High-Level Sensor Data Fusion Architecture for Vehicle Surround Environment Perception

Michael Aeberhard
BMW Group

Nico Kaempchen
BMW Group

High-Level Sensor Data Fusion Architecture for Vehicle Surround Environment Perception

Michael Aeberhard, Nico Kaempchen
ConnectedDrive
BMW Group Research and Technology
Munich, Germany
michael.aeberhard@bmw.de, nico.kaempchen@bmw.de

***Abstract*—Current driver assistance functions, such as Active Cruise Control or Lane Departure Warning, are usually composed of independent sensors and electrical control units for each function. However, new functions will require more sensors in order to increase the observable area and the robustness of the system. Therefore, an architecture must be realized for the fusion of data from several sensors. The goal of this paper is to give a brief overview of different types of sensor data fusion architectures, and then propose an architecture that can be best realized, in performance and in practice, for future advanced driver assistance systems. The proposed architecture is then demonstrated on a test vehicle designed for highly automated driver assistance systems, such as the emergency stop assistant.**

***Index Terms*—advanced driver assistance systems, high-level fusion, track-to-track fusion, surround environment perception**

I. Introduction

Today's production driver assistance systems, such as Active Cruise Control (ACC), Lane Departure Warning (LDW), Blind Spot Monitoring (BSM), etc. are usually independent systems, where the sensor and functionality of a system are located in one electronic control unit (ECU). For example, most ACC applications are based on a radar sensor, where the ECU for the radar sensor may also contain the application-specific logic of selecting the relevant vehicle for longitudinal control. Naab [1] provides a good state-of-the-art overview of production driver assistance systems and the information that they require. However, future advanced drive assistance systems (ADAS) will require more sensors, greater accuracy and better reliability due to the fact that these systems will provide increasingly complex functionality.

One such potential system is the emergency stop assistant, being developed by BMW Group Research and Technology, which can safely drive the vehicle in a highly automated driving mode onto the emergency lane in the event that the driver becomes incapacitated due to a medical emergency [2]. Such a highly automated driver assistance system (HADAS) requires a reliable vehicle surround environment perception in order to properly detect front, rear and neighboring vehicles for safe lane change maneuvers. This necessitates the use of many environment perception sensors, covering all possible angles. With so much data from many different sensors in different locations, designing the environment perception architecture is not trivial. In [3], a low-level modular fusion architecture is proposed for ADAS. A fusion architecture using JIPDA is described in [4].

The goal of this paper is to propose a sensor data fusion architecture that is optimized for the application of HADAS, such as the emergency stop assistant. This paper proposes a high-level sensor data fusion architecture, with the goal of simplifying the fusion process and making it more practical to implement in an automotive application.

Section II provides an overview of typical sensor data fusion architectures and their application in ADAS. The problems associated with a low or feature-level fusion architecture are described in Section III. The proposed high-level sensor data fusion architecture is described in detail in Section IV. Finally, the first results of the architecture are presented in Section V.

II. Sensor Data Fusion Architectures

Sensor data fusion in object tracking has been used for a long time in the aerospace industry for tracking aircraft with radar. The texts [5] and [6] provide a comprehensive overview of different sensor data fusion configurations and methods of fusion. In airborne radar applications, the detected objects are assumed to be point-targets, since the distance between the radar station and the detected aircraft is relatively large. However, this assumption does not hold in automotive applications, since a detected object, such as an overtaking vehicle, can fill the entire field-of-view of a sensor. Therefore, sensor data fusion

