# Vehicle Detection and Tracking for the Urban Challenge

Michael Darms*, Chris Baker†, Paul Rybksi‡, Chris Urmson§

**Abstract:** The Urban Challenge 2007 was a race of autonomous vehicles through an urban environment organized by the U.S. government. During the competition the vehicles encountered various typical scenarios of urban driving. Here they had to interact with other traffic which was human or machine driven. This paper describes the perception approach taken by Team Tartan Racing, winner of the competition. A focus is set on the detection and tracking of other vehicles around the robot. The presented approach allows a situation specific interpretation of perception data through situation assessment algorithms keeping the perception algorithms situation independent.

**Keywords:** Urban Challenge, Tartan Racing, Object Tracking, Situation Assessment

## 1 Introduction

The Urban Challenge is the third race for autonomous vehicles organized by DARPA [1]. During the first two races in the years 2004 and 2005, which are known as Grand Challenges, robots had to travel on a predefined route from start to finish through the desert. As defined by the rules the robots did not encounter any moving objects. If two robots came to close to each other one of them was paused, no interaction took place. The races were a milestone in history of autonomous vehicles [2][3].

As a next level of autonomy the third race known as Urban Challenge took place in an urban environment in November 2007. Autonomous vehicles had to accomplish missions where different locations in a road network had to be reached. The network contained for example multi-lane roads, intersections and open parking lots. The robots where simultaneously on the course together with human driven vehicles. All participants had to obey the rules of the California Driver Handbook [4]. In contrast to the first two races the robots had to have the ability to interact with other traffic and understand different scenarios. These ranged from simple scenarios like driving behind slower vehicles to complex ones like handling an intersection or driving on an open parking lot together with other vehicles.

As a consequence algorithms and sensors had to be found which enable the robot to perceive and interpret the environment in a way that allows safe decision making while still driving smooth and fast. This article describes the perception system of the autonomous

*M. Darms is with Continental, Lindau, Germany. (e-mail: michael.darms@us.contiautomotive.com)
†C. Baker is with Carnegie Mellon University, Pittsburgh, PA, USA. (e-mail: tallbaker@ri.cmu.edu)
‡P. Rybski is with Carnegie Mellon University, Pittsburgh, PA, USA. (e-mail: prybski@cs.cmu.edu)
§C. Urmson is with Carnegie Mellon University, Pittsburgh, PA, USA. (e-mail: curmson@ri.cmu.edu)

robot "BOSS" build by Team Tartan Racing, winner of the Urban Challenge 2007 (see also [10]). A focus is set on the detection and tracking of vehicles around the robot.

The described system implements a situation specific interpretation of sensor data which allows the robot to adapt its behavior to the capabilities of the perception system. For this the perception system does not directly translate the sensor data into a world model representation used by the behavior algorithms of the robot. It delivers just enough information so that situation assessment algorithms can interpret the data with respect to the current situation.

In section 2 the overall perception system will be described. First the definition of the world model which is used to represent the environment around the robot will be explained. Afterwards the software structure of the perception system will be described and it will be explained which information the perception system delivers as input for the situation assessment algorithms. Section 3 shows how the perception system's data is used in different scenarios by the situation assessment algorithms. As examples distance keeping and intersection handling are used. The section finishes with a short discussion about the need for an explicit vehicle classification for driving in an urban environment. The article ends with a short conclusion and a discussion of the presented approach.

# 2 Perception System

## 2.1 Definition of the World Model

The following section describes the world model used to represent an environment containing both static and moving obstacles. While the representation was tailored to the urban challenge, it can be also used to represent the more general environments. Three different domains are used: road structure, static obstacles and dynamic obstacles.

The road structure defines where and how vehicles are allowed to drive, encoding traffic rules as necessary. This includes information about the road boundaries, the number and position of lanes, lane widths and lane markings. Note that this does not necessarily correspond directly to physical boundaries such as walls or trees for example. The road structure is a logical interpretation of the environment, based on lane markings or curbs for example.

