Data Fusion of Environment-Perception Sensors for ADAS

23

Michael Darms

Contents

Abstract

More and more driver assistance systems are based on a fusion of multiple environment perception sensors. This chapter gives an overview about the objectives of sensor data fusion approaches, explains the main components involved in the perception process, and explains the special topics that need to

M. Darms (✉)
Continental AG, Lindau, Bavaria, Germany
e-mail: mdarms@gmx.de

H. Winner et al. (eds.), *Handbook of Driver Assistance Systems*,
DOI 10.1007/978-3-319-12352-3_24

be taken into consideration in developing a multi-sensor fusion system for driver assistance systems. Focus is put on the topics of data association, tracking, classification, and the underlying architecture. The architecture strongly influences the costs, performance, and the development process of a multi-sensor fusion system. As there are no deterministic methods that guarantee an optimal solution for developing an architecture, the chapter gives an overview of established, general architecture patterns in the field of sensor data fusion and discusses their benefits and drawbacks.

1 Introduction

Driver assistance systems exclusively based on single-sensor solutions are known from prior art. Examples include applications such as adaptive cruise control, which relies on a single RADAR or laser sensor, for example, or lane departure warning, which typically relies on a video sensor system.

As described in the previous chapters, the various sensor technologies all have specific advantages and disadvantages. For example, a RADAR sensor can be used to determine the longitudinal distance and velocity of a vehicle driving ahead with a degree of accuracy sufficient for the adaptive cruise control application (see ► Chaps. 45, "Adaptive Cruise Control," and ► 17, "Automotive RADAR"). However, the relevant object, to which a certain distance must be kept, can only be selected with a certain degree of precision due to the lateral resolution, the ambiguities in signal evaluation, and the lack of lane marker detection; interference from vehicles on adjacent lanes must be thus accepted in system operation. On top of this, there are limits to the ability to classify the detected object, so the control algorithm typically only uses objects for which motion has been detected.

The missing information can be provided, for example, by data from a video sensor (see ► Chaps. 19, "Automotive Camera (Hardware)," ► 20, "Fundamentals of Machine Vision," and ► 48, "Lateral Guidance Assistance"). Lane marker detection provides information which can be used for lane assignment. Classification algorithms allow vehicles in the video image to be distinguished from other objects, while image processing technologies enable to determine the position of vehicles in the video image. In contrast to RADAR sensor systems, the distance and speed cannot be measured and must therefore be estimated. The achievable precision is significantly lower with current sensor systems, especially in the long-distance range. The functionality of an adaptive cruise control system purely based on video sensors is thus restricted to a smaller speed range.

Combining the information from both sensors helps to leverage the benefits of both technologies. For example, the RADAR sensor's distance measurements can be combined with the classification information and vehicle position measurements in the video image. This makes it possible to reduce false interpretations and improve accuracy in terms of the lateral position and distance. At the same time, lane assignments, and thus the ability to detect the reference object with the help of the video sensor data, become more robust.

Various research works (see, e.g., Darms 2007; Holt 2004; Stüker 2004; Becker 2002; Bender et al. 2007), confirm the capability of data fusion approaches of this type; environmental sensor data fusion is used in production vehicles (see, e.g., Schopper et al. 2013). This means both the fusion of RADAR and camera sensors, as in the example given here, and other combinations, for example, short-range and far-range RADAR. The principle of fusion can be extended to other sensor technologies. The current research and development focuses include the fusion of various imaging sensors and the fusion of data from environmental sensors with stored map data.

The following sections provide an introduction to the basic principles of sensor data fusion in driver assistance systems. Firstly, the term sensor data fusion is defined, and the objectives of fusion are stated. The main components of environmental data processing are then explained with a view to fusion of data from multiple sensors. Finally, established architectural patterns for sensor data fusion are presented. Part of the text in this chapter is orientated on the text provided in Darms (2007).

2 Definition and Objectives of Sensor Data Fusion

2.1 Definition of Sensor Data Fusion

According to Steinberg et al., the process of data fusion is defined as follows:

> Data fusion is the process of combining data or information to estimate or predict entity states. (Steinberg et al. 1998)

The generic term "entity" is used to describe an abstract object to which information can be assigned. In the world of driver assistance systems, this can mean a physical object in the vehicle's environment, such as an observed vehicle, but also an individual state variable, such as the pitch angle.

The following text mainly refers to the former case and thus directly uses the term "object." The focus is on track estimation, which is also referred to as tracking, and on object discrimination (see also Klein 1999). Tracking means estimating the states of an object in terms of control theory (e.g., position and speed). Object discrimination is further broken down into detection and classification (Klein 1999). In the course of detection, a decision is made as to whether an object exists, while classification assigns the object to a predefined class (e.g., vehicle, pedestrian). However, the considerations presented here can also be generalized to apply to abstract objects (see also the discussion in Dietmayer et al. (2005)).

2.2 Objectives of Data Fusion

The primary objective of data fusion is to merge the data from individual sensors so as to combine their strengths in a beneficial way and reduce individual weaknesses.

