

Classification and Tracking of Dynamic Objects with Multiple Sensors for Autonomous Driving in Urban Environments

Michael Darms, Paul Rybski, Chris Urmson

Abstract—Future driver assistance systems are likely to use a multisensor approach with heterogeneous sensors for tracking dynamic objects around the vehicle. The quality and type of data available for a data fusion algorithm depends heavily on the sensors detecting an object. This article presents a general framework which allows the use sensor specific advantages while abstracting the specific details of a sensor. Different tracking models are used depending on the current set of sensors detecting the object. A sensor independent algorithm for classifying objects regarding their current and past movement state is presented. The described architecture and algorithms have been successfully implemented in Tartan Racing’s autonomous vehicle for the Urban Grand Challenge. Results are presented and discussed.

I. INTRODUCTION

Most of the existing commercial driver assistance systems with environmental perception are designed for longitudinal traffic in well structured environments (for example Adaptive Cruise Control [1]). Currently new driver assistance systems are being developed which work in more complex environments and use a multisensor fusion approach to process sensor data [2]. Examples are systems for intersection assistance or systems that assist drivers in construction sites on a highway.

One challenge that must be addressed is the distinction of static and moving obstacles in these environments. Some adaptive cruise control systems for example are designed to not react to objects which have not been detected moving by the system to avoid false reactions due to sensor artifacts (where artifacts are defined as erroneous detections or misinterpretations). On the other hand future driver assistance systems will be designed to work with static obstacles like poles or guardrails, too. Analogous to moving obstacle tracking [2], special algorithms exist to estimate the position and shape of static obstacles. To reduce artifacts with these algorithms it is important to distinguish static from dynamic obstacles, too [4].

In well structured environments like on a highway sensor artifacts are well understood and a distinction of static and

moving obstacles can be done reliably. In more complex environments like an urban setting new artifacts come into play. Here a multisensor approach can help to make the movement classification more robust using the available redundancy.

On the other hand more sensors lead to an increased complexity. Typical driver assistance systems today use a single environmental sensor and a tracking model tied to the characteristics of this sensor. When using heterogeneous sensors which are based on different sensor technologies the quality and type of the data now depends on the combination of sensors detecting an object [5][6]. Besides that, different sensors produce different kinds of misinterpretations of measurements, so called artifacts. These depend on sensor internals like the detection mechanism and feature extraction algorithms for example.

This article describes an architecture for sensor data fusion which allows to incorporate the information of different sensors in a generalized way. All sensor specific algorithms are encapsulated in sensor specific modules with a generalized interface. This reduces the effort needed to extend the system with new sensors or sensor technologies. Two sensor independent algorithms implemented in this architecture will be described which address the topics described above: An adaptive model switching approach which deals with the phenomenon of sensor dependent data quality, and a movement classification algorithm which robustly combines the information of different sensors to an overall movement classification.

The architecture and algorithms have been successfully implemented in Tartan Racing’s autonomous vehicle for the Urban Challenge 2007 . Data from thirteen environmental sensors with different detection modes has been fused into a model of composite object hypotheses. The data was used in a variety of applications including distance keeping, intersection handling and parking. The environment consisted of an urban road network with intersections, sharp curves and free traffic in open parking lots.

II. MULTISENSOR SETUP

A heterogeneous setup of thirteen different sensors were used on Tartan Racing’s autonomous vehicle. Fig. 1 shows the placement of these sensors and Table I lists the characteristics of those sensors.

The RADAR sensors enable the robot to detect the presence of objects with the sensitivity at up to a distance of 200m. The RADAR and LASER units mounted on the pan heads are used to detect cross traffic at intersections at a

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.

Passages about the model switching approach have been published on a local german conference "Steuerung und Regelung von Fahrzeugen und Motoren", 12./13. February 2008 in Baden-Baden [5].

M. Darms is with Continental Inc., Auburn Hills, MI, USA. Email: michael.darms@us.contiautomotive.com

P. Rybski is with Carnegie Mellon University, Pittsburgh, PA, USA. Email: prybski@cs.cmu.edu

C. Urmson is with Carnegie Mellon University, Pittsburgh, PA, USA. Email: curmson@ri.cmu.edu

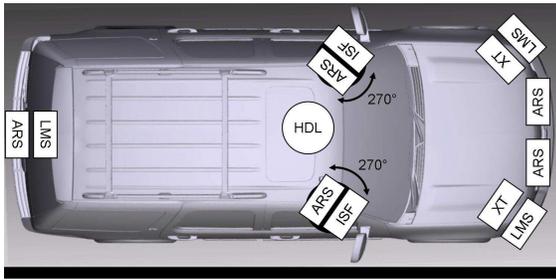


Fig. 1. Sensor setup on Tartan Racing’s autonomous vehicle [5].

