Technische Universität Wien

Magisterarbeit

# Extended Confidence-Weighted Averaging in Sensor Fusion

Ausgeführt am
Institut für Technische Informatik 182
der Technischen Universität Wien

unter der Anleitung von

O.Univ.-Prof. Dr. Hermann Kopetz
und
Univ.Ass.Dipl.-Ing. Dr.techn. Wilfried Elmenreich
als verantwortlich mitwirkenden Assistenten

durch

Angela Schörgendorfer
Matr.-Nr. 0125319
Kolbegasse 44 Haus 9, 1230 Wien

Wien, im Mai 2006 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

# Extended Confidence-Weighted Averaging in Sensor Fusion

Sensor fusion refers to the practice of combining information gathered by multiple sensors to receive a more accurate, dependable and comprehensive image of a system's environment. The need for such strategies arises from the fact that single sensors have limited dependability since they are subject to environmental interference or hardware noise, as well as possible sensor failure. A system depending on sensor information should therefore not rely on the information from a single sensor.

This thesis discusses the currently established techniques for sensor fusion, focussing on methods for merging raw sensor data. Analyzing the various shortcomings of established methods, a new method for stateless fusion of raw sensor data is presented — the technique of extended confidence-weighted averaging. It is based on the previously suggested confidence-weighted averaging method, but extends it by including known correlations between the error distributions of the sensors. The benefits expected from this new method are an increase in the accuracy of the fusion result as well as a more dependable estimate of the uncertainty associated to the fused result.

An evaluation on real sensor data collected using a mobile robot analyzes the improvements achieved by the newly presented method.

# Erweiterte Konfidenzgewichtete Mittelwerte in Sensor Fusion

Sensor Fusion bezeichnet die Zusammenführung von Informationen, die von mehreren verschiedenen Sensoren eines Systems ermittelt wurden, mit dem Ziel, eine genaueres, verlässlicheres und umfassenderes Bild der Umwelt zu erhalten. Jeder einzelne Sensor ist nur bedingt zuverlässig, da er von Faktoren wie Umwelteinflüssen oder zufälligem Rauschen beeinflusst wird, oder gar ausfallen kann. Relevante Funktionen und Entscheidungen sollten daher nicht auf Informationen eines einzigen Sensors basieren.

Diese Arbeit betrachtet derzeitige Techniken der Sensor Fusion, im Speziellen der Vereinigung von Sensorrohdaten. Ausgehend von dabei identifizierten Schwächen wird eine neue Methode für die zustandslose Fusion von Sensorrohdaten präsentiert — die der erweiterten konfidenzgewichteten Mittelwerte. Basierend auf der Methode der konfidenzgewichteten Mittelwerte berücksichtigt dieser neue Algorithmus zusätzlich Korrelationen zwischen den Fehlern der einzelnen Sensoren. Die Vorteile, die daraus zu erwarten sind, sind eine Verbesserung der Genauigkeit des Ergebnisses einerseits und eine verlässlichere Schätzung der verbleibenden Unsicherheit andererseits.

Die Verbesserungen durch diesen neuen Ansatz gegenüber bereits bestehenden werden schließlich anhand Analysen von Messungen, die von einem mobilen Roboter durchgeführt wurden, gezeigt.

# Acknowledgements

I want to thank Prof. Dr. Hermann Kopetz for giving me the opportunity to participate in the research conducted at the Institute of Computer Engineering, Real-Time Systems Group, at the Vienna University of Technology, where I realized the work of this thesis.

I am indebted to my faculty mentor Wilfried Elmenreich and want to thank him for his scientific advice and research suggestions, as well as for his never-failing ability to motivate and challenge me anew. Furthermore, I owe thanks to Gottfried Fuchs and Christian El Salloum for their valuable suggestions for improvements of this text.

I am also very much obliged to my family, who helped me during my work as a source of motivation and often shielded me from the concerns of everyday life that would have otherwise not made it possible for me to concentrate on my work like I did.

Finally, I want to thank Kyle Mullikin for his patience and support, and for his finishing refinements of the linguistic aspects of this work.

# Contents


**1 Introduction** **1**

1.1 Related Work . . . 2
1.2 Motivation and Objectives . . . 3
1.3 Structure of this Thesis . . . 4

**2 Basic Terms and Concepts** **5**

2.1 Dependability and Uncertainty . . . 5
2.1.1 Sensor Impairments . . . 7
2.1.2 Limitations of Sensor Information . . . 8
2.1.3 Representing Uncertainty . . . 10
2.1.4 Limitations of Sensor Fusion . . . 13
2.2 Sensor Fusion Strategies . . . 14
2.2.1 Types of Sensor Fusion . . . 16
2.2.2 Fusion Architectures . . . 19
2.2.3 Fusion Models . . . 22
2.3 Real-Time Aspects . . . 27
2.3.1 Model of Time . . . 27
2.3.2 The Time-Triggered Architecture . . . 29
2.4 Chapter Summary . . . 30

**3 Raw Data Fusion** **32**

3.1 Filtering Methods . . . 32
3.1.1 Kalman Filtering . . . 32
3.1.2 Covariance Intersection . . . 35
3.2 Interval Intersection . . . 38
3.2.1 Fault-Tolerant Sensor Averaging . . . 38
3.2.2 Fault-Tolerant Interval Intersection . . . 39
3.3 Confidence-Weighted Averaging . . . 40
3.4 Chapter Summary . . . 42

**4 Introducing Correlations** **43**
4.1 Motivation for Introducing Correlations . . . . . . . . . . . . . 43
4.2 Extended Confidence-Weighted Averaging . . . . . . . . . . . 46
4.3 Fault Tolerance . . . . . . . . . . . . . . . . . . . . . . . . . 48
4.4 Implementational Aspects . . . . . . . . . . . . . . . . . . . . 50
4.5 Chapter Summary . . . . . . . . . . . . . . . . . . . . . . . . 51

**5 Experimental Evaluation** **52**
5.1 Data Collection . . . . . . . . . . . . . . . . . . . . . . . . . 52
5.1.1 Smart Car Architecture . . . . . . . . . . . . . . . . . 52
5.1.2 Sensors . . . . . . . . . . . . . . . . . . . . . . . . . . 53
5.1.3 Experimental Setup . . . . . . . . . . . . . . . . . . . 57
5.2 Performance Evaluation . . . . . . . . . . . . . . . . . . . . . 59
5.2.1 Comparison of CWA and ECWA . . . . . . . . . . . . 59
5.2.2 Performance of Fault-Tolerant Approach . . . . . . . . 62
5.3 Chapter Summary . . . . . . . . . . . . . . . . . . . . . . . . 70

**6 Conclusion** **71**
6.1 Extended Confidence-Weighted Averaging . . . . . . . . . . . 71
6.2 Outlook . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72

**A Symbols** **73**
A.1 Algebraic Symbols . . . . . . . . . . . . . . . . . . . . . . . . 73
A.2 ECWA-specific notation for $n$ sensors . . . . . . . . . . . . . . 73

**B Results of experimental evaluation** **74**

**Bibliography** **77**

# List of Figures


2.1 Overlapping of fusion terminology . . . . . . . . . . . . . . . . 15
2.2 Fusion categorization based on input/output data . . . . . . . 19
2.3 Fusion architectures . . . . . . . . . . . . . . . . . . . . . . . . 20
2.4 The JDL fusion model . . . . . . . . . . . . . . . . . . . . . . . 23
2.5 The waterfall fusion process model . . . . . . . . . . . . . . . . 25
2.6 The Omnibus Model . . . . . . . . . . . . . . . . . . . . . . . . 27
2.7 The time-triggered sensor fusion model . . . . . . . . . . . . . 28
2.8 Structure of a TTP/A transducer cluster . . . . . . . . . . . . 29

3.1 Intersection of covariance ellipses . . . . . . . . . . . . . . . . 36
3.2 Covariance intersection vs. largest ullipsoid solution . . . . . . 37
3.3 Interval intersection . . . . . . . . . . . . . . . . . . . . . . . . 38
3.4 Interval intersection after a slight shift in inputs . . . . . . . . 40

4.1 Effect of independence or correlations on the fused distribution 45
4.2 Exclusion of outliers independent of their weight . . . . . . . . 48

5.1 The autonomous mobile robot Smart Car . . . . . . . . . . . . 53
5.2 Four-level architecture of the smart car . . . . . . . . . . . . . 54
5.3 Response of sensor IR 1 for given distances . . . . . . . . . . . 55
5.4 Response of ultrasonic sensor for given distances . . . . . . . . 57
5.5 Test Setup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
5.6 Deviation of estimated variance from true fused variance . . . 60
5.7 Error distributions over true distance . . . . . . . . . . . . . . 61
5.8 2-dimensional density distribution . . . . . . . . . . . . . . . . 62
5.9 Comparison of fusion performance on sensor data . . . . . . . 64
5.10 Performance of ECWA for different limits of $d_{ECWA}$ . . . . . . 66
5.11 Performance of ECWA for a varying number of tolerated faults 66
5.12 Performance of four methods on simulated data . . . . . . . . 68
5.13 Performance of ECWA on simulated data . . . . . . . . . . . . 69
5.14 Performance of ECWA for different fault-tolerance strategies . 69

B.1 Comparison of fusion performance for varying number of tolerated faults . . . . . . . . . . . . . . . . . . . . . . . . . . . 75

# List of Tables


4.1 Algorithm characteristics for the fusion of four inputs . . . . . 50

5.1 Sensor constants determined for calibration . . . . . . . . . . . . 56
5.2 Tested sensor configurations . . . . . . . . . . . . . . . . . . . . . 58
5.3 Error variance of each sensor for tested sensor configurations . 59
5.4 Comparison of fusion results for CWA and ECWA . . . . . . . . 60
5.5 Performance of tested fusion algorithms . . . . . . . . . . . . . . 65

B.1 Comparison of fusion results for different sensor configurations 74
B.2 Performance of tested fusion algorithms . . . . . . . . . . . . . . 76

*On résiste à l'invasion des armées;*
*on ne résiste pas à l'invasion des idées.*

Victor Hugo

# Chapter 1

# Introduction

Systems that are designed to function in an environment whose state can not previously be determined or that changes over the course of time rely on sensors to obtain information about their surroundings. Since sensors are subject to errors, uncertainties, and mechanical failures, a system should not base its perception of the environment on a single sensor. Sensor fusion addresses the problem of such impairments by employing multiple data sources and combining their information to a more accurate, reliable, and comprehensive representation of the environment.

Sensor fusion is not a merely artificial process, but is naturally performed by the brains of humans and animals: sensory information from all senses is combined to complete our perception of our surroundings. There have been attempts to model robotic sensor fusion applications according to neuropsychological and cognitive models of fusion processes in the human brain. But even those approaches that are not based on biological foundations, show some resemblance to observed cognitive processes in humans or animals, such as sensor-specific representation of information that is converted to a common representation before the fusion process, the coupling of sensor fusion with action, and the use of contextual information [Mur96].

Multisensor systems and techniques to combine sensor information are becoming increasingly relevant in various fields of application. Research and development in the field of sensor fusion was first mainly conducted in the military context. There, multisensor systems were employed to support automated detection of other military entities such as ships, tanks, or planes, for friend-foe-neutral identification (IFFN), and the location and tracking of objects. However, the approach has long since been adopted for non-military applications and today much research is conducted with strictly civilian objectives. Non-military applications of sensor fusion include systems for air traffic control, robotics, manufacturing, medical diagnosis, and remote sensing [Var00]. Accordingly, multisensor systems take a variety of shapes, ranging from large scale distributed wireless networks with possibly non-static topology, to safety-critical real-time embedded systems.

## 1.1 Related Work

Research on sensor fusion can roughly be categorized in terms of algorithms, fusion models or architectures, and applications.

Sensor fusion algorithms include filtering (e.g., the Kalman filter [Kal60] and extensions thereof [Jul97]), agreement (e.g., voting, approximate agreement [Aza96], fault-tolerant abstract sensors [Mar90, Sch01], sensor averaging [Elm02]), world modelling (e.g., occupancy grid [Elf89, Ste05], geometric map building [Mat98], or topological maps [Kün05]), and decision methods such as Bayesian methods, Dempster-Shafer reasoning, fuzzy decision methods [Bro98], as well as neural networks [Lew04].

The architectures of the systems implementing such fusion methods are various. In an effort to standardize the design process, various models have been proposed, among them the JDL fusion model [Wal90], the waterfall model [Mar97] and the time-triggered sensor fusion model [Elm01].

The applications of sensor fusion span from robotic navigation and control [Chr00, Kli06], across monitoring, e.g., in nuclear power plants [Meu95], automotive and aircraft control, and medical diagnosis [Ebr97], to large-scale distributed sensing systems such as for meteorological observation or plant and crop monitoring [Bri05]. Sensor fusion has also been employed for enhancing human-machine interfaces, like the combination of a camera and a microphone in audiovisual speech recognition [Bre94, Che98, Lew04] or that air- and bone-conductive microphones [Sub05]. Military applications mainly concentrate on battlefield surveillance, employing methods for object classification, target detection and tracking [She05] and decision support [Sar91].

## 1.2 Motivation and Objectives

The intention of this thesis is to develop a method for the stateless fusion of raw sensor data. The need for a new methods arises from the fact that established stateless fusion methods assume that error behavior does not show any cross-correlations between sensors. This assumption of independence simplifies the algorithms, but generally does not reflect the true sensor behavior.

Fusion without the knowledge about existing cross-correlations between sensor errors may lead to a suboptimal fusion result, and furthermore cause overly great confidence in the accuracy of the result. In the case of independent sources, the fused result can be expected to retain a reduced amount of uncertainty compared to the original data. If the condition of independence is violated, however, the informational value is altogether not as great as expected, so that in extreme cases uncertainty may not be reduced at all.

Filtering methods that do incorporate correlated error behavior, like the Kalman filter, have been proposed for sensor fusion. The drawbacks of such methods are for one the often intricate model of the observed system that the filters are based on. If the system environment is too dynamic or a significant factor has not been modelled or can not be observed, filter performance may deteriorate [Zha04]. Additionally, filtering has a delaying effect on the detection of changes, since variations in the location parameters are smoothed [Sch06].

Another drawback of filtering methods lies in their implementation in systems that employ replicated fusion nodes as means of achieving fault-tolerance. If the filter is to be applied by various nodes, all of them require the same information about previous states in order to deliver consistent fusion results. In the case of a node failure, and its subsequent repair and reintegration, the information that all other nodes have meanwhile accrued needs to be communicated to the integrating replica [Bau01]. Even though such reintegration schemes can be and have commonly been implemented in fault-tolerant systems, the communication effort may not be feasible for some applications in the light of other disadvantages of filtering methods.

## 1.3 Structure of this Thesis

This thesis is organized as follows:

Chapter 2 will give an overview on the basic concepts of sensor fusion. The main impairments of the quality of sensor information and the resulting motivation for sensor fusion, as well as its limitations are treated in section 2.1. Section 2.2 presents various established strategies for integrating sensor information. Aspects of the employment of sensor fusion in real-time systems are discussed in section 2.3.

Chapter 3 is devoted to the discussion of established methods for the particular field of low-level data fusion. These methods comprise filtering methods discussed in 3.1, as well as methods based on interval intersection in section 3.2 and a method of weighted averages in section 3.3.

Chapter 4 introduces extended confidence-weighted averaging as a new method for stateless data fusion. Section 4.1 justifies the development of such a method by pointing out the shortcomings of established methods. The derivation of the new approach is outlined in section 4.2. Sections 4.3 and 4.4 further present considerations concerning fault tolerance and implementational issues, respectively.

To compare the newly introduced method to other established approaches, chapter 5 presents an analysis of their performance on real sensor data provided by a set of distance sensors mounted on a mobile robot. Section 5.1 describes the experimental setup, while section 5.2 discusses the results of the experimental evaluation on real sensor data.

The thesis concludes with a summary of the presented results and an outlook on future research in chapter 6.

*Life is so constructed, that the event does not, cannot, will not, match the expectation.*

CHARLOTTE BRONTË

# Chapter 2

# Basic Terms and Concepts

This chapter will first present motivations for sensor fusion and its limitations and further give an overview of principles and strategies employed in sensor fusion systems.

## 2.1 Dependability and Uncertainty

Sensor fusion is one of various possible strategies to improve a system's dependability. *Dependability is that property of a computer system that allows reliance to be justifiably placed on the service it delivers*[Avi86, p.1], and encompasses the following five attributes [Avi04]:

- **Availability**, which refers to a system's readiness for correct service.
- **Reliability** concerning the continuity of correct service.
- **Safety** concerns the absence of catastrophic consequences.
- **Integrity** referring to the absence of improper system alterations.
- **Maintainability** as the ability to undergo modifications and repairs.

The concept of dependability reaches thus much further than the aspect of reliable and good quality data, which sensor fusion is mainly concerned with, and concerns software as much as hardware, architecture, security or communication issues.

Factors that impair the dependability of a system are faults, errors and failures. A *fault* is the cause of an error. Avižienis and Laprie suggest eight elementary fault classes which, among other criteria, distinguish faults according to *system boundaries* (internal or external faults), *phenomenological cause* (natural and human-made faults), *objective* (malicious and non-malicious faults) and *persistence* (permanent and transient faults).

An *error* is the part of a system's state that may cause the delivered service to deviate from correct service and thus cause a failure. Errors may be *detected* and their presence indicated by an error message or error signal, while otherwise they are *latent*. Kopetz [Kop97] identifies *transient* errors, which only exist for a short interval of time and do not require an explicit repair action, and *permanent* errors which persist until a repair action removes them.

*Failures* are deviations from the specified behavior of a system. The ways in which service may deviate can be summarized in four *service failure modes* [Avi04]:

**Failure Domain:** Failure may occur in the *content domain* or the *timing domain* or in both. When both information and timing are incorrect, two failure classes may be distinguished: *halt failures*, when the service of a system is halted or no service is delivered at all, and *erratic failures*, when a service is delivered but erratic (e.g. babbling).

**Failure Detectability:** Detectability addresses the signaling of failures to the user. If a failure is detected and indicated by a warning signal, it is a *signaled* failure, otherwise *unsignaled*.

**Consistency:** In the case that a system has more than one user, the way these users perceive failures allows to distinguish between *consistent* and *inconsistent* failures. Consistent failures are perceived identically by all users, while inconsistent failures or *Byzantine failures* have different appearance to different users.

**Consequences:** Depending on their severity, failures can be characterized by application-specific severity levels, ranging from minor to catastrophic failures.

To reduce the risk of degradation of a system's service due to such impairments, methods of fault prevention, fault tolerance, fault removal and fault forecasting can be employed. Particularly, sensor fusion is a practice intended to increase fault tolerance through redundancy. Sensor fusion algorithms may identify faulty sensors through specific mechanisms of fault detection such as proposed in [Tai02], but mainly mask the effect of undetected sensor errors through the combination of multiple measurements.

### 2.1.1 Sensor Impairments

Sensor measurements can be significantly impacted by a number of environmental effects and internal influences. The most commonly cited influences on sensors are temperature and humidity, though there are others such as electromagnetic fields, background noise, acoustic or seismic effects, etc. [Swa05].

The errors in sensor measurement are distinguished into systematic and random errors, which are coped with in different ways.

#### Systematic Errors

A systematic error or *bias* occurs when the measurement value shows a recurring deviation in one direction from the true value, so that the mean error is not equal to zero. The offset to the true value does not have to be constant, but can be the result of a time-invariant function of the correct value [Kou03]. For example, a sensor may always return a measurement 20% above the true value.

Systematic errors can be caused by an incorrect conversion of the measured attribute to the desired quantity, or changes over time like ageing of the sensors or a change of environmental conditions. They can be compensated through calibration of the sensing instrument or explicit consideration of bias parameters in consequent processing of the measurement.

#### Random Errors

Random error or *noise* in measurement values can be attributed to sources like random hardware noise, inaccuracies in the measurement technique, or environmental interference. The concrete magnitude and incidence of such an error is not predictable, but one can observe its statistical behavior. The *expected value* of random errors is zero, since any other value would indicate a systematic behavior of the error. Due to its stochastic nature it is not possible to eliminate random error of a measurement result, but its effects can be reduced by increasing the number of observations.

