A Multisensor Multiobject Tracking System for an Autonomous Vehicle Driving in an Urban Environment*

Michael Darms[1], Paul E. Rybski[2], Chris Urmson[2]
[1]Continental, Chassis & Safety Division, [2]Carnegie Mellon University, Robotics Institute

Michal Darms
Continental Chassis & Safety Division
Kemptener Str. 99, D-88131 Lindau am Bodensee, Germany
Phone: +49 (8382) 9699 0
Fax: +49 (8382) 9699 19
E-mail: Michael.Darms@contiautomotive.com

This paper presents the tracking system of Boss, Carnegie Mellon University's winning entry in the DARPA Urban Challenge in 2007. We present the key challenges for implementing the tracking system, the design principles that guided its implementation, the software architecture of the tracking system and the sensor setup used by Boss. The system has been shown to work robustly in many different situations, including intersection handling, distance keeping or driving on open parking lots. The design principles and tracking architecture are formulated in a general way and may be used for the development of driver assistance systems which have to deal with the same situations.

Driver Assist System, Intelligent Transport System, Advanced Project

1. INTRODUCTION

Autonomous driving [1] and driver assistance systems [2] are two fields of vehicle autonomy where advances in one directly affect advances in another. For instance, the fusion of environment sensor data in a driver assistance system can be directly applied the autonomous driving domain. Special fields of interest include system architecture design, environment perception and situation assessment algorithms.

This article describes the multisensor multiobject tracking system used by Boss. Boss, named after Charles F. "Boss" Kettering, is an autonomous self-driving car built by Tartan Racing [4] (see Figure 1). Boss competed in and won the 2007 Urban Challenge.

The Urban Challenge was a race of autonomous vehicles through an urban environment organized by DARPA [3]. It took place at the former George Air Force Base in Victorville, California on November 3rd 2007. Vehicles had to drive a distance of 60 miles spread over three autonomous missions where no direct human intervention from the teams was allowed. Before the race 11 finalists were selected from 35 teams in a qualification event. During the competition, all of the vehicles were simultaneously on the course with 50 human driven cars. Vehicles had to interact with each other in various situations including (but not limited to) passing other vehicles, handling intersections, driving on urban roads or on parking lots. Six vehicles finished the challenge, three without human intervention: Team Tartan Racing (1st) [4], Team Stanford Racing (2nd) [5], and Team Victor Tango (3rd) [6].

Fig. 1 Boss - autonomous robot of Team Tartan Racing at the 2007 Urban Challenge.

The paper is organized as follows. Section 2 describes the specific challenges for development of an object tracking system for the Urban Challenge from the viewpoint of Team Tartan Racing. Section 3 describes the design decisions that were made for implementing the system. The architecture of the tracking system is

* Parts of this paper have first been published on German conferences (see [7][8]).

presented in section 4, including an overview of the sensor layout, and the software architecture. Finally in section 5 conclusions drawn from this work are presented.

2. KEY CHALLENGES

This section summarizes the key challenges for developing an object tracking system for an autonomous vehicle participating in the Urban Challenge.

Determine which objects participate actively in traffic – For a correct situation assessment (e.g. at an intersection or while driving down a lane, see e.g. [9]) Boss must be able to identify all vehicles around it which actively participate in traffic so that it can interact with them according to the traffic rules.

The physical appearance of vehicles is not well defined. For the Urban Challenge most vehicles had extensive customizations such as external sensors which differentiate them from stock vehicles of the same class. This makes is hard (if not impossible) to use a classification algorithm which is based only on static properties of an object to identify all vehicles in a scene (such as a classification algorithm based on image processing only).

Information such as if an object is currently moving can provide cues to help decide if an object participates actively in traffic or not. Movement characteristics, however, have to be determined from noisy sensor data. A particular challenge is considering vegetation close to road boundaries (e.g. bushes) where it may be difficult to determine if they are moving or not because of the ambiguity in their shape.

Even if all vehicles around the robot are identified, many ambiguous situations exist. As an example, how should a vehicle parked near, but not at a stop line be treated. The vehicle could be stalled, or just sloppily waiting for precedence at the intersection.

