JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 4, December 2007

An Expandable Multi-Sensor Data-Fusion Concept for Autonomous Driving in Urban Environments

J. Effertz*
Technische Universität Braunschweig, Braunschweig, Germany

DOI: 10.2514/1.33934

We introduce a multi-target, multi-sensor data fusion concept for automotive applications in urban environments. Based on a sensor system consisting of laser scanners, radar and LIDAR sensors, a system architecture has been created which is capable of tracking objects of arbitrary contours. The fusion system is independent from the sensor types used and can be set up in a distributed architecture based on Ethernet communication. The classical Kalman filter algorithm has been extended to realize contour-functionality, removing the necessity to define a static object reference point. A specialized data association and pretracking stage effectively uses intentional sensor redundancy in order to suppress false alarms and increase track quality. The fusion system is implemented in the 2007 DARPA Urban Challenge vehicle of Team CarOLO, *Technische Universität Braunschweig*.

Nomenclature

s	scalar value
$\underline{v}$	vector
$\underline{\underline{M}}$	matrix
$\underline{x}$	track state vector
$\underline{y}$	sensor measurement vector
$\underline{p}$	position vector
$\underline{v}$	velocity vector
v	generalized time variable

I. Introduction

RELIABLE artificial perception of a vehicle's environment is one of the key issues in autonomous driving. In the past years, the field of autonomous mobile robots has spread from rather simple indoor applications to more complex environments. As a result, requirements for range and maneuvering speed for robotic perception have been increasing steadily. Recently, a variety of driver-assistance systems have been introduced into the automotive market, e.g. assisting the human driver in lane keeping, adapting the vehicle's speed to the preceding vehicles or even giving assistance in emergency breaking situations. Those systems are primarily targeting highway driving situations and using rather simple representations for the vehicle's surroundings. The complexity of urban environments however calls for a more sophisticated artificial view, since a variety of target types, static and dynamic are likely to be seen in an inner-city situation. So far, there is no single sensor technology available covering all necessities regarding range, view area, detection speed and reliability for operation in a rugged automotive environment. This makes it necessary to combine different sensor types in order to take advantage of each sensor technology's individual strength.

Received 13 August 2007; revision received 9 October 2007; accepted for publication 11 October 2007.
* Research Associate, Institute of Control Engineering, Hans-Sommer-Str. 66, 38106, Braunschweig, Germany

The combination of different sensors sharing the same view area brings along the need for sensor data fusion, delivering a single consistent artificial view of the real world. Regarding post-processing and fusion of sensor data, there are two main approaches common in the field of autonomous robots: object oriented and grid based.

With an object-oriented approach,[1] the environment is described by discrete objects carrying an assigned state vector, e.g. object position, velocity, acceleration or other trackable features. In order to associate incoming sensor readings with these tracked objects, it is necessary to reduce incoming raw measurement data to a compatible, object-oriented data description, which is typically done within sensor data preprocessing. A state estimator is then used to update known targets of the artificial surrounding with incoming sensor readings.

The grid-based approach[2] on the other hand discretizes the robot's environment into distinct grid cells while assigning certain features to each cell, such as probability of occupancy (occupancy grids). A sensor model is needed to assign feature updates for each grid cell based on raw measurement data of incorporated sensors such as range readings of a sonar sensor. Grid based approaches actively model the entire view area including regions free of targets, which leads to a richer view of the world but at a price of higher computational complexity.

This contribution discusses the sensor system, system architecture and data fusion concept of the 2007 Urban Challenge vehicle (further referred to as "Caroline") designed by the CarOLO-Team of the *Technische Universität Braunschweig*. The vehicle has successfully passed the DARPA Site Visit in June 2007 and will be part of the Urban Challenge National Qualification Event in Victorville, CA. While Caroline's overall perceptive system is based on a hybrid structure consisting of object and grid based approaches, the focus within this text lies on the object-based section.