Depending on their statistical properties, sensor errors can be classified as either nominal or artifactual [Ebr97]. A *nominal* error is within an acceptable error limit and has fixed statistical characteristics, as for example random measurement noise from sensor imprecisions would have. Such noise can be reduced by fusing observations from independent sources so that the noises may cancel each other out. An *artifactual* error or outlier, however, exceeds this acceptable error limit, is non-stationary and has unknown statistical properties. Among possible sources for such errors are arbitrary interference from the environment or deterministic effects caused by unobserved parameters. Fault tolerance or fault detection mechanisms may to be introduced into the fusion process to mask artifactual outliers or to detect and discard them.

### 2.1.2 Limitations of Sensor Information

There are fundamental limitations on any attempt to build a description of the environment based on the information from a single source beyond sensor errors or failure. The reliance on a single sensor suffers from the following shortcomings [DW88, Elm02]:

**Sensor Deprivation:** Due to a failure of a sensor the desired object can not be observed correctly.

**Limited Spatial Coverage:** An individual sensor can only render information about a limited region of the systems environment.

**Limited Temporal Coverage:** The frequency of measurements is limited by a sensor-specific set-up time to perform and transmit a measurement. It is therefore not possible to provide a continuous observation, but only measurements at discrete points in time.

**Imprecision:** The precision to which a quantity can be measured is limited by the resolution of the employed sensor. Resolution refers to the ability to detect and correctly report small enough changes of the observed variable [Hof05].

**Uncertainty:** Uncertainty is an effect caused by properties of the observed object. It arises when features may be missing, when not all relevant attributes can be observed, or if the observations are ambiguous.

Limitations on spatial and temporal coverage are known at the time of designing the system and can be explicitly compensated by employing multiple sensors and algorithms that combine sensor data to a more detailed or extensive model of the environment. Sensor deprivation and uncertainties in the observed object, on the other hand, can not be predicted. Also, random interference can cause noise in the measurement. To reduce a system's sensitivity to such random influences it is essential to employ fusion methods that are robust regarding missing or inaccurate data. Sensor fusion offers a solution to the problems listed above. The advantages that can be expected from using multiple sensors and combining their information are the following [Gro98, Mut98, Bos96]:

**Robustness and Reliability:** Through overlapping sensor domains sensor failure does not lead to catastrophic failure of the entire system, but the system undergoes graceful degradation.

**Extended spatial and temporal coverage:** Sensors can observe the same property in different regions of the environment, and their information can be combined to receive a more extensive picture of the property's distribution. On the other hand, if two sensors measure the same property with the same frequency but with an temporal offset, the number of measurements taken within a set interval of time can be doubled.

**Increased confidence:** A measurement taken by one sensor can be validated by other sensors' observations of the same object.

**Reduced ambiguity and uncertainty:** More information reduces the possibilities of ambiguous interpretation of the data.

**Robustness against interference:** By measuring the desired quantity with various sensors that employ different methods of measurement, e.g., ultrasonic and infrared sensors for gaging distances, the result is less sensitive to interference.

**Enhanced Resolution:** The combination of multiple independent measurements can yield a greater resolution than single sensors.

**Increased Dimensionality:** Different sensors may observe different features of the measurement space and reduce the vulnerability to missing or corrupted information of the measurement space [Wal90].

### 2.1.3 Representing Uncertainty

No sensor measurement is 100% reliable and exact but there are differences in how great the expected accuracy and dependability of different sources is. Sensor fusion is generally not a cause in itself, but provides its results as input to subsequent fusion or decision making processes. Any representation of sensor data should therefore contain some measure of the data's quality, as *"In many cases, knowing the reliability of a reading is just as important as knowing the exact value returned by a sensor."* [Bro98, p.159]

In a broad sense, *uncertainty* of measurement refers to doubt about the validity of the result of a measurement. It is defined as a parameter associated with the result of a measurement that characterizes the dispersion of the values that could reasonably be attributed to the measurand. The International Organization for Standardization (ISO) suggests the standard deviation of a parameter as its measure of *standard uncertainty* [Int93]. Related, but more traditional concepts are that uncertainty is a measure of possible error in the estimated value of the measurand or that it is an estimate characterizing the range of values within which the true value of a measurand lies. However, the latter two definitions focus on generally unknown quantities, the *true value* of the measurand and the *error* of the measurement.

The form of representing uncertainty in sensor fusion depends on the data type as well as on the inference methods applied and application-specific demands. The distinction between representations is not always clean cut, and often different representations and inference methods lead to equivalent results [Dil92]. Uncertainty may be expressed using one of the four following representations, each employing different methods of fusing information and its associated uncertainty [Bro98].

### Explicit Accuracy Bounds

If uncertainty is modelled using accuracy bounds, a sensor does not return a single exact value but rather a range of values. If the sensor works correctly, the correct value is always contained by that range. The size of the range can be derived using statistical information like the standard deviation of a sensor's error distribution. An extension of the range implies a greater probability that the correct value is within the range but at the same time increases uncertainty.

Representations using explicit accuracy bounds allow deterministic methods and geometric reasoning to be employed for fusion, like those that will be discussed in section 3.2. However, the crisp boundaries associated to the fusion result may not always reflect the underlying uncertainty properly, since additional statistical and probabilistic properties of the inputs are not considered.

### Probabilistic Methods

Probabilistic methods assume that elementary propositions, like a certain observation or the true value of system state variables, occur with a known probability and have known joint probabilities. Parting from that knowledge, the probability of an event or the likelihood of hypotheses can be derived. The basic formula for such methods is Bayes's rule

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.1}$$

which quantifies the *a posteriori* probability of the occurrence of event $A$ given that event $B$ has already occurred. If $P(A)$, the *a priori* probability of a correct value $A$ and $P(B|A)$, the probability of receiving reading $B$ when the correct value is $A$, are known, this rule allows for the probability of the correct value being $B$ given a measurement to be estimated.

The drawback of this method is that the required prior probabilities are generally difficult to determine, and may have to be established subjectively. The estimation is especially complex if there are multiple potential hypotheses and multiple dependent conditional events. Bayesian methods also require that all hypotheses be mutually exclusive and do not provide any measure of general uncertainty [Hal92].

Dempster and Shafer introduced a reasoning approach that extends traditional Bayesian theory and overcomes some of its disadvantages. Their approach does not require any prior probability distributions and includes an explicit measure of lack of information or general uncertainty. Dempster-Shafer reasoning leads to identical results as Bayesian inference if the set of hypotheses is exhaustive and all hypotheses are mutually exclusive [Dem67, Sha76].

### Statistical Methods

Statistical methods operate on random variables which are characterized by a parameter and a probability distribution. The basic model of such methods is that readings from a sensor follow a known distribution. Several measurements can then be combined to make an inference about the fused parameter and the variance of its distribution.

Generally, information provided by the sensor can be divided into two components: signal and noise. Even though the noise may follow any of a multitude of distribution families and can affect the signal additively, multiplicatively or in any other shape, most statistical models assume *additive Gaussian noise*. This choice can be justified by the *central limit theorem*, which implies that the distribution of the noise tends toward the normal distribution if the overall noise is the sum of a large number of independent stochastic factors that influence the sensor reading, that may each have arbitrary probability distributions [Gel74].

### Fuzzy Logic

Fuzzy logic is a multi-valued logic that was presented by Zadeh as an alternative to boolean logic to account for imprecision generally found in reasoning processes. A fuzzy set is a class of objects that each have a grade of membership assigned by a membership function, and there exist functions equivalent to inclusion, union, intersection and complement in traditional set theory [Zad65].

Applied to sensor measurements, each observed (*fuzzy*) variable has one or more assigned membership functions and each reading has thus assigned grades of membership to the respective fuzzy sets. As an alternative to random variables, a fuzzy variable can thus represent the distribution of values associated to the result of a measurement [Fer05].

### 2.1.4 Limitations of Sensor Fusion

Sensor fusion has enriched the development of complex systems, but at the same time its employment does meet limitations so that the expected gains in performance may not always materialize.

The application of sensor fusion methods is expected to improve the quality of information that a system produces and processes. To meet these expectations, the fusion method to be employed needs to be chosen carefully, since an inappropriate fuser may perform worse than even the worst sensor [Rao01]. Movellan and Mineiro identify this phenomenon as *catastrophic fusion*, which occurs when single components of a system outperform the outcome of the overall system [Mov98]. This may occur even in a well-designed system if the fuser makes implicit assumptions about its context but has to operate in a context inconsistent with these assumptions. This phenomenon can also be observed outside of the technical domain: If a person hears the sound "ba" dubbed over the video of a person mouthing the sound "ga", the listener will apprehend the sound "da"because the brain fuses the two conflicting impressions [McG76].

Sensor fusion cannot be expected to work wonders when confronted with bad inputs. It can only extract as much information as is contained in the input data, and if that is too corrupted or incomplete, even a well-chosen fusion method can not deliver a satisfactory outcome. Even iterating a sensor-fusion algorithm is not able to achieve higher precision or accuracy without additional information [Bro96]. Fowler therefore criticized overly great enthusiasm for such systems:

> *"One of the grabbiest concepts around is synergism. Conceptual application of synergism is spread throughout military systems but is most prevalent in the "multisensor"concept. This is a great idea provided the input data are a good quality. Massaging a lot of crummy data doesn't produce good data; it just requires a lot of extra equipment and may even reduce the quality of the output by introducing time delays and/or unwarranted confidence.*[...] *Note: It takes more than correlation and fusion to turn sows' ears into silk purses."* [Fow79, p.5]

Despite Fowler's scepticism, it is widely recognized that an increase in the information that is input to the fusion process leads to an improvement of the fusion result. Nahin and Pokoski proved that the results of decision fusion using majority vote or maximum likelihood decision rules do benefit from additional independent sensors. They do however admit, that independence between information sources can not be guaranteed in practical applications [Nah80].

Dasarathy showed for the case of a two-sensor suite for decision fusion that a third sensor can only benefit the overall performance if the additional sensor's behavior meets certain prerequisites. Otherwise adding sensors may even lead to a loss of performance [Das00]. Aldosari and Moura addressed the choice of quality and number of sensors depending on the signal to noise ratio (SNR) and concluded that the number of required sensors may be in the order of hundreds in realistic low-SNR environments [Ald04].

To objectively evaluate the benefits of fusion and compare the effectiveness of different algorithms, Theil, Kester and Bossé proposed a set of measures of performance with which the effectiveness of fusion methods can be assessed [The00]. The performance characteristics based on the accuracy and precision of the results and the system's reaction time for fusion methods performing detection, tracking or classification tasks.

## 2.2 Sensor Fusion Strategies

The combination of sensor data from various sources has been denominated with various terms, some of them misleading or conflicting. There have been attempts to define standardized terms, but until today the utilization of them is generally ambiguous. In the early years research in this field the terms *multisensor integration* [Ten81] or *multisensor correlation* [Cha90] were often employed, which have been criticized for the use of a terminology that refers to specific concepts employed as fusion strategies [Wal95].

The general term of *data fusion* has been defined as *"the process of combining data to refine state estimate and predictions"*. [Ste98, p.4] According to this definition, data fusion is a very broad concept that encompasses the integration of data from various sources, among which may be sensors, and which subsumes or partially coincides with terms such as *sensor fusion*, *in-*

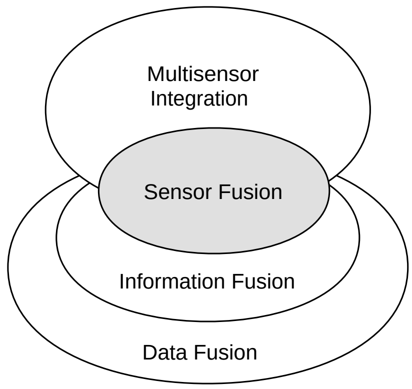



Figure 2.1: Overlapping of fusion terminology

*formation fusion* and *multisensor integration* (figure 2.1). More specifically, *multisensor data fusion* combines data from multiple sensors [Hal92]. In contrast, however, some researchers use the term of data fusion simply for fusion of raw data [Das97, Ben02].

To avoid the ambiguously employed term of data fusion, the general term of *information fusion* has been used to denote a similar concept. The International Society of Information Fusion defines it as encompassing *"the theory, techniques and tools conceived and employed for exploiting the synergy in the information acquired from multiple sources (sensor, databases, information gathered by human, etc.)"* [oIF06].

If all the merged information stems from sensors, the fusion process is termed *sensor fusion*. This term will be used in this thesis according to this definition, i.e., implying that direct or indirect sensor data is being processed and combined using no other information than the sensor measurements and certain meta-information about their sources, without additional knowledge from databases, human input or other sources.

Sensor fusion is distinct from the concept of *multisensor integration*. While (multi-)sensor fusion designates the actual combination of sensory data from different sources, multisensor integration denotes a more general concept of a system that incorporates information provided by multiple sensors [Luo90]. A system based on multisensor integration, though relying on various sensors, may therefore process the received data separately in its control application, while sensor fusion would generally preprocess the data so that only fused information would be communicated to the control process.

### 2.2.1 Types of Sensor Fusion

#### Categorization based on Sensor Configuration

Sensor fusion models may be categorized into three types based on the configuration of the sensors in the system and the form of integration of their information [DW88]:

**Competitive:** In a *competitive* configuration, different sensors measure the same property of an observed object. The fusion process aims to find a consensus about the observed quantity from this redundant information. A configuration of that kind provides greater reliability, or fault-tolerance to a system since a certain number of faulty sensors can be tolerated. Competitive fusion generally leads to increased confidence when sensor readings confirm each other, but may also result in lower confidence if they strongly disagree.

**Complementary:** *Complementary* fusion occurs when two or more independent information sources deliver partial information about an object which is then combined to achieve a more extensive representation. When fusing images taken by various cameras, for example, each camera may cover a different region of the environment to allow a greater scope of observation [Chr00]. The observation space may be disjunct, in which case the observations may simply be concatenated to a single vector, or the may overlap so that more sophisticated fusion methods are required for the information about overlapping regions.

**Cooperative:** A *cooperative* configuration implies that either one sensor relies on another's information to obtain new information about a feature of interest [Nas93], or that information from more than one source is necessary to derive a desired feature [Bro98]. As an example, overlapping images from at least two cameras are necessary to reconstruct a three-dimensional model of the environment [Lab05].

### Categorization based on Level of Abstraction

Fusion methods can be distinguished based on the point in the information processing at which the fusion occurs. The categorization stems from considering the fusion as occurring at different levels of abstraction (*low, intermediate* or *high*) [Elm02], or as occurring at different points in time (*early* or *late*) [Lew04]. Depending on the amount of data processing prior to the fusion step, the communication requirements may be reduced at the price of information loss [Hal92].

**Raw Data Fusion** or *low-level fusion* generates data by combining the unprocessed data obtained from various sources. For this purpose, the available data needs to be *commensurate* in that it observes the same physical attribute of an entity. The resulting data is expected to be more accurate than the single inputs with as little loss of information as possible.

The most commonly employed method for data fusion is *filtering*, which estimates the current system state based on measurements taken up to the current point in time. Related concepts are *smoothing* and *filtering*, which estimate the system state for previous or future points in time respectively, and can be implemented using filtering methods as well [Sar91]. Stateless data fusion methods in contrast do not filter observations over time but only combine measurements of the same point in time. For further discussion of methods for raw data fusion see chapter 3.

**Feature Fusion,** also called *early fusion* or *intermediate-level fusion*, takes features that have been extracted from their respective sensor signals and combines them to a single feature vector. Extracting features leads to a loss in information, but at the same time reduces communications requirements between sensors and fusion process compared to fusion of raw data.

A simple form of feature fusion is feature vector concatenation [Chi97], which however leads to an increase in dimensions of the information space so that algorithms for feature selection may be applied additionally [Jim99, Zha05].

**Decision Fusion,** *late fusion* or *high-level fusion*, occurs when classifications or decisions that have been made separately are combined to a final result. Fusing at this level leads to the greatest information loss which may result in a local optimization rather than a general optimized solution. On the other hand, this type of fusion allows for information from noncommensurate sensors to be combined.

Methods for fusion on this level may be based on voting algorithms such as majority voting, on opinion pools [Dil92], Bayesian decision methods [Kit00, Kri04], various optimal fusion methods for different assumptions for known or unknown error behavior [Rao97, Rao04, Ald04] as well as artificial neural networks or Hidden Markov Models [Lew04].

These categories are not necessarily mutually exclusive. Instead, hybrid fusion methods that combine two or all of the above levels may be implemented. As an example, decision fusion may be supported by direct input of additional raw data. However, more complex fusion logic and greater communication bandwidth than for any of the three separate categories may be required [Hal92].

### Categorization based on Input/Output Data

Dasarathy suggested an expansion of the three level categorization of data-, feature- and decision-level fusion into five process modes depending on the type of the process's input and output data [Das97]. This categorization avoids the ambiguous perspectives in the definition of processing levels, which sometimes have been characterized by the nature of the input data or in other cases by that of the output data. Figure 2.2 shows the five fusion levels in a schematic relation to their input and output data, compared to a three-level categorization.

According to Dasarathy's model, the most basic form of fusion is *Data In-Data Out (DAI-DAO)* fusion, commonly referred to as *data fusion*. Methods applied in this category of fusion are generally based on techniques from traditional signal and image processing. The second level is the *Data In-Feature Out (DAI-FEO)* fusion in which raw data is combined to derive a feature of the observed object. Depending on the perspective of the categorization, such fusion steps have been categorized as pertaining to either

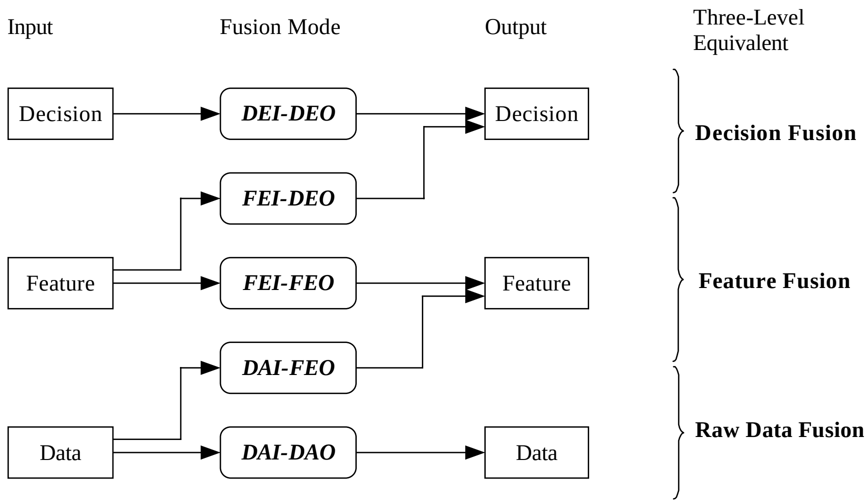
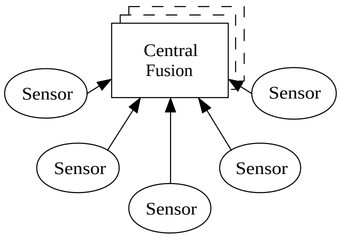



Figure 2.2: Fusion categorization based on input/output data

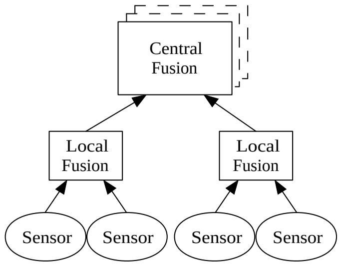
raw data fusion or feature fusion. The next fusion mode is the *Feature In-Feature Out (FEI-FEO)* fusion. Generally, features extracted from various information sources are combined to derive more reliable information or to form a multidimensional feature space. The fourth fusion level, the *Feature In-Decision Out (FEI-DEO)* fusion, derives a decision or classification of the object based on the observed features employing pattern recognition and pattern processing methods. This level has been categorized ambiguously as well as either feature or decision fusion. Finally, *Decision In-Decision Out (DEI-DEO)* fusion is what is commonly referred to as decision fusion.