Static obstacles are defined as obstacles which are assumed not to move during the observation period. This includes obstacles off and on the road. Example static obstacles are buildings, traffic cones or parked cars which do not participate actively in traffic. In contrast to this dynamic obstacles are defined as objects which potentially move during the observation period. With this definition all vehicles participating actively in traffic are dynamic obstacles even if they are temporarily stopped at an intersection for example.

With the definitions given above an object is either static or dynamic, it cannot not be represented in both classes. Note however that an object can change from static to dynamic and vice versa: A person may for example get into a parked car and participate from this moment on actively in traffic. The information if an object is static or dynamic is not tied to its physics, it is like the road structure an interpretation of the current state of the environment around the robot.

One technical challenge within this approach is distinguishing between static and dynamic obstacles. Sensor measurements can only provide clues to answer this question based on physical observations (e.g. engine is running). Even if it would be possible to classify an object as a vehicle reliably it would still not be clear if it actively participates in traffic or not. Knowledge about the scene and the situation is needed to fully answer this question.

## 2.2 Architecture

### 2.2.1 Overview

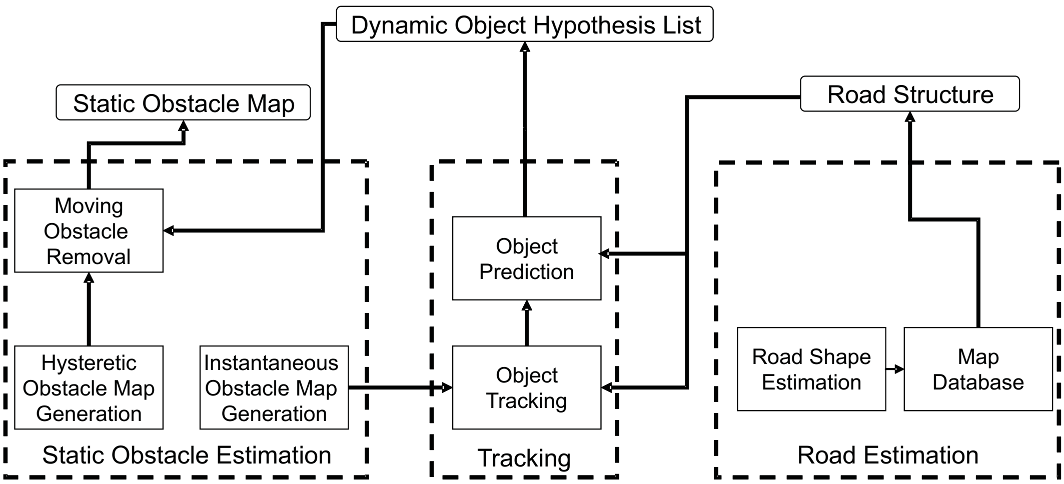


Figure 1: Architecture of Perception System. Encapsulated in dashed lines the subsystems. Arrows show the data flow.

Figure 1 shows the software architecture of the Tartan Racing's perception system. It provides all information necessary for an situation assessment algorithm to interpret a scene as described in the previous section (2.1). All information generated by the perception system is solely based on physical sensor readings. "Scene understanding" is not used to interpret the data. This way the perception system is kept situation independent, which makes the overall system more flexible regarding and extensible (see also [5]).

The perception system is divided into three subsystems: A road estimation subsystem which generates information about the road structure, a tracking subsystem which generates information about dynamic objects (see e.g. [6], [7]) and a static obstacle estimation subsystem which generates information about static obstacles. Every subsystem uses a multisensor approach. More than a dozen sensors were used to generate the data. Table 1 gives an overview. The following subsections will explain which data is generated by each subsystem.