Track and predict the behaviour of observed objects – The higher level reasoning algorithms of Boss need information about the current states of observed vehicles around the robot. This includes not only their instantaneous velocity and position but also a decent estimate of their future state both on roads and in free movement zones (e.g. parking lots). The autonomous vehicle uses this information for its strategic decisions (e.g. merging into moving traffic, see e.g. [9]) and for calculating collision free trajectories (e.g. while driving on a road or through a parking lot, see e.g. [10]).

Provide information about the environment in multiple application scenarios **–** Boss needs to be able to deal with various application scenarios, like distance keeping, merging into moving traffic or intersection handling. The characteristics of these situations lead to different possible optimization strategies for dealing with uncertainties regarding the detection of obstacles around the robot.

For smooth distance keeping while driving down a lane very few false detections can be tolerated. Every erroneous object which is reported to the higher level algorithms by the tracking system can cause the robot to slow down or even stop unnecessarily. In this scenario, there is generally sufficient time to validate measurements to reject false positives. In contrast to distance keeping, while merging into moving traffic some false detections are acceptable in favor of minimizing false negatives, and causing an unsafe merge maneuver. Here the tracking system should report object as soon as possible accepting the probability of a false positive.

Deal with sensor variety **–** There is no single sensor that can provide all relevant data for driving in an urban environment. The main reason is the limited field of view of sensors, but additionally redundant sensors are required to deal with uncertainties regarding the detection of objects and the interpretation of sensor data. A surface that is hard to measure at range with a laser sensor may be readily detectable with a radar, and vice versa. Additionally individual sensor interpretation algorithms might misinterpret raw data and filter out information that is relevant (e.g. wrong suppression of noise).

Because of this the tracking system must be designed to be able to incorporate sensors that use different detection principles and interpretation algorithms without being inflexible regarding changes in the sensor configuration. Additionally the tracking system must be extensible with new sensors and sensor technologies since the development process often generates new insights. These insights may lead to a reconfiguration of the sensor system or the addition of new sensors to deal with the shortcomings of an existing configuration.

3. DESIGN PRINCIPLES

The following design principles guided the implementation of the tracking system for the Urban Challenge. The concepts helped to structure the software of the autonomous vehicle.

7.1 Classification of Vehicles

The tracking system does not classify objects as vehicles. Instead it puts out a list of dynamic obstacle hypotheses. All objects in this list are assumed to potentially move during the observation period and as such may be vehicles.

Every hypothesis is accompanied by a movement classification. The current movement is classified into *moving* or *not moving* respectively; the past moving is classified as either *observed moving* or *not observed moving*. A hypothesis is classified as *moving* once the tracking system decides that the object is currently not stopped. It is classified as *observed moving* once the tracking system decides that the object has changed its position over time.

The final decision of when a dynamic object hypothesis is interpreted as a vehicle which participates actively in traffic is left to the situation assessment algorithms encapsulated in the higher level reasoning system which is responsible for the behaviour of the robot. Thus the dynamic obstacle hypothesis list can be interpreted with respect to the current situation.

At an intersection, for example, all dynamic obstacle hypotheses (regardless of the movement state classification) are interpreted as vehicles which participate actively in traffic. This information is used to determine precedence order.

Alternatively, while driving down a lane, only object hypotheses which have the observed moving flag set and objects hypotheses which are close to the center of a lane are interpreted as vehicles which participate actively in traffic (for details see [11]).

In the intersection example the interpretation is conservative in order to reduce the risk of falsely taking precedence. Error recovery algorithms in the higher level reasoning system deal with cases where the situation is misunderstood by the robot (e.g. a static traffic cone is interpreted as a vehicle, see also [9]).

While driving down a lane the interpretation is less conservative allowing the robot to drive smoothly even with parked vehicles at the road boundaries. Again misunderstandings are handled by error recovery algorithms such as in the case where a traffic cone in the middle of a road is interpreted as stopped vehicle.

7.2 Dealing with detection uncertainties

The tracking system only puts out a particular dynamic object hypothesis as long as sensor data supports the existence of the hypothesis. In case no sensor data currently supports the hypothesis an object prediction only occurs for durations typical of sensor measurement dropouts (e.g. caused by sensor noise). In all other cases the object hypothesis is removed from the hypotheses list and the higher level reasoning algorithms (situation assessment algorithms) have to deal with the uncertainty explicitly.