The following aspects can be distinguished (see also Lou and Kay 1991; Joerg 1994).

Redundancy Redundant sensors provide information relating to the same object. This helps to improve the quality of the estimation. An estimation algorithm must take the measuring error dependencies into consideration (see, e.g., Bar-Shalom and Li 1995). One risk is the multiple introductions of artifacts and misinterpretations into the fusion process (see below).

Redundancy can also help to improve the error tolerance and availability of the system in cases of individual sensor failure on the one hand – assuming that the system can still provide data of sufficient quality without the information from the failed sensor – and for artifacts or misinterpretations by individual sensors on the other. Redundancy can reduce the influence of an individual single error on the system as a whole.

Complementarity Complementary sensors deliver different, supplementary information into the fusion process. This can happen from a spatial point of view, where the same sensors deliver information with different field of views. Particular attention should be focused on data processing of the peripheral zones of the detection area in this case (see, e.g., Stüker 2004). This can also mean data that relate to the same object. The information content can be enhanced by detecting different properties. It is possible that a combination of the individual items of data is required to provide the information required by the application.

The use of different sensor technologies can also improve the robustness of the overall system in terms of detecting individual objects that may not be reliably detected by single-sensor technology. For example, the beam from a laser sensor penetrates glass, or the beam from a RADAR sensor penetrates various plastic materials without detecting the object in question. Combining the sensors reduces the probability of not detecting the object at all.

Temporal Aspects The overall system's speed of acquisition can be improved by a fusion approach. This can be achieved firstly by parallel processing of information from the individual sensors and secondly by appropriate timing of the acquisition process (e.g., by sensors measuring alternately).

Improved precision, or the introduction of complementary information, also influences the dynamic of the estimation. It must also be noted that different applications can pose different requirements in terms of estimation dynamic and accuracy and that it can still make sense, even in a sensor fusion system, to use different estimation algorithms for different applications (see Sect. 3.2).

Costs When designing any sensor system, the costs are a decisive factor in deciding its practical feasibility. The use of a fusion system can help to reduce the costs, compared with an individual sensor. However, this is not true in all cases because, for example, improvements can also be achieved by developing new algorithms for evaluating the data from a single sensor or by hardware advances.

The decision to develop a single- or multi-sensor system will thus always be multidimensional and must be based on the aspects stated above.

The costs of a sensor fusion system are substantively influenced by the architectural structure of the system (see, e.g., Hall 2001; Klaus 2004). Thus far, a uniform architecture has not been specified in the automotive industry in the form of a mandatory or de facto standard. This makes cross-enterprise cooperation between suppliers and vehicle manufacturers, strategic development of sensors and algorithms adapted for a common architecture, and the migration to new assistance functions and sensor generations more difficult (see also Hall 2001).

Modularity and the ability to economically extend the system are critical to its practical feasibility. The aim is to realize migration to new assistance functions economically and make it possible to source sensors and modules from various suppliers, an aspect which is especially important to vehicle manufacturers.

3 Main Components in Sensor Data Processing

3.1 Overview

The following section summarizes the main components in environment sensor data processing. The structure is generic and applies also to single-sensor systems. The special features that need to be taken into consideration in developing a multi-sensor system are pointed out at the appropriate places.

3.2 Signal Processing and Feature Extraction

In the scope of signal processing and feature extraction (see also Hall and McMullen 2004), information from the vehicle's environment is acquired by sensors. Figure 1 shows the process. In the first step, which is referred to as measure, the receiver element of the sensor (signal reception) receives payload signals (energy) overlaid with interference signals (noise) and converts them into raw signals (e.g., voltages, currents). The raw signals are interpreted as physical measurements (e.g., intensities, frequencies, etc.), which finally form the sensor's

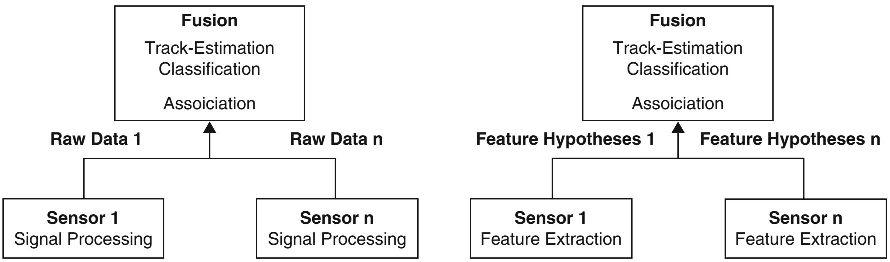

Fig. 1 Perception process: measure and perceive (see Darms 2007, p. 9 and Darms et al. 2009)

raw data. During signal processing, (physical) assumptions for interpretation are made (e.g., maximum reception level, impulse forms, etc.). Where these assumptions are breached, artifacts (system-specific weaknesses) occur.

In the second step, termed perceive, features (e.g., edges, extreme values) are extracted from the raw data on the basis of assumptions and models/heuristics. An object hypothesis, an assumed object, is derived from these feature hypotheses. Misinterpretations can occur due to the use of heuristics.