TABLE I
SENSORS CHARACTERISTICS [5]

Sensor	Sensor Type	Max. Range*	Vertic. Angle*	Horiz. Angle*	Features used for tracking
Continental ARS300	Scanning Radar (near/far)	60m/200m	4.3°	56/18	2D coordinates of detection/velocity
Continental ISF172	Fixed Beam Laser	150m	4°	14°	2D coordinates of detection
SICK LMS291	Scanning Laser, 1 level	80 m	0.25°	180°	Edge Target/2D coordinates of detection
IBEO AlascaXT	Scanning Laser, 4 level	200m	3.2°	240°	Edge Target/2D coordinates of detection
Velodyne HDL-64E	Scanning Laser, 64 beams	120m	26.8°	360°	Edge Target/2D coordinates, height information target for validation

*according to specification

sufficient distance to merge safely with traffic moving at speeds up to 30mph. The RADARs measure Doppler shift which allows direct measure of velocity which is more accurate than having to estimate velocity from multiple sequential distance measurements like the LASER sensors. The features extracted from RADAR data are reported as a set of 2D position coordinates that reflect the center of the object along with an associated velocity of that point. The fixed beam LASER sensors only return a 2D point, no direct velocity measurement is possible.

The planar scanning LASER sensors return information about the (2D) shape and orientation of a vehicle in the near range [8]. The shape is computed from the intersection of the plane of the beam with the object. The returned shape is used to estimate the yaw angle and yaw rate of the detected vehicle. Due to the fixed angular resolution of the scanning lasers, the shape information at long distances is not good enough to perform yaw estimation with the required accuracy. As with the RADAR sensors the data however is accurate enough to associate the information to lanes on the road.

The 3D LASER scanner is the only sensor on the robot that provides information about the shape of the object in height as well as width and length. However, the effective detection range of the sensor is not sufficient for autonomous driving with merging and passing maneuvers in 30mph traffic. Furthermore, the top of the vehicle occludes some of the laser’s field of view and as such the laser cannot detect objects that are extremely close to the vehicle (including a large blind spot in the rear).

For the Urban Challenge no one sensor or sensor modality was sufficient to provide enough information to track objects

around the vehicle. The multiple sensors provided complete sensing coverage all around the vehicle and the combination of RADARs and LIDARs allowed for long-range detection of vehicles while still being able to estimate shape and orientation as the vehicle approaches the robot. The multiple sensors also provided redundancy in case of sensor failures and artifacts.

III. MODELING MOVING OBSTACLES

A. Tracking Model

Tracking algorithms like the Kalman filter use a model to describe the dynamic properties of an object hypothesis [3]. The detail of the tracking model is defined both by the needs of the application and the capabilities of the used sensors.

We propose an approach where the tracking model is switched according to the available sensor information. Two models were sufficient for all application scenarios of the Urban Challenge and serve as an example in the following part of the article: a box model and a point model.

The box model uses a fixed width w and length $l > w$ to represent the shape of a vehicle whereas the point model has no shape information. The box model uses the position of the center of the box x, y the yaw angle ϕ , yaw rate $\dot{\phi}$, a velocity v , and an acceleration a as state variables. The velocity and acceleration are always in the direction of the longer edge of length l . The state propagation equations are based on a simple bicycle model [7] which couple the x and y coordinates via the yaw angle and yaw rate.

The point model is described by 2 coordinates in the 2D plane x, y the corresponding velocities \dot{x}, \dot{y} and accelerations \ddot{x}, \ddot{y} . A constant acceleration model is used for state propagation [3]. The noise parameters adapt with the length and direction of the velocity vector (\dot{x}, \dot{y}) . This again couples the x and y coordinates and—similar to the bicycle model—does not constrain the model to a predefined direction defined by the coordinate system. This way the model is usable for tracking vehicles driving in an arbitrary and previously unknown direction.