### 2.2.2 Fusion Architectures

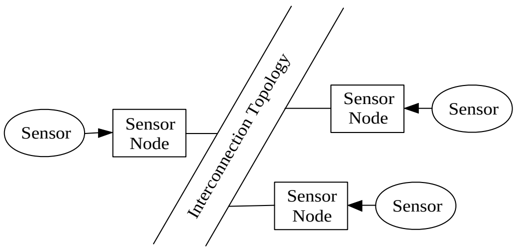
The design of fusion architectures depends on the decision of where to fuse the data in the processing flow within the system. This choice affects the bandwidth of required communication, the complexity of the processing logic, the algorithms that may be employed and finally the quality of the fusion product [Hal92]. The schemes according to which sensor information is processed and combined within a system can be categorized into three groups — *centralized*, *hierarchical* and *decentralized* architectures [Mut98], which are schematically represented in figure 2.3.




(a) Centralized architecture




(b) Hierarchical architecture




(c) Decentralized architecture

Figure 2.3: Fusion architectures

### Centralized Architectures

In a centralized architecture, all sensor devices report their readings to a central processor as indicated in figure 2.3(a), which collects and processes the information. The computational requirements on the processor may be high due to complex physical models or cross-correlation models on which the specific fusion techniques are based [Hal92]. Even though it may be convenient in such a configuration that all information is available in one location for optimal fusion, the communication demand may be not be feasible when integrating a large number of sensors [Ant95].

Apart from communication concerns, the main disadvantages of this configuration are the possibility of a failure of the entire systems in case the central node fails, and a great computational load on the central processor which may render it a computational bottleneck [Rao91]. The threat posed by failures could however be overcome by introducing replicas of the central node which perform the same calculations simultaneously.

### Hierarchical Architectures

The principle behind a hierarchical architecture, as depicted in figure 2.3(b), is to increase the processing speed by distributing fusion tasks to various processing nodes. Data is obtained by different subsystems which fuse it locally in various stages and communicate the results to a central processing node. The required communication bandwidth between the nodes is thus reduced. The central processor then aggregates the separate fusion results to a global estimate.

Even though the computational load is distributed among various nodes, this architecture still relies on one central node to derive the final fusion result, which retains the same susceptibility to failures as the centralized architecture unless the function of the central processor is replicated. Additionally, more complex fusion algorithms are necessary to guarantee the consistent combination of partial results stemming from local models [Has88] and maintain synchronization of parallel processes in the subsystems [Wal90].

Unlike in the centralized case, a hierarchical architecture can not guarantee optimal estimation from the point of view of quality of the fusion result. Each intermediate fusion step in a hierarchical structure involves loss of information, so that the final fusion process only operates on abstractions of the original data. Therefore the global result may turn out suboptimal even though local fusion processes reach local optima [Ten81].

### Decentralized Architectures

A decentralized architecture (figure 2.3(c)) consists of a network of sensing nodes that are fully or partially interconnected. All information is processed locally by each node without a central processing site and fusion is based on local information as well as that received from surrounding nodes. There is no center for global decisions or supervision, instead the nodes communicate their observation or decisions to their neighbors.

Decentralization overcomes the problems associated with centralized and hierarchical systems in the way that the redundant nodes running the same algorithms make the system resilient to the failure of nodes. Overall processing speed can be increased since potential computational bottlenecks may be avoided by executing different algorithms in parallel. [Rao91]

Decentralized systems are not without drawbacks, however. Communication issues in fully decentralized systems are complex and communication overheads are higher than in centralized systems. The number of processors is generally higher, and monitoring of the system is harder than in the other two architecture models [Mut98].

### 2.2.3 Fusion Models

The approaches to information fusion have been diverse due to the various fields of applications and employed methods. As attempts to standardize the design of fusion systems, several fusion models have be proposed that should each guide the design process by partitioning the fusion process into subtasks.

#### The JDL Fusion Model

The Joint Directors of Laboratories (JDL) Data Fusion Subpanel developed the following model in 1986 as a framework for designing data fusion systems [Wal90]. The model identifies a data base and five processing levels, even though these levels do not indicate a chronological processing order and can be interleaving and executed concurrently. The following are the components of the process model [Hal97]:

**Sources:** The input to the fusion process may originate from various sources such as sensors, *a priori* information or human input.

**Man-machine interaction:** This part of the model provides an interface for human input, such as commands, information requests or reports from an operator, and is the mechanism by which the system communicates the data fusion results the user.

**Source preprocessing (Level 0):** Through pre-screening of the data and its allocation to the appropriate processes, the step of source preprocessing reduces the data fusion processing load.

**Object refinement (Level 1):** At this level locational, parametric, and identity information is combined to represent individual objects. The main functions of this level are *alignment* to a consistent reference

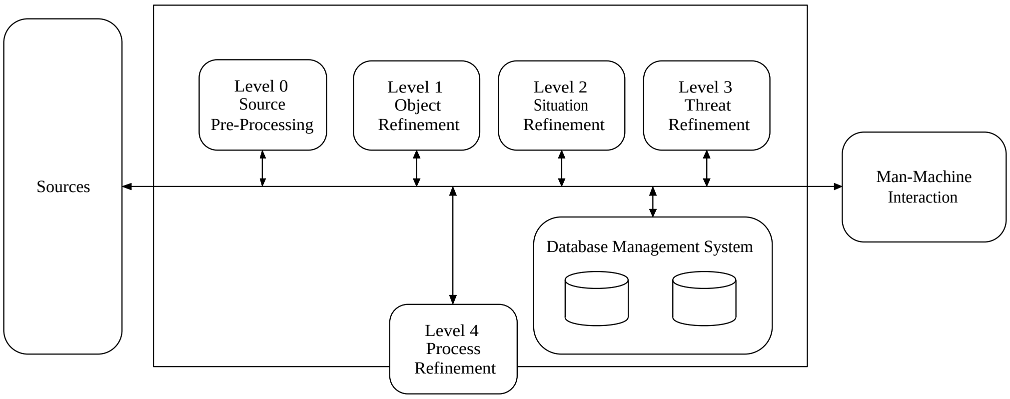



Figure 2.4: The JDL fusion model (from [Hal97])

frame and units, *estimation* and/or prediction of an object's attributes, *assignment* of data to the respective object and *identification* through classification methods.

**Situation refinement (Level 2):** This level incorporates objects, observed events, environmental information and *a priori* knowledge to find a contextual description of the relationship between objects and events.

**Threat refinement (Level 3):** Based on the current situation and predictions about the future this process level attempts to deduce threats, vulnerabilities as well as opportunities for operation.

**Process refinement (Level 4):** As a meta-process this level monitors system performance, e.g., concerning real-time constraints or long-term data fusion performance, identifies what information is required for further tasks, and allocates sensors and sources to achieve particular mission goals.

**Database management system:** The database management system supports the data fusion process by monitoring, storing, updating and providing a large amount and often great variety of data, e.g., images, signal data, etc.

The JDL is one of the most widely used methods for categorizing functions related to data fusion. Even though the model was originally developed for military applications — a fact that the definitions of the levels clearly reflect — it can be applied to the design of commercial systems as well.

The model however does have its shortcomings. Waltz pointed out that the JDL model does not address multi-image fusion problems and proposed an extension to the model that would cover such applications as well [Wal95]. Steinberg, Bowman, and White identified other problems in the application of the model, like that it is very general, and has been interpreted differently in varying applications, which is a contradiction to the original intention of the model which was to unify the design process of multisensor systems. Also, it tends to be interpreted as a guide to a stepwise rather than an interlaced process.

Its focus on tactical targeting applications does not always make the extension to other applications obvious to developers. Steinberg et al. therefore proposed a revision of the JDL model that extends the functional model and the taxonomy to fields beyond the original military focus and integrates a data fusion tree architecture model for system description, design and development [Ste98]. A few years later the model was once again extended to account for different types of input data, models, outputs and inference methods in the various classes of data fusion problems. Additionally, levels of resource management corresponding to each level of the fusion process were suggested [Ste04]. Further proposed extensions include the interaction between processes of different levels and the integration of information exploitation processes related to sensor fusion, such as pattern recognition and explanation or agent-based data retrieval [Lli04].

### The Waterfall Model

The waterfall model presents an alternative to the JDL model which focuses on the processing functions at the lower levels of the fusion process [Mar97]. The processing stages that the model identifies are depicted in figure 2.5. They can be related to the levels of the JDL model in the way that sensing and signal processing matches JDL level 0 (source pre-processing), feature extraction and pattern processing correspond to JDL level 1 (object refinement), situation assessment to JDL level 2 (situation refinement), and decision making coincides with level 3 (threat refinement).

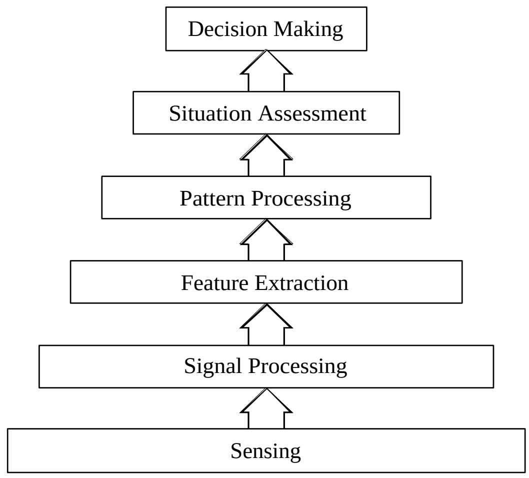



Figure 2.5: The waterfall fusion process model (from [Bed00])

Even though the waterfall model puts more emphasis on a fine division of the fusion process, it is still very similar to the JDL model and suffers from similar shortcomings. An additional drawback of the model is that it does not explicitly include any feedback dataflow. The model has been widely used in UK defence data fusion community, but has not been able to establish itself elsewhere [Bed00]

### The Boyd Control Loop

The Boyd control loop, or *Observe-Orientate-Decide-Act* (OODA) loop, was originally used to model general military decision making and command processes [Boy86], but has been widely employed for data fusion. The model identifies four phases that are iterated and compare to the JDL model as follows [Bed00]:

**Observe:** The observation stage is comparable to the JDL level 0 of pre-processing.

**Orientate:** This stage subsumes levels 1,2 and 3 of the JDL model.

**Decide:** The decision stage corresponds to JDL level which encompasses process refinement and resource management.

**Act:** This stage has no directly corresponding part in the JDL model. It closes the loop by taking into account the effect of decision and actions of the system.

The Boyd model explicitly considers the iterative nature of a fusion process and includes a phase of putting into action the decisions that have been taken by the system. On the other hand, it does not identify the different fusion tasks necessary in the *orientate*-phase in detail. It may therefore be a general guideline for reasoning processes rather than a model for specific data fusion strategies.

### The Omnibus Model

The Omnibus Model was suggested by Bedworth and O'Brien in 2000 [Bed00]. It combines aspects of various older data fusion process models and compensates for their various disadvantages. Figure 2.6 presents a diagram of the omnibus model architecture. The model is based on the cyclic nature of the Boyd control loop but incorporates the finer distinction of fusion steps in the waterfall model. It uses a general terminology that does not assume that applications are defence oriented. The model is intended to structure a system on two levels. First, it characterizes and partitions the system's aims to obtain an ordered list of tasks. In a second abstraction step the model may be used to further subdivide each such task.

### The Time-Triggered Sensor Fusion Model

Finding that most existing fusion models had very abstract timing and communication properties, Elmenreich and Pitzek proposed the *time-triggered sensor fusion model* adequate for real-time systems [Elm01]. The model distinguishes three abstraction layers with well-defined interfaces and explicitly contains the cyclic process of data acquisition, data processing and acting upon deduced information. Figure 2.7 depicts the model layers and their interfaces and indicates the control loop [Elm02].

**Transducer level:** This level subsumes the transducer nodes consisting of sensors and/or actuators that interact with the observed object. Sensors provide information about properties of interest, while actuators execute a control value.

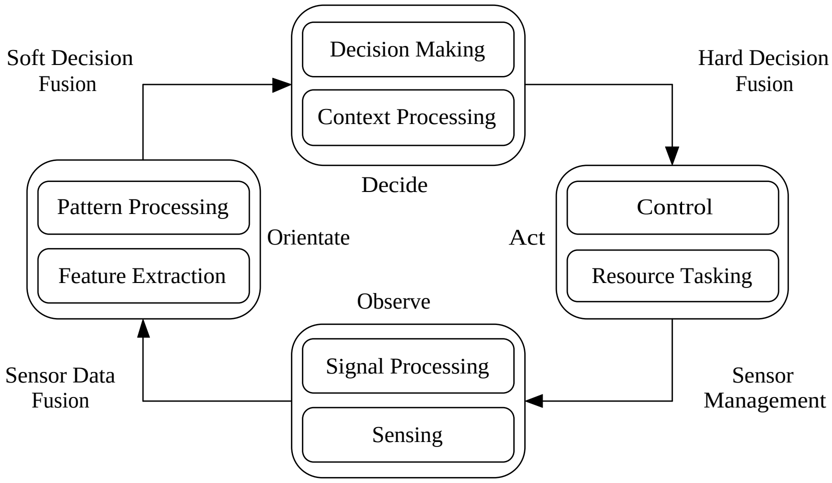



Figure 2.6: The Omnibus Model (from [Bed00])

**Fusion/Dissemination level:** Sensor fusion tasks and fault tolerance mechanisms are implemented at this level. The representation of the fused data is independent from the number and types of employed sensors in the transducer level. It further distributes control information from the control level to the actuators.

**Control level:** This level receives the information about the environment deduced by the previous level and parting from that derives control decisions which in turn are communicated to the lower levels.

## 2.3 Real-Time Aspects

### 2.3.1 Model of Time

For the correct integration of data collected by distributed nodes knowing the point of time at which a measurement was taken is essential. Only if each observation has an associated timestamp, it is possible to determine the temporal order of measurement. If the timestamps are assigned by distributed clocks, the problem of establishing temporal order gains in complexity.

The occurrence of events in a distributed system is measured by local clocks in the system nodes. To record at which instant an event occurs at a specific node, each event is assigned a timestamp by the local clock. The clocks of all nodes in a distributed system can never be perfectly synchronized

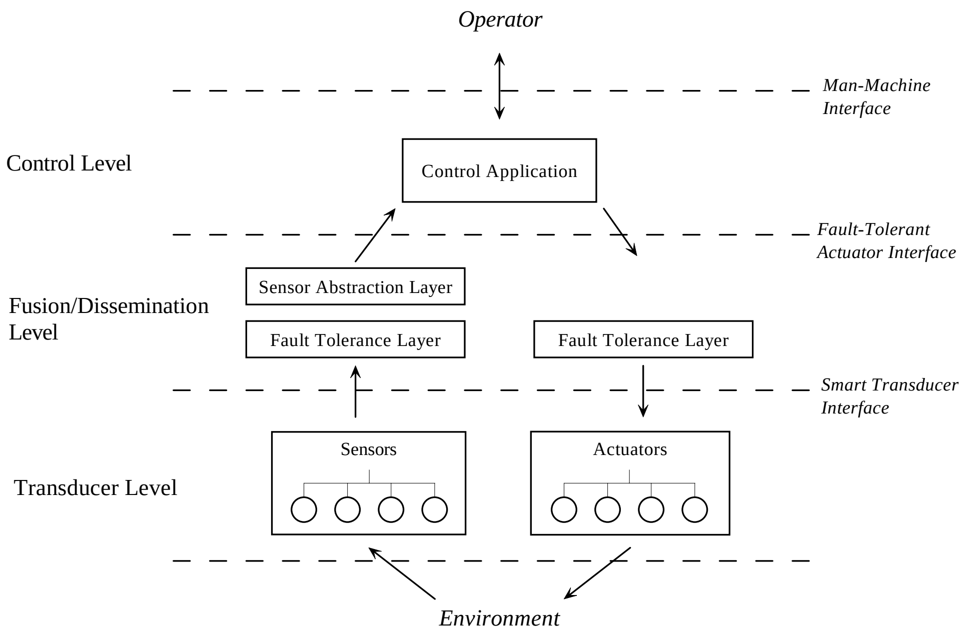



Figure 2.7: The time-triggered sensor fusion model (from [Elm02])

so that two events that happen at the same instant at two different nodes may therefore not have the same timestamps.

To ensure a precise and accurate measure of time and to reconstruct the chronological order of events, the local clocks are synchronized periodically. A global notion of time is established by properly selecting ticks from the synchronized physical clocks, these selected ticks being called *macroticks*. [Kop97]. A reasonable global time ensures that the timestamps two different clocks assign to the same event differ at most by one macrotick. A fundamental limitation that cannot be overcome, however, is that the true chronological order of two events can not be determined if their timestamps differ by less than two macroticks.

Due to imperfect precision in clock synchronization, different nodes may come to differing views in the ordering of events. In a system at which events may occur at any point in time, that is on a *dense* timeline, the nodes observing events need to execute an agreement protocol to achieve a consistent view of the order of occurrence. Such agreement protocols however

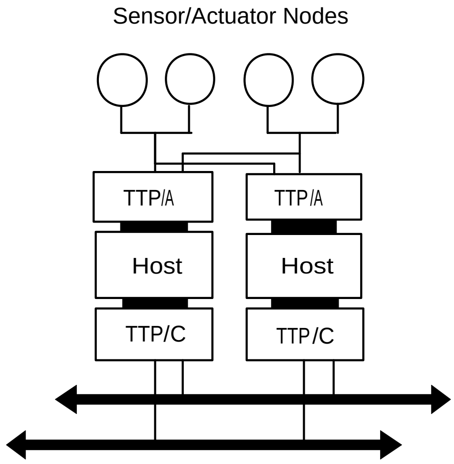



Figure 2.8: Structure of a TTP/A transducer cluster

are costly in communication and processing requirements and impose an additional delay on the information processing.

To avoid the need for an agreement protocol, events can be coerced to occur on a *sparse* time base. In the sparse time model, time is partitioned into alternating intervals of activity and silence. Events are only allowed to occur during an interval of activity, and all events happening during the same period of activity are considered to have occurred at the same time. It is therefore still not possible to establish the exact temporal order for all events, however, the view of ordering is consistent in all nodes without additional communication overhead. Events that are separated by at least one interval of silence can be temporally ordered. Obviously, a sparse time can only be realized if the respective system has control over the occurrence of events. Events outside of the system's sphere of control will still happen along a dense timeline. [Kop02]

### 2.3.2 The Time-Triggered Architecture

The time-triggered architecture (TTA) provides an infrastructure for dependable distributed embedded systems. Its communication is based on and controlled by a global sparse time base which guarantees that signals are only transmitted at predefined points in time. It comprises nodes each equipped with memory, input/output system, communication controller, operating system and the specific application software. Nodes may be grouped and interconnected to form clusters. The communication system between the nodes works autonomously and fetches messages and delivers them at predefined points in time [Kop03].

The two communication protocols employed in such systems are the fault-tolerant TTP/C and fieldbus protocol TTP/A. TTP/C provides such services as fault-tolerant and timely message transport and clock synchronization and interconnects nodes within the TTA. In contrast, TTP/A is employed to connect transducers to nodes of the TTA. It offers mechanisms for on-line diagnosis and reconfiguration of transducers and supports a "plug-and-play" mode in which new sensors are detected, configured, and integrated on-line and thus provides great flexibility.

The employment of sensors or transducers in in a TTA would be realized by forming such a transducer cluster as a fault-tolerant unit which is interconnected by TTP/A fieldbus. To achieve fault tolerance, at least two buses need to be installed, as indicated in figure 2.8. Each fieldbus is controlled by a TTP/A master which is connected to the rest of the TTA via a TTP/C gateway. Each transducer node holds its own calibration data, diagnostic data and configuration data in its local memory and can thus deliver an already calibrated observation with self-diagnosed uncertainty. The sensor fusion process is then executed in the controllers of the TTA nodes of the cluster and only a single fused value is communicated to other clusters in the system.