II. Sensor System

A. Robotic Perception

An analysis of the Urban Challenge competition requirements[3] points out the necessity to detect obstacles in a view area of nearly 360 degrees. The focus is set on the front and rear sections in order to detect fast oncoming traffic early enough for correct maneuver decisions. Within Caroline, three different sensor technologies are used in order to fulfill these requirements (Fig. 1).

Three automotive laser scanners mounted on the front and rear section of the vehicle deliver detailed information about existing static and dynamic obstacles up to a range of approx. 50 meters. Each one is based on a four-plane

Fig. 1 Sensor concept.

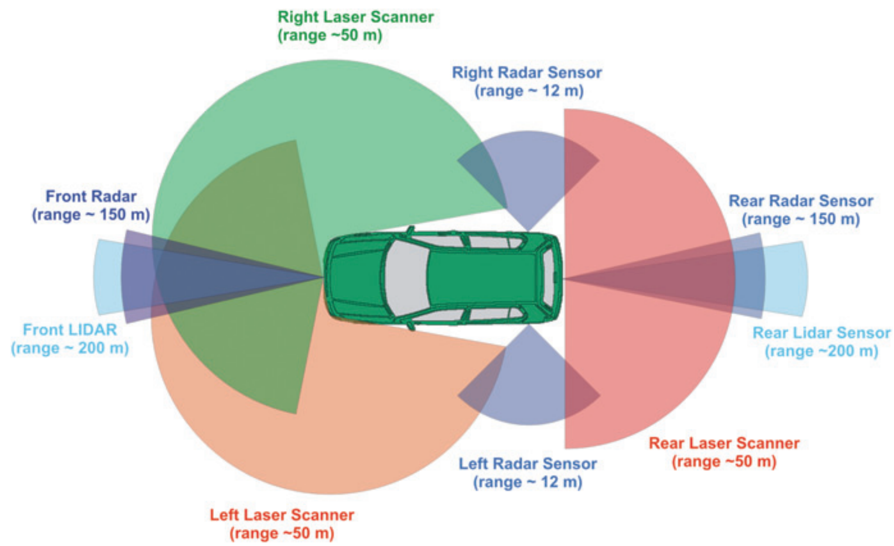

Fig. 2 View areas.

time-of-flight scanning principle, providing an additional vertical opening angle of 3.2 degrees. This way, small pitch movements of the vehicle can be compensated. The chosen mounting positions lead to almost 360 degrees of viewing area around the vehicle except for a small blind spot on the rear left and right sides.

In addition to these rotating laser scanners, a stationary multi-beam LIDAR sensor is mounted on the front and rear of the vehicle, extending the view area up to a range of 200 meters but with a considerably small horizontal opening angle of 12 degrees.

Two mid-range radar sensors are mounted on the front and rear section as well, covering a range of 150 meters with a horizontal opening angle of 40 degrees. Two short-range radar units on the rear left and right sides are used to close the blind spot left from the laser scanners.

All sensor systems provide sensor-internal data preprocessing and deliver their measurements in an object-oriented fashion, each with a different level of complexity:

1) The laser scanners are capable of describing complex contours of detected targets consisting of a variety of contour points and a common velocity vector.
2) The LIDAR system delivers line-oriented data consisting of target position, target width and target velocity.
3) The radar sensor is only capable of delivering point-shaped targets consisting of a target position and a target velocity vector.

All readings are delivered with respect to the individual sensor coordinate system. Since the sensors have been chosen from standard automotive driver assistance applications, they provide a CAN-bus interface (Controller Area Network) for delivering their object data. Due to the limited bandwidth of that bus interface, a private sensor CAN has been chosen for each sensor. This avoids conflicts in bus arbitration which would lead to higher cycle times and therefore affect the overall system's detection speed.

As seen in Fig. 2, a high level of redundancy has been established in the front and rear section of the vehicle. This redundancy is used within the vehicle's tracking system in order to effectively suppress false alarms during track initialization. Since false alarms, e.g., a misinterpreted reflection seen as a target by the radar sensor, are not likely to be issued by all sensors at the same time, each measurement can be justified by the readings of other sensors within the same area of redundancy. Track initialization therefore is key in the accuracy of a tracking system and will be further described in section IV.

