

# **An Adaptive Model Switching Approach for a Multisensor Tracking System used for Autonomous Driving in an Urban Environment**

## **The Approach used by Tartan Racing in the Urban Grand Challenge**

**Dr.-Ing. Michael Darms**, Continental AG, Pittsburgh, PA (USA)

**Paul Rybski**, PhD, Carnegie Mellon University, Pittsburgh, PA (USA)

**Chris Urmson**, PhD, Carnegie Mellon University, Pittsburgh, PA (USA)

### **Abstract**

For most of the existing commercial driver assistance systems the use of a single environmental sensor and a tracking model tied to the characteristics of this sensor is sufficient. When using a multi-sensor fusion approach with heterogeneous sensors the information available for tracking depends on the sensors detecting the object. This paper describes an approach where multiple models are used for tracking moving objects. The best model for tracking is chosen based on the available sensor information. The architecture of the tracking system along with the tracking models and algorithm for model selection are presented. The design of the architecture and algorithms allows an extension of the system with new sensors and tracking models without changing existing software. The approach was implemented and successfully used in Tartan Racing's autonomous vehicle for the Urban Grand Challenge. The advantages of the multisensor approach are explained and practical results of a representative scenario are presented.

### **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]). Thus the use of a single environmental sensor and a tracking model tied to the characteristics of this sensor is sufficient. New driver assistance systems will use a multi-sensor fusion approach to process sensor data [2]. Data generated from different sensor technologies will be combined, so that depending on the number and type of sensors detecting an object the quality of the data will change. Even when only a single sensor is used, the quality of the data can change. As an example, the shape information extracted from laser scanner data will change with the distance to a detected object.

This article describes an adaptive model switching approach to address this phenomenon. It has been successfully implemented in Tartan Racing's autonomous robot [3][4] for the Urban Challenge 2007 [5]. 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.

The first section of this paper describes the sensors used for object tracking on Tartan Racing's autonomous vehicle. The advantages of the multisensor approach with different sensor technologies will be described along with the differences in features and data quality obtained from the different sensors. The next section explains the architecture of the tracking system. The implemented system on the robot uses two distinct models for object tracking, which are explained next. The selection of these models depends on the information available from each sensor. A voting scheme where individual sensors vote on the best tracking model is used to determine the model that best fits the available data. The algorithm to determine the best model is also presented. The paper concludes with results in a representative scenario.

### Multisensor Setup for Object Tracking

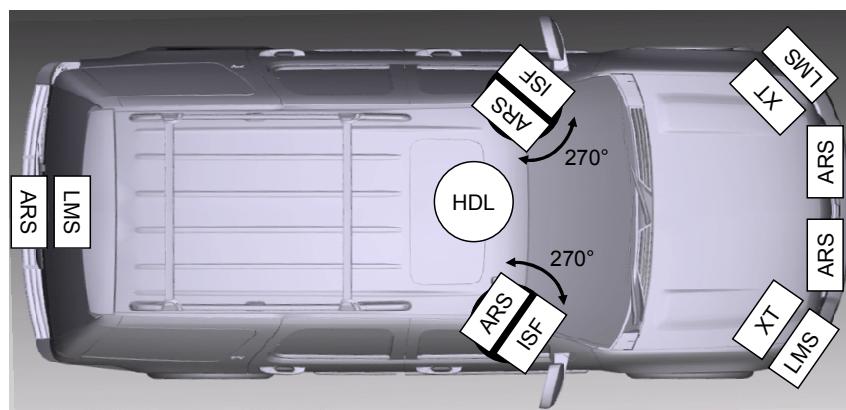


Fig. 1: Environment sensor setup used for object tracking on Tartan Racing's autonomous robot.

Figure 1 shows the sensor configuration that is used on Tartan Racing's autonomous vehicle. Table 1 summarizes the characteristics of the thirteen sensors (for a general overview see for example [5]).

The radar sensors enable the robot to detect objects reliably at near and far distances. The units mounted on the pan heads are used to detect cross traffic at intersections early enough

to merge safely with traffic moving at speeds up to 30mph. The radars are able to measure Doppler shift which allows direct measure of velocity and gives a higher precision and lower latency than an estimate based on distance measurements as it is required with the laser sensors. This direct velocity measurement can also be used to distinguish moving and static objects more robustly than with sensors measuring distances only. In the configuration used on the robot the information extracted from the sensors raw data consisted of 2D coordinates of the detection center and the velocity in radial sensor direction. The accuracy of the measured position allows an association of the detected vehicle to the lane it travels in up to the detection range of 200m.