## 2.4 Chapter Summary

The purpose of this chapter was to present basic concepts of sensor fusion. Due to the limited dependability of single sensors as a result of measurement noise, the possibility of sensor failure, environmental effects and cross-sensitivities, systems employ multiple sensors to derive information about their environment. Fusion is generally able to reduce uncertainty about observed variables, expand spatial and temporal coverage and increase robustness to interference and failures.

In the wide field of sensor fusion research diverse approaches have evolved and this chapter has attempted to categorize them based on the type of data that is being processed, processing architectures or models for formulating the fusion process and surrounding processing and reasoning steps. Finally, it was pointed out that for the consistent integration of sensor information the time at which observations are made is essential. Due to the fact that distributed clocks can never be synchronized perfectly, some mechanism has to be employed to determine the temporal order of observations. The time-triggered architecture was cited as an example for solving this problem without an explicit agreement protocol, by implementing time-triggered communication based on a sparse time base.

*Not to be absolutely certain is, I think,*
*one of the essential things in rationality.*

Bertrand Russell

# Chapter 3

# Raw Data Fusion

The method to be developed in this thesis will provide a method for low-level data fusion. This chapter presents an overview of currently established approaches for such tasks. Most commonly, filtering methods as discussed in sections 3.1.1 and 3.1.2 are employed. They fuse observations over time based on detailed models of dependencies between attributes of the system state. In contrast, the stateless fusion methods presented in sections 3.2 and 3.3 do not make any model assumptions and require no information about previous states to derive their fusion results.

## 3.1 Filtering Methods

### 3.1.1 Kalman Filtering

The *Kalman Filter* was introduced by Kalman and Bucy in 1960 [Kal60, Kal61]. It assumes detailed knowledge about a model of the system under observation. Even though the fulfilment of this premise is questionable in many practical applications, the method and its extensions have been used in fields like aerospace navigation, robotics or process control [Man98].

A Kalman Filter is a recursive linear estimator which combines periodic observations up to time $k$ to obtain an estimate of the current system state with minimum mean squared estimation error. The dependencies between observations and system state have to be linear, and errors are assumed to follow a Gaussian distribution and to be uncorrelated between different sensors and over time [Mut98].

The observed system is described as follows:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \omega \tag{3.1}$$

where $\mathbf{x}_k$ is the state of the system at time $k$ and $\mathbf{A}$ is the state transition matrix from time $k$ to $k+1$. $\omega \sim N(\mathbf{0}, \mathbf{Q})$ is the associated process noise modelled as uncorrelated, zero-mean white noise with process covariance $\mathbf{Q}$. To account for interaction of the control system with the observed phenomenons, $\mathbf{u}_k$ designates the input control vector and $\mathbf{B}$ is the matrix that describes its relation to the system state vector [Mut98]. This additive term, however, is not always explicitly part of the model ([Kal60, Man98, Kay93]).

The observations of the system are described by the following equation:

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \upsilon \tag{3.2}$$

where $\mathbf{y}_k$ is the vector of observations at time $k$, $\mathbf{H}$ is the model matrix that relates the system state to the observations and $\upsilon \sim N(\mathbf{0}, \mathbf{R})$ is the associated observation noise assumed to be uncorrelated white noise with covariance $\mathbf{R}$.

Equations 3.1 and 3.2 represent the two basic filter equations and are a very general model for combining measurements over time to an optimal estimate. They are the grounds upon which the two stages of the filter algorithm — *prediction* and *estimation* — are based.

In the prediction stage, the system state at time $k+1$ is predicted based upon the information collected up to time $k$ (equation 3.3). The corresponding predicted error covariance matrix $\mathbf{P}$ is calculated according to equation (3.4).

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{A}\hat{\mathbf{x}}_{k|k} + \mathbf{B}\mathbf{u}_k \tag{3.3}$$

$$\mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^{\mathbf{T}} + \mathbf{Q} \tag{3.4}$$

From there the Kalman gain matrix $\mathbf{K}_{k+1}$ and the innovation covariance matrix $\mathbf{S}$ are derived:

$$\mathbf{K}_{\mathbf{k+1}} = \mathbf{P}_{k+1|k}\mathbf{H}^T\mathbf{S}_{k+1}^{-1} \tag{3.5}$$

$$\mathbf{S_{k+1}} = \mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^T + \mathbf{R} \tag{3.6}$$

The estimation stage updates the estimate of the current state and the error covariance matrix using the new observation $\mathbf{y}_{k+1}$:

$$\hat{\mathbf{x}}_{k+1|k+1} = [\mathbf{1} - \mathbf{K}_{k+1}\mathbf{H}]\hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\mathbf{y}_{k+1} \tag{3.7}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}^T_{k+1} \tag{3.8}$$

At the beginning of the iterations, $\mathbf{x}_0$ and $\mathbf{P}_{0|0}$ need to be initialized with an estimate that is not yet based on any data. An inaccurate initialization will not affect the performance of the filter in the long run, since the repeated update of $\mathbf{P}_{k|k}$ will let it reach its true value. In the same manner $\mathbf{K}_{k+1}$ and $\mathbf{S}_{k+1}$ will converge to a steady state as long as $\mathbf{A}, \mathbf{B}$ and $\mathbf{H}$ are time-invariant [Man98].

The standard Kalman filter is designed to estimate states based on linear relationships between observations and system states. However, in many real data fusion applications, the true workings of a system can not be described with linear models. To solve such non-linear estimation problems, the *Extended Kalman Filter* (EKF) has been derived [Mut98]. The EKF is a linear estimator for nonlinear systems obtained by a linearization of nonlinear state and observations equations.

The extended filter can not guarantee any of the properties the standard filter has [Man98]. It no longer provides a minimum variance estimate but only an approximation thereof [Kay93]. As another shortcoming, the matrices $\mathbf{K}$ and $\mathbf{S}$ will not tend to constant values but have to be updated with every observation, which leads to an increased computational load. Finally, the performance depends on an accurate initialization of the parameters to ensure that the resulting models are valid, and if a prediction is too far from the true state, the true covariance may be greatly underestimated which leads to instability of the filter [Mut98].

To overcome these problems, Julier et al. elaborated the *Unscented Kalman Filter* (UKF) as a new approach to generalizing the Kalman filter, which does not make as strong linearizing assumptions as the traditional EKF. They show that their modified filter predicts the system state and the covariances more accurately than the EKF with far less time consuming calculations [Jul95].

The Kalman filter has been an inspiration for the development of various other filters. One related family is that of $H_\infty$-optimal filters which use a $H_\infty$-norm instead of the minimization of the squared error as optimization criteria [Sha92]. Furthermore, mixed performance filters combine Kalman filtering algorithms with such $H_\infty$-optimal methods [Kha96]. As another alternative to the Kalman filter, Gan and Harris show that an *information filter*, which is functionally equivalent, can reduce the computational load if the measurement matrices of the data sources are identical [Gan01]. Hashemipour [Has88] adapted the filter to apply it to hierarchical architectures, Rao and Durrant-Whyte later extended it for fully decentralized sensor fusion systems [Rao91].

### 3.1.2 Covariance Intersection

The Kalman filter delivers optimal results in the case of accurately defined covariances of sensor errors in the matrix $\mathbf{R}$. In a complex system, however, it may not always be possible to determine these covariances. A result may be that it is optimistic in the sense that it underestimates the actual error covariance of the fusion result [Ben02]. Julier and Uhlmann proposed the *Covariance Intersection Algorithm* to find a robust solution to the problem of data fusion in the presence of unknown correlations based on the Kalman filter [Jul97]. The algorithm yields consistent estimates for any degree of correlation between the sources in the sense that the variance and covariance of the result is never underestimated. To fuse input variables, it calculates a convex combination of the means and covariances between measurements of the same source.

The approach is based on a geometric interpretation of the linear combination of two input sources $a$ and $b$ with covariance matrices $\mathbf{P_a}$ and $\mathbf{P_b}$. Plotting out the covariance ellipse for each input source, which is the set of points $\{\mathbf{p} : \mathbf{p^T P^{-1} p} = k\}$ where $k$ is a constant, the covariance ellipse of the fused result $c$ is always contained by the intersection of the ellipses of the inputs for any type of correlation between the inputs. Figure 3.1 visualizes this concept. The solid lines represent the two covariance ellipses of two two-dimensional random variables, and the faint dashed lines within their intersection are a few of the possible fused covariance ellipses for different correlations between the two variables. The bold dashed line indicates the

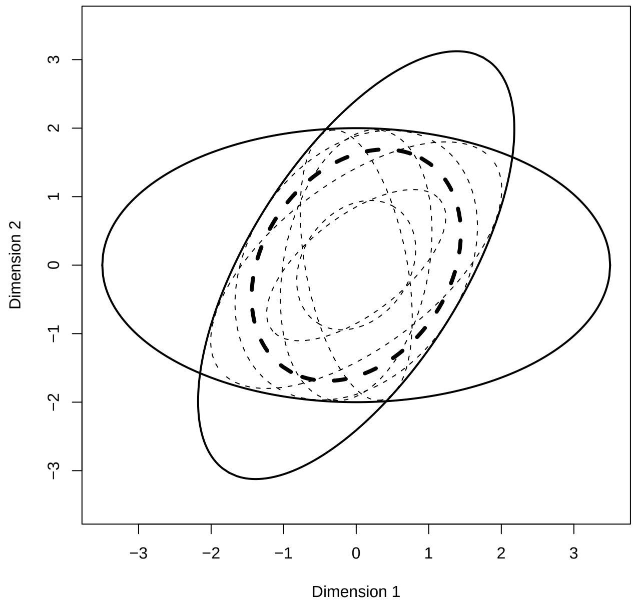



Figure 3.1: For any possible correlation setting between two variables, the fused covariance ellipse lies within the intersection of the two single ellipses.

covariance ellipse a Kalman filter would calculate without information about correlations.

The algorithm finds the covariance matrix $\mathbf{P_c}$ of the fused result which encloses the intersection region as tightly as possible by employing equations 3.9 and 3.10. The parameter $\omega$ manipulates the weights assigned to $a$ and $b$. It should be chosen so that it optimizes some performance criteria, such as the trace of $\mathbf{P}_c$. The resulting covariance matrix is then employed in an incremental estimation of the system state analogous to the Kalman filter. To solve this nonlinear optimization problem while avoiding a possibly high numerical effort, an approximation method for fast covariance intersection was proposed [Nie02] which was later improved to consider the relative variances of inputs [Fra05].

$$\mathbf{P}_c^{-1} = \omega \mathbf{P}_a^{-1} + (1-\omega)\mathbf{P}_b^{-1} \tag{3.9}$$

$$\mathbf{P}_c^{-1}\bar{c} = \omega \mathbf{P}_a^{-1}\bar{a} + (1-\omega)\mathbf{P}_b^{-1}\bar{b} \tag{3.10}$$

The algorithm is designed to never underestimate the true fused covariance matrix by systematically overestimating it. The resulting covariance

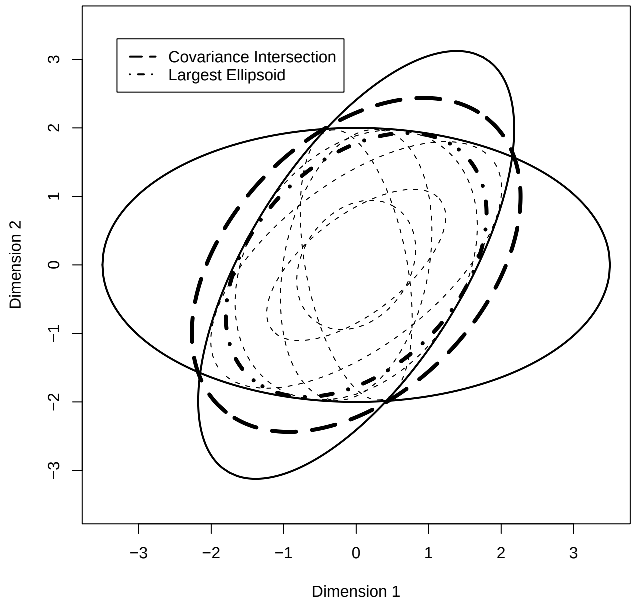



Figure 3.2: Covariance Intersection and Largest Ellipsoid solution for unknown correlations between the sources.

matrix is an upper bound for the actual covariance, and therefore a pessimistic estimate which leads to a decrease in performance. Trying to find a compromise between this method and the Kalman filter that underestimates correlations, Benaskeur introduced the *Largest Ellipsoid Algorithm*, that is based on a similar approach and finds the largest ellipse contained in the intersection region of the two covariance ellipses [Ben02]. Figure 3.2 compares the results of the two algorithms. The bold dashed line represent a possible solution found by the covariance intersection method, which includes the entire intersection region. The dot and dash line indicates a solution found by the largest ellipsoid algorithm, which is completely contained by the intersection region, but could possibly underestimate the actual covariance.

Both the Covariance Intersection method and the Largest Ellipse Algorithm offer a way of considering the increased uncertainty stemming from correlations between data sources. However, in the case of known correlations they may both lead to pessimistic estimates of the fused results reliability, since they automatically assume the worst possible configuration of correlations.

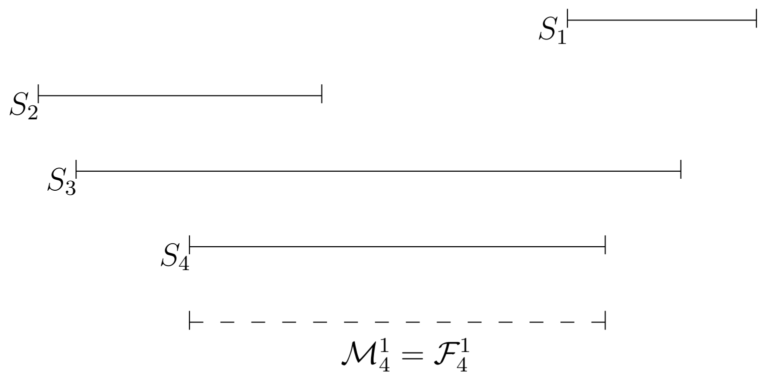



Figure 3.3: Intersection of four sensors, assuming $f = 1$. $\mathcal{M}_4^1$ equals $\mathcal{F}_4^1$ and contains values that are only covered by two intervals and can therefore not be correct values.

## 3.2 Interval Intersection

### 3.2.1 Fault-Tolerant Sensor Averaging

Marzullo [Mar90] introduced a method for combining measurements of various sensors to obtain a more accurate estimate of the observed variable through interval intersection. He defines an *abstract sensor* as a piecewise continuous function that maps a measured physical state variable into the center of an interval of real numbers. Unlike a *concrete sensor*, which is the device that takes the actual measurement of a state variable, an abstract sensor can represent imprecision of the measurement caused by limited accuracy of the sensor. Also, it uses a unique representation of measurements from sensors which may differ in certain properties, which differences are however irrelevant to the fusion and the following control process.

The width of the abstract sensor's interval around the measured value gives an upper and lower bound for the true value and represents the accuracy of the sensor. If the abstract sensor is *correct*, the interval always contains the true value of the observed variable and its accuracy does not exceed a reasonable upper bound. The assignment of the actual confidence interval to each measurement is taken to be part of the application-specific design, and is not part of Marzullo's theoretical model.

To combine the measurements of such abstract sensors, Marzullo introduces *Fault-tolerant Sensor Averaging*, a method that calculates a correct fused interval if at least half of all sensors work correctly. It is founded on the idea that if any two abstract sensors contain the correct value of a variable, the intersection of these two intervals must contain the correct value. If one assumes that out of $n$ sensors no more than $f$ sensors are faulty, that is their interval may not include the correct value, then any value that is not contained in at least $n-f$ intervals cannot be the correct value. On the other hand, the intersection of any $n-f$ intervals could include the true value.

The interval $\mathcal{M}_n^f$ of the fused abstract sensor is derived by calculating the smallest and the largest value contained in at least $n-f$ intervals respectively. As shown in Figure 3.3, the resulting interval may however contain values that cannot be the correct value because they are contained in less than $n-f$ intervals. In the case of $f \geq \lfloor (n+1)/2 \rfloor$, the resulting interval still contains the correct value, however it is larger and therefore less accurate than any of the original abstract sensors.

### 3.2.2 Fault-Tolerant Interval Intersection

As shown in [Sch01], Marzullo's method is susceptible to even small changes of the input intervals. A slight shift of one interval can result in it not intersecting with others anymore, and can thereby change the size of the fused interval significantly, as the comparison of figures 3.3 and 3.4 shows.

The Lipschitz Condition applied to sensor fusion states that small changes in sensor readings should produce only small changes in the fusion function's output [Rus02]. Schmid and Schossmaier [Sch01] introduce a method based on Marzullo's that does satisfy the Lipschitz condition. Their *fault-tolerant interval intersection* function calculates a new interval $\mathcal{F}_n^f$ from a set of $n$ abstract sensors of which at most $f$ are faulty by defining its left and right limits as the $f+1$-largest left edge and the $f+1$-smallest right edge respectively.

The comparison of the intervals $\mathcal{F}_n^f$ and ${\mathcal{F}'}_n^f$ in figures 3.3 and 3.4 respectively show that they are not at all affected by the small change in one interval. Concerning the relationship between $\mathcal{F}_n^f$ and $\mathcal{M}_n^f$, Schmid and Schossmaier show that $\mathcal{M}_n^f$ is always contained in $\mathcal{F}_n^f$, and that $\mathcal{F}_n^f$ is optimal for all fault-tolerant intersections that satisfy the Lipschitz condition.

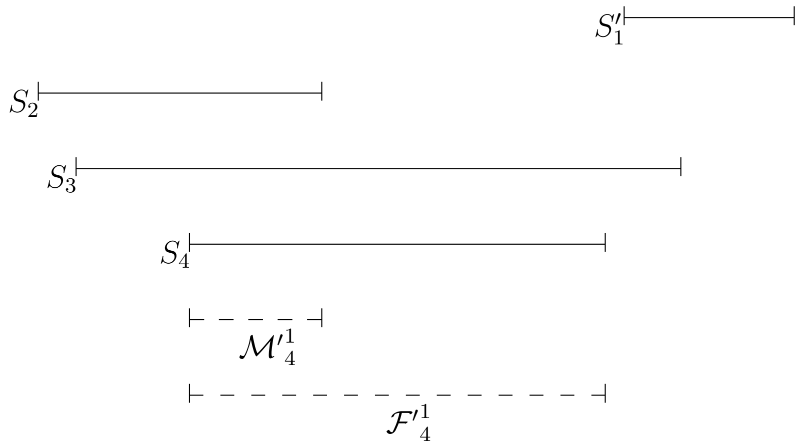



Figure 3.4: Intersection of four sensors, assuming $f = 1$. The shift of interval $S_1$ to $S_1'$, results in a drastic change of $\mathcal{M}'^1_4$ while $\mathcal{F}'^1_4$ remains unchanged.

## 3.3 Confidence-Weighted Averaging

In his thesis Elmenreich [Elm02] suggests an algorithm for fusing data from replicated sensors based on weighted averages. The inputs to the fusion process are weighted according to the uncertainty associated to their sources: sensors that show less variation in their measurement errors than others are taken to be more dependable and are therefore allowed to have a greater influence on the fusion outcome than less dependable sources. The measurement errors of all inputs are assumed to be independent and have a mean of zero.

The fused value $x_{FUSED}$ is calculated as the weighted average of all measurements $x_i$ (3.11). The weights should be chosen so that they minimize the expected variance of the fused value (3.12).

$$x_{FUSED} = \sum_{i=1}^{n} w_i x_i \tag{3.11}$$

$$\sigma^2_{FUSED} = \sum_{i=1}^{n} w_i^2 \sigma_i^2 \tag{3.12}$$

Thus the optimal weights $w_i$ are the solution to the minimization problem

$$\arg\min_{w_i} \sum_{i=1}^{n} w_i^2 \sigma_i^2 \tag{3.13}$$

under the condition that the sum of all weights is equal to 1 to obtain an undistorted result.