B. Movement Classification

Some applications need reliable information if an object is potentially moving (like a vehicle) or if it is a static object (like a pole or traffic cone). For an autonomous vehicle like in the Urban Challenge this information is needed for the behavior algorithms of the system. These are responsible for the decisions of the vehicle, like safe driving through an intersection (“Is this object a parked vehicle or waiting for precedence?”). Driver assistance systems make use of this information, too. Examples are systems for intersection assistance or collision avoidance algorithms.

To get this information a detailed scene understanding is necessary. Even if a classification algorithm classifies an object as a vehicle, ambiguous cases exist such as if the vehicle is parked or just about to start moving.

As a simplification which was sufficient for the scenarios of the Urban Challenge, we propose an algorithm which classifies all object hypothesis into:

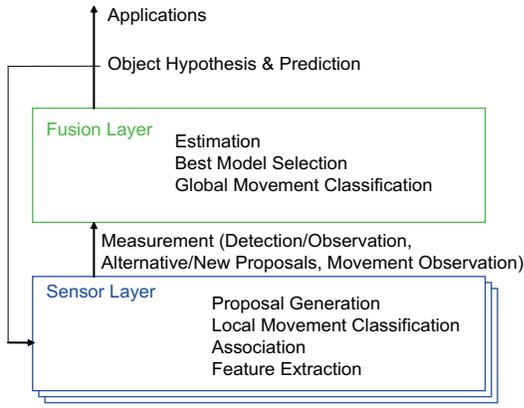


Fig. 2. Sensor Fusion Architecture [4], [5].

- The current movement state: Moving and Not Moving
- The past movement state: Observed Moving and Not Observed Moving

The current movement state is set to moving once the object tracking system decides that the object is not stopped. The past movement state is set to observed moving, once the object hypothesis has significantly changed its position. A similar classification is used in current adaptive cruise control implementations, which only react to objects, that have been observed moving in the past.

IV. SYSTEM ARCHITECTURE FOR OBJECT TRACKING

Fig. 2 shows the architecture of a sensor fusion system that implements the two approaches described in the previous sections (see also [5], [9]). It is divided into two layers, a Sensor Layer and a Fusion Layer. For each sensor type (e.g. RADAR, scanning LASER, etc.) a specialized sensor layer is implemented. For each sensor an instance of its particular sensor layer runs on the system. This way all sensor type specific operations are encapsulated in specialized modules. New sensor types can be added without changing existing sensor modules and the implementation of the fusion layer. This simplifies the extensibility of the system.

At the fusion layer all general functions for object tracking are performed. These include state estimation, best model selection, and movement classification (Fig. 3). State estimation is done with a probabilistic estimator using a prediction and an update step [5]. The algorithms for best model selection and the movement classification will be described in the following sections.

The current set of best object hypothesis is provided to the applications and is also fed back to the sensor layer. To be compatible with this decomposition the algorithms inside the fusion layer must have the following properties:

- Be independent of sensor types.
- Be independent of the number of sensors used in the tracking system.
- Be independent of the number of tracking models used in the tracking system.

All information about how states are propagated is encapsulated in the fusion layer, the state propagation equations

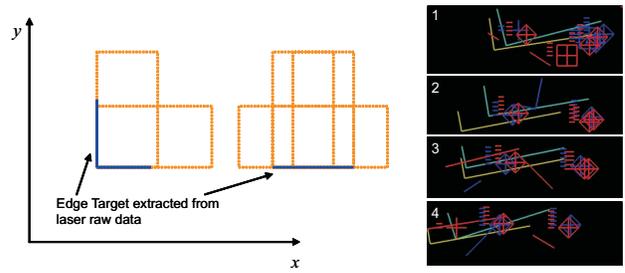


Fig. 3. Left: Possible Box Model interpretations of Edge Targets. Right: Snapshots of a vehicle driving perpendicular to the robot through an intersection. Edge Targets: Laser scanner features; Diamonds: Radar features [5].

are hidden to the sensor layer. An application specific filtering can be achieved this way [9], however this was not implemented for the Urban Grand Challenge.

Each time a sensor has new raw data it requests a prediction of the current best set of object hypothesis at the current measurement time and associates the raw data to these predicted object hypothesis [9]. For each extracted feature a set of possible interpretations is created by using a heuristic which takes the sensor specific characteristics of the raw data into account. Examples for these characteristics are the resolution of the sensor, the noise level of distance measurements, the maximum angle under which an object is detectable or a special treatment of the boundaries of the field of view.