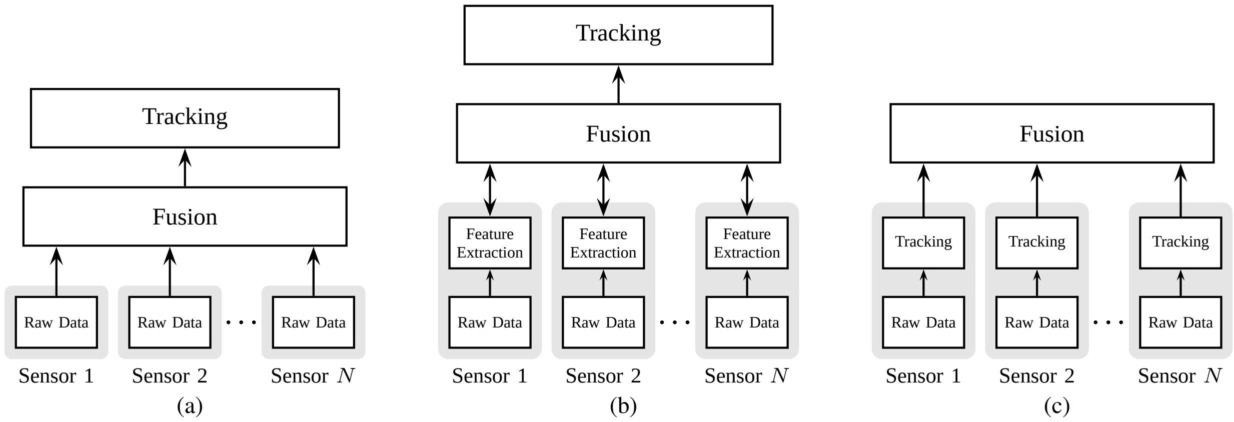


Fig. 1. Basic structure of common sensor data fusion architectures for advanced driver assistance systems: (a) low-level fusion, (b) feature-level fusion and (c) high-level fusion.

algorithms and architectures must be newly investigated for ADAS and HADAS applications.

This section provides a quick overview of the basic fusion architectures and their application in ADAS. Recent work has favored the low and feature-level fusion approach, however, the high-level fusion approach, if implemented correctly, has large potential and many advantages over low/feature-level fusion.

A. *Low-Level Fusion*

In a low-level fusion architecture, no pre-processing of raw data takes place at the sensor-level. Each sensor simply transmits its raw data to the fusion module, which then performs a low-level fusion of the raw data from all of the sensors. The fused raw data is then given as input to a central tracking algorithm. This architecture is sometimes also called a centralized tracking architecture. A basic low-level fusion architecture is illustrated in Figure 1a. Low-level fusion methods have been successfully demonstrated for automotive applications. In [7], a low-level measurement vector fusion is used to combine data from different sensors for a pre-crash application.

The advantage of low-level fusion is that it is possible to classify data at a very early stage through the fusion of raw data from different sources. The fused measurements that are selected for input to the tracking algorithm will already have a high likelihood of being a valid measurement of a relevant object for a specific application. However, low-level fusion requires high data bandwidth and can be complex to implement in practice. Adding a new sensor to the architecture requires significant changes to the fusion module, due to the fact that raw data comes in different formats and from different sensor types.

B. *Feature-Level Fusion*

The idea of feature-level fusion was introduced in [8]. Instead of directly using raw data from different sensors, as done in low-level fusion, feature-level fusion attempts to extract certain features from raw data through a pre-processing step, before carrying out the data fusion. The extracted features from a specific object model are then used as the input to the tracking algorithm. The feature-level fusion architecture is depicted in Figure 1b.

The main advantage of feature-level fusion is that it is able to reduce the bandwidth of sensor data to the fusion module, since extracted feature data is much smaller than raw sensor data. Feature-level fusion retains the same classification and pre-processing capabilities of low-level fusion, allowing for a similar efficient integration of relevant data into the tracking algorithm. Feature-level fusion has been successfully used in automotive applications, as demonstrated in [8] and [9].

C. *High-Level Fusion*

The high-level fusion architecture is the opposite of low-level fusion. Each sensor independently carries out a tracking algorithm and produces an object list. The fusion model then associates these objects with one another and performs a high-level track-to-track fusion of the sensor-independent objects. A basic high-level fusion architecture is shown in Figure 1c.