Equation (3.14) expresses these optimal weights, where $n$ is the number of observations to be fused, $x_i$ represents a measurement taken by sensor $i$, and $\sigma_i^2$ is the estimated variance associated to that sensor.

$$w_i = \frac{1}{\sigma_i^2 \sum_{j=1}^{n} \frac{1}{\sigma_j^2}} \tag{3.14}$$

The fused value $x_{FUSED}$ and its associated fused variance $\sigma_{FUSED}^2$ are thus calculated as in (3.15) and (3.16). For $n \geq 2$ the variance of the fused result is always smaller than any of the input variances.

$$x_{FUSED} = \frac{\sum_{i=1}^{n} \frac{x_i}{\sigma_i^2}}{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}} \tag{3.15}$$

$$\sigma_{FUSED}^2 = \frac{1}{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}}. \tag{3.16}$$

Under the assumption of independence of errors between sensors and supposing the measurements are unbiased, i.e., the expected deviation from the true value is equal to 0, this method minimizes the expected variance of the fused value [Elm06].

In contrast to the methods using interval intersection introduced in 3.2, the transition between a faulty and an accurate measurement is smooth. A measurement is not either completely included or excluded, but rather has varying influence on the result of the fusion process depending on the amount of confidence associated to it. A faulty measurement from a source identified as unreliable thus has only little influence on the outcome, but is not completely excluded from the calculations. Should an otherwise reliable source contribute a faulty measurement, however, the great weight assigned to this input may result in a distortion of the fused value. It is therefore necessary to employ means to identify such faulty measurements beforehand and discard them.

## 3.4 Chapter Summary

This chapter presented an overview of established approaches to the low-level data fusion problem. The most commonly used method for filtering observation over time are the Kalman filter and its many offsprings. These filtering methods are based on several restrictive assumptions such as a known model of dependencies between observed variables. The Kalman filter is able to incorporate known correlations. For the case of unknown correlation behavior, extensions which assume a worst case correlation to correct the error estimate have been proposed.

For stateless fusion, two conceptually different methods have been proposed — that of interval intersection and that of weighted averaging. Both methods use some measure of confidence for each input value representing the uncertainty associated to the sensor measurement. Fault-tolerant interval intersection completely excludes certain measurements from the fusion process and can even tolerate multiple faulty measurements at the cost of possible loss in performance in the non-faulty case. Confidence-weighted averaging does not explicitly exclude any observations from the calculation but rather relies on the estimated confidence in the measurement sources. To assume a certain measure of fault tolerance, it therefore has to rely on some identification of faulty measurements previous to the fusion process.

Neither of the two presented methods for stateless fusion considers possible dependencies between measurement errors of different sensors. Since confidence-weighted averaging is based on statistical properties of the sensor behavior, and error covariance or correlation are statistical measures related to that of variance, this method can be easily extended to take such relationships into consideration.

*A creative man is motivated by the desire to achieve,*
*not by the desire to beat others.*

AYN RAND

# Chapter 4

# Introducing Correlations

In this chapter, an extension to confidence-weighted averaging will be presented that improves the fusion process by considering known correlations between the fusion inputs.

## 4.1 Motivation for Introducing Correlations

Most competitive fusion applications assume that errors committed by the sensors are independent [Kit00]. In many important situations, however, it is impossible to guarantee that the sensor noise is independent.

One reason for correlated errors may be that the stochastic noises acting on a physical system may affect different sensors in the same way, or may be correlated themselves. Another cause may be a lack of knowledge about the dependencies within the system. The observation noise of sensors on the same vehicle may for example be correlated due to vehicle motion [Jul97]. Finally, non-stochastic influences that are not modelled by the system may affect the observations deterministically but *appear* to cause random deviations.

The standard measure of *covariance* is defined as the expected value of the product of the deviation of two random variables from their means [Gel74]:

$$Cov(X, Y) = E[(X - E[X])(Y - E[Y])] \tag{4.1}$$

The covariance is a measure whose magnitude depends on the variance of each random variable. The *correlation coefficient* $\rho$ (4.2) is the covariance normalized by the standard deviation of each variable, and allows for a comparison of degrees of correlation between random variables with different variances. It assumes values from -1 for perfectly negatively correlated variables to +1 for perfectly positively correlated variables.

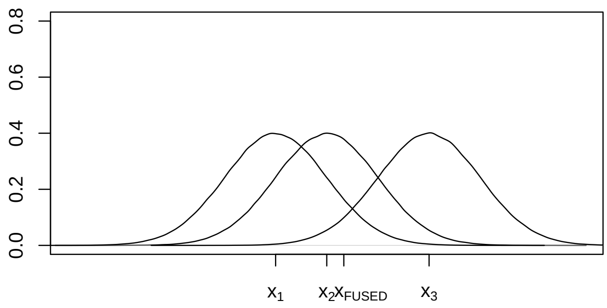
$$\rho(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X)}\sqrt{Var(Y)}} \tag{4.2}$$

Errors that are independent are also uncorrelated, i.e., their correlation coefficient is equal to zero. The reverse is not true in the general case, since covariance and correlation coefficient only measure linear interrelations between variables. A positive correlation coefficient indicates that two sensors tend to commit similar errors, i.e., if one sensor overestimates a parameter, the other tends to overestimate its value as well.

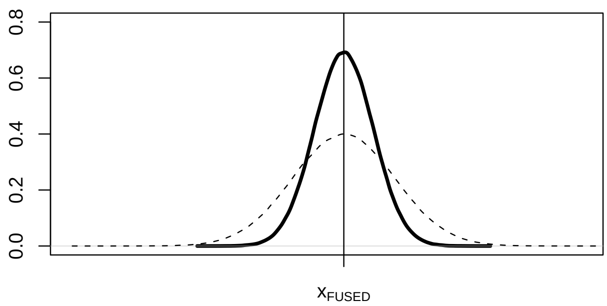
If the assumption of independence is correct, the noise can be cancelled out and on average a better result can be expected. In the case of correlated errors, however, the error of the fused result can turn out worse than that of some inputs [Lew04]. At the same time, increased confidence is associated with the fused information, since it is expected to be based on independent observations and therefore more dependable.

In the case of confidence-weighted averaging (section 3.3), this effect can be demonstrated quite easily. If we consider three sensors return data with a statistical error variance of 1, the fusion according to confidence-weighted averaging will assign a variance of $\frac{1}{3}$ for the fused result. If we assume, however, that the three data sources return exact replicas of each other so that the errors they commit are exactly the same and therefore perfectly positively correlated, the improvement in the variance of the result is illusory. Due to the strong correlation between the input errors the fusion process is not really able to reduce the uncertainty in the result and the fused variance in reality is once again equal to 1. The graphs in figure 4.1 visualize the effects of such correlations on the fused error distribution.

A similar problem arises with the approach of interval intersection. If sensors show highly correlated behavior in the errors they commit, their assigned intervals will be more likely to intersect. Since the fused interval depends on the number of overlapping intervals, such correlated intervals will be more likely to be assumed as part of the correct fused interval, even though their congruency does not stem so much from accurate measurements




(a) Distribution of error for two input variables




(b) Fused distribution of error in case of independent and correlated inputs

Figure 4.1: Effect of independence or correlations between error distribution on the fused distribution: distribution for uncorrelated (solid line) and positively correlated inputs (dotted line).

as from correlated errors.

In reality sensor measurements will hardly ever be perfectly correlated since the scenario of fusing perfect replicas is of little use in a sensor network. However, as mentioned, sensors can show significantly correlated error behavior under certain conditions, so that the fused variance will be found somewhere between the two scenarios of completely uncorrelated and perfectly positively correlated sensors.

Including known correlations between the inputs of a fusion process can be expected to yield two significant benefits. Firstly, considering dependencies between measurements can improve the accuracy of the fusion result. The additional information can help gauge the influence of each input on the fusion process more carefully. As a second advantage, the knowledge about correlated errors will help to more accurately estimate the reliability of the fused value. In the case of positive correlations, the fact that the gain in confidence is not as great as it would be if the errors were uncorrelated can be adequately incorporated in the measure of uncertainty.

## 4.2 Extended Confidence-Weighted Averaging

As in the case of confidence-weighted averaging, in our new approach the fused value is to be calculated as a weighted average of all input measurements (4.3). The weights are to be determined according to the reliability of each source as measured by its error variance as well as considering dependencies between the errors committed by each sensor. The sensors are assumed to be calibrated so that the expected error is 0 for each sensor, and the fusion result shall be unbiased as well, with minimum expected error variance.

$$x_{FUSED} = \sum_{i=1}^{n} w_i x_i = \mathbf{w}^T \mathbf{x} \tag{4.3}$$

The $n \times n$ covariance matrix $\mathbf{\Sigma} = \sigma_{ij}$ contains the error variances for each of the $n$ sensors in $\sigma_{ii}$, and the respective covariances between two sensors $i$ and $j$ in $\sigma_{ij}, i \neq j$. The variance of the weighted sum of $n$ such correlated inputs is derived according to (4.4). The method introduced in section 3.3 is thus only a special case of the approach to be developed, in which $\sigma_{ij} = 0$ for $i \neq j$.

$$\sigma^2_{FUSED} = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \sigma_{ij} \tag{4.4}$$

The weight vector $\mathbf{w} = w_i$ should be chosen so that the expected fused variance be minimized, i.e., so that it solves the minimization problem

$$\arg\min_{\mathbf{w}} \mathbf{w}^T \mathbf{\Sigma} \mathbf{w} \tag{4.5}$$

under the condition that the sum of all weights is 1.

The following notation will be used: $\mathbf{c_1}$ is the vector containing the elements of the first column of $\mathbf{\Sigma}$ except its first element $\sigma_{11}$. $\mathbf{C}^*$ is a matrix of dimensions $(n-1)\times(n-1)$ equivalent to $\mathbf{\Sigma}$ without its first row and column. $\mathbf{w}^*$ designates the weight vector $\mathbf{w}$ without its first element $w_1$. Finally, $\boldsymbol{\sigma_{11}}$ represents a column vector of $n-1$ replications of $\sigma_{11}$, and $\mathbf{1}$ is a column vector of $n-1$ replications of 1.

The optimal weights are obtained by calculating the first partial derivative of $\sigma^2_{FUSED}$ and setting them equal to 0. The sum of all weights has to equal 1, which implies that $w_1 = 1 - \sum_{i=2}^{n} w_i$ or $w_1 = 1 - \mathbf{1}^T\mathbf{w}^*$, and the fused variance can be expressed as

$$\sigma^2_{FUSED} = \sigma_{11}(1-\mathbf{1}^T\mathbf{w}^*)^2 + 2(1-\mathbf{1}^T\mathbf{w}^*)\mathbf{c_1}^T\mathbf{w}^* + \mathbf{w}^{*T}\mathbf{C}^*\mathbf{w}^* \tag{4.6}$$

which is equivalent to

$$\sigma^2_{FUSED} = \sigma_{11} - 2\boldsymbol{\sigma_{11}}^T\mathbf{w}^* + \sigma_{11}(\mathbf{1}^T\mathbf{w}^*)^2 + 2\mathbf{c_1}^T\mathbf{w}^* - 2\cdot\mathbf{1}^T\mathbf{w}^*\mathbf{c_1}^T\mathbf{w}^* + \mathbf{w}^{*T}\mathbf{C}^*\mathbf{w}^* \tag{4.7}$$

The partial derivative respective to $\mathbf{w}^*$ of this expression is

$$\frac{\partial\sigma^2_{FUSED}}{\partial\mathbf{w}^*} = -2\cdot\boldsymbol{\sigma_{11}} + 2\cdot\boldsymbol{\sigma_{11}}\mathbf{1}^T\mathbf{w}^* + 2\cdot\mathbf{c_1} - 2\cdot\mathbf{1}^T\mathbf{w}^*\mathbf{c_1} - 2\cdot\mathbf{1}\mathbf{c_1}^T\mathbf{w}^* + \mathbf{C}^*\mathbf{w}^* + \mathbf{w}^{*T}\mathbf{C}^* \tag{4.8}$$

$\mathbf{C}^*$ is a symmetrical matrix, therefore the expressions $\mathbf{C}^*\mathbf{w}^*$ and $\mathbf{w}^{*T}\mathbf{C}^*$ are equivalent. Equation (4.8) is thus set equal to 0 and can be simplified to the following expression:

$$\frac{\partial\,\mathbb{V}\left(X_{FUSED}\right)}{\partial\mathbf{w}^*} = -\boldsymbol{\sigma_{11}} + \boldsymbol{\sigma_{11}}\mathbf{1}^T\mathbf{w}^* + \mathbf{c_1} - \mathbf{1}^T\mathbf{w}^*\mathbf{c_1} - \mathbf{1}\mathbf{c_1}^T\mathbf{w}^* + \mathbf{C}^*\mathbf{w}^* = 0 \tag{4.9}$$

From this $\mathbf{w}^*$ can easily be isolated and we receive

$$\mathbf{w}^* = (\boldsymbol{\sigma_{11}}\mathbf{1}^{\boldsymbol{T}} - \mathbf{c_1}\mathbf{1}^T - \mathbf{1}\mathbf{c_1}^T + \mathbf{C}^*)^{-1}(\boldsymbol{\sigma_{11}} - \boldsymbol{c_1}) \tag{4.10}$$

from which in turn we can calculate the last required weight $w_1$ as $w_1 = 1 - \mathbf{1}^T\mathbf{w}^*$ so that $\mathbf{w} = [w_1, \mathbf{w}^*]^T$.

## 4.3 Fault Tolerance

Extended confidence-weighted averaging, like its precursor, does not make a binary decision between faulty and accurate sensors, but implements a smooth transition from dependable to very undependable sources. This decision is based on statistical information about previous sensor behavior, and compensates for *nominal* errors as defined in section 2.1.2.

However, *artifactual* errors may occur and not follow the general behavior of the sensor. Due to some fault a generally reliable sensor may return one measurement that is completely inaccurate. If this sensor assigns great confidence to its measurement, the faulty observation has great influence on the fused result, and may thus distort the result significantly. Such cases call for a method to identify distorting observations and exclude them from the calculations.

An optimal evaluation of distorting influences would be using a *leave-one-out* strategy, i.e., comparing the result based on all observations to those results obtained excluding one source at a time and excluding that source which caused the result to degenerate most. In a realistic sensor fusion application, however, the computational effort for such an analysis will generally not be feasible. Therefore the need for a simple approximation to the optimal approach arises.

A simple option would be to calculate a trimmed weighted mean by sorting all inputs and excluding the $t$ largest and smallest values respectively. This way the weights the observation have in the average are not considered, so that the following scenario could occur (figure 4.2): Two outliers lie close to each other, but the more extreme value has a significantly smaller weight than the second outlier. Still, if two observations are to be omitted, this most extreme value is excluded, while the outlier remains as input to the fusion process and distorts the result greatly due to its great weight.


Figure 4.2: Exclusion of outliers independent of their weight

In the case of (simple) confidence-weighted averaging (CWA), a measure that may be used to compare the distorting effect of single observations on the fused result is the *Mahalanobis distance* as defined in (4.11) [Fil03]. It measures the distance between an observation and the fused value obtained based on all observations in terms of standard deviations of that observation.

$$d_M(x_i, x_{FUSED}) = \frac{(x_i - x_{FUSED})}{\sigma_i} \tag{4.11}$$

To avoid that extreme values affect the quality of the result, a fixed number of the most distorting input values could be excluded. The preliminary fusion result is calculated to determine the magnitude of $d_M$ for each input, then $t$ observations with the greatest value for $d_M$ and $t$ with the smallest (negative) value are removed and the final fusion result is calculated from the remaining inputs [Elm02].

In extended confidence-weighted averaging (ECWA), however, an observation's influence on the fusion process does not only depend on its associated variance, but also on all its covariances with other inputs. The Mahalanobis distance can therefore not be adopted directly. It is however obvious from equation (4.11), that if taken to the second power this measure is inversely proportional to a source's variance, and consequently directly proportional to its CWA-weight. We can therefore state

$$d_M^2(x_i, x_{FUSED}) \propto w_i(x_i - x_{FUSED})^2 \tag{4.12}$$

and define a measure of distortion $d_{ECWA}$ for extended confidence-weighted averaging as

$$d_{ECWA}^2(x_i, x_{FUSED}) = w_i(x_i - x_{FUSED})^2 \tag{4.13}$$

The drawback of excluding a fixed number of observations is that faulty as well as non-faulty sensor measurements are excluded. The effects of the exceptional case of faulty data are mitigated at the cost of a loss of valuable information in the general case of well-functioning sensors. The tradeoff between fault-tolerance and loss of reliable data can be reduced by not setting a fixed number of observations to be excluded and instead defining a limit up to which an observation is not considered "too distorting", resulting in graceful degradation of the fusion results.

## 4.4 Implementational Aspects

The algorithm of extended confidence-weighted averaging is centered around the calculation of weights which entails a matrix inversion. If variances and covariances can assumed to be constant, the weights only have to be calculated once. With the employment of *smart sensors* which possess the capability of self-diagnosis [Sch05], the estimate of a sensor's error may vary, so that variances, covariances and, as a result, the fusion weights need to be updated continually.

The problem of matrix inversion generally has a complexity of $O(n^3)$, which can be reduced to $O(n \log^2 n)$ through an approximation using hierarchical matrices [Hac01]. However, this approximation requires preprocessing of the matrices, an additional computation which may only result practical for very large matrices. The memory requirement for storing a matrix grows quadratically with the number of sensors. Confidence-weighted averaging, on the other hand, has linear complexity and memory requirements.

To compare the computational requirements of the two weighted average algorithms in a realistic application, they were implemented for the fusion of four sensors on an AVR ATmega 8515 microcontroller clocked at 7.3728MHz with no floating point unit. The implementation was realized in C using 32-bit floating point numbers and compiled by the AVR-GCC 3.3.2 compiler.

To emulate the employment of the algorithms in on-line sensor fusion, the correlation matrix was taken to be constant and updated with every set of observations using a confidence factor obtained from each sensor, which corresponded to an estimated variance in a look-up table. Table 4.1 lists the observed characteristics for the fusion of four observations. They show that for this setup ECWA needed about six times as long as CWA for the execution of one fusion step and required 2.5 times as much flash memory.

| Algorithm | Execution Time | Code Size |
|---:|:---:|:---:|
| CWA | 10.8 ms | 1,638 B |
| ECWA | 59.6 ms | 4,114 B |

Table 4.1: Algorithm characteristics for the fusion of four inputs

## 4.5 Chapter Summary

A new method for the stateless fusion of sensors with correlated error behavior has been presented in this chapter. The methods calculates a weighted average of all inputs, assigning the weights according to the variance of each source and existing cross-correlations. Its result is optimal in the sense that it minimizes the expected variance of the result. In the case of unbiased sensors, this variance is equivalent to the mean squared error. If there is a remaining bias in the measurements, and the magnitude of this bias can be estimated, the estimated mean squared error can be employed instead of variances to achieve a fusion result minimizing the mean squared error.

To forestall that the fusion result be distorted by outliers, a mechanism for outlier detection similar to the Mahalanobis distance has been proposed. A fixed number of faults can thus be removed by identifying the possibly most distorting observations and excluding them from the fusion process.

In the case that sensor performance can be evaluated dynamically and the estimate of variance or mean squared error, as a result, varies, the covariance matrix is not static. The dependencies between the sensors, however, are assumed to be unchanging. In this case, only the matrix of correlation coefficients would be constant and from this the covariance matrix would be updated each time the newly estimated variances are reported to the fusion process.

*By far the best proof is experience.*

SIR FRANCIS BACON

# Chapter 5

# Experimental Evaluation

To demonstrate the improvements achieved through extended confidence-weighted averaging, this new method has been evaluated on real sensor data. This chapter presents the experimental setup and compares the performance of the proposed method to that of other established approaches.

## 5.1 Data Collection

For the generation of real sensor data on which to analyze the performance of the new proposed approach, the *Smart Car* - an autonomous mobile robot equipped with various distance sensors, depicted in figure 5.1 - was employed. The robot's task is to navigate through an obstacle course by employing sensor fusion and path planning methods [Köß06].

### 5.1.1 Smart Car Architecture

The architecture of the Smart Car can be categorized into four levels as depicted in figure 5.2 [Kli06]. The first level is the mechanical hardware layer and consists of a four wheeled model car fitted with a wooden board on which its electrical and electromechanical hardware are mounted. The electromechanical hardware layer comprises the sensors detailed in section 5.1.2, power supplies, LED indicators, and servos for controlling the alignment of the infrared sensors and the steering direction.