### 2.2.2 Road Structure

To describe the road structure a vector format is used. The road structure is represented as a topological network of segments, intersections, and zones. A segment contains a number of road lanes where each lane has a specified shape and direction of travel. Intersections are junctions that explicitly connect lanes from different segments. Zones are free form open areas, such as parking lots, which have no explicit restrictions on where vehicles are

Table 1: Sensors used on the autonomous vehicle Boss. All numbers presented in the table are based on specifications.

| Sensor | Characteristics |
|---|---|
| Applanix POS-LV 220/420 GPS/IMU | sub meter accuracy GPS system with Omnistar VBS corrections, used to position the robot within the digital map |
| Velodyne HDL-64 Lidar | scanning laser with $360^o$ x $26^o$ FOV, 120m maximum range, used for object tracking and static/instantaneous map generation |
| SICK LMS 291-S05/S14 Lidar | scanning laser with $180°/90°$ x $0.9°$ FOV and 80m maximum range, used for object tracking, static/instantaneous map generation and road estimation |
| Continental ARS 300 Radar | scanning radar with $60°/17°$ x $3.2°$ FOV and 60m/200m maximum range, used for object tracking |
| Continental ISF 172 Lidar | fixed beam laser with $14°$ x $4°$ FOV and 150m maximum range, used for object tracking |
| IBEO Alasca XT Lidar | scanning laser with $240°$ x $3.2°$ FOV and 200m maximum range, used for object tracking |
| Point Grey Firefly | High dynamic range camera with 45 FOV, used for road estimation |

allowed to travel. With these definitions an urban network can be described sufficiently for autonomous driving. The road structure is also used by the tracking subsystem for computational efficiency and to reduce the number of sensor artifacts (see e.g. [7]).

### 2.2.3 Dynamic Obstacle Hypothesis List

Since distinguishing static and dynamic obstacles requires scene understanding only a list of dynamic obstacle hypothesis - all objects which potentially belong to the class of dynamic obstacles - is provided. For the special case of the Urban Challenge it was known that all dynamic obstacles were vehicles of a known maximum size. This allows the use of two simple models do describe dynamic objects: a point model, which does not include any information about the shape of an object, and a box model, which describes dynamic objects by a box with a predefined length and width. The model is chosen based on the available sensor information (see [6] for details).

In addition to estimated state variables like position and velocity the following information is included in the form of discrete flags for each dynamic object hypothesis a) the current movement state: Moving or Not Moving and b) the past movement state: Observed Moving and Not Observed Moving. The flag Moving is set once the tracking subsystem determines that the object hypothesis is currently moving. The flag Observed Moving is set once the tracking system has determined that the hypothesis has been moving in the past.

All hypothesis which have the Moving and Observed Moving flag set are directly known to be dynamic obstacles. Obstacles which only have the Observed Moving property might be dynamic obstacles which have become static (e.g. a car that moved into a parking spot and will not move again). In practice, we treat objects that have only the Observed Moving flag as dynamic obstacles since if an object was recently moving, it is likely it will move again soon. Testing showed that this assumption is a good approximation for short observation periods.

Dynamic obstacles can by definition move during the observation period. Because of this objects hypothesis are only kept in the list as long as sensor data can be used to support the estimation of the objects state variables. This means that once an object hypothesis is not visible to a sensor anymore (e.g. it becomes occluded) it is removed from the list. Only sensor specific artifacts ("measurement dropouts") are bridged by the tracking system over a time period, where a model based prediction deliverers meaningful data (see [5]).

#### 2.2.4 Static Obstacle Map

Static obstacles are represented in a map with equally spaced cells. The advantage of this format is a description without the concept of an object or a particular object model. The disadvantage however is the amount of data that is needed to represent the obstacles (see e.g. [8] for a discussion). Data associated to dynamic object hypotheses having the Observed Moving flag set are removed from the static obstacle map as the interpretation is obvious in this case.

In contrast to dynamic obstacles the information about static obstacles is kept in the map, even when it becomes occluded. Data is removed from the map once a sensor actively confirms, that there is no obstacle at the given location. An algorithm with a time constant long enough to reduce sensor noise sufficiently for the robots motion planning algorithms (see e.g. [9]) is used.

As an intermediate result an instantaneous obstacle map is generated, too. The map contains static and dynamic obstacles - data associated to dynamic obstacle hypotheses with the Observed Moving flag set are not removed. Similar to the static map there is no notion of objects, the map is basically a 3D snap shot of the scene at a given moment in time. The data is only used inside the perception system for validating sensor information inside the tracking subsystem (see [7]). Because this data does not have to be as precise as for the static obstacle map a short filtering time constant can be used.