B. Self Localization and Coordinate Transformation

Another important part of the sensor system is the vehicle's self-localization. A GPS-INS (global positioning coupled with inertial navigation) unit is mounted on the vehicle's roof providing accurate absolute position and orientation with respect to an earth-fixed reference frame. The acquired ego state is distributed to all perceptional sensor data processing units via fast Ethernet in order to ensure minimum latency. All processing units are synchronized to the master GPS clock via Network Time Protocol (NTP). Upon reception of new sensor data, readings are immediately time-stamped and then transformed from the sensor reference frame (SC) into a common earth fixed frame (WC), using the sensor system's own calibration information and acquired ego position and orientation,

$$\begin{aligned} \underline{x}_{\mathrm{WC}} &= \underline{\underline{R}}_{\mathrm{VC,WC}} \cdot \left(\underline{\underline{R}}_{\mathrm{SC,VC}} \cdot \underline{p}_{\mathrm{SC}} + \underline{t}_{\mathrm{SC,VC}}\right) + \underline{t}_{\mathrm{VC,WC}} \\ \underline{v}_{\mathrm{WC}} &= \underline{\underline{R}}_{\mathrm{VC,WC}} \cdot \left(\underline{\underline{R}}_{\mathrm{SC,VC}} \cdot \underline{v}_{\mathrm{SC}}\right) \end{aligned} \tag{1}$$

with $\underline{\underline{R}}$ and $\underline{t}$ being the rotation matrix and translation vector from sensor coordinates to vehicle coordinates (SC, VC) and on to inertial world coordinates (VC, WC). As a result, it is possible to carry out all post processing with respect to a common earth-fixed coordinate frame, so that static objects will have zero velocity, which eases their state estimation during object tracking and data fusion.

III. System and Software Architecture

A focus has been put on a linear signal flow and a sensor-independent software architecture in order to easily integrate alternative sensor technologies in the future (Fig. 3). Sensor measurement data is read from the sensor's CAN-busses and converted into a generalized sensor object structure, which can then be post-processed by sensor data fusion. Within this acquisition stage, all coordinate transformations and time-stamping operations are carried out. Additionally, sensor communication is monitored regarding expected cycle times, transfer errors or other malfunctions. Any unintended behavior of the sensor system is sent to a central safety surveillance unit (watchdog) which can then actively stop the vehicle and reboot single sensor- or computing units in order to provide an error-recovery functionality.

After acquisition, sensor data is sent to the data fusion units via fast Ethernet. This network-based architecture enables the fusion system to acquire sensor data from any computing unit in the vehicle, so the integration of further sensors can easily be accomplished without changes to the existing hard- and software structure.

All computing units in the vehicle are based on the same hardware platform, a 2 GHz Intel Pentium Mobile processor in a passively cooled case. This hardware choice has been primarily driven by necessity for low energy consumption and avoidance of moving parts (e.g., cooling fans) in order to maximize the system's reliability for