The main advantage of high-level fusion is its modularity and encapsulation of sensor specific details. All sensor relevant details are kept at the sensor-level, allowing the fusion module to process the data abstractly. This makes the high-level fusion architecture favorable in applications where modularity and simplicity of design is required. However, classification becomes more difficult because the sensor-level tracking algorithms have less information when associating raw data measurements to relevant objects. Track-to-track fusion can also become a complex task, because sensors can vary drastically in their tracking capability and reliability. Despite these disadvantages, high-level fusion architectures have been succesfully demonstrated in automotive applications, e.g. for ACC [10] and for safety systems [11].

III. Problems with Low/Feature-Level Fusion

Low/feature-level fusion architectures have proven to be effective in recent works. However, the applications are limited, since low/feature-level fusion architectures are usually developed for a single application, such as a two-sensor ACC. Once more complex applications are considered, the increase in sensors and their various locations on the vehicle becomes problematic in a low/feature-level fusion architecture.

A. Temporal and Spatial Alignment

Depending on the type of low-level fusion carried out, temporal alignment of sensor data can be problematic. There is no filtered dynamic information about measurements available, making it difficult to temporally align highly asynchronous data. This can create inaccuracies in a low-level measurement-to-measurement association and fusion. Measurements being directly fed into a central filtering algorithm must rely on highly accurate timestamps. Temporal jitter due to global timestamp synchronization and communication delays can lead to inaccurate timestamps, which can cause significant noise in the estimated states of a filter.

In addition to being temporally aligned, data must also be spatially aligned to a common coordinate system. Inaccuracies in spatial alignment can also create noise in the estimated states of a filter. Spatial misalignment can occur when two sensor are biased against one another, either due to an internal hardware bias (e.g. due to temperature variations) or inaccurate sensor calibration.

B. Modularity

For future ADAS, modularity is of high importance, since different sensor configurations for different vehicle models should be supported in the same fusion architecture. The more sensors there are, the more complex the fusion task becomes at a low-level. Most low-level fusion architectures also rely on some sort of feedback mechanism with the fusion module, thereby increasing the complexity of the system. Finally, with the large amount of sensors required for some functions, vehicle bus communication bandwidth is increased, requiring more buses and more hardware, all of which increase complexity and cost of the system.

IV. Proposed Fusion Architecture

The proposed fusion architecture for ADAS consists of three main levels of sensor data processing: sensor, fusion and application-level processing (Figure 2). At the sensor-level, each sensor produces an object list that is locally calculated by the sensor. The object lists from all of the sensors are then processed at the fusion-level, where they are combined to construct a global object list. At the application-level, the global object list is further processed and combined with other data sources in order to determine which objects are relevant for a specific ADAS function.

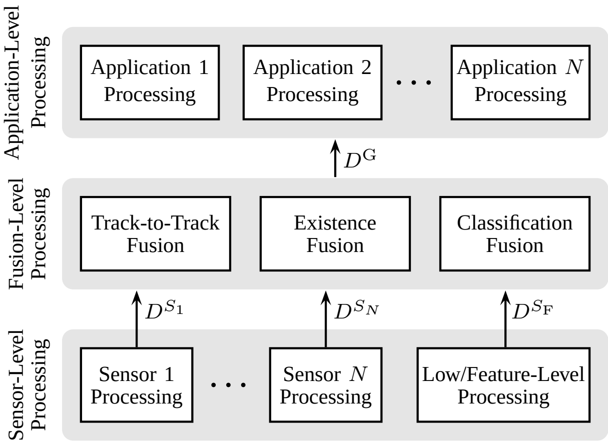


Fig. 2. Proposed fusion architecture for advanced driver assistance functions.

The proposed fusion architecture uses a standard communication protocol between the sensors and the fusion module, consisting of the object state vector, uncertainty, state mask, existence probability and classification. An object is fully described with the state vector

$$\mathbf{x} = \begin{bmatrix} x & y & v_x & v_y & a_x & a_y & l & w \end{bmatrix}^{\mathrm{T}} \tag{1}$$

where x, y is the position, v_x, v_y is the velocity, a_x, a_y is the acceleration, l is the length and w is the width of an object. Along with the state vector, the sensors must also provide the accuracy of the estimation in the form of a covariance matrix, $\mathbf{P}$. However, since not every sensor is able to observe or estimate the full state vector, an additional masking matrix, $\mathbf{M}$, must be provided, through which the state vector from a sensor can be defined as

$$\hat{\mathbf{x}}^S = \mathbf{M}^S \hat{\mathbf{x}} \tag{2}$$

$$\mathbf{P}^S = \mathbf{M}^S \mathbf{P} (\mathbf{M}^S)^{\mathrm{T}} \tag{3}$$

If the sensor-level kinematic model is different than the global model (e.g. constant turn model, instead of a constant velocity or acceleration model), then $\mathbf{M}^S$ would be a function of the sensor-level states for calculating the states of the global object model.