# 3 Situation Assessment

## 3.1 Intersection Handling

As the robot approaches and waits at an intersection, the precedence order among vehicles at that intersection must be determined in order to identify the point in time when the robot may proceed. Other vehicles may be static on arrival, or for long periods of time after arrival, so all dynamic object hypotheses are used to determine the precedence order, regardless of their movement flags. This requires situational treatment of false-positives in some cases, such as a parked vehicle or any other obstacle close to the intersection. Given these possible sources of error like not detecting an object due to an occlusion the robot uses an intersection-centric (as opposed to vehicle-centric) precedence estimation

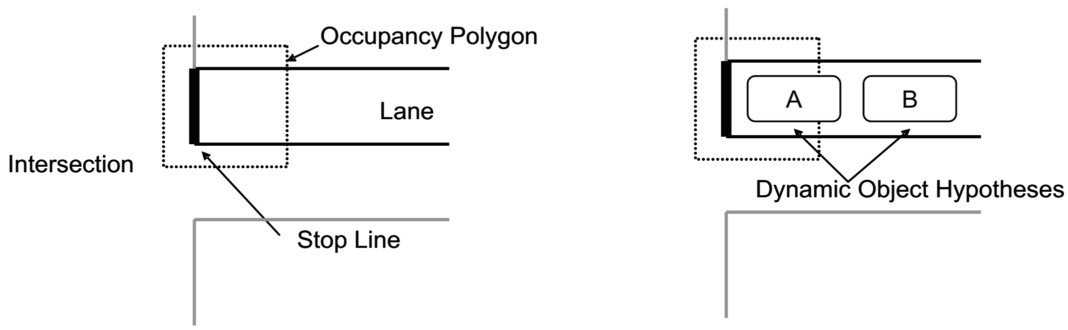


Figure 2: Example of an occupancy polygon. The dynamic object hypothesis “A” causes the polygon to be considered as occupied, hypothesis “B” not.

algorithm, described in detail in [10].

For each lane entering the intersection, an occupancy polygon is generated based on the road structure, encompassing the area backward along that lane for some configurable distance, as shown in Figure 2. A given occupancy polygon is considered ”occupied” when the estimated front bumper of any vehicle, either the robot or a dynamic object hypothesis, is inside the polygon. When in the ”occupied” state, the occupancy polygon maintains two pieces of temporal data: the time of first occupancy and the time of most-recent occupancy. The former is used in the determination of the precedence order, which is a simple matter of sorting the occupied polygons in ascending order by first occupancy. The latter is used to used to implement a temporal delay on when the polygon becomes ”unoccupied”, such that it must be persistently empty over some span of time for it to transition to the ”unoccupied” state. This delay maintains the estimated precedence ordering even if the dynamic object hypothesis briefly disappears or drifts outside the polygon, providing robustness to transient tracking errors such as sensor artifacts (”measurement dropout”) or short-time occlusions caused by vehicles traveling through the intersection.

Beyond strict adherence to precedence at an intersection, the Urban Challenge rules also specified explicit treatment of stalled vehicles and misperceived precedence ordering, requiring the robot to proceed after ten seconds of inactivity. The robust implementation of this special case also provides a path forward for the potential false-positives described above. If an occupied polygon retains precedence for ten consecutive seconds, it is subsequently ignored for the purposes of precedence estimation. This simultaneously implements the required treatment of stalled vehicles and discards any false-positive dynamic obstacle hypotheses at the cost of a ten-second delay. If the robot proceeds through an intersection under this premise, its speed and acceleration are reduced in an attempt to guarantee the safety of the maneuver.

Tests showed that the probability of an obstacle close to an intersection triggering a false-positive is comparatively low, especially if all four entrances of a typical intersection are taken into consideration. Moreover, the difference in behavior, a ten-second delay followed by reduced speed, is acceptable from an outside perspective and basically reflects cautious action by the robot.