The planar scanning laser sensors provide information about the planar (2D) shape and orientation of a vehicle in the near range (see e.g. [6][9]). This information can be used to predict the movement of a tracked vehicle including an estimated yaw angle and rate which is not possible with the measurements from the radars alone (see the next section for details) The information is precise enough to calculate a path for the autonomous robot close to other moving traffic while keeping a reasonable safety distance (e.g. oncoming traffic on roads or free traffic in an open parking lot). 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 height in addition to the planar shape of objects. In the configuration used on the vehicle the effective detection range of the sensor is not sufficient for autonomous driving with merging and passing maneuvers in 30mph traffic. Furthermore, due to the mounting position the sensor cannot acquire measurements in various occluded regions close to the vehicle (especially the rear). Analogous to the planar laser scanners the resolution of the data points decreases with distance and direct measurement of velocities is not possible.

From this description, it is clear that no one of the sensors alone was sufficient to provide enough information to track objects around the vehicle. By fusing the fields of view of each of the different sensors, complete sensor coverage around the vehicle could be achieved and the described advantages of the different sensor technologies could be combined. Moreover redundancy makes the system more robust to possible sensor failures and artefacts of single sensors.

Table 1: Sensor Characteristics and features extracted from sensor raw data.

Sensor	Sensor Type	Max. Range*	Vertical Angle	Horizontal Angle	Features used for tracking
Continental ARS300	Scanning Radar (near/far)	60/200m	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

### Tracking Architecture

Each sensor produces a different kind, amount, and quality of information. To handle this disparity, different tracking modes are used. Depending on the available information the tracking model is switched to the model with the highest precision currently supported by sensor data. For the applications scenarios of the Urban Challenge two tracking models were sufficient: a box model and a point model (see Figure 2).

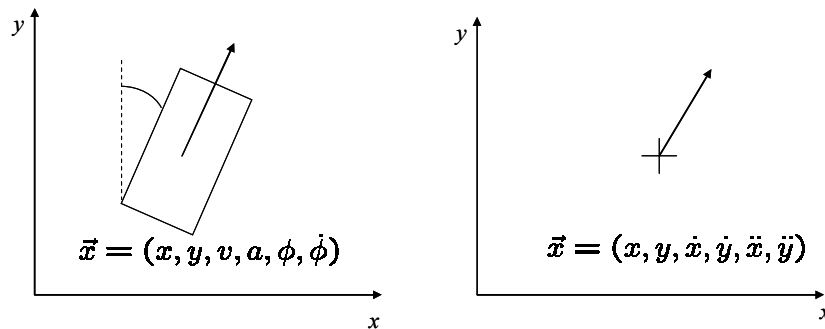


Fig. 2: Tracking models: box model with fixed shape (left) and point model without shape information (right).

The box model uses a fixed length and width to represent the shape of a vehicle whereas the point model has no shape information. For the box model the velocity and acceleration vector is always parallel to the longer edge. The orientation is described by a yaw angle and a yaw rate. The state propagation equations couple the x and y coordinates via the yaw angle and yaw rate (simple bicycle model, see e.g. [6]). The point model is described by 2 coordinates in the 2D plane and the corresponding velocities and accelerations. A constant acceleration model is used for state propagation (see e.g. [7]). The noise parameters adapt with the

length and direction of the velocity vector. 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.

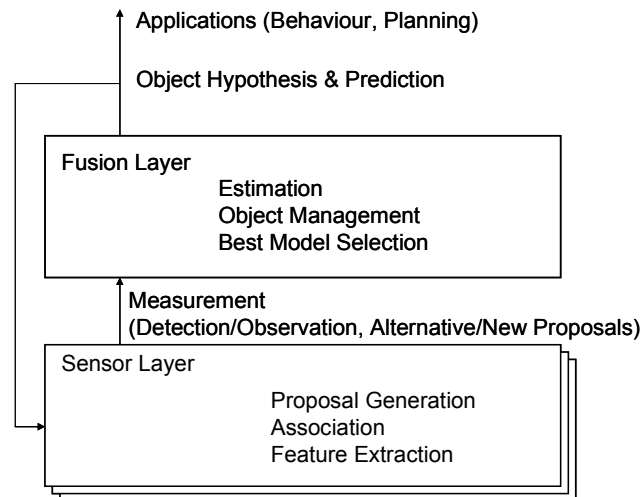


Fig 3: Simplified architecture of the tracking system.

Figure 3 shows a simplified architecture of the tracking system (see also [8]). It is divided into two layers, the *Sensor Layer* and the *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. The most important are state estimation, object management and model selection (see Figure 3 and [8]). State estimation is done with a probabilistic estimator using a prediction and an update step (see e.g. [8]). The current set of best object hypothesis is provided to the applications (behavior and planning algorithms) and is also fed back to the sensor layer. To be compatible with this decomposition the tracking algorithm in 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.

By maintaining these requirements, the independence of the fusion layer can be guaranteed and thus the tracking system can be easily extended to new sensors. Details of the fusion layer are explained in the following section.