The electrical hardware layer consists of a distributed fieldbus network, in which TTP/A nodes are connected by a TTP/A communication fieldbus.

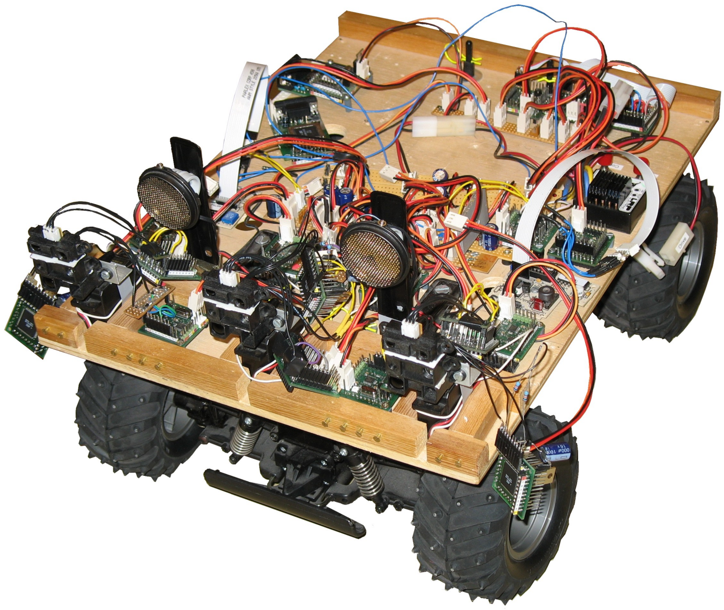


Figure 5.1: The autonomous mobile robot Smart Car

The nodes include sensor nodes which collect raw data from the sensors, servo nodes which control the servos on which the sensors are mounted and the servo controlling the steering direction. An motor control node regulates the speed of the vehicle, the navigation node performs fusion and path planning algorithms, and a display node shows debugging information. Finally, the software layer implements the TTP/A protocol software and the application software which performs sensor fusion and applies path planning techniques.

### 5.1.2 Sensors

Ten distance sensors of various types have been mounted on the car, nine of which are infrared sensors and the last one an ultrasonic sensor.

#### Infrared Sensors

Three groups, each comprising three identical sensors, make up the set of infrared sensors. The first group, designated IR 1 - IR 3, is of the type Sharp GP2Y0A02 which is designed for measuring distances between 20 cm and 150 cm. This type of sensor takes a continuous distance reading and returns an

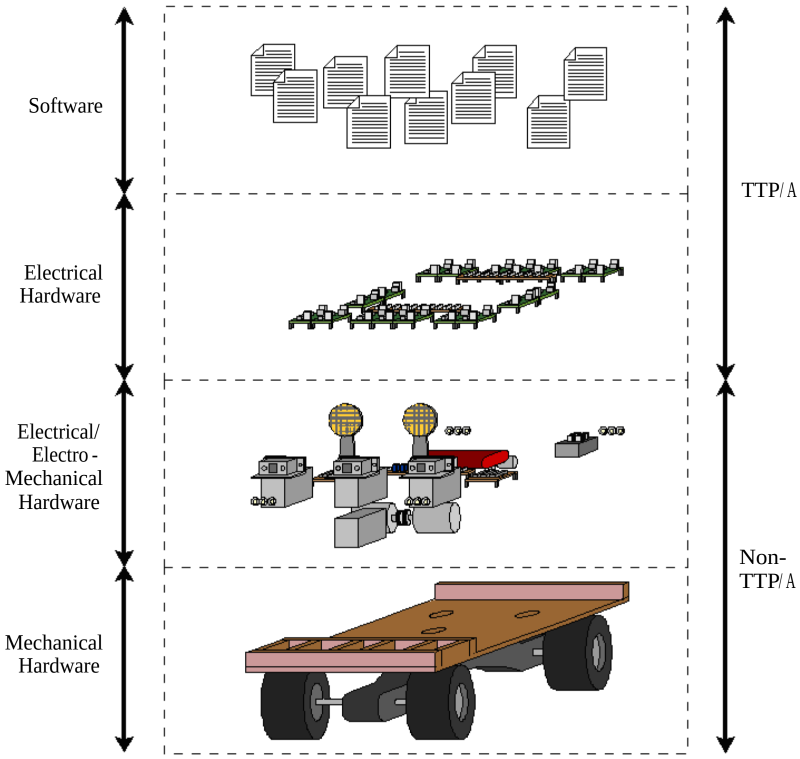



Figure 5.2: Four-level architecture of the smart car

analog voltage corresponding to the reciprocal measured distance. The sensor can provide reliable distance measurements under varying light conditions and shows little susceptibility to the color and reflectivity of the observed object. The Sharp GP2D12 model (IR 4 - IR 6) is an analog sensor as well and has the same properties as the first, but has a detection range of 10-80 cm. The third group (IR 7 - IR 9) consists of three Sharp GP2D02 digital sensors, designed for distances of 10-80 cm. They only take measurements when requested and return their measurements as digital values via a serial bus.

The technique employed by these sensors to determine distances is triangulation. The sensor emits infrared light and measures the angle at which it returns to the detector if it has been reflected by an object within the sensor's range. Since the position of the emitter relative to the detector is known, the distance to the object can be inferred from that angle. Figure 5.3 shows the exemplary measurement values returned by sensor IR 1 for given distances. According to its specifications, the sensor follows a visible trend

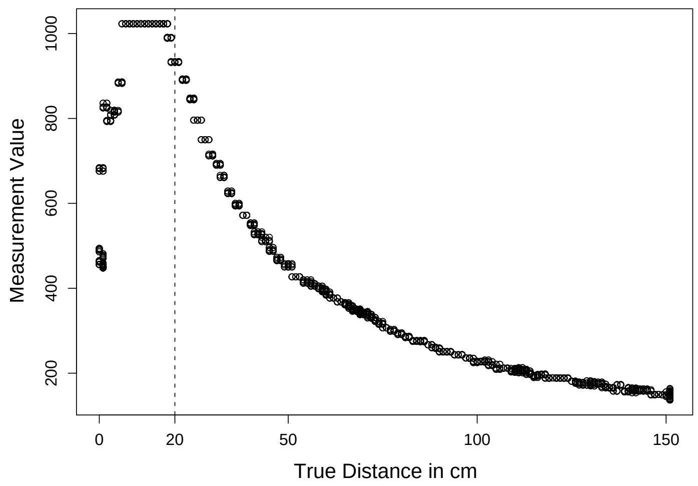



Figure 5.3: Response of sensor IR 1 for given distances

for distances greater than 20 cm. In the case of shorter ranges outside the specified measurement range, however, the returned values can not be distinguished from measurements for greater distances and are therefore excluded from the calibration process.

Equation (5.1) expresses the hyperbolic function that approximates the conversion function by which the actual distance $dist$ can be derived from the sensor measurement $x$ [Acr06]. The sensor dependent parameters $a, b$ and $c$ have to be established by fitting a nonlinear regression line minimizing the squared error of the observations. Table 5.1 lists the constants determined for each sensor during calibration. The data used for calibration were measurements obtained while only one sensor at a time was active. The calibration parameters for sensors IR 1 through IR 6 proved constant for all further tested sensor configurations. IR 7 though IR 9, however, showed significant deviations from the original parameters if they were employed together with any of the analog sensors, so they had to be recalibrated for each test setup.

$$dist = \frac{a}{x - b} + c \tag{5.1}$$

The values returned by the sensors are integers, so that the resolution of the measurements is limited and worsens with increasing distance due to the shape of the transformation function. For sensors of type GP2Y0A02, distances can be distinguished with a maximum precision of 0.03 cm if the

| Sensor Type | Sensor ID | $a$ | $b$ | $c$ |
|---|---|---|---|---|
| GP2Y0A02 | IR 1 | 27099 | -22 | -7 |
| | IR 2 | 28472 | -30 | -8 |
| | IR 3 | 20746 | -1 | -5 |
| GP2D12 | IR 4 | 9604 | 12 | -4 |
| | IR 5 | 6086 | 59 | 0 |
| | IR 6 | 7441 | 34 | -3 |
| GP2D02 | IR 7 | 1494 | 71 | -4 |
| | IR 8 | 2103 | 47 | -9 |
| | IR 9 | 2018 | 69 | -12 |

Table 5.1: Sensor constants determined for calibration

object and the sensor are 20 cm apart, while at a distance of 80 cm the precision is 0.3 cm. For GP2D12 sensors, the precision ranges from 0.06 cm to 1 cm for the respective distances, and for GP2D02 from 0.4 cm to 4.5 cm.

### Ultrasonic Sensor

The tenth sensor is a Polaroid 6500 Series ultrasonic sensor. It is designed for ranges from approximately 20 cm to 10.5 m [Sen04]. The value returned by the sensor represents the time that has elapsed between the emitting of an ultrasonic ping and its return to the sensor. Therefore, there is a linear relation between the distance to an object and the value returned by the sensor.

Figure 5.4 plots the measurements of the ultrasonic sensor for given distances. For ranges smaller than 35 cm the sensor returns close to the same signal so that no information can be gathered. For distances greater than 35 cm the measurement values show the expected linear relation with the true distance. Intercept and slope of the linear conversion function in (5.2) are determined by fitting a linear regression line that minimizes the squared error.

$$dist = 0.076x - 12.78 \tag{5.2}$$

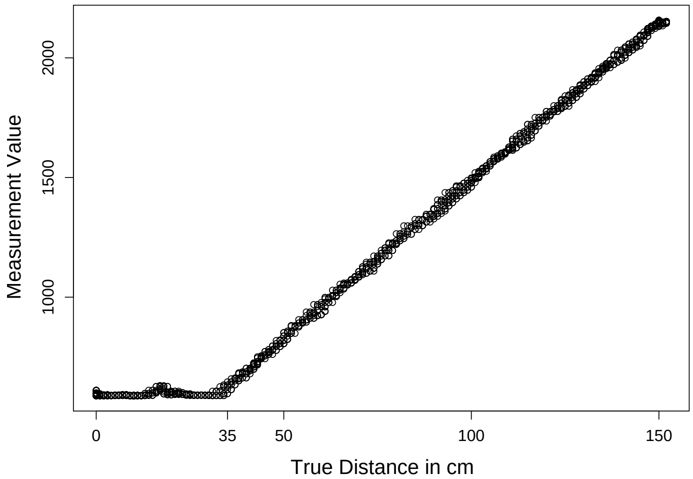



Figure 5.4: Response of ultrasonic sensor for given distances

#### Odometric Reference Sensor

A shaft encoding sensor was employed as a reference sensor for the distance measurements. It measures the distance the smart car has travelled from a known starting point for each instant in which the distance sensors measured the distance to an obstacle. Using this sensor, the reference distance to the observed obstacle was determined with sufficient accuracy. The sensor is equipped with a measuring wheel of 6.23 cm in diameter, delivering 100 ticks per revolution, so that every tick corresponds to $\frac{1}{9.75}$ cm travelled.

### 5.1.3 Experimental Setup

To evaluate and compare the performance of different fusion algorithms, the sensor on the smart car took concurrent measurements of the distance between them and a test obstacle. The smart car was positioned at approximately 150cm from the obstacle and slowly moved towards it, measuring the travelled distance. The sensor measurements were taken at a frequency of 25 observations per second in the case of the infrared sensors and at 6.25 observations per second by the ultrasonic sensor. The data was transmitted via the TTP/A fieldbus to a PC (see figure 5.5), where the fusion process and analysis were performed off-line using the statistical software **R** [R D05].

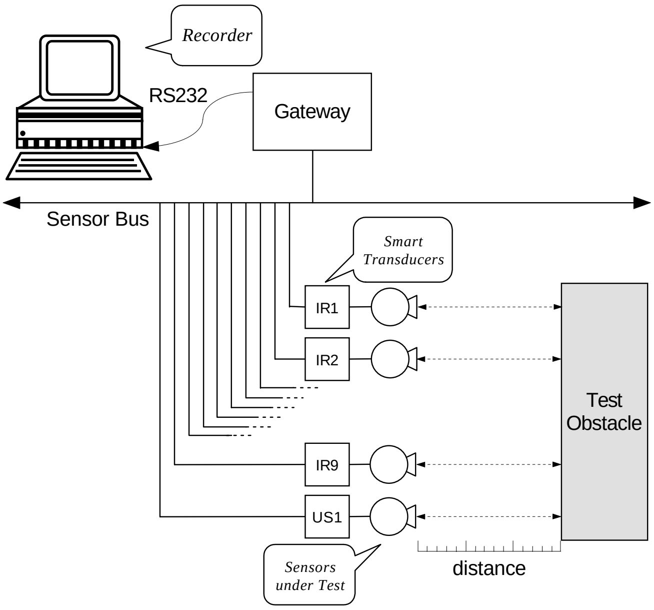



Figure 5.5: Test Setup

| Configuration | Active Sensors |
|---|---|
| *A* | IR 1 - 3 |
| *B* | IR 4 - 6 |
| *C* | IR 7 - 9 |
| *AC* | IR 1 - 3, IR 7 - 9 |
| *BC* | IR 4 - 9 |

Table 5.2: Tested sensor configurations

Different sets of sensors were employed concurrently, to observe interactions between the different groups. Table 5.2 lists the tested infrared sensor constellations and their label which will be used to refer to them in the following analysis. At first each group of identical sensors was tested, then each set of analog sensors was coupled with the set of digital infrared sensors. Additionally to the infrared sensors, the ultrasonic sensor was employed in all test series.

To simulate the fusion process for each sensors set, the reported data was split into two independent data sets. The covariance matrices were estimated from only part of the data. Based on these parameters, the performance of the fusion methods was evaluated on the remaining observations.

| | Variance | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Test | Infrared Sensors | | | | | | | | | US |
| set | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| *A* | 2.06 | 1.83 | 2.24 | - | - | - | - | - | - | 1.67 |
| *B* | - | - | - | 7.53 | 13.79 | 6.93 | - | - | - | 0.90 |
| *C* | - | - | - | - | - | - | 1.71 | 7.51 | 6.23 | 0.54 |
| *AC* | 4.88 | 3.34 | 1.13 | - | - | - | 1272.92 | 11.09 | 15.29 | 1.86 |
| *BC* | - | - | - | 19.43 | 4.69 | 11.67 | 35.34 | 21.69 | 16.77 | 6.20 |

Table 5.3: Error variance of each sensor for tested sensor configurations

## 5.2 Performance Evaluation

Analyzing the uncertainty of measurement of each sensor we found great differences between the various sensors. Additionally, we found that depending on the sensor configuration the uncertainty would vary since apparently the sensor signals interfered with each other. Particularly in the configuration *AC* the variance of sensor IR 7 increases greatly, due to great measurement errors for ranges above 50 cm. Table 5.3 therefore summarizes the uncertainty of each sensor expressed as its estimated variance for the different sensor configurations tested.

The algorithms to be evaluated and compared are the fault-tolerant sensor averaging algorithm, the fault-tolerant interval intersection, confidence-weighted averaging (CWA) and extended confidence-weighted averaging (ECWA). In comparing the results of various fusion methods, we considered the true *mean squared error* (MSE) of the results, the true *mean absolute error* (MAE) as well as the true variance of the result and the estimated variance as calculated by the fusion method.

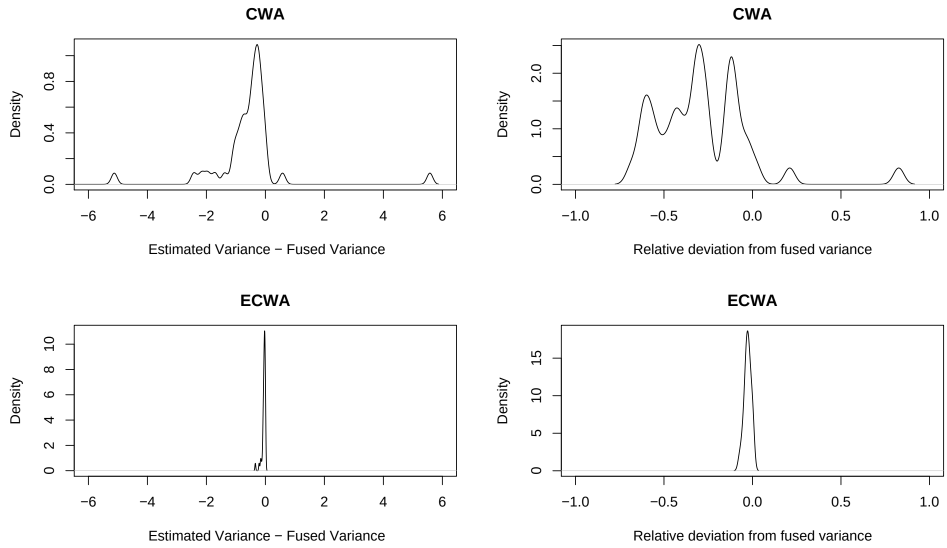
### 5.2.1 Comparison of CWA and ECWA

Table 5.4 lists exemplary results for the fusion methods CWA and ECWA for fusing the data obtained from test set *A*. For results of other sensor configurations see appendix B.

The results show that ECWA generally performs better in terms of the mean squared error of the fusion result. In the case of CWA, the variance is significantly underestimated for all tested configurations. On the other hand, ECWA delivers more accurate estimates. In terms of relative error,

| Sensors | Method | MSE | MAE | Fused Variance | Estimated Variance |
|---|---|---|---|---|---|
| IR1+2 | CWA | 5.50 | 1.71 | 5.29 | 3.33 |
| | ECWA | 5.35 | 1.76 | 5.35 | 5.13 |
| IR1+2 + US | CWA | 0.65 | 0.64 | 0.64 | 0.46 |
| | ECWA | 0.61 | 0.62 | 0.55 | 0.52 |
| IR1+2+3 | CWA | 1.79 | 1.07 | 1.52 | 1.14 |
| | ECWA | 1.67 | 1.05 | 1.37 | 1.30 |
| IR1+2+3 +US | CWA | 0.58 | 0.6 | 0.48 | 0.36 |
| | ECWA | 0.58 | 0.6 | 0.43 | 0.40 |

Table 5.4: Comparison of fusion results for CWA and ECWA




(a) Absolute deviation (b) Relative deviation

Figure 5.6: Deviation of estimated variance from true fused variance over all tested configurations

for all tested setups the mean relative deviation from the true fused variance is -28.9% in the case of CWA, but only -2.7% in the case of ECWA. Figure 5.6(a) visualizes the absolute deviation from the true fused variance, which takes much greater values for CWA than for ECWA. The distribution of the relative deviation depicted in figure 5.6(b), shows an underestimation of up to -68.6% for CWA, while the greatest deviation committed by ECWA is -7.4%.

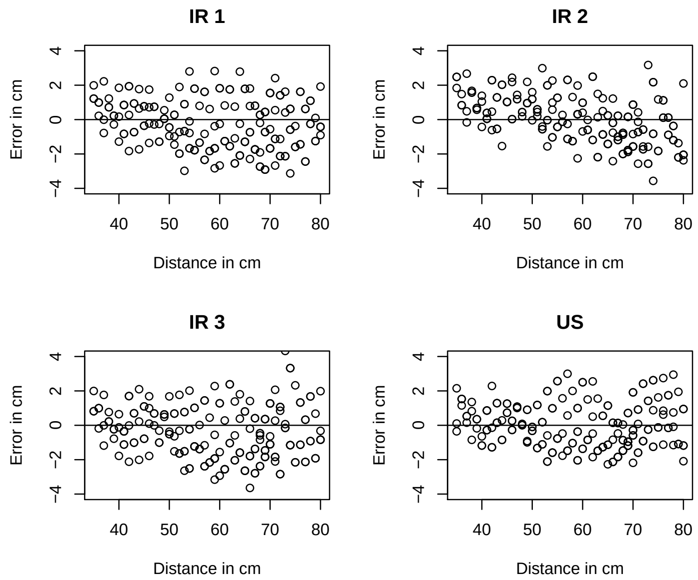



Figure 5.7: Error distributions over true distance for three infrared and one ultrasonic sensor

The great discrepancy between estimated and true variance in the fusion result stems from error correlations within the data that are not considered in CWA. In the case of sensor set $A$, these correlations exist not only between the infrared sensors, but also between infrared sensors and the ultrasonic sensor.