Both the size of the occupancy polygon and the hysteretic delay may be adjusted according to the known limitations of the perception system. Larger polygons and longer delays increase robustness to perception errors and sensor noise at the cost of increasingly conservative behavior, requiring more time to decide a polygon is empty and making it

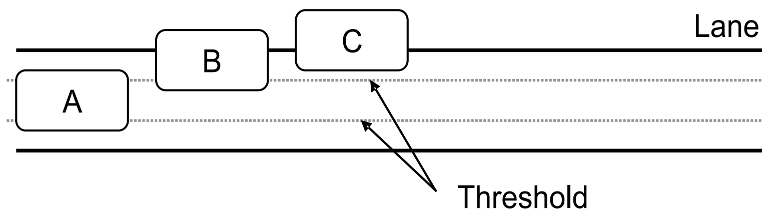


Figure 3: Different interpretations of dynamic obstacle hypothesis in dependence of the position in a lane. All dynamic obstacle hypothesis would be taken into consideration for distance keeping if they have the Observed Moving property. If not, A and B would still be considered, C not.

more difficult to differentiate between multiple vehicles queued at a stop line. Smaller values yield more aggressive decision-making at the cost of possibly proceeding out-of-turn. These values adjust the overall behavior of the robot at intersections without requiring changes in the perception system, reflecting an effective decoupling of dynamic obstacle detection from situation-specific interpretation.

## 3.2 Distance Keeping

The Distance Keeping behavior is active while driving along a road behind a slower vehicle. It aims to preserve a minimum safety gap to the lead vehicle, eventually matching speed to maintain that gap in the steady-state. The lead vehicle is selected from a subset of the list of dynamic obstacle hypotheses to limit the occurrence of spurious braking due to false-positives from the perception system in this situation. All hypotheses overlapping the lane of travel with the observed-moving property are included, reflecting a high confidence in each being an active participant in local traffic. Hypotheses which do not have the observed-moving property are only included if they are close enough to the center of the lane according to an adaptive threshold corresponding to the local width of the road (see Figure 3). The goal is to reject parked vehicles and other curbside obstacles while still taking vehicles stopped on the road into consideration for distance keeping.

There is still the possibility of falsely interpreting a dynamic obstacle hypothesis as relevant. For example, a traffic cone or barricade in the center of the lane may generate a dynamic obstacle hypothesis that will be considered as a stopped vehicle by the distance keeping behavior. Similar to the requirement of overriding the precedence order at intersection, the Urban Challenge rules required that the robot handles a stalled vehicle on the road by coming to a safe stop behind it, waiting to determine it has stalled, then safely circumventing the vehicle through an adjacent lane.

The time delay for a circumvention maneuver must be adjusted under the assumption that a vehicle is validly stopped on the road and may start moving again, as in stop-and-go traffic leading up to an intersection. In the case of a traffic cone standing on the road, this causes the robot to wait somewhat longer than preferable, but as an optimization, the location of the blocking object with respect to intersections, curves, etc. as known from the road structure is used to adjust the time delay. Once again, tests showed that the difference in behavior was small and acceptable.

The absolute position and velocity of the closest relevant obstacle hypothesis are retained over an adaptive span of time to provide robustness to sensor noise and measurement dropouts similar to the occupancy hysteresis discussed in Section 3.1. The obstacle's distance and velocity relative to the robot is considered in the computation of the hysteretic delay as an estimated time-to-collision. A dynamic object hypothesis close to the robot with a small estimated time-to-collision is retained for a comparatively long span of time, reflecting an increased risk associated with transient measurement dropouts. Farther hypotheses with longer expected collision times represent lower risks to the robot, and they are retained for accordingly shorter durations.

To illustrate the case requiring a longer hysteretic delay, consider that vehicles close to the robot are detected only by SICK laser sensors (with all other sensors being disabled), scanning in the horizontal plane with a very small vertical field of view. Under certain circumstances, depending mainly on the pitch angle of the robot and the local shape of the road, a vehicle close to the robot may not be reliably detected. This is exacerbated if the vehicle is detected comparatively late and the robot pitches forward under heavy deceleration. In this case, a longer hysteretic delay gives the perception system more of an opportunity to re-acquire the vehicle.