Fig. 3 (right) shows edge targets which are extracted from the raw data of the scanning lasers (a heuristic for the planar lasers is described in [8]). Edge target features are "L" shaped features, which describe objects which either have two edges with a near 90 angle or objects where only one edge is visible. Fig. 3 (left) shows possible interpretations of edge targets as a box model. As there is a great deal of uncertainty in the edge target features all possible interpretations are generated. If the data is not sufficient to be interpreted as a box model (e.g. at larger distances or because the raw data does not represent a vehicle) a point object interpretation is used instead.

Based on a sensor specific heuristic a measure for the compatibility of the generated interpretations with the associated prediction is computed. Then it is checked if any of the interpretations differs significantly from the current tracking model used on the fusion layer.

If this is not the case then the best interpretation will be used to generate an observation. An observation holds all information necessary for the update step of the estimation at the fusion layer. It encapsulates the measurement equations and the information about the measurement noise. Analogous to the state propagation encapsulated in the fusion layer, all of the observation information is encapsulated in the sensor layer. Thus the algorithm which updates the state estimate at the fusion layer does not need to interpret the data from the sensor layer. This makes the fusion layer independent of the sensors modules implemented in the system.

If any of the interpretation differs significantly from the

prediction provided by the fusion layer the sensor initializes a new object hypothesis for each differing interpretation. Any of these new hypothesis can potentially replace the current model hypothesis used on the fusion layer. A set of hypothesis is called proposal. A proposal can be provided in addition to an observation or - if there is no interpretation compatible with the current best object hypothesis - without an observation. In this case the associated data is only called a detection to reflect the fact that the sensor detected the object, but cannot provide any meaningful information for the state estimation. For features which cannot be associated to any object hypothesis a sensor module provides an unassociated proposal per extracted feature with an ordinal ordering of the quality of the contained new object hypotheses.

In the fusion layer the best tracking model is selected based on the proposals provided from the different sensors and any other information available. The implementation used during the Urban Challenge uses information about road shape to bias the selection of the best proposal in on road scenarios. In parking lots scenarios the best proposal according to the ordinal ordering of the new object hypotheses is selected.

Depending on the sensor capabilities a sensor specific movement classification algorithm is implemented inside the sensor layer. The result of the sensor internal classification is a so called movement observation. This can be one out of three possibilities: *movement confirmation*, *no movement confirmation*, *no information available*.

A *movement confirmation* tells the fusion layer that the sensor assumes that the associated object hypothesis is currently moving based on the sensor internal judgement. In our configuration the RADAR module could measure the velocity via the Doppler shift. The movement of objects not moving perpendicular to the RADAR measurement direction can be detected reliably this way.

A *no movement confirmation* tells the fusion layer that the associated object hypothesis is currently not moving. As certain sensors can detect a movement only in a specific directions (like the RADAR sensor described above) the information is passed in form of a so called *no movement vector*, which is a unit vector pointing in the direction in which the sensor detected no movement. A special case is a null vector which can be used if based in the sensor information the object is not moving at all. This approach could be used with vehicle to vehicle communication available for example, however it was not used during the Urban Challenge.

Finally *no information available* tells the fusion layer that the sensor module cannot provide any information regarding the current movement state for the associated object.

V. MODEL SELECTION ALGORITHM

A heuristic is used inside the fusion layer to determine the best model for tracking [5]. It is only based on the proposals and observations from sensors which currently detect or observe a given object.

A sensor *supports* a model if by using only observations from this particular sensor the model is observable in princi-

ple. For instance, the planar and 3D LASER sensors support both the box and the point model while the RADAR and fixed beam LASER sensors can only support the point model.

A sensor *proposes* a particular model if it provides an proposal with this model as an alternative to the model currently in use by the fusion layer. The model may be different in either the state vector (e.g. a different yaw angle for a box model, for example); or it may be a different model entirely (e.g. a point model instead of a box model). A sensor will propose an entirely different model if it cannot support the current model at all (e.g. a RADAR sensor will always propose a point model, if the current model is a box model) or if the sensor does not support the current model based on the internally computed quality measure for the interpretation (e.g. a LASER sensor can propose a point model if the detected vehicle is at a distance where yaw cannot be estimated anymore).