Where the information from multiple sensors is used in the estimation process, it is necessary to find a common reference for the information. This task is in particular made more difficult if the information is not orthogonal, that is, statistically independent.

One fundamental problem here is that of transferring the data to a coordinate system with a common reference point. In the case of a single sensor, the effect of adjustment errors can only be a negligible offset. However, maladjustment of a multi-sensor system can make it impossible to align the data from various sensors or cause systematic errors and deviations. This can impair the quality of the evaluation (see below). Suitable adjustment processes and (online) adaption algorithms are thus a central development focus for a multi-sensor system.

On top of this, various sensors can measure different attributes, even if this is not desired. This occurs in particular with non-orthogonal sensors. For example, if the distance to a vehicle is measured by a laser sensor and a RADAR sensor, it is possible that the sensors detect different parts: the laser sensor might detect the rear reflectors on a truck, while the RADAR sensor detects the rear axle. This effect can also be observed for identical sensors. One reason for this is that an object is detected from different angles of view. It is aggravated by sensor-specific artifacts during measurement and feature extraction, which can also have an effect despite the use of identical sensors.

Special care needs to be taken for the perceive part in multi-sensor systems. For example, the extracted feature hypotheses from various sensors will ideally relate to the same physical object. Due to different sensor resolutions, and misinterpretations, for example, in data segmentation (see, e.g., Holt 2004; Streller 2006), the object hypothesis can differ between sensors. For a system with unsynchronized sensors, the extracted features can also originate at different points in time. To be able to combine the data from the various sensors, one thus at least needs a mutual time base and sufficiently accurate time stamping (see also Kampchen and Dietmayer 2003).

The topic of temporal and spatial association of data from various sensors is also summarized in the referenced literature under “sensor registration” (see, e.g., Hall 2001).

3.3 Data Association

The feature hypotheses gained from signal processing and feature extraction are associated with object hypotheses already known to the system in the data

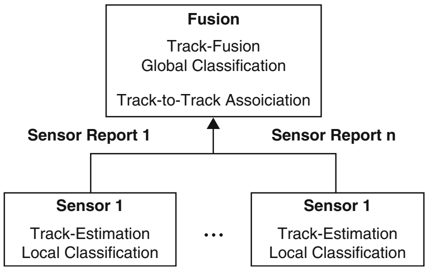

Fig. 2 Breakdown of the data association process (see Darms 2007, p. 45)

association step (see, e.g., Bar-Shalom and Li 1995). The quality of the estimation is significantly influenced by the data association process (see Holt 2004; Stüker 2004; Bar-Shalom and Li 1995). If an incorrect association is made, information loss occurs or false information is introduced in the estimation process (see, e.g., Stüker 2004).

Hall and Llinas break the data association process down into the following three steps (see Hall and Llinas 1997 and Fig. 2); special algorithms used in automotive applications can be found in Holt (2004), Becker (2002), and Streller (2006), for example.

1. **Generating association hypotheses**. Theoretically possible associations of feature hypotheses to object hypotheses are found. The results are one or multiple matrices with theoretically possible associations (association matrices).
2. **Evaluating the association hypotheses**. The association hypotheses found are evaluated with the aim of quantitative evaluation or ranking. The results are quantitative values (e.g., costs) in the association matrix or matrices.
3. **Selection of association hypotheses**. A selection is made from the evaluated association options; downstream data processing and thus, in particular, data filtering are based on this.

The three processing steps do not need to be implemented separately; on the contrary, they can depend on one another. However, it is advisable to decouple the steps in the development process (see Hall and McMullen 2004). The quality and performance of the available resources (e.g., computing capacity, resolution and usable raw data of a specific sensor, artifacts and potentially false interpretations) play a role in designing the algorithms. Depending on these boundary conditions, various solutions are possible (see Hall 2001).

Hypothesis generation itself can be broken down into two sub-steps: postulating the association hypotheses and selecting the theoretically possible hypotheses. Various methods can be used for postulating the association hypotheses. They include (see Hall and McMullen 2004):

Physical models. Fields of view and occlusions of the sensors used can be calculated. Object hypotheses that lie significantly outside the field of view are not considered in generating hypotheses.

Scenario knowledge. The behavior and potential location of objects on the basis of the observed scenario can be leveraged, for example, areas for finding road markers or traffic signs.

Probabilistic models. The expected number of false detections can be factored into the process.

Ad hoc methods. One example of this is postulating all possible association options. No prior knowledge needs to exist for this. However, it does make the process of selecting the correct associations more difficult.

The following methods are possible for selecting possible hypotheses (see, e.g., Hall and McMullen 2004):

Pattern detection algorithms. Associations can be ruled out using the raw signals and raw data (e.g., via correlation techniques).

Gating techniques. Physical models, for example, can be used to compute an area in which object hypotheses, or the feature hypotheses derived from them, can exist with a specific probability at the current time of measuring (prediction). Feature hypotheses originating from the current measurement cycle that lie outside of such an area are not associated with the corresponding object hypothesis.