In the case of hypotheses farther from the robot, the perception system relies heavily on long-range radar sensors, as they are more effective at detecting distant objects than the laser sensors that contribute to the instantaneous obstacle map, which is used to validate the data (see section 2.2.4). However, they are also sensitive to metal objects on the ground, such as a manhole cover, which can be safely driven over by the robot. If such an object is detected with the radar sensors at long range, the robot slows down slightly until the object enters the validation area of the instantaneous obstacle map. Then, the radar feature will be invalidated and the associated dynamic obstacle hypothesis removed from the list (see also [7]). Since the object is still far away from the robot, the associated risk of collision is low, the time span for retaining it as the relevant obstacle is small, and the robot is allowed accelerate again after a comparatively short delay.

From an outside perspective this is again perceived as acceptably cautious behavior, and tuning the parameters to the computation of the retention delay again allows for selection between cautious and aggressive driving. In the former case, the robots reacts very conservatively to false-positives at long ranges, and fast autonomous driving is often not possible. In the latter case, the robot more readily discards long-range readings and can often be forced to perform abrupt braking maneuvers when approaching real vehicles. As above, this behavior can be tuned independently from the underlying perception system according to the needs of the specific situation.

## 3.3 Including an Explicit Vehicle Classification

It is important to note that the described approach is implemented without explicitly classifying dynamic obstacle hypotheses as vehicles (or not). The approach only interprets state variables of object hypothesis with respect to the current sitation. No classification algorithm exists which can guarantee the correct classification of every vehicle (ideal Receiver Operator Curve, ROC). But even with a nearly ideal ROC curve for classifying real word vehicles autonomous vehicles participating in the Urban Challenge differed significantly from typical vehicles encountered on the road. For example, most participants added substantial and numerous external sensors, significantly altering the visual and geometric properties of the vehicle.

An classification algorithm that may be used to separate vehicles from other objects in order to select the correct dynamic obstacle hypothesis for an situation assessment algorithm must be tuned to incorporate these deviations, but it is difficult to tune such algorithm without having data available for a validation. Since the characteristics of the other vehicles were not known before the race, the worst case would have to be assumed to ensure that no valid vehicles would be rejected. As the robot is completely autonomous, the risk of rejecting a true vehicle is very high, in contrast to driver assistance systems which are meant to supplement, not replace, the driver's abilities. In the limit this means that nearly every object has to be accepted as vehicle, limiting the usefulness of a vehicle classifier as a hard constraint on the validity of an obstacle hypothesis.

Additionally in a real world application the assumption that only vehicles are moving objects is unreasonable. There also exist motor cycles, bicycles, pedestrians, etc. To reliably select all of these objects out of the list of object hypotheses an algorithm would need to have a global knowledge of all these classes. This however is not feasible.

A vehicle classification algorithm can, however, be used to enhance the situational reasoning algorithms by supplementing the motive properties of an obstacle hypothesis. If the classification algorithm is tuned to minimize the probability of falsely classifying an object as a vehicle then adding this information to the hypothesis would convey a high confidence in its being a real vehicle. In turn, this would potentially allow more intelligent decision-making.

For example, a stopped object in the middle of the lane. If it is known the object is a vehicle, the autonomous vehicle could be conservative in how long it waits before passing since there is a reasonable chance the object will start moving. If, however, the object is not known with high confidence to be a vehicle, the old strategy may still to be used. Conversely, the old strategy may be tuned to be slightly more aggressive depending on the overall confidence in vehicle classification algorithm.

Explicit knowledge that an object is a vehicle can also be used to bias situation assessment algorithms in intersection handling scenarios. If it is known for example that the object hypothesis located inside an occupancy polygon is a vehicle, the threshold for waiting for the object to take precedence can be raised. Note that, in principle, movement properties of an object hypothesis can also be used to bias this decision. If, for instance, an object inside an occupancy polygon has the "observed moving" property, it is more likely to be a vehicle which might move again. However, tests done in preparation for the race showed that this does not change the robot's behavior significantly, especially given vehicles that are already stopped when they enter sensor range.