A model is *currently supported* by a sensor if the sensor is currently directly observing the object with the model (e.g. LASER sensor observes a box model) or if the sensor is currently proposing the model as an alternative (e.g. LASER sensor proposes a box model while the point model is the current model). Before a proposal can be considered by the fusion layer, the sensor must continually propose that model for a pre-defined number of consecutive cycles. This helps to mitigate rapid model switching in the case where the sensor is receiving noisy returns.

Algorithm 1 Model selection algorithm

```

1:  $j = 0$ ,  $\text{modelPref}[0 \dots \text{numberOfModels}] = 0$ 
2: for all possible models  $i = 1 \dots \text{numberOfModels}$  do
3:    $\text{relSupport}[i] \leftarrow \text{numCurrentlySupportingSensors}/\text{numSupportingSensors}$ 
4:   if  $\text{relSupport}[i] \geq \text{minRelSupport}$  then
5:      $\text{modelPref}[i] \leftarrow \text{modelPref}[j] + 1$ 
6:      $j = i$ 
7:   end if
8: end for
9:  $\text{maxModelPref} \leftarrow \text{max}(\text{modelPref}[])$ 
10:  $\text{maxModelPrefIdx} \leftarrow \text{max arg}(\text{modelPref}[])$ 
11: if  $0 == \text{maxModelPref}$  then
12:   do nothing (no decision is possible, keep the current best model)
13:   exit the algorithm
14: else
15:    $\text{bestModel} \leftarrow \text{model}[\text{maxModelPrefIdx}]$ 
16: end if
17: if  $\text{bestModel} == \text{currentModel}$  then
18:    $\text{thresh} \leftarrow \text{floor}(\text{numSupportingSensors} * \text{thresholdReinit})$ 
19:   if  $\text{numProposingSensors} > \text{thresh}$  then
20:     reinitialize model (model is OK but states need to be reinitialized)
21:   else
22:     do nothing (may be a false alarm)
23:   end if
24: else
25:   change model to  $\text{bestModel}$ 
26: end if

```

In the current implementation the box model is the preferred model above the point model and the system will attempt to use that model whenever possible due to the additional information in conveys. Alg. 1 describes the decision algorithm [5]. Lines 1-16 of the algorithm decide which model type has the highest support by the sensors. By varying the value *minRelSupport* the point at which models are switched can be adjusted. A higher value ensures that a switch to model with a higher accuracy will be performed

only if there are enough sensors supporting it. In the Tartan Racing system the number of supporting sensors increases as the tracked vehicle gets closer to the robot. For example, at a range closer than 30m up to four sensors can support the box model, which helps to suppress artifacts as shown in Fig 4, right. Lines 17-26 of the algorithm determine if the model needs to be reinitialized. Here again a minimum number of sensors is needed to support the request for a reinitialization. The floor function ensures that not all sensors need to agree to a reinitialization unless *thresholdReinit* is set to 1.

VI. MOVEMENT CLASSIFICATION

A. Moving vs. Not Moving

The following heuristic inside the fusion layer combines the movement observations from all sensors detecting an object to determine the current movement state.

First how often an object has been confirmed moving is determined by polling all sensors. If the total number is above a threshold th_{moving} then the object is called *potentially moving*. As all sensors are counted the time that is necessary to get to the result *potentially moving* is dependent on the number of sensors currently detecting the object. If only one sensor detects the object th_{moving} consecutive confirmations are needed.

If the object is not classified as *potentially moving* based on the movement confirmations a statistical test on the velocity estimation is performed. It is checked if the hypothesis H_0 : "Absolute velocity is smaller than v_{min} ." can be rejected. If this is the case, then the object is classified as *potentially moving*.

The statistical test can always be performed as it is based on the state variables only. This is the case for situations where none of the sensors has the capability to perform a sensor based movement confirmation as described above. In our system this was true for the LASER sensors as no sensor specific algorithm was implemented. There may however be ways, to exploit information included in the raw data to detect a movement.

To get robust results the significance level α of the test and v_{min} can be tuned. With a higher v_{min} and a lower α the test becomes more conservative. Lowering α leads to longer times until enough data is available to reject H_0 , increasing v_{min} leads to less false positives at low speeds, however slow objects are not classified correctly. Hardware wise the test can be influenced by increasing the number of sensors detecting an object or with a higher accuracy of a single sensor.

If an object is classified as *potentially moving* a cross check against the no movement vectors provided by the sensors is performed. For this the dot product between the normalized current velocity estimate and the no movement vector is build and checked against a threshold. If the result from any of the dot products reveals that the object is not moving then the object is classified as not moving, otherwise it is classified as moving.