This gives the robot the flexibility to react to object loss in a situationally dependent manner. Furthermore, the tracking system is separated from the situation assessment algorithms and can be developed independently.

If for example an object is not detected anymore while the robot waits for precedence at an intersection (e.g. caused by a sensor occlusion) the robot will wait for some amount of time to check if the object is detected again. During that time the previously determined precedence order is kept.

A different strategy with different timeouts is used while driving down a lane. Here the distance and the relative velocity of the vanishing object is used to determine if it makes more sense to slow down, stop or keep on driving smoothly (see also [11]).

7.3 Modeling Dynamic Objects

Two discrete models are used to model dynamic object hypothesis: a simple point model and a complex box model. The box model uses a fixed length and width to represent the shape of a vehicle. Estimated state variables are the position of the box, the velocity and acceleration in the longitudinal direction of the box, a yaw angle and a yaw rate. A reduced bicycle model is used for state propagation. The point model has no information about the shape of an object, only the position, velocity and acceleration in the 2D plane are estimated. A constant acceleration model with adaptive noise is used for state propagation (for details see [7]).

The models are switched depending on the currently available sensor information (see [7]). This allows using the more complex model whenever enough sensor information is available. The probability that enough information is available is directly influenced by the physical sensor setup on the robot, and how much redundancy is built into the configuration.

7.4 Extrapolation of observed vehicles

The extrapolation of dynamic object hypotheses is generally based on logical constrains defined by the road network. Only object hypotheses which are classified as moving and observed moving are extrapolated. A multi-hypothesis approach is taken. Future positions and velocities of object hypotheses are extrapolated based on the current position on the road and the estimated velocity. At every point where a driver has an obvious choice to change his action (e.g. intersections) multiple hypotheses are generated. In regions where no environment structures can be exploited for an extrapolation (e.g. open parking lots) a prediction is based only on state variables.

On roads this approach allows dealing with uncertainties in the estimated state variables in a robust way. Even if state variables could be estimated without any error, our approach generates a better prediction since a prediction based on state variables alone makes sense only for short periods of time. Human drivers use a similar model of other vehicles since if it could not be assumed that other vehicles behave at least to a certain degree according to the traffic rules (e.g stopping at a stop line, driving within a lane) smooth driving would be impossible.

In open areas like parking lots, the increased freedom of the autonomous vehicle allows it to deal with higher uncertainties in the prediction of observed objects (e.g. a large distance can be kept to other vehicles). If the sensor configuration is chosen appropriately (see section 4.1) the box model can be used in regions close to the robot for object tracking. This allows a sufficient accuracy for the prediction of observed vehicles so that the robot can avoid collision with other vehicles and drive safely (see [10]).

4. TRACKING SYSTEM ARCHITECTURE

The tracking system is a subsystem of the robot's overall perception system which also includes a static obstacle estimation module, a road estimation module and an instantaneous map estimation module (see also [10]). The static obstacle estimation module provides information about all obstacles in a scene that are assumed to never move during the observation period (see figure 4b), the instantaneous map provides untracked 3D information about objects around the robot (see figure 4f), and the road estimation module provides information about the roads around the robot (see figure 4a). Information from these modules is used within the tracking system (see section 4.2).

4.1 Sensor Configuration

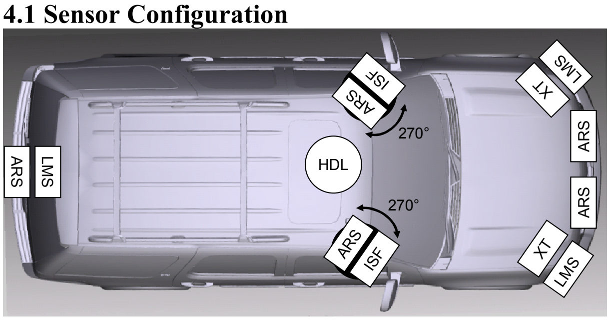

Fig. 2: Sensor Configuration for Object Tracking [8].