All information about how states are propagated is encapsulated in the fusion layer, the state propagation equations are hidden to the sensor layer. 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 (see also [8]). 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.

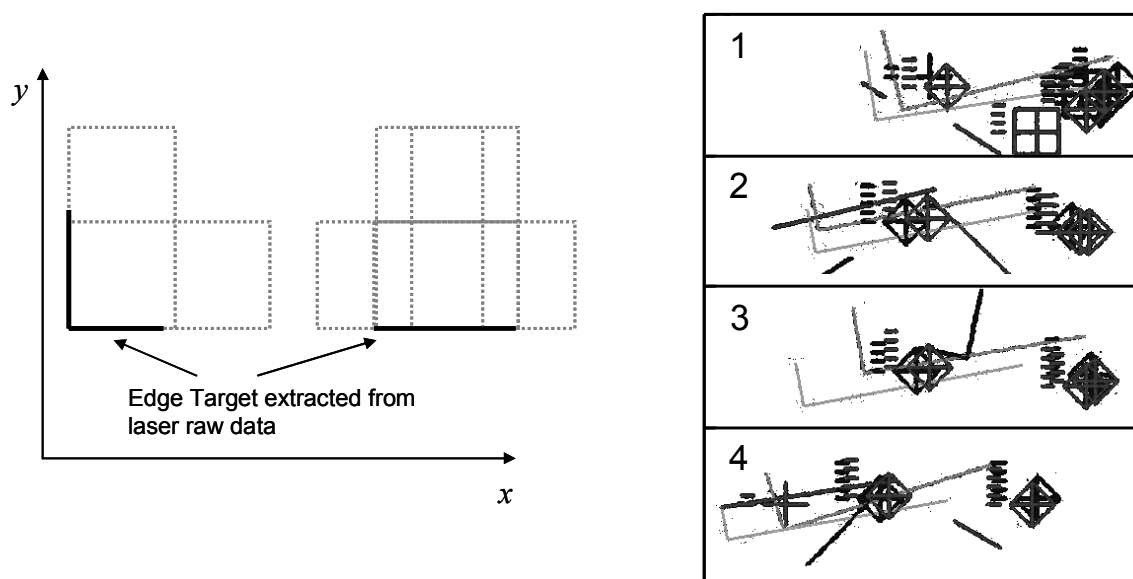


Fig 4: Left: Possible Box Model interpretations of Edge Targets. Right: Snapshots of a vehicle driving perpendicular to the robot through an intersection. Edge Targets: Laser scanner features; Diamonds: Radar features. Artefacts are mainly caused by ground detections due to pitch of the robot and shape of the ground (Displayed features are not synchronized - up to 100ms difference per snap shot).

Figure 4 shows *Edge Targets* which are extracted from the raw data of the scanning lasers (a heuristic for the planar lasers is described in [9]). Edge target features describe objects which either have two edges with a near  $90^\circ$  angle or objects where only one edge is visible. Figure 4 (left) shows possible interpretations of edge targets as a box model. As there is a great deal of uncertainty in the edge target features (see Fig 4, right) 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 and it is checked if any of the interpretations differs significantly from the current tracking model used on fusion layer.

If this is not the case then the best interpretation will be used to generate an observation. The 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 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 sensor layer. This makes the fusion layer independent of the sensors modules implemented in the system.

If an interpretation differs significantly from the prediction provided by the fusion layer the sensor initializes a new object hypothesis. Each of these new hypotheses can potentially replace the current model hypothesis used on fusion layer. The set of hypothesis is called proposal. Proposals 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 a set of unassociated proposals per extracted feature with an ordinal ordering of the quality of the proposals.

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 on roads. In parking lots the best proposal according to the ordinal ordering is selected.

### **Voting Algorithm for Model Selection**

The algorithm to select the best tracking model only depends on the proposals and observations from sensors which currently detect or observe a given object. A sensor counts as *supporting* a model if by using only observations from this particular sensor the model is observable. With the sensor configuration described in the first section the laser scanners (planar and 3D) support both, the box and the point model, the radar and fixed beam laser sensors can support only the point model.

A model counts as *currently supported* by a sensor if the sensor observes it directly or proposes the model as an alternative. To make the algorithm less sensitive to single false alarms of single sensors a minimum number of consecutive cycles with proposals for the specific model type can be defined before the proposal of a particular sensor is actually counted.

A sensor counts as *proposing* a model if it proposes this model as an alternative to the current model in use in the fusion layer. The model may be different in either the state estimation (e.g. wrong yaw angle in the vehicle model for example); or it may be a different model, if either the sensor cannot support the current model (e.g. a radar sensor and box model) or it does not support the current model based on the internally computed quality measure for the interpretation (e.g. a laser sensor and box model: detected vehicle in a distance where the yaw estimation is not meaningful anymore).