If the cross check can be performed depends on the sensors detecting the object. In our system the cross check could only

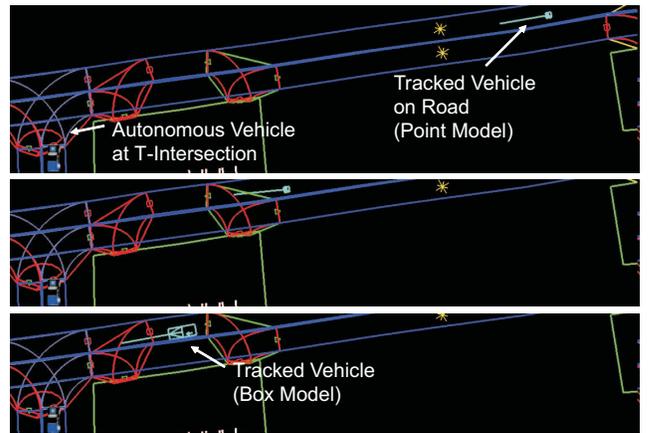


Fig. 4. The model is switched from point to box once the vehicle is detected with the scanning LASER sensors [5].

be performed when at least one RADAR sensor detected the object. In all other cases only the result of the statistical test can be used to determine the movement state.

B. Observed Moving vs. Not Observed Moving

To determine if an object should be classified as *observed moving* the following heuristic works robustly. First the distance traveled is estimated from the last time the object has been classified as *not observed moving* is evaluated. If this distance is above a threshold it is checked if the object is classified as *moving* based on the movement observations. If this is true, then the object is classified as *observed moving*. If it is not true then the object has to be classified as *moving* for a time period $t_{min,1}^{obs}$ to be classified as *observed moving*.

The mandatory check against the distance traveled increased the robustness against short term artifacts, especially during the initialization phase of the filters. It is possible to not perform this check for objects which are confirmed moving by the sensors alone, however this did not increase the performance significantly, so that the more conservative approach makes sense.

Because of artifacts and noise every approach will lead to wrong classifications in certain cases. This implies that there has to be a way to revise the current decision regarding the *observed moving* classification. The following heuristic takes short term artifact into account. The *observed moving* decision is only kept if the object is classified as moving for a time period $t_{min,2}^{obs} > t_{min,1}^{obs}$. Once this time period is over the observed moving flag is kept for $t_{max}^{obs} > t_{min,2}^{obs}$. As a result a wrong *observed moving* classification is kept for a maximum of t_{max}^{obs} .

VII. RESULTS

Fig. 4 shows a vehicle driving up to an intersection with the autonomous vehicle waiting for precedence. At a distance of more than 150m only the pointed RADAR sensor detects the approaching vehicle—the point model is used for tracking. The adaptation of the noise with respect to the velocity vector stabilizes velocity estimation in the direction of travel. As soon as the vehicle is close enough for the

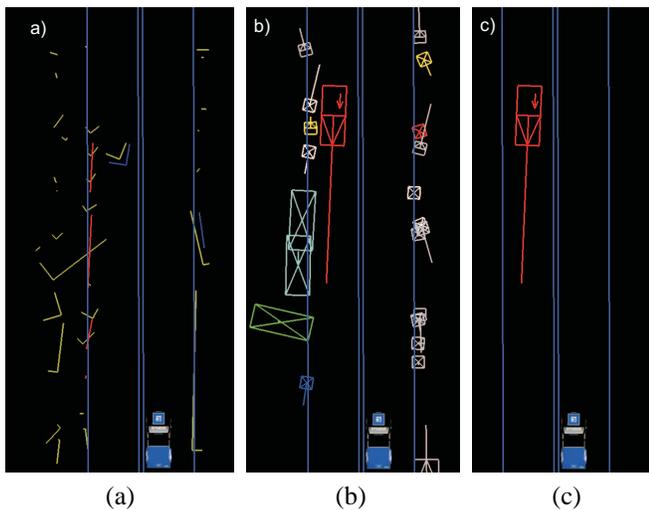


Fig. 5. (a) Edge Targets. The targets left and right to the road originate from bushes. (b) Object hypotheses. (c) Only object hypotheses with the observed moving flag are shown.

LASER sensors to generate box model proposals the tracking model is changed. The RADAR sensors still provide accurate velocity measurements which allow a precise estimation of the time gap for merging—the position measurements however are represented only with a very low weight in the observation. Due to the information provided by the LASER sensors the yaw angle of the object can now be estimated.