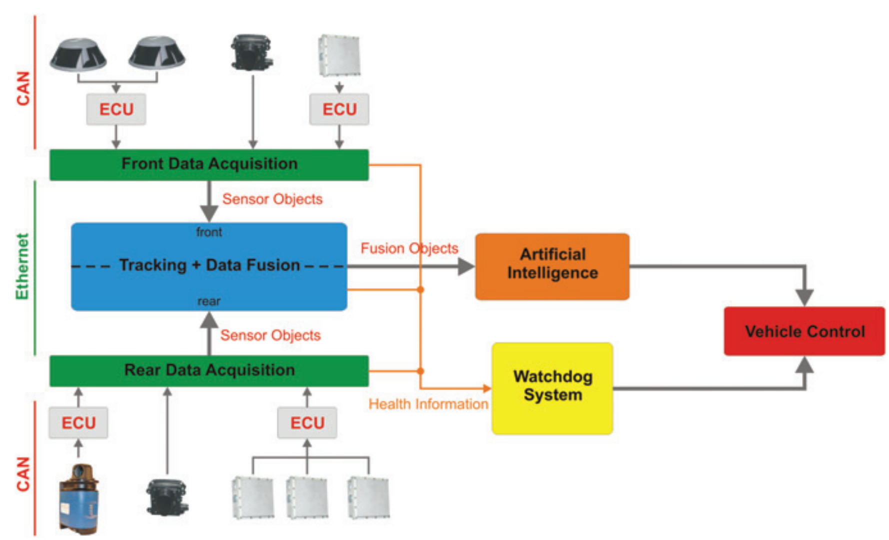

Fig. 3 System architecture.

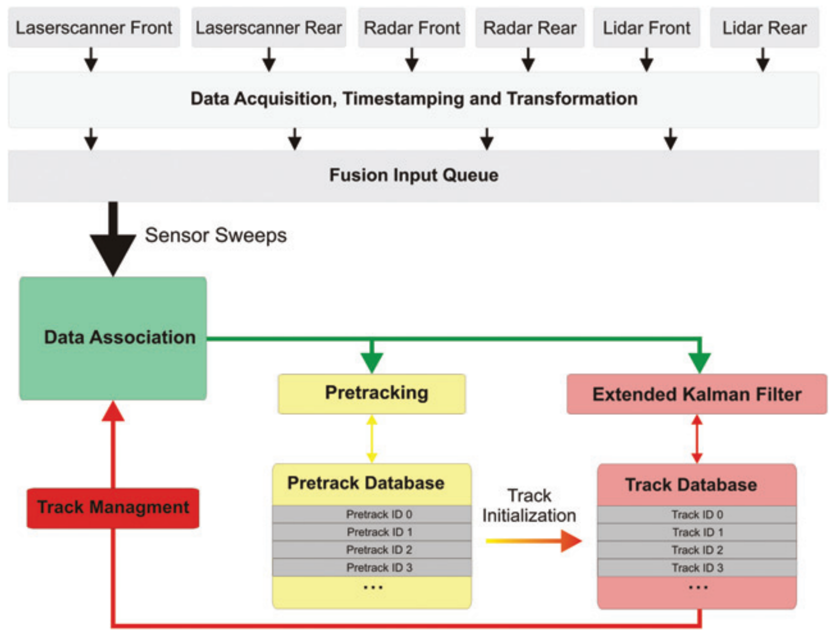

Fig. 4 Fusion system architecture.

long-term usage in an automotive environment. For sensor acquisition and fusion, two of those units have each been equipped with four CAN channels for low-level communication. To equally balance the computational load on the fusion system, it has been split into a front and rear section, each handling the associated sensors. Resulting fused sensor data (fusion objects) is transferred to Caroline's artificial intelligence unit via fast Ethernet for further mission planning.

The fusion system itself is based on a pipes and filters pattern (Fig. 4). All incoming sensor data is queued and then processed sequentially using a first in – first out strategy. Within the first step, data association is carried out in order to assign incoming sensor objects to their corresponding tracks in the fusion system, taken from a real-time track database. In case of a positive match between an existing track and incoming sensor object, this pairing is then pushed into the processing queue of the systems Extended Kalman Filter in order to correct the track with new measurement data. If no match can be found, the sensor object is regarded as a potentially new target and pushed into the pretracking system. Within pretracking, sensor data is justified against time and all other sensors taking into account sensor redundancy where applicable. Pretracking and data association will be further described in section IV.

If a sensor object has reached a certain level of justification, a new track will be instantiated and pushed into the real-time track database. Parallel to data association, pretracking and final object tracking, a track management unit periodically scans the track database for "dead" tracks – i.e., fusion objects that have not been updated for a certain amount of time. In addition to this garbage collection, all valid tracks are compared to each other in order to realize track merging and track splitting, which is necessary to handle situations like a passenger entering or leaving his vehicle or any other situation where two objects in the real world converge or split. Instead of transferring a whole track database image to downstream modules, create, update and delete messages of the track database are issued via the network. Every client is then capable of maintaining it's own track database. This way, network load can be significantly reduced without any loss of information.