Each object is also defined by an existence probability, $p(\exists \mathbf{x})$, which calculates the probability that an estimated object actually exists. Lastly, objects are also defined by a classification vector, where objects may be classified as

$$\mathbf{c} = \begin{bmatrix} C_{\text{Car}} \\ C_{\text{Truck}} \\ C_{\text{Bike}} \\ C_{\text{Stationary}} \\ C_{\text{Unknown}} \end{bmatrix} \tag{4}$$

A. Sensor-Level Data Processing

Each sensor is treated as a separate entity at the sensor-level and is expected to produce an object list that is compatible with the fusion-level processing unit, as described in the previous section. The sensor-level processing unit for each sensor must therefore produce the following data:

$$D^S = \{\hat{\mathbf{x}}^S, \mathbf{P}^S, \mathbf{M}^S, p(\exists \mathbf{x})^S, \mathbf{c}^S\} \tag{5}$$

where $\hat{\mathbf{x}}^S$ and $\mathbf{P}^S$ is the state vector and covariance matrix from the tracking algorithm, $\mathbf{M}^S$ is the sensor-level masking matrix with respect to the global object model, $p(\exists \mathbf{x})^S$ is the object existence probability and $\mathbf{c}^S$ is the object classification vector. An example of sensor-level processing is illustrated in Figure 3.

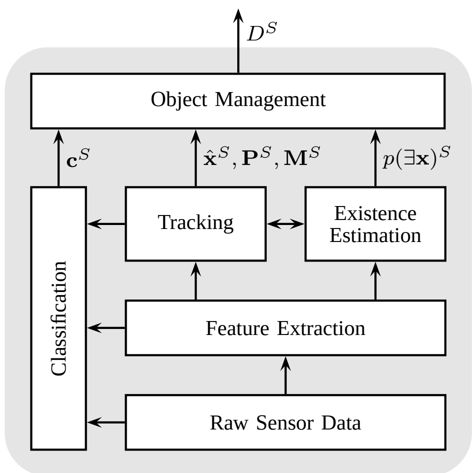


Fig. 3. Sensor-level processing structure for producing D^S for a single sensor.

In general, sensor-level processing is a standard tracking algorithm, consisting of raw data pre-processing (e.g. segmentation, feature extraction), data association, filtering and object management. The object existence probability, $p(\exists \mathbf{x})^S$, takes into account the sensor's field-of-view and any applicable sensor-specific properties. Additionally, objects are classified at the sensor-level whenever possible, based on the properties of the raw data or features, or from the state estimation.

The proposed architecture assumes nothing about the tracking algorithm used at the sensor-level; only the dimension and description of the state vector is known through the masking matrix, $\mathbf{M}^S$. Making this assumption allows for a highly modular architecture that requires little knowledge about the sensor-specific algorithms. For an automotive application, this type of modularity has a huge advantage, since only one fusion module needs to be designed, but can then be used with several different types of sensor configurations and applications.

In order to allow the architecture to benefit from the advantages of a low/feature-level fusion architecture, the sensor-level processing could in itself consist of a low/feature-level fusion, where it can be considered as a virtual sensor in the proposed architecture, as shown in the bottom-left of Figure 2. This makes the proposed architecture compatible with an object list that is generated from a low/feature-level fusion, when such a fusion strategy may be necessary for a specific application.

B. Fusion-Level Data Processing

At the fusion-level, the independent object lists from the different sensors are fused together to generate a global object list by means of a track-to-track fusion algorithm. The fusion processing module accepts object lists in the standard format described earlier. The fusion-level processing steps are depicted in Figure 4.