In open parking lots the sensor configuration can generate a box model with sufficient accuracy to predict the movement of a tracked vehicle for up to three seconds based on estimated states only. This makes the robot able to drive in an open parking lot together with other vehicles—human or robot driven.

The distinction of static and moving obstacles was a key concept of the perception system of Tartan Racing's autonomous vehicle for the Urban Challenge 2007 [4]. Fig. 5 (a) shows features extracted from LASER scanner data on a road which has dense bushes and a fence next to the road boundaries. Fig. 5 (b) shows the according object hypothesis and Fig. 5 (c) only object hypothesis which are classified as observed moving. Only the vehicle driving down the road is classified as *observed moving*.

In environments encountered during the Urban Challenge the approach allowed smooth autonomous driving. The configuration used on the robot however was a compromise between the delay to classify an object hypothesis as moving, the minimum classification velocity and the suppression of false alarms. In areas like shown in Fig. 5 (a) not all false alarms for moving obstacles could be suppressed. Additional information about the location of obstacles relative to the road was used to bias the decision and reduce the number of false alarms to nearly zero.

VIII. CONCLUSIONS AND FUTURE WORK

In a data fusion system with heterogeneous sensors the sensors contribute different levels of information. To make

the system extensible and flexible for changes an architecture and algorithms have to be found which allow an abstraction of the sensors.

In this paper we proposed an architecture which encapsulates all sensor specific algorithms in a sensor layer and sensor independent algorithms in a fusion layer. The architecture allowed an efficient development of our obstacle tracking system for the Urban Challenge.

For object tracking we proposed an adaptive model switching approach, where the tracking model is selected based on the available sensor information. The selection is based on votes from sensors detecting the object and is independent of the underlying sensors and tracking models. The practical realization showed that the approach works robustly for a combination of RADAR and LASER sensors (fixed beam and scanning).

For classifying the movement state of detected dynamic obstacles a sensor independent algorithm has been presented, which combines sensor specific movement observations and a sensor independent hypothesis test into a robust classification of the movement state. We showed results of the algorithm which allow smooth autonomous driving of our robot.

The feature extraction algorithms used for the LASER sensors searched for "L" shapes only. In areas with bushes artifacts caused by this approach can lead to false positives regarding the movement classification. Future work will include the development of more sophisticated feature extraction algorithms which are able to reject this data in an early stage. The additional use of information about the road already showed very good results.

REFERENCES

- [1] Winner, H.: Adaptive Cruise Control. In Jurgen, R. K. (Hrsg.): Automotive Electronics Handbook. 2. Auflage. New York, London: McGraw-Hill, 1999, 30.1-30.30.
- [2] Bishop, R.: "Intelligent Vehicle Technology and Trends", Boston, London: Artech House Publishers, 2005.
- [3] Bar-Shalom, Y.; Li, X.-R.: "Multitarget multisensor tracking - principles and techniques", YBS, 1995, ISBN 0964831201.
- [4] Darms, M., Rybski, P., Urmson, C.: "Vehicle Detection and Tracking for the Urban Grand Challenge", AAET - Automatisierungs-, Assistenz- und eingebettete Systeme fr Transportmittel, Symposium 13./14. Feb 2008, Braunschweig, 2008.
- [5] 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, Dsseldorf: VDI-Verlag, 2008.
- [6] Stker, D.: Heterogene Sensordatenfusion zur robusten Objektverfolgung im automobilen Straenverkehr. Oldenburg, Univ., Diss. (Online Publication), 2004 URL: <http://deposit.d-nb.de/cgi-bin/dokserv?idn=972494464i>.
- [7] Kaempchen, N. et al.: "IMM object tracking for high dynamic driving maneuvers", IEEE Intelligent Vehicles Symposium 2004, 825 - 830.
- [8] MacLachlan R.: "Tracking Moving Objects From a Moving Vehicle Using a Laser Scanner", tech. report CMU-RI-TR-05-07, Carnegie Mellon University, June, 2005.
- [9] Darms, M.; Winner, H. [Modular System Architecture]: A modular system architecture for sensor data processing of ADAS applications. In Intelligent Vehicles Symposium, 2005. Proceedings. IEEE. Las Vegas, USA, 6.-8. Juni 2005 2005, 729 - 734