Table 1: Sensor Characteristics

Sensor	Sensor Type	Max. Range*	Vert. Angle	Horiz. Angle
Continental ARS300	Scanning Radar (near/far)	60/200m	4.3°	56°/18°
Continental ISF172	Fixed Beam Laser	150m	4°	14°
SICK LMS291	Scanning Laser, 1 level	80 m	0.25°	180°
IBEO AlascaXT	Scanning Laser, 4 level	200m	3.2°	240°
Velodyne HDL-64E	Scanning Laser, 64 beams	120m	26.8°	360°

*according to specification.

Figure 2 shows the sensor configuration which is installed on Boss for object tracking. All sensors run asynchronously. The combined field of view provides complete coverage around the robot. The sensors on the panheads are pointed with respect to the current driving situation (e.g. to the left and right at an intersection).

The two most important types of sensors on the vehicle for object tracking are the HDL-64E laser scanner and the ARS300 scanning radars. Due to the large vertical opening angle the HDL-64E is the only sensor on the vehicle that provides 3D information about objects. The opening angle of all other sensors on the robot is not large enough to reliably distinguish detections originating from the ground from detections originating from vehicles. This problem is generally due to rapid slope changes of the ground, which can occur in urban environments. The effective range of the HDL-64E in the configuration used on the robot is however not sufficient for all autonomous maneuvers, especially merging and passing maneuvers at 30mph.

The radars robustly detect objects in the near and far range. By using the Doppler shift the relative velocity of objects can be measured directly. This gives a low latency and high accuracy for velocity estimation and can be used to distinguish dynamic from static objects. Additionally, for all detections where an absolute velocity is measured it can be inferred that the measurement does not originate from the ground. This makes the system more robust against misinterpretations of sensor data.

The sensor setup is designed for redundancy regarding the detection principles, raw data interpretation and single sensor failures. On the panheads for example laser and radar technology is used to minimize the probability of not detecting an object during merge maneuvers. The software architecture allows the system to continue working even if single sensors stop working completely. In the front of the robot the sensor setup keeps the probability low that a real vehicle cannot be tracked with the complex a box model (see also [8]).

4.2 Software Architecture

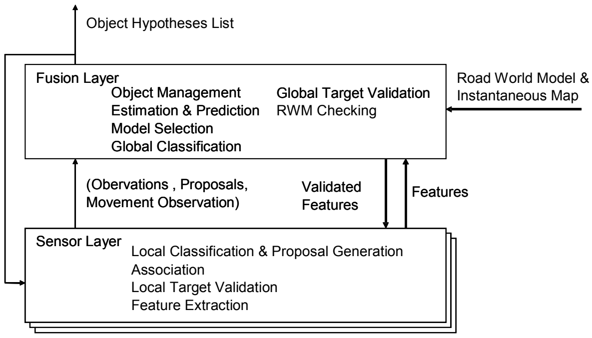

Fig. 3: Software Architecture for Object Tracking

The tracking system is divided into two layers: a sensor and a fusion layer. The sensor layer encapsulates sensor specific algorithms. For each type of sensor a sensor layer module is implemented. One instance per physical sensor device runs on the robot. The fusion layer is responsible for combining the data of the different sensors to the list of dynamic obstacle hypotheses. The following functionality is implemented inside the layers (see figure 3 and [11]).

- **Sensor Layer**
  - *Feature extraction* Features which potentially correspond to vehicles are extracted from the sensor raw data.
  - *Local Feature Validation*. Features are validated using a sensor specific heuristic to reject misinterpretations (e.g. ground detections).