Hypothesis evaluation can be based on probabilistic models based on Bayes' theorem, possibilistic models based on the Dempster–Shafer theory, neuronal networks, or even ad hoc techniques, such as unweighted distance computation between a prediction of the features and the features themselves (see, e.g., Hall and McMullen 2004).

Finally, a variety of mathematical algorithms exists for *hypothesis selection* (see Hall and McMullen 2004). This solution requires a large amount of computing time with increasing dimensions and in particular if data from multiple cycles are considered in the selection algorithm.

Simple approaches, where hypothesis selection only considers the data from the current cycle, are manageable in terms of complexity. Stüker provides an overview of various association methods (see Stüker 2004). A problem that is frequently found here is that of associating n object hypotheses with m feature hypotheses where $m \geq n$, and where one object hypothesis is associated with precisely one feature hypothesis.

Precise methods exist for this that minimize the aggregated costs in the association matrix. One example is the Munkres algorithm which has a complexity of $O(n^2 m)$) (see Becker 2002). Less complex algorithms also exist, but they only provide approximated solutions. One example is the iterative nearest neighbor method, which successively selects associations with the lowest cost, or the highest probability, at a complexity of $O(m^2 \log 2m)$ (see Becker 2002). Depending on the sensor technology, various algorithms are used (see Darms 2007).

As the discussion shows, data association can also be optimized by means of sensor-specific algorithms. Without access to the raw data, and if the sensor-specific conditions are not taken into account, the quality of the data association can degrade (see also Darms 2007).

Data association is also related to feature extraction and object hypothesis generation. Again, a variety of sensor-specific options for optimizing or reconciling individual processes exist with a view to achieving the best possible association of feature hypotheses to object hypotheses given the existing resources. This approach makes it possible to identify artifacts, e.g., duplicate measurements in the scope of data association, and to exclude them from the fusion process (see, e.g., Darms et al. 2008).

Knowledge of the way the data are generated, such as potential artifacts and typical misinterpretations, can thus be used for optimizing the algorithms. In addition, special properties of a sensor technology, such as the resolution capability, can be taken into account when designing the algorithms. The data association algorithm design is thus related to the knowledge of how the data are generated and thus of the hardware of the sensor being used. In a modular setup, it can thus be useful to encapsulate the association algorithms in sensor-specific modules (see Darms 2007).

3.4 Data Filtering

The feature hypotheses that have been extracted and associated with an object hypothesis are processed downstream by a filter or estimation algorithm. This algorithm is used to improve the information, but also to gain new information (see Bar-Shalom et al. 2001; Hänsler 1997). Examples include:

– Signal and noise separation
– Reconstructing state variables that cannot be measured directly

For an overview of filter algorithms for sensor data fusion, see Holt (2004), Klein (1999), and Bar-Shalom and Li (1995). The filter parameters are designed and configured to suit the optimization criteria that need to be defined for the individual application (see Hänsler 1997). If the filter is part of the control loop, it influences the dynamic behavior of the entire system (see, e.g., Lunze 2006; Föllinger 1990). In this case, the filter parameters must be adapted to suit the control loop's requirements (e.g., ACC). It is important to find a compromise between the filter dynamic and the achievable estimation accuracy (see Lunze 2006). If a state controller is used, the separation theorem (Lunze 2006; Föllinger 1990) at least ensures the stability of the overall system, assuming that the estimator is stable. The control and estimation parameters can be designed separately (see Lunze 2006; Föllinger 1990); this offers benefits in terms of architecture.

To save costs, the data from a multi-sensor system can be provided to various applications (see, e.g., Darms 2007; Dietmayer et al. 2005). It is important to consider the fact that, depending on the sensor accuracy, areas can exist in which various applications cannot be operated with a common filter algorithm or in which shared operation of the application with one filter necessitates finding a compromise that is not optimal for individual applications in terms of dynamics (see Darms 2007).

Development of data filtering algorithms cannot be completely abstracted from the data association design. This is true of the design process, in which mutually compatible algorithms must be found (see Bar-Shalom and Li 1995), and also of the runtime behavior, given that the data filtering dynamic influences the quality of the association process. Again, depending on the sensor accuracy, it is possible that different filter algorithms for applications and data association make sense (see Darms 2007).

3.5 Classification

During classification, object hypotheses are assigned to a predefined class on the basis of associated properties (see, e.g., Klein 1999). The properties can come from the sensor's raw data, but also from the estimated state variables of the object hypothesis.

In a multi-sensor system, the input data from various sensors are available. In terms of the architectural design, it is beneficial for the data included in the fusion process to be mutually orthogonal. A multiple implementation of a classification on the basis of state variables can be avoided, given an appropriate architecture design (see Sect. 4).

3.6 Situation Analysis

Situation analysis determines the overall behavior of the driver assistance system. For example, adaptive cruise control (ACC) is backed up by a state machine that defines the application's behavior in various scenarios (see, e.g., Mayr 2001).