IV. Data Association and Pretracking

Data association and pretracking have a key functionality within Caroline's fusion system. Imperfect data association leads inevitably to incorrect tracks, whereas incorrect track initialization during pretracking leads to imperfect data association, since correct tracks and false alarms will then compete for incoming measurement data. With this central position, the association and pretracking stage dominates the state estimator in the main tracking stage, since no state estimator can transform falsely associated sensor readings into useful update information for a track. In classical tracking approaches where objects are mostly described through a state vector consisting of a generalized object position, velocity and, if applicable, further derivatives of these quantities, data association can be performed in a point-to-point matching process.[1]

Within Caroline's fusion system, these approaches had to be extended in order to handle complex-shaped spread objects. Three different types of sensor objects have to be processed: complex contours delivered by laser scanners, line-shaped objects delivered by the LIDAR system and classical point-shaped objects received from radar sensors. It is not possible to define a common general object position seen by all sensors, since each sensor will most likely see the target differently (most obviously is the unknown point of reflection delivered from a radar compared to precise contour measurements gained from a laser scanner). Additionally, as the vehicle moves through the real world, the point of reflection of each sensor type moves on the outline of a real-world object. Therefore a multi-point track model has been chosen, describing a detected object by an arbitrary number of contour points and postulating a common movement vector following a rigid body assumption. This way, each contour measurement can be matched to the best fitting tracked contour point. A two-staged data association process has been set up, with the first stage serving as a justification whether or not track and measurement describe the same real-world object and in the second stage then calculating the optimal contour association between measured and tracked object points. Within stage one, a weighting function counting for the minimum Euclidian distance and similarity of velocities is calculated,

$$w_{i,j} = a \cdot \min\left[\left|\underline{x}_k^i - \underline{x}_l^j\right| \middle| \forall k, l\right] + b \cdot \left|\underline{v}_i - \underline{v}_j\right| \tag{2}$$

with $w_{i,j}$ being a scalar weight for association between track i and measurement j with tracked and measured velocity vectors $\underline{v}_i$, $\underline{v}_j$, $\underline{x}_k^i$, $\underline{x}_l^j$ being the kth and lth contour point position of track i and measurement j and a, b serving as tuning parameters. A threshold for this weight is further defined and an association below that threshold level will be pushed into stage two.

In stage two, an optimal match between all measured and tracked contour points is calculated based on an association matrix Ω,

$$\Omega = \begin{bmatrix} \left|\underline{x}_1^i - \underline{x}_1^j\right| & \dots & \left|\underline{x}_1^i - \underline{x}_l^j\right| \\ \dots & \dots & \dots \\ \left|\underline{x}_k^i - \underline{x}_1^j\right| & \dots & \left|\underline{x}_k^i - \underline{x}_l^j\right| \end{bmatrix} \tag{3}$$

Optimization can be carried out by standard algorithms such as the Hungarian / Munkres method, Nearest Neighbor or similar approaches.[2,4] This two-staged association process avoids unnecessary computational load on the system, since unlikely associations will be filtered out in stage one while the computational challenging minimization is only carried out for positive matches.

During pretracking, incoming sensor data is first associated with preliminary track objects (pretracks) using similar methods as described above. A pretrack carries along a vector of sensor assignments, storing for each sensor type the last assigned sensor object id – if applicable. A simple Kalman filter based on a constant-velocity motion model is calculated for each pretrack to update its position given incoming sensor data. In addition to the vector of sensor assignments, an update counter is carried along storing the number of positive association events. Taking into account the sensor redundancies read from a configuration file, a threshold for track activation is evaluated based on this update counter, dependent on the level of redundancy in the affected observation area of that object. A simple description language has been implemented to efficiently model these redundancies and to influence the update count threshold for track activation, e.g.:

```
polygon={0,2;10,2;10,-2;0,-2}
modifyCount=2000
condition=(RADARFront && !(LASERFront||LIDARFront)),
```

which means for the fusion system "Activate track in a 2×10 meter, box-shaped view area after 2000 positive matches when it is only seen by the front radar system and not by the laser scanners or LIDAR sensors", which in this case serves as a protection against random, unstable false alarms of the radar sensor directly in front of the vehicle.