Figure 5.7 shows the distribution or measurement error over the true distance for each sensor in the set. With the exception of sensor IR 2, for which a slight bias may be diagnosed, the error distributions seem to be fully stochastic, and in fact there is no significant correlation between the true distance and the error. Still, there are correlations between the errors committed by each sensor, as figure 5.8 indicates for the case of IR 1 and the ultrasonic sensor. Though both sensor have have mean error of zero, if they overestimate the measurement, they tend to do so at the same time. The cause for this may be a factor that influences both sensors but is not observed. Admittedly, sensor set $A$ showed the strongest correlations. Though weaker, there were dependencies between errors in all tested sets that lead to an improvement through ECWA, as the results in table B.1 indicate.

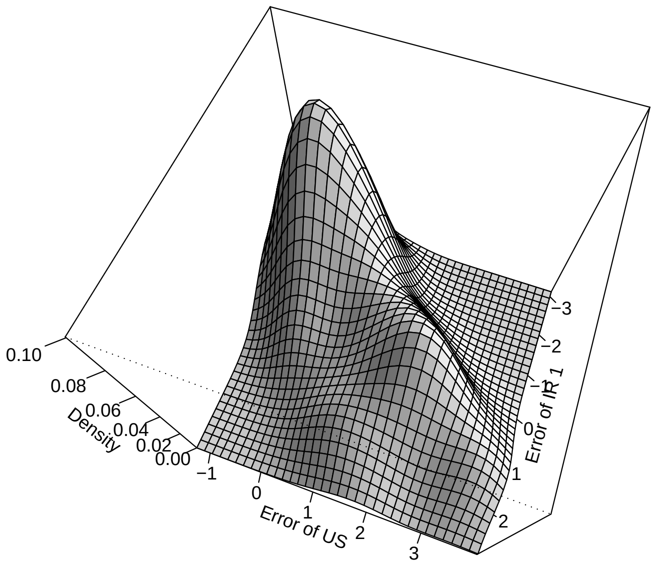



Figure 5.8: 2-dimensional density distribution for IR 1 and the ultrasonic sensor

### 5.2.2 Performance of Fault-Tolerant Approach

The interval intersection methods proposed for sensor fusion (see section 3.2) both explicitly implement fault tolerance in the sense that they can still return correct fused intervals as long as less than half the inputs are faulty. CWA and ECWA can be modified to tolerate outliers as well employing the algorithm outlined in 4.3.

Table 5.5 summarizes the results for the four discussed methods for an increasing number of explicitly tolerated faults $t$. Faults in this analysis were assumed to be contained in the data, no additional errors were added. The analyzed sensors were those of set $BC$. In the case of the two interval-based methods, a great improvement in terms of mean squared error is already noted when tolerating only one fault. The MSE changes only little for $t$ between 1 and 5, however the smallest maximum squared error was achieved by both methods for $t = 3$.

The average performance and maximum error of CWA is best regarding all listed characteristics when it does not tolerate any faults at all. Even though for this data set fault-tolerance has not lead to an improvement in the fused result, the mean squared errors are still in all cases better than for any of the interval intersection methods. In the case of the ECWA the lowest MSE is achieved when the method tolerates a single fault and with increasing $t$ the performance worsens, though the committed error is still smaller than that of the interval based methods.

Figure 5.9 visualizes the discussed results. Even though fault-tolerant averaging and interval intersection can often improve their results up to a certain number of faults, they still always perform worse than CWA or ECWA. The latter two, however, can generally not improve their performance but may even worsen it.

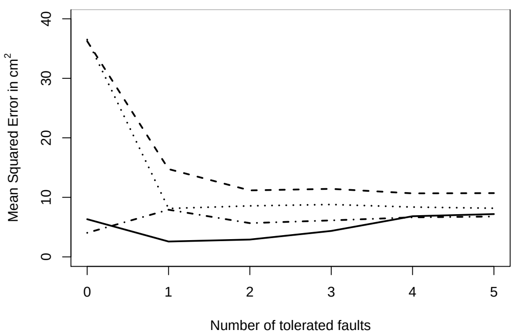
In terms of estimating the variance of the result, ECWA outperforms all methods. The difference between the estimated and the true value is close to zero for all settings (see figure 5.9(c)). CWA again underestimates the variance and thus assumes too great a quality of the fusion result. The two interval based methods, on the other hand, generally return a very pessimistic estimate of the variance.

**Varying the point of exclusion of faulty measurements**

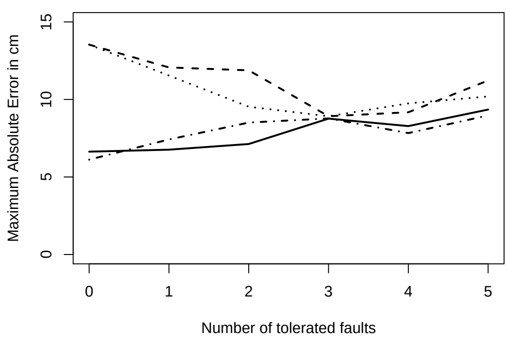
The previously discussed performance of ECWA for varying numbers of faults is contrary to our expectations. The elimination of observations seems to lead to a loss of valuable information that causes its performance to degrade.

To avoid the tradeoff between fault-tolerance and information loss, the proposed approach of excluding only "too distorting" values shall be employed and its effect on the fusion result evaluated. The results of this technique applied to ECWA are summarized in figure 5.10. In each figure, the axis of abscissae represents the limit for $d_{ECWA}$, which was evaluated in steps of 1 within the interval [0,20]. As many as $t$ observation that exceed the limit could be excluded by the algorithm.

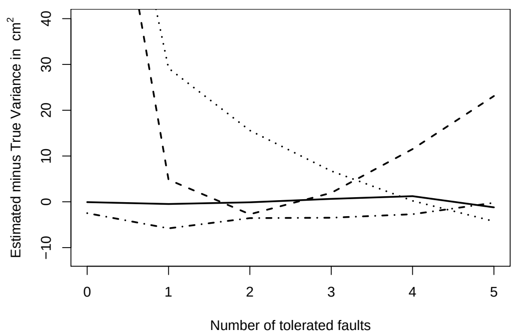
The results show that unlike expected, the performance could not be improved by this strategy. We found that outlying observations from dependable sensors had a greater chance of being excluded than faulty observations from sensors with great error variance, so that valuable information would be excluded instead of the true error. With increasing numbers of faults, performance improves as the limit for $d_{ECWA}$ increases, until the level of error achieved by excluding one observation is reached.




(a) Mean squared error




(b) Maximum absolute error



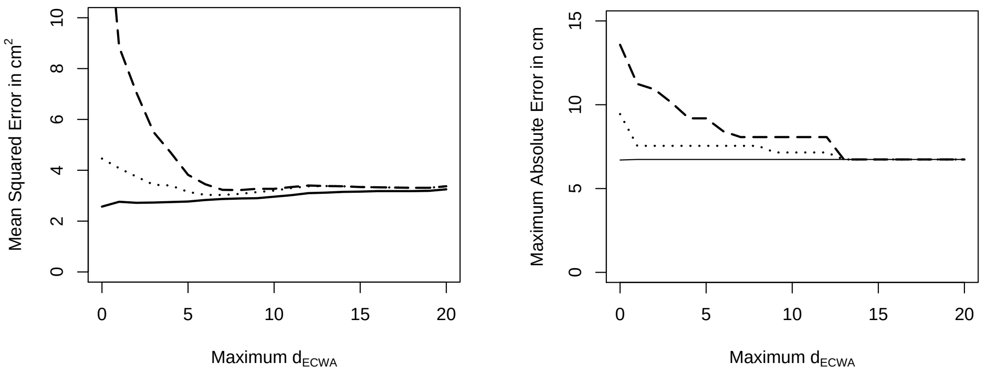

(c) Difference between estimated and true variance

Figure 5.9: Performance in terms of MSE, maximum absolute error and estimated variance for different $t$ on sensor set $BC$: Fault-tolerant averaging (dashed line), fault-tolerant interval intersection (dotted line), CWA (dot-and-dash line) and ECWA (solid line).

| Method | $t$ | True MSE $cm^2$ | True MAE $cm$ | True Variance $cm^2$ | Estimated Variance $cm^2$ | Maximum squared error $cm^2$ |
|---|---|---|---|---|---|---|
| Fault-tolerant Sensor Averaging | 0 | 36.24 | 5.54 | 5.60 | 113.08 | 183.15 |
| | 1 | 14.75 | 3.14 | 14.71 | 19.54 | 145.52 |
| | 2 | 11.17 | 2.74 | 10.80 | 8.08 | 141.04 |
| | 3 | 11.43 | 2.79 | 10.78 | 12.73 | 79.50 |
| | 4 | 10.67 | 2.69 | 10.13 | 21.64 | 84.14 |
| | 5 | 10.71 | 2.70 | 10.25 | 33.35 | 125.93 |
| Fault-tolerant Interval Intersection | 0 | 36.51 | 5.56 | 5.69 | 118.65 | 183.15 |
| | 1 | 8.13 | 2.24 | 8.11 | 37.28 | 133.19 |
| | 2 | 8.57 | 2.31 | 8.51 | 24.09 | 90.79 |
| | 3 | 8.80 | 2.27 | 8.75 | 15.48 | 79.50 |
| | 4 | 8.37 | 2.21 | 8.34 | 8.55 | 94.98 |
| | 5 | 8.17 | 2.19 | 8.19 | 3.93 | 103.90 |
| CWA | 0 | 4.04 | 1.55 | 4.02 | 1.54 | 37.42 |
| | 1 | 7.92 | 2.30 | 7.50 | 1.70 | 55.05 |
| | 2 | 5.67 | 1.83 | 5.64 | 2.08 | 72.38 |
| | 3 | 6.13 | 1.81 | 6.09 | 2.61 | 77.00 |
| | 4 | 6.64 | 1.93 | 6.61 | 3.90 | 61.30 |
| | 5 | 6.78 | 1.88 | 6.75 | 6.54 | 80.43 |
| ECWA | 0 | 6.33 | 2.30 | 1.05 | 0.98 | 44.00 |
| | 1 | 2.58 | 1.20 | 1.95 | 1.47 | 45.70 |
| | 2 | 2.91 | 1.27 | 2.25 | 2.14 | 50.77 |
| | 3 | 4.36 | 1.61 | 3.49 | 4.13 | 76.79 |
| | 4 | 6.83 | 2.72 | 10.52 | 11.75 | 68.58 |
| | 5 | 7.19 | 3.28 | 17.75 | 16.54 | 87.36 |

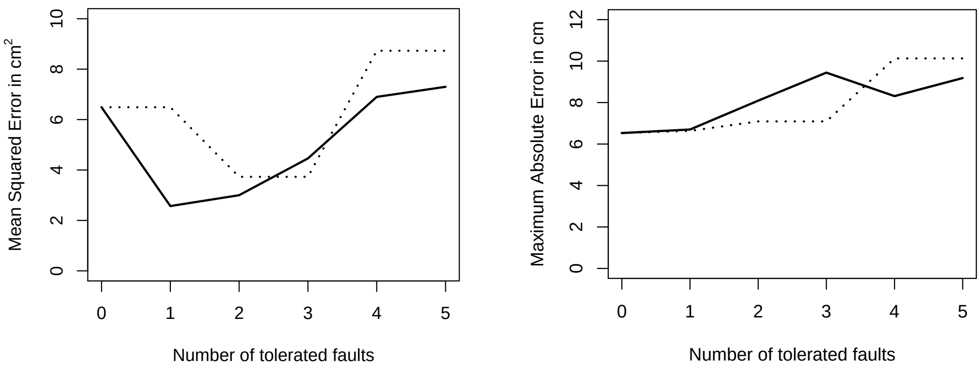
Table 5.5: Performance of tested fusion algorithms on sensor set $BC$, for a varying number of tolerated faults $t$

Employing the strategy of a trimmed mean would not improve the results greatly either (figure 5.11). The achieved mean squared error was generally of the same magnitude or greater than the previous exclusion method, however, the maximum absolute error could be improved.




(a) Mean squared error (b) Maximum absolute error

Figure 5.10: Performance of ECWA for different limits of $d_{ECWA}$ for tolerating $t$ faults: $t = 1$ (solid line), $t = 3$ (dotted line), $t = 5$ (dashed line).




(a) Mean squared error (b) Maximum absolute error

Figure 5.11: Performance of ECWA for a varying number of tolerated faults to be excluded, employing $d_{ECWA}$ (solid line) and a trimmed mean (dotted line) as exclusion criteria.

**Simulation**

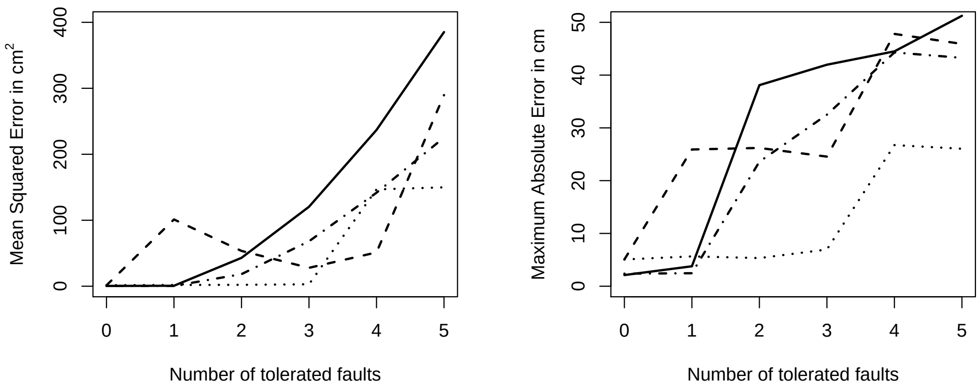
The data collected by the robot may be of too good a quality to make general conclusions due to it having been collected in a testing environment. It may not contain enough faulty data to reliably investigate the methods' behavior in the presence of a greater number and magnitude of faults. We therefore generated artificial data based on the following assumptions to investigate the performance in a general fault case.

Data for seven calibrated sources with different variances (equation(5.3)) and cross-correlations (equation(5.4)) was generated with zero-mean noise following a normal distribution. Faults were then introduced into each dataset by randomly manipulating $f$ out of the seven sensors with equal probability of selection, generating random faults for these selected sensors following a uniform distribution on the interval [-50,50] but outside of the respective sensor's range in the non-faulty case. This data emulates low-quality sensor information in the sense that every single set of "concurrent" measurements contains $f$ faulty observations and no *a-priori* information about a sensor's probability of delivering faulty data outside of its usual range of fluctuation is available.

$$\sigma_{\mathbf{i}} = \begin{pmatrix} 1.07 & 6.76 & 2.39 & 6.78 & 0.9 & 1.18 & 5.32 \end{pmatrix} \tag{5.3}$$

$$\rho_{ij} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 0.6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.9 & 0.4 \\ 0 & 0 & 0 & 0 & 0.9 & 1 & 0.4 \\ 0 & 0 & 0 & 0 & 0.4 & 0.4 & 1 \end{pmatrix} \tag{5.4}$$

On this simulated data the performance of the averaging methods is not as good as could be expected (figure 5.12). With an increasing number of faults, the mean squared as well as the maximum error increases dramatically. We therefore have to admit that on extremely faulty data for which no information about fault behavior is available, ECWA does not perform well. It would be necessary to introduce other fault-tolerant methods such as filtering previous to the fusion process to avoid such behavior.

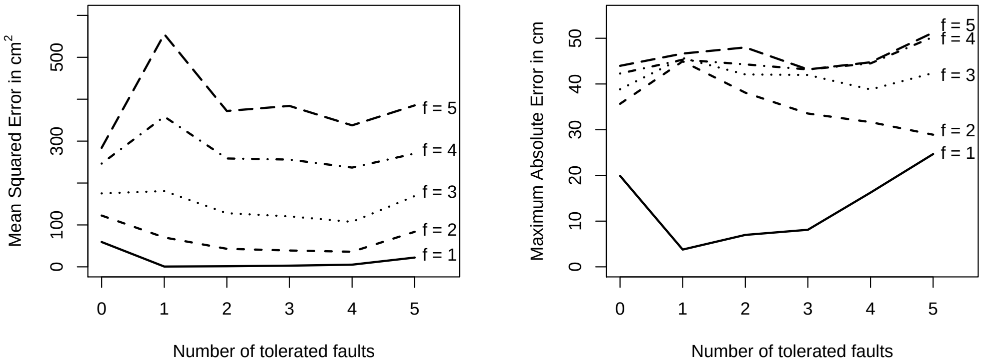



(a) Mean squared error (b) Maximum absolute error

Figure 5.12: Performance of four methods on simulated data with $f$ generated faults for $t = f$ tolerated faults:Fault-tolerant averaging (dashed line), fault-tolerant interval intersection (dotted line), CWA (dot-and-dash line) and ECWA (solid line).

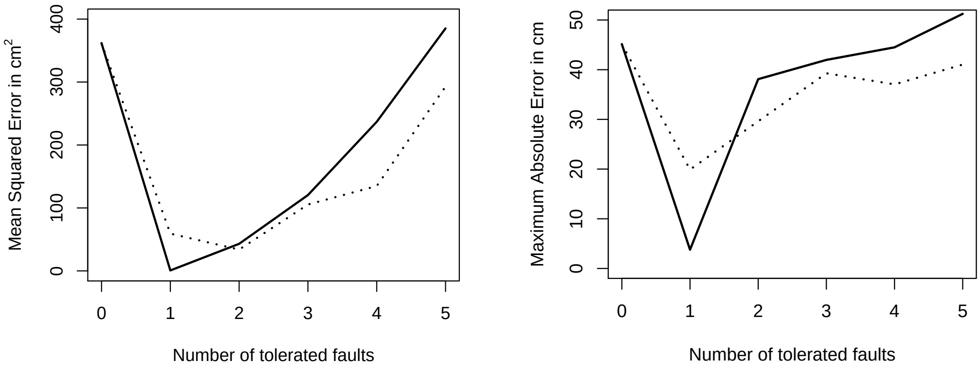
The graphs in figure 5.13 visualize the performance of ECWA for $f$ purposely created faults when $t$ faults are tolerated by the method. Both mean squared error and maximum error increase with the number of faults. As long as less than half of the inputs are faulty ($f \leq 3$), the mean squared error does improve with an increasing number of tolerated faults, though for $t = 5$ the number of excluded observation seems to be to great so that the performance worsens again.

In comparing the suggested measure $d_{ECWA}$ to a regular trimmed mean, the results on the simulated data are contrary to the findings on the real sensor data. On the artificially generated observations, the trimmed mean outperforms the other exclusion technique (figure 5.14).




(a) Mean squared error (b) Maximum absolute error

Figure 5.13: Performance of ECWA on simulated data with $f$ generated faults for varying number of tolerated faults.




(a) Mean squared error (b) Maximum absolute error

Figure 5.14: Performance of ECWA on simulated data with $f$ generated faults and $t = f$ tolerated faults, comparing $d_{ECWA}$ (solid line) and a trimmed mean (dotted line) as exclusion criteria.

## 5.3 Chapter Summary

This chapter has presented the results of an evaluation of the four discussed methods for stateless fusion on real sensor data as well as on simulated data. The sensor data was collected by the smart car, a mobile robot equipped with ten distance sensors of various types.

The analysis of performance on real sensor data showed that the two averaging methods outperformed the interval intersection methods in terms of mean and maximum error, and the approach of extended confidence-weighted averaging particularly stands out by accurately estimating the quality of the fusion result. Employing mechanisms of fault-tolerance with ECWA did not lead to any improvements.