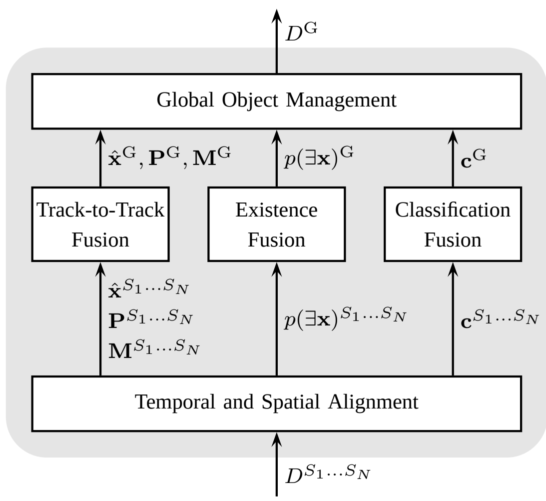


Fig. 4. Fusion-level processing structure for producing D^{G}, the global object list.

Before fusion, the object lists from each sensor must be spatially and temporally aligned to a common coordinate system. However, the alignment is less problematic than in a low-level fusion architecture, due to the fact that the objects already have filtered kinematic information, and therefore a more accurate alignment is possible. The fusion module then performs a track-to-track association with the aligned objects in order to determine which objects from different sensors represent the same object in reality. The associated objects' state vector and covariance matrix are then fused into one object using a track-to-track fusion algorithm. Through the use of the masking matrix, $\mathbf{M}^S$, only the common states of the objects are fused. New objects are assigned a new and unique identification number at the fusion-level. This allows objects to be uniquely identified throughout their history, as they pass through the field-of-view of different sensors.

Additionally, the object existence probabilities and classification vectors are also fused together. The new, global, object data, D^{G}, is then transmitted to the application-level processing unit, completely independent of any sensor-specific details.

C. Application-Level Data Processing

At the application-level, the global object list from the fusion-level is used to extract the relevant objects required for a specific application. Additionally, the application-level processing unit can also be connected to other high-level information, such as a digital map, camera or laserscanner based lane detection, occupancy grid map, or ego-vehicle dynamics or status data. From all of this data, an application can perform its function-specific situational assessment algorithm.

For HADAS, such as the emergency stop assistant, the application-level processing unit needs to identify relevant traffic participants in the ego-lane and in neighboring lanes in order to determine if a lane change maneuver can be safely executed, as described in [12]. This requires the global object list from the fusion-level processing unit and a reliable road boundary detection algorithm, such as the one described in [13], or a highly-accurate digital map. The objects are then associated to the left lane, ego lane or the right lane. Based on the location, velocity and quality of the objects in the ego or neighboring lane, an assessment is made whether or not a lane change is possible and safe in order drive the vehicle onto the emergency lane in case the driver becomes incapacitated.

V. Results

The test vehicle used for evaluation of the environment perception architecture and the emergency stop assistant

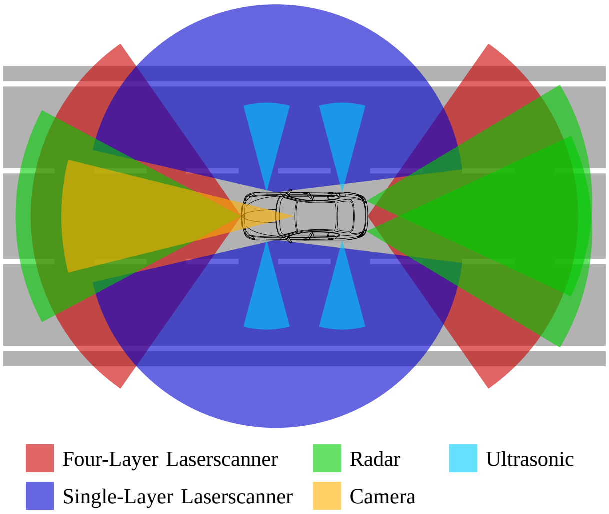


Fig. 5. BMW 5 Series test vehicle sensor configuration.

is a BMW 5 Series, equipped with laserscanner, radar, video and ultrasonic sensors, as shown in Figure 5. This sensor configuration is able to detect objects +/- 200 m longitudinally and +/- 120 m laterally, making it an ideal configuration for testing ADAS and HADAS. Redundant sensors in each direction provide the system with robustness and reliability.