V. Tracking and Data Fusion

For the main tracking algorithm, a model-switching Extended Kalman Filter has been implemented, based on two track motion models. A six-dimensional motion model describes fast moving objects using a state vector $\underline{x}$,

$$\underline{x^{6D}} = \begin{pmatrix} x_{1,\ldots,n} \\ y_{1,\ldots,n} \\ v \\ a \\ \alpha \\ \omega \end{pmatrix} \tag{4}$$

with $x_{1\ldots n}$, $y_{1\ldots n}$, v, a, α, ω being the x and y coordinate of the n contour points, the common velocity, acceleration, course angle and course angle velocity with respect to the global earth-fixed reference frame. For slow or static objects, a simpler four-dimensional state vector has been chosen,

$$\underline{x^{4D}} = \begin{pmatrix} x_{1,\ldots,n} \\ y_{1,\ldots,n} \\ v \\ \alpha \end{pmatrix} \tag{5}$$

thus taking into account that the majority of detected objects are of a rather static nature and distribution of available sensor information in unnecessary many state variables is suboptimal in that case. As seen in Eqs. (4) and (5), the classical state vector has been enriched by the number of contour points, thus making it necessary to extend the Kalman Filter algorithm to handle multiple positions within the same state vector. Similarly, we define the sensor measurement vector $\underline{y}$ for a sensor object consisting of m contour points,

$$\underline{y} = \begin{pmatrix} x_{1,\ldots,m} \\ y_{1,\ldots,m} \\ v_x \\ v_y \end{pmatrix} \tag{6}$$

with x_l, y_l, v_x, v_y being measured contour point x- and y-coordinates as well as x- and y-velocity components with respect to the global earth fixed reference frame.

Postulating a common position noise covariance for all contour points within track and measurement, the update algorithm[4,5,6] can be extended as follows:

$$\begin{aligned}
\underline{x}_k(v+1|v) &= f(\underline{x}_k(v)) \\
\underline{\underline{P}}(v+1|v) &= \underline{\underline{F}}^T \cdot \underline{\underline{P}} \cdot \underline{\underline{F}} + \underline{\underline{Q}} \\
\underline{s}_{k,l}(v+1) &= \underline{y}_l(v+1) - h(\underline{x}_k(v+1|v)) \\
\underline{\underline{S}}(v+1) &= \underline{\underline{H}} \cdot \underline{\underline{P}}(v+1|v) \cdot \underline{\underline{H}}^T + \underline{\underline{R}} \\
\underline{\underline{K}}(v+1) &= \underline{\underline{P}}(v+1|v) \cdot \underline{\underline{H}}^T \cdot \underline{\underline{S}}(v+1)^{-1} \\
r_{k,l}(v+1) &= K(v+1) \cdot s_{k,l}(v+1)
\end{aligned} \tag{7}$$

with $\underline{x}_k$ being the track state vector regarding contour point k, $f(\underline{x})$ the nonlinear system transfer function, $\underline{\underline{P}}$ the common state covariance matrix, $\underline{\underline{F}}$ the system transfer Jacobian, $\underline{\underline{Q}}$ the system noise covariance, $\underline{s}_{k,l}$ the innovation vector of tracked contour point k compared with measured point l of the associated sensor object, $\underline{y}_l$ the sensor measurement vector regarding measured point l, $h(\underline{x})$ the nonlinear system output function, $\underline{\underline{S}}$ the common innovation covariance matrix, $\underline{\underline{H}}$ the system output Jacobian, $\underline{\underline{R}}$ the estimated measurement noise, $\underline{\underline{K}}$ the Kalman gain in this update cycle and $\underline{r}_{k,l}$ the correction vector for tracked contour point k getting updated with measured point l.