Situation analysis is thus the link between environment sensor data processing and the assistance function. Algorithms for situation analysis need to consider both the capability of the environmental data acquisition system and the application's boundary conditions. In the case of automatic emergency braking, for example, a decision to intervene is taken as part of the situation analysis; this decision is driven both by the accuracy with which the potential collision object is estimated and by the potential, vehicle-specific evasion trajectories.

4 Architecture Patterns for Sensor Data Fusion

4.1 General Overview

The architecture documents the structure and the interactions between the individual components for the persons involved in developing the system (see Starke 2005). The architecture of the system also contributes toward structuring the development process (see Starke 2005). This is also true beyond corporate boundaries, as the architecture and the degree of coupling (see Vogel 2005) within the system influence the extent to which components can be manufactured by various suppliers.

There is no deterministic method that guarantees an optimal solution for developing an architecture (Starke 2005). The following section lists established, general architecture patterns in the field of sensor data fusion and discusses the benefits and drawbacks.

4.2 Decentralized–Centralized–Hybrid

The distinction into decentralized, centralized, and hybrid fusion relates to the module view of the system (Vogel 2005). It is based on the degree of data processing in the sensors, the results of data processing in the sensors, and the point at which the data are merged in the fusion process (Klein 1999). It is typically used in conjunction with tracking (Hall and Llinas 1997).

Figure 3 shows a **decentralized architecture**. This approach is referred to in the referenced literature as *sensor-level fusion, autonomous fusion, distributed fusion, or post-individual sensor processing fusion* (Klein 1999). The individual sensor modules handle object discrimination and tracking. The results are merged in a central module, possibly involving feedback of results from the centralized fusion to the sensors (Bar-Shalom and Li 1995). In this case, each decentralized module can additionally handle the central module functions, thus achieving redundancy (Bar-Shalom and Li 1995).

In terms of object discrimination, this type of architecture is optimal, given that the sensors are mutually orthogonal for this operation. This is the case, for example,

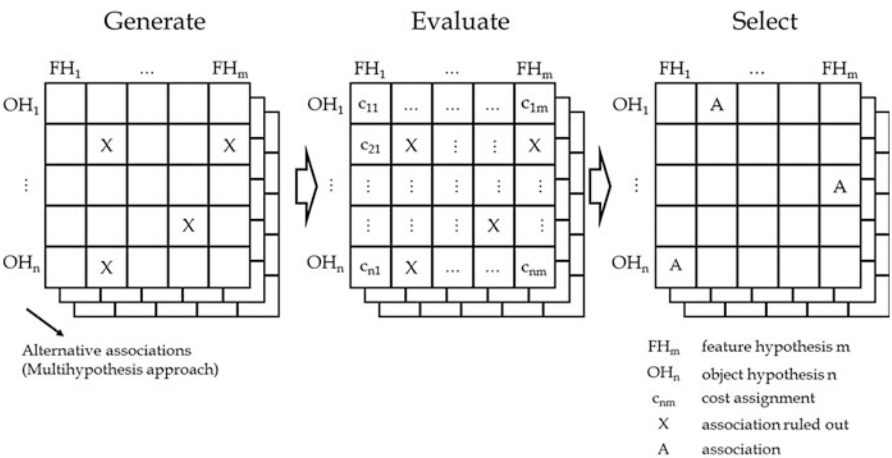

Fig. 3 Decentralized architecture (see Darms 2007, p. 16)

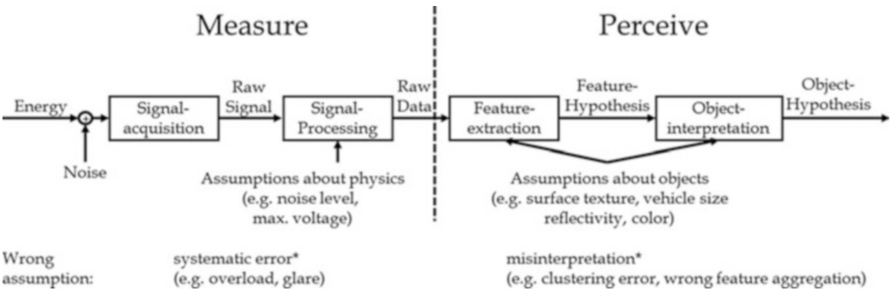

Fig. 4 Centralized architecture. (**a**) Fusion at raw data level. (**b**) Fusion at feature level (see Darms 2007, p. 17)

if sensor principles based on different physical effects are used that do not cause artifacts due to identical phenomena (Robinson and Aboutalib 1990). Two pieces of information are required for fusion: firstly the discrimination decision and secondly a metric for the decision quality (Klaus 2004).

The architecture can also be optimal for tracking, in the sense of minimizing the estimation error (Bar-Shalom and Li 1995). However, this is only true given relatively restrictive preconditions, which rarely exist in practical applications. If the sensors' measuring times also differ, again, the solutions are only approximately ideal in terms of the achievable accuracy (Bar-Shalom and Li 1995).

Figure 4 shows a **centralized architecture**. This is referred to in the referenced literature as *central-level fusion*, *centralized fusion*, or *pre-individual sensor processing fusion* (Klein 1999). The data only go through minimal preprocessing in the sensor modules (feature or raw data level) and are then merged in a centralized module, possibly involving feedback to the sensor modules (Klein 1999).