# 4 Conclusions

Situation assessment algorithms build the bridge between the perception system and the behavior algorithms in an autonomous vehicle. This is similar for driver assistance systems where the vehicle does not drive autonomously but a system assists the driver in different situations.

This article shows how the perception system can be kept independent of the current situation by providing sufficient information for for necessary situational assessments. The situation dependent interpretation of dynamic object hypotheses generated by the perception system presented in this article has been used for other driving situations including yielding to other traffic and changing lanes/merging. The presented approach allows tuning each of these different scenarios independently allowing a balance in behavior of the robot between caution and aggression. The underlying perception system however does not have to be changed.

The approach to tie information to specific situations instead of to dynamic objects detected in the environment (for example the intersection centric approach versus vehicle centric approach for intersection handling) is another step in decoupling the perception system from specific situations. Situation assessment algorithms can also apply situational reasoning to handle occlusions or sensor artifacts. This approach was key to efficiently tuning the performance of the overall system without altering the perception system.

It has been shown that an explicit vehicle classification is not necessary for autonomous driving in urban environments. That said, the availability of classification information improves the performance of the overall system though. The information however can only be used to augment an algorithm that works without the explicit classification since there are no guarantees that the classification algorithm will correctly classify obstacles.

The architecture presented here allows perception system improvements to transparently enhance situational awareness algorithms without further development effort and vice versa. This holds true for improvements in the reliability of the detection of obstacles and for algorithmic extensions of the system, like an explicit classification of objects.

# References


[1] Committee on Army Unmanned Ground Vehicle Technology and the National Research Council. Technology Development for Army Unmanned Ground Vehicles. Washington, D.C. 2002.

[2] C. Urmson et al.: "High Speed Navigation of Unrehearsed Terrain: Red Team Technology for Grand Challenge 2004", Tech. report CMU-RI-TR-04-37, Robotics Institute, Carnegie Mellon University, June, 2004.

[3] S. Thrun et al.: "Stanley, the robot that won the DARPA Grand Challenge", Journal of Field Robotics, Vol 23, No. 9, September 2006, pp 661-692.

[4] California Driver Handbook http://www.dmv.ca.gov/pubs/dl600.pdf

[5] Darms, M., Rybski, P., Urmson, C.: "Vehicle Detection and Tracking for the Urban Grand Challenge", AAET - Automatisierungs-, Assistenz- und eingebettete Systeme für Transportmittel, Symposium 13./14. Feb 2008, Braunschweig, 2008, pp. 260-277.

[6] Darms, M., Rybski, P., Urmson, C.: "An Adaptive Model Switching Approach for a Multisensor Tracking System used for Autonomous Driving in an Urban Environment", in Steuerung und Regelung von Fahrzeugen und Motoren - AUTOREG 2008. VDI-Berichte, Düsseldorf: VDI-Verlag, 2008, pp 521-530.

[7] Darms, M., Rybski, P., Urmson, C.: "Classification and Tracking of Dynamic Objects with Multiple Sensors for Autonomous Driving in Urban Environments", to be published in Intelligent Vehicles Symposium, 2008. Proceedings. IEEE. Eindhoven, NL, June 4.-6. 2008.

[8] Effertz, J.: "Tracking und Datenfusion im urbanen Fahrzeugumfeld", AAET - Automatisierungs-, Assistenz- und eingebettete Systeme für Transportmittel, Symposium 13./14. Feb 2008, Braunschweig, 2008, pp. 181-197.

[9] Ferguson, D., Baker C. Likhachev M., Dolan, J.: "A Reasoning Framework for Autonomous Urban Driving", to be published in Intelligent Vehicles Symposium, 2008. Proceedings. IEEE. Eindhoven, NL, June 4.-6. 2008.

[10] Urmson, C., Baker C., Darms M. et al.: "Autonomous Driving in Urban Environments: Boss and the DARPA UrbanChallenge", Submitted to Journal of Field Robotics, 2008.