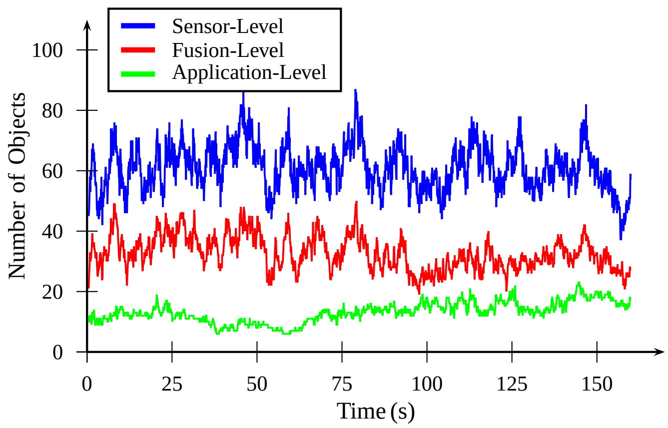
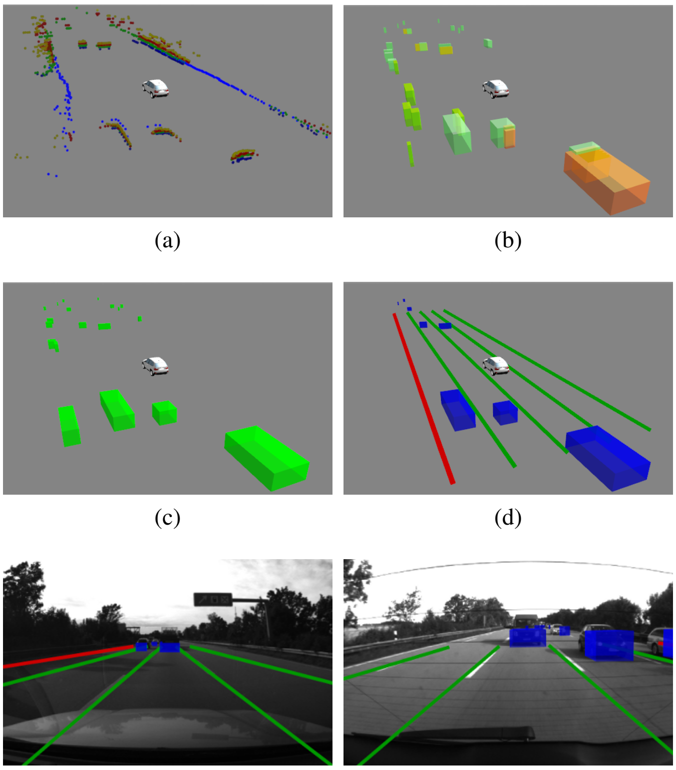


Fig. 6. Number of objects at the output of each level in the architecture for a typical highway scenario.

Preliminary results show that the proposed architecture is a successful means of organizing and processing data from many sensors that are mounted at different positions on the vehicle. The architecture has the effect of reducing the number of objects at each level, where the application-level produces a small number of objects that are relevant for a specific application. For a typical highway scenario, the number of objects at the output of each level in the architecture is shown in Figure 6. Figure 7 visualizes the sensor data at every step of the proposed architecture, from the raw data (Figure 7a) to



(a) (b) (c) (d) (e) (f)

Fig. 7. Progression of object detection in the proposed sensor data fusion architecture, as shown in the environment perception visualization tool: (a) raw data (laserscanner only), (b) sensor-level objects, (c) fusion-level objects and (d) application-level objects. The last two images, (e) and (f), show the detected application-relevant objects overlaid on the front and rear reference images, respectively.

the application-relevant objects (Figure 7d). The proposed architecture is currently being successfully used as the environment perception framework for demonstrating the emergency stop assistant.