In terms of object discrimination, this type of architecture is superior to a decentralized architecture if the sensors are not mutually orthogonal. If the sensors are orthogonal, the results do not differ (Klein 1999).

A centralized architecture is optimal for tracking, without the restricting prerequisites that apply for a decentralized architecture. Additionally, measurements not taken at the same time can be optimally merged (Bar-Shalom and Li 1995).

The main drawbacks of a centralized architecture are firstly restrictions in terms of flexibility, as the internal algorithms of the central module may need to be modified to accommodate extensions, and secondly a higher data volume that occurs at the interfaces between the sensor modules and the fusion module (Klein 1999).

A **hybrid architecture** combines the centralized and decentralized approaches. In addition to minimally preprocessed data (raw data), data preprocessed by the sensors (tracks) can be fed to the central fusion module. Tracks can in turn provide input for a decentralized fusion module in the same system. The results from this decentralized module can flow into the central fusion module's fusion algorithm (Klein 1999).

As an example of the use of hybrid architecture, Bar-Shalom and Li describe a scenario that is broken down into various acquisition areas, each of which is covered by a multi-sensor platform. A centralized architecture is used within the platform, while the overall estimation is determined by a decentralized architecture across the areas (Bar-Shalom and Li 1995).

4.3 Raw Data Level–Feature Level–Decision-Making Level

The distinction into fusion at raw data level, feature level, and decision-making level relates to the resolution of the data fed to the fusion algorithm and the degree of sensor data preprocessing (Klein 1999). It thus relates to the runtime view (Starke 2005) and is typically used in the context of object discrimination algorithms (Hall and Llinas 1997).

In the case of **fusion at raw data level**, minimally preprocessed data that exist at the resolution of the sensors involved (e.g., pixels in image processing) are fused in a centralized architecture. This means that, for example, information from various spectra (infrared, visible light) can be fused prior to image processing (Klein 1999). The advantage this approach offers is the availability of complete sensor information to which the fusion algorithm can be adapted. The main disadvantages are the large data volume between the sensors and the centralized module, as well as the difficulty of changing and extending the optimized algorithms in the centralized module.

In the case of **fusion at feature level**, the features are first extracted before fusion is performed. In a centralized architecture, this reduces the communication bandwidth between the sensor modules and the central module at a price of losing information.

Fusion at decision-making level is equivalent to a decentralized architecture. In contrast to fusion at feature level, object discrimination is already performed in the sensor modules. The results are then merged together with the tracking information in a centralized module (Klein 1999). Tracking in this case does not need to follow the decentralized architecture principle.

Table 1 summarizes the architectural principles decentralized–centralized–hybrid and raw data level–feature level–decision-making level, as well as their dependencies.

4.4 Synchronized–Unsynchronized

In terms of the system's dynamic interaction, a distinction can be made between synchronized and unsynchronized sensors. The distinction relates to the temporal sequence in which the data are acquired by the sensors (see, e.g., Bar-Shalom and Li 1995; Narbe et al. 2003a; Narbe et al. 2003b; Mauthener et al. 2006).

In **synchronized sensors** data acquisition is temporally aligned. Synchronous sensors are a special case of synchronized sensors in which data acquisition occurs

Table 1 Fusion architectures (see Darms 2007, p. 19, following Hall and McMullen 2004, pp. 360–361; see also Klein 1999, p. 73)

Type	Description	Fusion level	Comment
Centralized	Raw data fusion	Raw data	Minimal information loss In comparison, needs the greatest communication bandwidth between the sensor modules and the centralized module Optimal for orthogonal and non-orthogonal sensors
	Feature fusion	Feature	Requires lower communication bandwidth than fusion at raw data level Information loss due to feature extraction The benefits of fusion at raw data level cannot be leveraged for non-orthogonal sensors
Decentralized	Fusion of state variables and discrimination decisions	Decision-making level	Information loss due to feature extraction Optimal object discrimination for orthogonal sensors Optimal tracking only under restrictive conditions Dependency on the results determined in the sensor modules must be taken into consideration on fusion Redundancy can be achieved by allowing multiple decentralized modules to compute the fusion
Hybrid	Combination of centralized and decentralized	Combination possible on all levels	Combines the properties of centralized and decentralized architecture High complexity of the architecture in comparison

simultaneously. With **unsynchronized sensors**, data acquisition occurs in an individual sensor cycle that is not aligned with the other sensors and does not need to be constant.

The drawback of synchronization is the additional overhead in terms of hardware and possibly software; the advantage is that the system's timing behavior is already known at the design stage (see also Kampchen and Dietmayer 2003).

4.5 New Data–Data Constellation–External Event

Events which cause data fusion to be performed can be grouped into three classes: the occurrence of new data, the occurrence of a specific data constellation, and the occurrence of an external event.