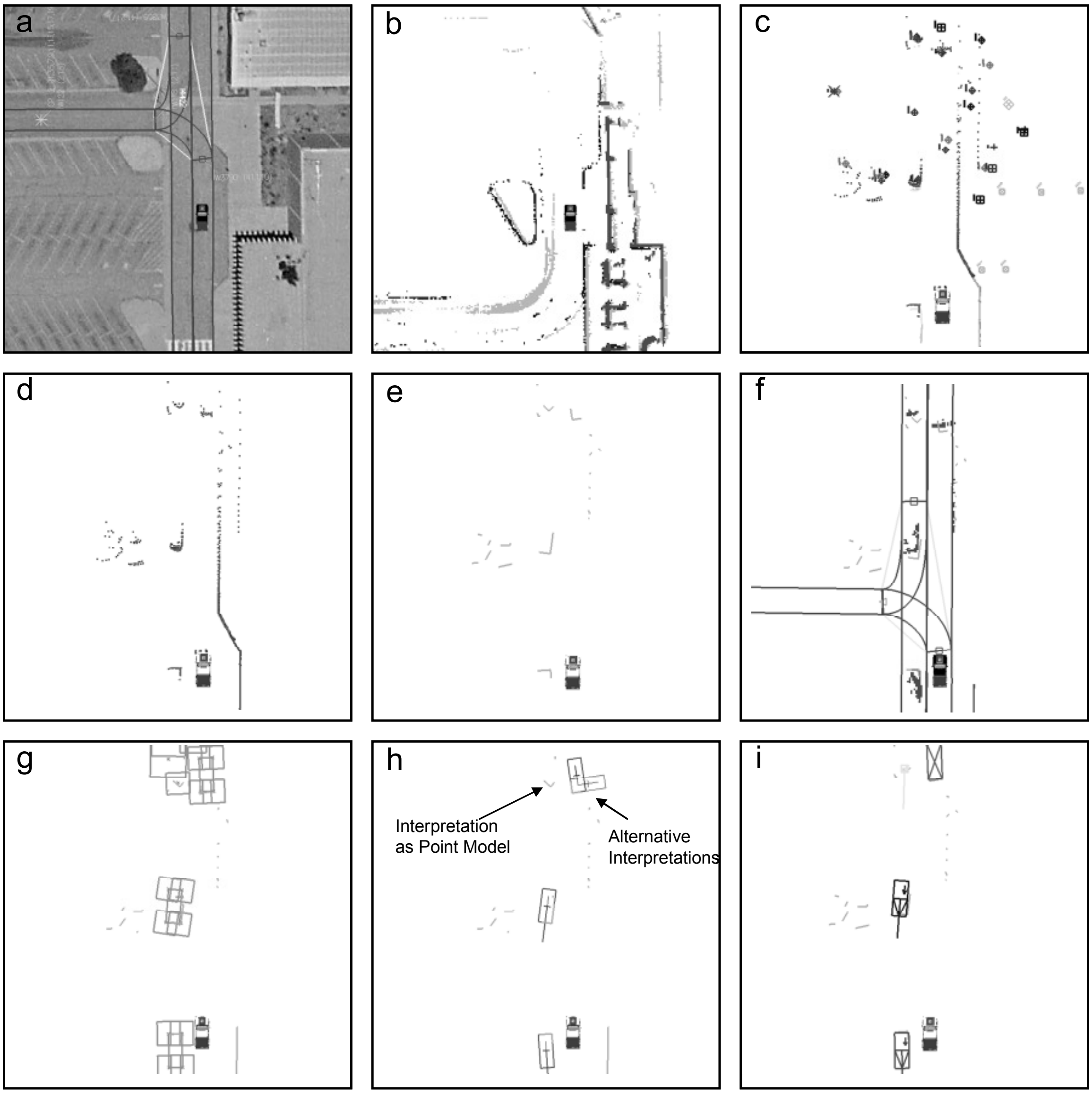

Fig. 4: a) Road Structure b) Static Map; Illustration of object tracking algorithm: c) Raw data (diamonds represent radar detections, dots originate from laser scanners) d) only laser scanner data e) features extracted from laser scanner data f) feature validation against road structure and instantaneous obstacle map g) data association h) interpretation of features and proposal generation i) Model selection and update of state estimation (see [8]).

- *Data Association*. Features are associated to object hypotheses by a sensor type specific algorithm (taking into consideration e.g. detection capabilities, potential false detections, sensor resolution, field of view).
- *Local Movement Classification.* Based on sensor specific data it is decided if an object moves or not (e.g. Doppler effect).
- *Proposal Generation.* New object hypotheses for unassociated features or as alternative to existing hypotheses are generated.
- *Observation Generation.* All information necessary to update the state estimation for an associated object hypothesis on fusion level is generated.