VI. Conclusion

This paper proposed a high-level sensor data fusion architecture designed for future highly automated driver assistance functions, such as the emergency stop assistant being developed by BMW Group Research and Technology. The fundamental aspects of the architecture have been implemented in a test vehicle and have proven to show high potential for a high-level sensor data fusion architecture.

However, optimal algorithms for track-to-track fusion, existence fusion and classification fusion still need to be investigated and developed. A sensor-independent probability of object existence at the sensor-level and fusion-level is currently being developed. Future work will dive deeper into each level of the architecture in order to improve the surround environment perception for HADAS. Once these steps have been completed, the system can then be fully evaluated over many different types of highway driving scenarios.

Acknowledgments

The authors would like to acknowledge the suggestions and support from the many colleagues of the BMW Group Research and Technology ConnectedDrive group.

References

[1] K. Naab, "Sensorik- und Signalverarbeitungsarchitektur für Fahrerassistenz und Aktive Sicherheit," in *Tagung Aktive Sicherheit durch Fahrerassitenz*, Garching, 2004.

[2] N. Kaempchen, P. Waldmann, and F. Homm, "Umfelderfassung für den Nothalteassistent – ein System zum automatischen Anhalten bei plötzlich reduzierter Fahrfäigkeit des Fahrers," in *Automatisierungs-, Assistenzsysteme und eingebettete Systeme für Transportmittel*, Braunschweig, Germany, February 2010, pp. 121–137.

[3] M. Darms and H. Winner, "A modular system architecture for sensor data processing of ADAS applications," in *IEEE Intelligent Vehicles Symposium*, Las Vegas, Nevada, U.S.A, June 2005, pp. 729–734.

[4] M. Munz, M. Mählisch, and K. Dietmayer, "A probabilistic sensor-independent fusion framework for automotive driver assistance systems," in *6th International Workshop on Intelligent Transportation*, Hamburg, Germany, March 2009.

[5] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Norwood, MA, 1999.

[6] Y. Bar-Shalom, *Multitarget-Multisensor Tracking: Principles and Techniques*, Yaakkov Bar-Shalom, Storrs, CT, 3rd edition, 1995.

[7] S. Pietzsch, T. D. Vu, O. Aycard, T. Hackbarth, N. Appenrodt, J. Dickmann, and B. Radig, "Results of a precrash application based on laser scanner and short-range radars," in *IEEE Transactions on Intelligent Transportation Systems*, December 2009, vol. 10, pp. 584–593.

[8] N. Kaempchen, M. Bühler, and K. Dietmayer, "Feature-level fusion for free-form object tracking using laserscanner and video," in *IEEE Intelligent Vehicles Symposium*, Las Vegas, Nevada, U.S.A, June 2005, pp. 453–458.

[9] M. Mählisch, R. Schweiger, W. Ritter, and K. Dietmayer, "Sensorfusion using spatio-temporal aligned video and lidar for improved vehicle detection," in *IEEE Intelligent Vehicles Symposium*, Tokyo, Japan, June 2006, pp. 424–429.

[10] H. Takizawa, K. Yamada, and T. Ito, "Vehicles detection using sensor fusion," in *IEEE Intelligent Vehicles Symposium*, Parma, Italy, June 2004, pp. 238–243.

[11] N. Floudas, A. Polychronopoulos, O. Aycard, J. Burlet, and M. Ahrholdt, "High level sensor data fusion approaches for object recognition in road environment," in *IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, June 2007, pp. 136–141.

[12] M. Ardelt, P. Waldmann, F. Homm, and N. Kaempchen, "Strategic decision-making process in advanced driver assistance systems," in *6th IFAC Symposium Advances in Automotive Control*, Munich, Germany, July 2010.

[13] F. Homm, N. Kaempchen, J. Ota, and D. Burschka, "Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection," in *Intelligent Vehicles Symposium*, San Diego, CA, U.S.A, June 2010, pp. 1006–1013.