If fusion occurs whenever **new data** occur, no information is lost. Depending on whether synchronized or unsynchronized sensors are used, the fusion process needs to find solutions for processing data that do not arrive at the fusion modules in the temporal sequence of data acquisition (Stüker 2004; Bar-Shalom 2002). In a decentralized structure, the latest fusion data can be fed back to the sensors so that the sensors always have the latest prediction, for example, for preconditioning algorithms.

If the fusion process always occurs for **specific data** constellations, for example, whenever the data for specific sensors occur, then data caching resources must be reserved. Additionally, the fusion data are not available at the earliest possible point in time. If unsynchronized sensors are used, a decision as to the filtering state in which the data are input into the fusion process must be made (see Sect. 4.6).

If the results of a centralized fusion module are not fed back to the sensors, fusion can be triggered at arbitrary points in time by an **external event**. This allows the data rate to be accommodated to match downstream processing, thus allowing for the resources to be accommodated. However, in terms of tracking accuracy, this is not an optimal solution (Bar-Shalom and Li 1995).

4.6 Original Data–Filtered Data–Predicted Data

In terms of the filtering state of data input into the fusion process, a distinction can be made between original data, filtered data, and predicted data.

In case of **original data**, the temporally unfiltered data are fed into the fusion process. This allows for optimal tracking.

If **filtered data** are used (e.g., in a decentralized architecture), optimal tracking can be achieved under restrictive conditions. However, if the filtered data are treated like unfiltered data and passed to a further filter for estimation, a chain of filters is established. This generally leads to higher signal propagation delay. Additionally, errors are now correlated; for an optimal estimation result, the filter model needs to take this into consideration.

It is also possible to use **predicted data** (e.g., on the basis of models). This approach is often used to relate the acquired measurement data to a point in time when a specific data constellation occurs and to consolidate the different measurements to so-called super measurements. Bar-Shalom and Li are of the opinion that this approach does not lead to optimal results in terms of the achievable estimation error for unsynchronized sensors (Bar-Shalom and Li 1995).

4.7 Parallel–Sequential

Another distinction which can be found in the referenced literature relates to the fusion algorithm. A distinction is made here between **parallel fusion**, where fusion of the existing measurements occurs in a single step, and **sequential fusion**, where the measurements are merged in multiple, sequential steps. If the systems are linear and the sensors are synchronized, then the two methods are equivalent (Bar-Shalom and Li 1995).

Dietmayer et al. also refer to explicit fusion in the case of synchronized sensors and parallel fusion and to implicit fusion for unsynchronized sensors and sequential fusion (Dietmayer et al. 2005).

5 Conclusions

In the author's opinion, data fusion is essential for meeting the requirements for future driver assistance systems and automated vehicles. This is particularly true of systems designed to improve safety.

Given an appropriate architecture design, the sensor fusion data system can represent an abstraction of the environmental perception of the deployed sensors. The applications can thus be developed independently of the environment acquisition system. The design of the situational analysis as an interface between fusion and application plays a central role then.

However, experience also shows that the equation "more sensors equals a better system" does not apply without restrictions in practical applications. For example, the overall system complexity increases with each sensor. Each sensor adds sensor-specific properties to the system. If this is not modeled or taken into consideration with a sufficient degree of accuracy, it may still be possible to partially improve certain aspects, but will at the same time impact the overall performance. Hall (2001) provides an overview of the typical pitfalls in designing a multi-sensor system.

References

Bar-Shalom Y (2002) Update with out-of-sequence measurements in tracking: exact solution. IEEE Trans Aerosp Electron Syst 3:769–777

Bar-Shalom Y, Li XR (1995) Multitarget-multisensor tracking – principles and techniques. YBS, Storrs

Bar-Shalom Y, Li XR, Kirubarajan T (2001) Estimation with applications to tracking and navigation – theory, algorithms and software. Wiley, New York

Becker JC (2002) Fusion der Daten der objekterkennenden Sensoren eines autonomen Straßenfahrzeugs (Fusion of data from object detecting sensors in an autonomous road vehicle). VDI-Verl, Düsseldorf

Bender E et al (2007) Anti collision system proreta – on the way to the collision avoiding vehicle – part 1: basics of the system. ATZ Wordwide 109(4):20–23

Darms M (2007) Eine Basis-Systemarchitektur zur Sensordatenfusion von Umfeldsensoren für Fahrerassistenzsysteme (A baseline system architecture for sensor data fusion of environment sensors for driver assistance systems) Fortschrittberichte VDI, Series 12, vol 653

Darms M, Rybski P, Urmson C (2008) Vehicle detection and tracking for the urban challenge, AAET 2008, 9th symposium, 13/14 Feb 2008, Braunschweig

Darms M, Rybski P, Baker C, Urmson C (2009) Obstacle detection and tracking for the urban challenge. IEEE Trans Intell Transp Syst 10(3):475–485

Dietmayer K, Kirchner A, Kampchen N (2005) Fusionsarchitekturen zur Umfeldwahrnehmung für zukünftige Fahrerassistenzsysteme (Fusion architectures for environmental perception in future driver assistance systems). In: Maurerer M (ed) Fahrerassistenzsysteme mit maschineller Wahrnehmung (Driver assistance systems with mechanical perception). Springer, New York, pp 59–87