The tracked contour points can then be updated by adding the first to components of the associated vector $\underline{r}_{k,l}$. In order to calculate updated common velocity, acceleration, course angle and course angle velocity (in the six dimensional movement model), the mean value for vector $\underline{r}_{k,l}$ is calculated over all given contour point associations,

$$\underline{r}_{\text{mean}} = \frac{1}{N} \sum_{k,l=1}^{N} \underline{r}_{k,l} \tag{8}$$

with N being the total number of acquired contour point matches within the second stage of data association. Corrected common values can then be acquired by adding the last four components of vector $\underline{r}_{\text{mean}}$ to the corresponding elements in the track state vector.

Obviously, by postulating a common system and measurement noise covariance for all contour points, Kalman gain can be computed once per update cycle. While it would theoretically be possible to calculate a separate Kalman gain for each tracked contour point and therefore removing the limitations to system and measurement covariance, this would lead to a N-times bigger computational load, since matrix inversion of the system innovation covariance matrix is the most costly part of the algorithm. In this case, the algorithm would simply calculate a separate Kalman filter for each contour point, which isn't practically realizable in a real-time application. In the described approach we have no significantly higher computational effort compared to a standard EKF while at the same time realizing spread-contour functionality and removing the need for a stable point of reference for tracked objects.

In order to prevent the track from being flooded with contour points, a garbage collection mechanism has been installed by carrying along update counters for each contour point which store the last update timestamp and the overall number of updates so far. This way, inactive contour points can be detected easily and removed from the tracks point list.

VI. Summary and Outlook

This contribution introduced an expandable architecture for multi-sensor data fusion in urban environments based on standard automotive sensor technology. The system architecture allows network-distributed operation and parallel processing on separate computational units, thus giving a maximum amount of flexibility and interchangeability regarding applied sensor technology. The tracking system has been designed to manage objects of arbitrary shapes; a common static object reference point is not required. Sensor redundancy can effectively be used to increase tracking quality and suppress false alarms. The underlying Kalman filters have been adapted in order to allow arbitrary numbers of contour points for each target.

Further improvements can be achieved by incorporating three-dimensional terrain data in order to effectively detect false sensor measurements from unwanted ground-reflections, which are a special issue concerning the laser based sensors. For this purpose, a hybrid approach consisting of a grid-based method to model the drivable terrain in order to identify unwanted ground reflections, which can then be suppressed in the object based tracking system is currently developed. Higher computational power could also be used to implement more sophisticated state estimators, e.g. the IMM Kalman Filter,[7] instead of the simple model switching carried out so far. Currently, the large number of tracks expected in an urban environment and the requirement for real-time operation limit the choice of more complex state estimators.

References

[1]Becker, J.-C., "Fusion der Daten der objekterkennenden Sensoren eines autonomen Straßenfahrzeugs *(German)*", Ph.D. Thesis, Institute of Control Engineering, Braunschweig, Germany, March 2002.

[2]Thrun, S., Burgard, W., and Fox, D., *Probabilistic Robotics*, MIT-Press, 2005.

[3]DARPA, "Technical Evaluation Criteria," 2006. [Online]. URL: http://www.darpa.mil/grandchallenge/docs/Technical_Evaluation_Criteria_031607.pdf

[4]Bar-Shalom, Y., *Estimation and Tracking*, Artech House, 1993.

[5]Bar-Shalom, Y. *Multitarget-Multisensor Tracking: Advanced Applications*, Artec House, 1990.

[6]Bar-Shalom, Y. and Hoffmann, T. E., *Tracking and Data Association*, Academic Press Inc., 1988.
[7]Shawn Michael Herman, *A Particle Filtering Approach to Joint Passive Radar Tracking and Target Classification*, University of Illinois, 2002.

Christopher Rouff
Associate Editior