Finally a preferential ranking of the different models has to be defined. In the current implementation the box model is the preferred model above the point model. This order may change with the addition of new models however the algorithm itself does not need to change. The following pseudo code describes the decision algorithm:

```
compute for each possible model relSupport=currentSupport/numberOfSupportingSensors
If for all possible models relSupport<minRelSupport
    do nothing (no decision is possible, keep the current best model)
else
    bestModel is the model with highest preference && relSupport>=minRelSupport

if bestModel==currentModel
    if numberOfProposingSensors>floor(numberOfSupportingSensors*thresholdReinit)
        reinitialize model (model is OK but states need to be reinitialized)
    else
        do nothing (may be a false alarm)
else
    change model
```

The first portion of the algorithm decides 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 artefacts as shown in Fig 4, right.



The second part of the algorithm determines 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.

## Results

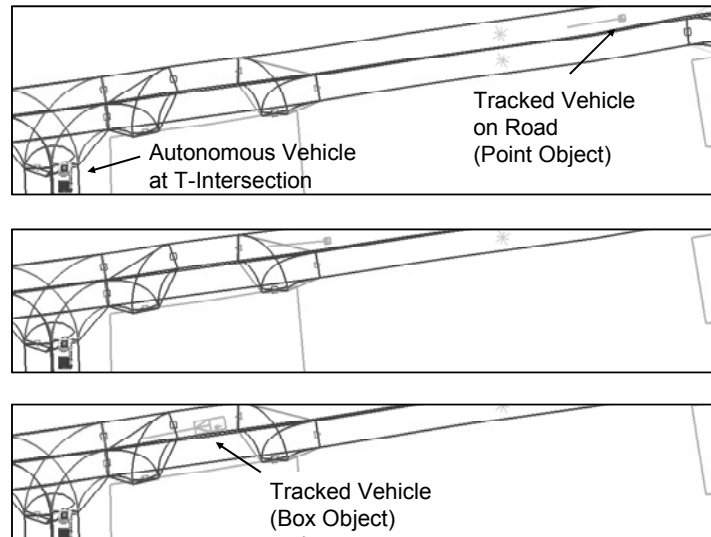


Fig. 5: Autonomous vehicle waiting for precedence. The model is switched from point to box once the vehicle is detected with the scanning laser sensors.

Figure 5 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 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.

## Conclusions

When fusing data from heterogeneous sensors for tracking moving objects a single tracking model cannot reflect the different levels of information provided by the sensors. Depending

on the resolution of a sensor for example it is possible to estimate a yaw angle of a tracked vehicle (scanning laser) or not (fixed beam laser). A way to make use of the different sensor capabilities is to use different tracking models which are switched according to the available information. In this paper an architecture has been presented which incorporates this approach. The architecture encapsulates all sensor specific algorithms in a *Sensor Layer* and sensor independent algorithms in a *Fusion Layer*. This makes it possible to add new sensors to the system without changing existing code. It has been shown that two tracking models are sufficient to track vehicles in an urban environment as specified for the Urban Challenge. An algorithm which selects the tracking model has been presented. 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).

### **Acknowledgements**

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.

- [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] Press Release CMU: "Carnegie Mellon Tartan Racing Wins \$2 Million DARPA UGC", [http://www.cmu.edu/news/archive/2007/November/nov4\\_tartanracingwins.shtml](http://www.cmu.edu/news/archive/2007/November/nov4_tartanracingwins.shtml), 2007.
- [4] Website Team Tartan Racing: [www.tartanracing.org](http://www.tartanracing.org), 2007.
- [5] Website DARPA Urban Challenge: [www.darpa.org/grandchallenge](http://www.darpa.org/grandchallenge), 2007.
- [6] Dietmayer, K. et al. „Fusionsarchitekturen zur Umfeldwahrnehmung für Zukünftige Fahrerassistenzsysteme. In Mauerer, M. (Hrsg.): Fahrerassistenzsysteme mit maschineller Wahrnehmung. New York: Springer, 2005, 59–87.
- [7] Kaempchen, N. et al.: "IMM object tracking for high dynamic driving maneuvers.", IEEE Intelligent Vehicles Symposium 2004, 825 – 830.
- [8] Bar-Shalom, Y.; Li, X.-R.; Kirubarajan, T.: Estimation with applications to tracking and navigation - theory, algorithms and software. New York, NY [u.a.]: Wiley, 2001.
- [9] Darms, M.: „Eine Basis-Systemarchitektur zur Sensordatenfusion von Umfoldsensoren für Fahrerassistenzsysteme“, Fortschritt. VDI: R12, Nr. 653, 2007.
- [10] 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.