Föllinger O (1990) Regelungstechnik – Einführung in die Methoden und ihre Anwendung (Control theory – introduction to methods and their application), 6th issue. Hüthig Buch Verlag, Heidelberg

Hall DL (2001) Handbook of multisensor data fusion. CRC Press, Boca Raton

Hall D, Llinas J (1997) An introduction to multisensor data fusion. Proc IEEE 85(1):6–23

Hall DL, McMullen SA (2004) Mathematical techniques in multisensor data fusion, 2nd edn. Artech House, Boston

Hänsler E (1997) Statistische Signale: Grundlagen und Anwendungen (Statistical signals: basics and applications), 2nd issue. Springer, Berlin

Holt Vv (2004) Integrale multisensorielle Fahrumgebungserfassung nach dem 4D-Ansatz (Integral multi-sensorial environmental acquisition based on the 4-D approach), Diss. Univ. der Bundeswehr, Munich (online publication), URL: urn:nbn:de:bvb:706–1072. http://athene-forschung.unibw.de/doc/85338/85338.pdf. Accessed 31 Jan 2015

Joerg KW (1994) Echtzeitfähige Multisensorintegration für autonome Mobile Roboter (Real-time capable multi-sensor integration for autonomous mobile robots). BI-Wiss.-Verl., Mannheim

Kampchen N, Dietmayer K (2003) Data synchronization strategies for multi-sensor fusion. In: 10th world congress on intelligent transport systems. Band Proceedings of IST 2003. Madrid

Klaus F (2004) Einführung in Techniken und Methoden der Multisensor-Datenfusion (Introduction to the techniques and methods of multi-sensor data fusion). Habil.-Schr. Univ. Siegen, online publication, URL: urn:nbn:de:hbz:467–575; http://www.ub.uni-siegen.de/pub/diss/fb12/2003/klaus/klaus.pdf. Accessed 31 Jan 2015

Klein LA (1999) Sensor and data fusion concepts and applications, 2nd issue, vol 35. SPIE, Bellingham

Lou RC, Kay MK (1991) Multisensor integration and fusion in intelligent systems. In: Autonomous mobile robots, Bd. 1. IEEE Computer Society Press, Los Alamitos

Lunze J (2006) Regelungstechnik Mehrgrößensysteme, digitale Regelung (Control theory on multiple value systems, digital control), vol 2. Springer, Berlin

Mauthener M et al (2006) Out-of-sequence measurements treatment in sensor fusion applications: buffering versus advances algorithms. In: Stiller C, Maurer M (Hrsg) 4. Workshop Fahrerassistenzsysteme FAS2006 (Driver assistance systems workshop), pp 20–30. fmrt, Karlsruhe

Mayr R (2001) Regelungsstrategien fur die automatische Fahrzeugführung: Längs- und Querregelung, Spurwechselund Überholmanöver (Control strategies for automatic vehicle guidance: longitudinal and latitudinal control, lane changing and overtaking manoeuvres). Springer, Tokyo

Narbe B et al (2003a) Datennetzkonzepte für die Sensordatenfusion – Teil 1 (Data network concepts for sensor data fusion – part 1). Elektronik Automotive 4:54–59

Narbe B et al (2003b) Datennetzkonzepte für die Sensordatenfusion – Teil 2 (Data network concepts for sensor data fusion – part 2). Elektronik Automotive 5:40–44

Robinson G, Aboutalib A (1990) Trade-off analysis of multisensor fusion levels. In: Proceedings of the 2nd national symposium on sensors and sensor fusion, No 2, Chicago, IL, pp 21–34

Schopper M, Henle L, Wohland T (2013) Intelligent Drive Vernetzte Intelligenz für mehr Sicherheit (Intelligent Drive – networked intelligence for improved safety). ATZExtra 5:106–114

Starke G (2005) Effektive Software-Architekturen – Ein praktischer Leitfaden (Effective software architectures – a practical guide), 2nd issue Hanser, Wien

Steinberg A, Bowman C, White F (1998) Revisions to the JDL data fusion model. Quebec City

Streller D (2006) Multi-Hypothesen-Ansatz zur Erkennung und Verfolgung von Objekten in Verkehrsszenen mit Laserscannern (Multi-hypotheses approach for detection and tracking of objects in traffic scenarios with laser scanners). VDI-Verl, Düsseldorf

Stüker D (2004) Heterogene Sensordatenfusion zur robusten Objektverfolgung im automobilen Straßenverkehr. (Heterogenous sensor data fusion for robust object tracking in automotive road traffic), Diss. Univ. Oldenburg, (online publication), URL: urn:nbn:de:gbv:715-oops-2319; http://oops.uni-oldenburg.de/201/1/stuhet04.pdf. Accessed 31 Jan 2015

Vogel O (2005) Software-Architektur – Grundlagen – Konzepte – Praxis (Software architecture – basics – concepts – practical applications), 1st issue Elsevier, Spektrum, Akad. Verl., Munich