- **Fusion Layer**
  - *Global Feature Validation.* Features are validated using non-sensor specific algorithms (e.g. checks against the location relative to the road).
  - *Model Selection.* Based on the proposals the best tracking model is selected via a voting algorithm (see [8]).
  - *Estimation & Prediction/Extrapolation.* The state estimate is updated using available observations. The state are predicted: a) (short term) to the sensor measurement times for data association and b) (long term) for the higher level reasoning algorithms (e.g. collision avoidance).
  - *Global Movement Classification.* The movement state is classified using the local

movement classifications and the state estimates.

- *Object Management*. Object hypotheses are added or deleted from list

Figure 4 illustrates the tracking algorithm with data taken during the national qualification event of the Urban Challenge. The architecture allows adding new sensor types to the system with minimal changes to existing code. Adding more sensors of an already implemented sensor type requires no modifications to the source code at all. This makes the system extensible with little effort.

5. Conclusions

This paper presented the key challenges for implementing a tracking system for the Urban Challenge. The design principles which were used to cope with these challenges were explained and the tracking system which was built according to these design principles has been presented. The sensor setup which was used for object tracking during the Urban Challenge has been shown and an example which illustrates the tracking algorithm with real sensor data was given.

A key concept of the approach is the separation of the situation assessment algorithms from the tracking system. This simplifies the implementation of the overall system and allows optimizing the behaviour of the robot with respect to different situations without modifying the perception system. The behaviour of the robot can be adapted to the capabilities of the perception system.

The tracking system does not directly classify objects as vehicles. Instead it provides a list of dynamic object hypotheses which potentially correspond to vehicles. The situation assessment algorithms are responsible for interpreting this list with respect to the current situation.

The situation assessment algorithms are also responsible for dealing with uncertainties regarding the detection of vehicles. The perception system only provides information about objects which are currently detected by sensors and filters out only sensor specific short term effects caused for example by sensor noise.

The tracking system was used successfully at the 2007 Urban Challenge for various application scenarios. These included intersection handling, distance keeping, merging into moving traffic and driving on open parking lots. The concepts presented in this paper may be used to further improve Driver Assistance Systems which assist the driver in such situations.

ACKNOWLEDGEMENT

This work would not have been possible without the dedicated efforts of the Tartan Racing team and the generous support of our sponsors including General Motors, Caterpillar, and Continental. This work was further supported by DARPA under contract HR0011-06-C-0142.

REFERENCES

[1] C. Thorpe, R.C. Coulter, M. Hebert, T. Jochem, D. Langer, D. Pomerleau, J. Rosenblatt, W. Ross, and A. Stentz, "Smart Cars: The CMU Navlab", Proceedings of WORLD MED93, October, 1993.

[2] Bishop, R.: "Intelligent Vehicle Technology and Trends", Bosten, London: Artech House Publishers, ISBN 1580539114, 2005.

[3] DARPA Urban Challenge: www.darpa.org/grandchallenge, 2008.

[4] Team Tartan Racing: www.tartanracing.org, 2008.

[5] Team Stanford Racing: http://cs.stanford.edu/group/roadrunner/, 2008.

[6] Team Victor Tango: http://www.victortango.org/, 2008.

[7] 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.

[8] Darms, M., Rybski, P., Urmson, C.: "Vehicle Detection and Tracking for the Urban Challenge", 9th Symposium AAET 2008 - "Automatisierungs-, Assistenzsysteme und eingebettete Systeme für Transportmittel", Braunschweig, Feburary 13./14. 2008.

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

[10] Ferguson, D., Darms, M., Urmson, C., Kolski, S.: "Detection, Prediction, and Avoidance of Dynamic Obstacles in Urban Environments", Proceedings of the IEEE Intelligent Vehicles Symposium 2008, Eindhoven June 4-6, 2008.

[11] Darms, M., Rybski, P., Baker, C., Urmson, C.: "Obstacle Detection and Tracking for the Urban Challenge", to be published in IEEE Transactions on Intelligent Transportation Systems, Special Issue on the DARPA Urban Challenge Autonomous Vehicle Competition, 2008.