E. D. Dickmanns, “The development of machine vision for road vehicles in the last decade,” in Intelligent Vehicles Symposium, 2002. IEEE, vol. 1, 2002, pp. 268–281 vol.1.
[10] C. Stiller, “Cooperative environment perception,” in Proc. 1st Autocom Workshop on Preventive and Active Safety Systems for Road Vehicles, Istanbul, Turkey, Sept. 2005, pp. 39 – 41.
[11] S. E. Palmer, Vision Science: Photons to Phenomenology, 3rd ed. The MIT Press, Cambridge, (MA), USA, 1999.
[12] G. Färber, “Biological Aspects in Technical Sensor Systems,” in AMAA05, Berlin, Mar. 2005.
[13] S. Neumaier, P. Harms, and G. Färber, “Videobasierte Umfelderfassung zur Fahrerassistenz,” in 4. Workshop Fahrerassistenzsysteme FAS2006, L¨owenstein, Oct. 2006.
[14] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, 2nd ed. Morgan Kaufmann Publishers, San Francisco, (CA), USA, 1988.
[15] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” IEEE Trans. Pattern Aanalysis and Machine Intelligence, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.
[16] M. Richardson and P. Domingos, “Markov logic networks,” Machine Learning, vol. 62, pp. 107 – 136, 2006.
[17] H.-H. Nagel, “Steps toward a cognitive vision system,” AI-Magazine,
vol. 25, no. 2, pp. 31 – 50, 2004.
[18] I. Dagli, M. Brost, and G. Breuel, “Action recognition and prediction for driver assistance systems using dynamic belief networks,” in Conf. Agent Technologies, Infrastructures, Tools, and Applications for EServices, 2002, pp. 179 – 194.
[19] K. Tischler, M. Clauss, Y. Guenter, N. Kaempchen, R. Schreier, and M. Stiegeler, “Net worked environment description for advanced driver assistance systems,” in Proc. IEEE Intelligent Transportation Systems Conference, Vienna, Austria, Sept. 2005.
[20] S. Chakrabarti and A. Mishra, “Qos issues in ad hoc wireless networks,” IEEE Communications Magazine, vol. 39, no. 2, pp. 142–148, Feb. 2001.
[21] H. Xiaoyan, X. Kaixin, and M. Gerla, “Scalable Routing Protocols for Mobile Ad Hoc Networks,” IEEE Network, vol. 16, no. 4, pp. 11–21, July/August 2002.
[22] J. Baber, J. Kolodko, T. Noel, M. Parent, and L. Vlacic, “Cooperative autonomous driving - Intelligent vehicles sharing city roads,” IEEE Robotics and Automation Magazine, vol. 12, no. 1, 2005.