On simulated data we showed that in a pessimistic fault scenario, the performance of ECWA can be improved through fault tolerance as long as at least half the inputs are non-faulty. In fact, on this data the simple strategy of employing a trimmed mean performed better than that of the more complex method of determining an observation's distorting influence. However, for inputs of very low quality, without any additional fault-tolerant strategies such as filtering the performance is worse than that of fault-tolerant intersection methods. The additional effort for such a fault-tolerant preprocessing is however justified by the subsequent improvement by ECWA showed on real filter data.

*If you have knowledge, let others light their candles in it.*

MARGARET FULLER

# Chapter 6

# Conclusion

## 6.1 Extended Confidence-Weighted Averaging

In this thesis a new algorithm for sensor fusion — *extended confidence-weighted averaging* (ECWA) — has been elaborated. The goal was to develop a method for stateless fusion of concurrent real-valued observations. Unlike other methods for the stateless fusion problem, it does not assume independent error behavior, an assumption that simplifies the fusion process but generally does not accurately represent sensor behavior. ECWA combines all observation by calculating a weighted average based on the uncertainty associated to each sensors and known cross-correlations between sensor errors.

In the case of uncorrelated sources, this method is equivalent to simple confidence-weighted averaging, otherwise the extended version has shown to improve the fusion results and particularly deliver a more accurate estimate of the uncertainty that remains after fusing. The advantage thus lies not only in a more accurate estimate of the observed variable, but also a more truthful estimate of confidence that subsequent fusion or decision making processes may associate to the product of the fusion process.

To avoid that faulty data from an otherwise dependable source distorts the fusion result, two mechanism for fault-tolerance have been proposed: that of a simple trimmed mean and a more sophisticated strategy of excluding observations based on their distorting influence on the fusion result. On simulated data we have shown that the trimmed mean may perform better than the second approach, however both methods may not deliver results of as good quality as the method of fault-tolerant interval intersection if the data contains many faults with unknown behavior. To attain the advantages of ECWA, it may thus be necessary to subject the data to simple preprocessing steps such as filtering.

## 6.2 Outlook

To achieve fault tolerance in ECWA, the suggested methods of excluding distorting observation have proven to be non-satisfactory, since they often overlook the true fault. To still benefit from the improvements of ECWA, additional fault-tolerant mechanisms may have to be introduced previous to the fusion process. Averaging a fixed number of sequential observations, as could be performed by a smart sensor, would be a simple method for such purpose and would be a small trade-off respecting computational effort to the benefits of ECWA. This strategy has proven feasible in the analysis of real data in our study, however, a systematic investigation into the possible improvement may be of interest.

To employ ECWA in flexible fusion architectures, e.g., as would be implementable with components that provide *plug-and-play*-capabilities, it will be of great interest to investigate the feasibility of online-estimation of correlations for newly integrated sensors. This could be realized by observing the error of already established sensors relative to the fusion result and the measurements of the sensor to be integrated. The same aspect will be relevant for recognizing changes in correlation behavior or online re-calibration of sensors.

For the implementation in embedded real-time systems, it will further be necessary to conduct a more in-depth analysis than has been presented here of the proposed algorithm's performance regarding time and resource requirements.

# Appendix A

# Symbols

## A.1 Algebraic Symbols

| Symbol | Explanation |
|---|---|
| $x$ | One-dimensional variable |
| $\mathbf{x}$ | Vector |
| $\mathbf{X}$ | Matrix |
| $\mathbf{\Sigma}$ | Covariance matrix with elements $\sigma_{ij}$ |
| $\sigma_{ij}$ | Covariance between variables $i$ and $j$ |
| $\sigma_i^2$ | Variance of variable $i$ |

## A.2 ECWA-specific notation for $n$ sensors

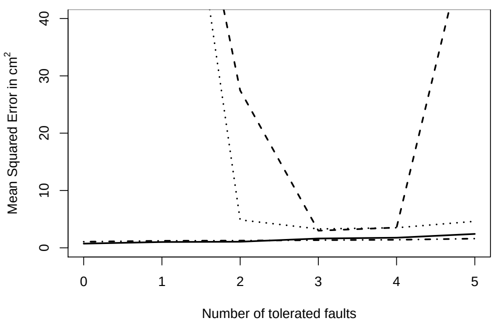
| Symbol | Explanation |
|---|---|
| $\mathbf{1}$ | Vector of $n-1$ replicas of 1 |
| $\mathbf{C}^*$ | Covariance matrix $\mathbf{\Sigma}$ without first column and row |
| $\mathbf{c_1}$ | First column of covariance matrix $\mathbf{\Sigma}$ without element $\sigma_{11}$ |
| $\boldsymbol{\sigma}_{\mathbf{11}}$ | Vector of $n-1$ replicas of $\sigma_{11}$ |

# Appendix B

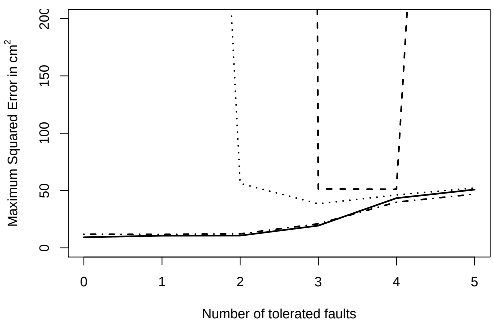
# Results of experimental evaluation

| Sensors | Method | MSE | MAE | varfused | varestim |
|---|---|---|---|---|---|
| IR1+2 | CWA | 5.50 | 1.71 | 5.29 | 3.33 |
| | ECWA | 5.35 | 1.76 | 5.35 | 5.13 |
| IR1+2 | CWA | 0.65 | 0.64 | 0.64 | 0.46 |
| + US | ECWA | 0.61 | 0.62 | 0.55 | 0.52 |
| IR1+2+3 | CWA | 1.79 | 1.07 | 1.52 | 1.14 |
| | ECWA | 1.67 | 1.05 | 1.37 | 1.30 |
| IR1+2+3 | CWA | 0.58 | 0.6 | 0.48 | 0.36 |
| + US | ECWA | 0.58 | 0.6 | 0.43 | 0.40 |
| IR4+6 | CWA | 6.37 | 2.23 | 5.32 | 3.62 |
| | ECWA | 6.46 | 2.23 | 5.33 | 5.34 |
| IR4+6 | CWA | 1.17 | 0.87 | 1.16 | 0.73 |
| + US | ECWA | 1.01 | 0.8 | 0.87 | 0.88 |
| IR4+5+6 | CWA | 4.23 | 1.66 | 4.23 | 2.85 |
| | ECWA | 4.30 | 1.64 | 4.22 | 4.22 |
| IR4+5+6 | CWA | 1.28 | 0.90 | 1.20 | 0.69 |
| + US | ECWA | 0.95 | 0.77 | 0.88 | 0.86 |
| IR8+9 | CWA | 2.13 | 1.26 | 2.00 | 1.08 |
| | ECWA | 2.14 | 1.26 | 2.01 | 2.00 |
| IR8+9 | CWA | 1.71 | 1.13 | 1.67 | 0.65 |
| + US | ECWA | 1.92 | 1.13 | 1.59 | 1.54 |
| IR7+8+9 | CWA | 1.82 | 1.16 | 1.77 | 0.68 |
| | ECWA | 1.77 | 1.12 | 1.75 | 1.69 |
| IR7+8+9 | CWA | 1.55 | 1.08 | 1.54 | 0.48 |
| + US | ECWA | 1.83 | 1.10 | 1.40 | 1.34 |

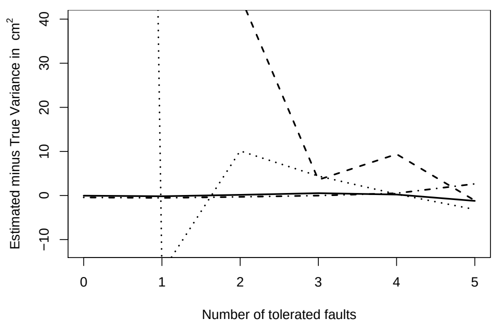
Table B.1: Comparison of fusion results for different sensor configurations




(a) Mean Squared Error




(b) Maximum Squared Error




(c) Difference between estimated and true variance

Figure B.1: Performance in terms of MSE, maximum squared error and variance for different $t$ on sensor set $AC$: Fault-tolerant averaging (dashed line), fault-tolerant interval intersection (dotted line), CWA (dot-and-dash line) and ECWA (solid line).

| Method | $t$ | True MSE $cm^2$ | True MAE $cm$ | True Variance $cm^2$ | Estimated Variance $cm^2$ | Maximum squared error $cm^2$ |
|---|---|---|---|---|---|---|
| Fault-tolerant Averaging | 0 | 134.09 | 4.18 | 131.77 | 906.65 | 14046.9 |
| | 1 | 94.58 | 3.06 | 93.54 | 383.29 | 15011.37 |
| | 2 | 27.47 | 1.80 | 27.30 | 72.48 | 13588.23 |
| | 3 | 2.99 | 1.35 | 2.46 | 6.03 | 51.38 |
| | 4 | 3.53 | 1.44 | 3.09 | 12.47 | 51.11 |
| | 5 | 59.99 | 4.95 | 59.75 | 58.62 | 1187.96 |
| Fault-tolerant Interval Intersection | 0 | 156.08 | 5.06 | 154.33 | 1319.14 | 14046.9 |
| | 1 | 99.16 | 6.39 | 97.82 | 80.45 | 1376.89 |
| | 2 | 4.87 | 1.72 | 4.47 | 14.58 | 56.21 |
| | 3 | 3.29 | 1.45 | 2.89 | 7.30 | 38.46 |
| | 4 | 3.50 | 1.47 | 3.07 | 3.39 | 46.16 |
| | 5 | 4.62 | 1.67 | 4.33 | 1.17 | 52.25 |
| CWA | 0 | 1.06 | 0.83 | 0.91 | 0.47 | 11.92 |
| | 1 | 1.22 | 0.91 | 1.09 | 0.55 | 11.78 |
| | 2 | 1.25 | 0.90 | 1.06 | 0.75 | 12.22 |
| | 3 | 1.34 | 0.94 | 1.11 | 1.09 | 20.96 |
| | 4 | 1.40 | 0.95 | 1.19 | 1.70 | 39.87 |
| | 5 | 1.60 | 0.98 | 1.34 | 3.98 | 46.88 |
| ECWA | 0 | 0.73 | 0.68 | 0.57 | 0.53 | 9.24 |
| | 1 | 1.01 | 0.82 | 0.87 | 0.70 | 10.63 |
| | 2 | 1.06 | 0.83 | 0.83 | 0.98 | 10.72 |
| | 3 | 1.62 | 1.00 | 1.15 | 1.68 | 19.39 |
| | 4 | 1.62 | 1.43 | 2.49 | 2.71 | 43.37 |
| | 5 | 2.42 | 1.90 | 5.22 | 4.00 | 50.84 |

Table B.2: Performance of tested fusion algorithms on sensor set $AC$, for a varying number of tolerated faults $t$

# Bibliography

[Acr06] Acroname. *Sharp IR Range Finder Voltage-to-Range Conversion*, 2006. Available at `http://www.acroname.com/robotics/info/articles/irlinear/irlinear.html`.

[Ald04] S.A. Aldosari and J.M.F. Moura. Fusion in sensor networks with communication constraints. In *IPSN'04: Proceedings of the third international symposium on Information processing in sensor networks*, pages 108–115, New York, NY, USA, 2004. ACM Press.

[Ant95] R.T. Antony. *Principles of data fusion automation.* Artech House, Inc., Norwood, Massachusetts, 1995.

[Avi86] A. Avižienis and J.-C. Laprie. Dependable computing: From concepts to design diversity. In *Proceedings of the IEEE*, volume 74, pages 629–638, May 1986.

[Avi04] A. Avižienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. In *IEEE Transactions on Dependable and Secure Computing*, volume 1, pages 11–33, January 2004.

[Aza96] M.H. Azadmanesh and R.M. Kieckhafer. New hybrid fault models for asynchronous approximate agreement. *IEEE Transactions on Computers*, 45(4):439–449, April 1996.

[Bau01] G. Bauer. *Transparent Fault Tolerance in a Time-Triggered Architecture.* PhD thesis, Vienna University of Technology, Institut für Technische Informatik, Vienna, Austria, April 2001.

[Bed00] M. Bedworth and J. O'Brien. The Omnibus model: a new model of data fusion? *Aerospace and Electronic Systems Magazine, IEEE*, 15(4):30–36, April 2000.

[Ben02] A.R. Benaskeur. Consistent Fusion of Correlated Data Sources. In *28th Annual Conference of the Industrial Electronics Society*, pages 2653–2656, November 2002.

[Bos96] É. Bossé, J.Roy, and D.Grenier. Data Fusion Concepts Applied to a Suite of Dissimilar Sensors. In *Canadian Conference on Electrical and Computer Engineering*, pages 692–695, May 1996.

[Boy86] J.R. Boyd. Patterns of Conflict. Set of briefing slides *A Discourse on Winning and Losing*, 1986. Available at `http://www.d-n-i.net/second_level/boyd_military.htm`.

[Bre94] C. Bregler, S.M. Omohundro, and Y. Konig. A hybrid approach to bimodal speech recognition. In *Conference Record of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 556–560, October 1994.

[Bri05] K.W. Bridges and E. Biagioni. In S.S. Iyengar and R.R. Brooks, Editors, *Distributed Sensor Networks*, chapter Plant Monitoring with Special Reference to Endangered Species, pages 1039–1047. Chapman & Hall/CRC, Boca Raton, Florida, 2005.

[Bro96] R.R. Brooks and S.S. Iyengar. Robust distributed computing and sensing algorithm. *IEEE Computer*, 29(6):53–60, June 1996.

[Bro98] R.R. Brooks and S.S. Iyengar. *Multi-Sensor Fusion: Fundamentals and Applications with Software*. Prentice Hall, New Jersey, 1998.

[Cha90] Y.A. Chau and E. Geraniotis. Multisensor correlation and quantization in distributed detection systems. In *Proceedings of the 29th IEEE Conference on Decision and Control*, volume 5, pages 2692–2697, December 1990.

[Che98] T. Chen and R.R. Rao. Audio-visual integration in multimodal communication. In *Proceedings of the IEEE*, volume 86, pages 837–852, May 1998.

[Chi97] C.C. Chibelushi, J.S.D. Mason, and F. Deravi. Feature-level data fusion for bimodal person recognition. In *Sixth International Conference on Image Processing and Its Applications*, volume 1, pages 399–403, July 1997.

[Chr00] T. Christensen, M. Noergaard, C. Madsen, and A. Hoover. Sensor networked mobile robotics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 782–783, June 2000.

[Das97] B.V. Dasarathy. Sensor fusion potential exploitation - innovative architectures and illustrative applications. In *Proceedings of the IEEE*, volume 85, pages 24–38, January 1997.

[Das00] B.V. Dasarathy. More the merrier... or is it? Sensor suite augmentation benefits assessment. In *Proceedings of the Third International Conference on Information Fusion*, volume 20-25, July 2000.

[Dem67] A.P. Dempster. Upper and Lower Probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, 38:325–339, 1967.

[Dil92] R.A. Dillard. Using data quality measures in decision-making algorithms. *IEEE Exper*, 7(6):63–72, December 1992.

[DW88] H.F. Durrant-Whyte. Sensor Models and Multisensor Integration. *International Journal of Robotics*, 7(6):97–113, December 1988.

[Ebr97] M. Ebrahim, J. Feldman, and I. Bar-Kana. A Robust Sensor Fusion Method for Heart Rate Estimation. *Journal of Clinical Monitoring and Computing*, 13(6):385–393, November 1997.

[Elf89] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, June 1989.

[Elm01] W. Elmenreich and S. Pitzek. The Time-Triggered Sensor Fusion Model. In *Proceedings of the 5th IEEE International Conference on Intelligent Engineering Systems*, pages 297–300, September 2001.

[Elm02] W. Elmenreich. *Sensor Fusion in Time-Triggered Systems*. PhD thesis, Vienna University of Technology, Institut für Technische Informatik, Vienna, Austria, October 2002.

[Elm06] W. Elmenreich and A. Schörgendorfer. Fusion of Continuous-Valued Sensor Measurements using Statistical Analysis. In *International Symposium on Mathematical Methods in Engineering (MME-06)*, Ankara, Turkey, April 2006.

[Fer05] A. Ferrero and S. Salicone. A fully-comprehensive mathematical approach to the expression of uncertainty in measurement. In *International Workshop on Advanced Methods for Uncertainty Estimation in Measurement*, pages 8–13, May 2005.

[Fil03] P. Filzmoser. Multivariate Statistik. Vorlesungsskriptum, Institute of Statistics and Probability Theory, Vienna University of Technology, January 2003.

[Fow79] C.A. Fowler. Comments on the Cost and Performance of Military Systems. *IEEE Transactions on Aerospace and Electronic Systems*, 15(1):2–10, January 1979.

[Fra05] D. Franken and A. Hupper. Improved fast covariance intersection for distributed data fusion. In *8th International Conference on Information Fusion*, volume 1, pages 154–160, July 2005.

[Gan01] Q. Gan and C.J. Harris. Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 37(1):273–279, January 2001.

[Gel74] Arthur Gelb, Editor. *Applied Optimal Estimation.* MIT Press, Cambrigde, Massachusetts, and London, England, 1974.

[Gro98] P. Grossmann. Multisensor Data Fusion. *GEC Journal of Technology*, 15:27–37, 1998.

[Hac01] W. Hackbusch. *Jahrbuch 2001*, chapter Hierarchische Matrizen, pages 549–557. Max Planck Gesellschaft zur Förderung der Wissenschaften e.V., München, November 2001. Available at `http://www.mis.mpg.de/mpi/jb-wh/index.html`.

[Hal92] D.L. Hall. *Mathematical techniques in multi-sensor data fusion.* Artech House, Inc., Norwood, Massachussetts, 1992.

[Hal97] D.L. Hall and J. Llinas. An introduction to multisensor data fusion. In *Proceedings of the IEEE*, volume 85, pages 6 – 23, January 1997.

[Has88] H.R. Hashemipour, S. Roy, and A.J. Laub. Decentralized structures for parallel Kalman filtering. *IEEE Transactions on Automatic Control*, 33(1):88–94, January 1988.

[Hof05] D. Hofmann. In P. Sydenham and R. Thorn, Editors, *Handbook of Measuring System Design*, chapter 48: Common Sources of Errors in Measurement Systems. John Wiley & Sons, Ltd, 2005.

[Int93] International Organization for Standardization (ISO), Geneva, Switzerland. *Guide to the Expression of Uncertainty in Measurement*, $1^{\text{st}}$ edition, 1993.

[Jim99] L.O. Jimenez, A. Morales-Morell, and A. Creus. Classification of hyperdimensional data based on feature and decision fusion approaches using projection pursuit, majority voting, and neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 37(3):1360–1366, May 1999.

[Jul95] S.J. Julier, J.K. Uhlmann, and H.F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference*, volume 3, pages 1628–1632, June 1995.

[Jul97] S.J. Julier and J.K. Uhlmann. A non-divergent Estimation Algorithm in the Presence of Unknown Correlations. In *Proceedings of the American Control Conference*, volume 4, pages 2369–2373, June 1997.

[Kal60] R.E. Kalman. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, (82):35–45, March 1960.

[Kal61] R.E. Kalman and R.S. Bucy. New Results in Linear Filtering and Prediction Theory. *ASME Journal of Basic Engineering*, (83):95–108, March 1961.

[Kay93] S.M. Kay. *Fundamentals of statistical signal processing: Estimation Theory.* Prentice-Hall, Engelwood Cliffs, New Jersey, 1993.

[Kha96] P.P. Khargonekar, M.A. Rotea, and E. Baeyens. Mixed $H_2/H_\infty$ filtering. *International Journal of Robust and Nonlinear Control*, 6(4):313–330, May 1996.

[Kit00] J. Kittler. A Framework for Classifier Fusion: Is It Still Needed?. In F.J. Ferri, J.M. Iñesta Quereda, A. Amin, and P. Pudil, Editors, *SSPR/SPR*, volume 1876 of *Lecture Notes in Computer Science*, pages 45–56. Springer, 2000.

[Kli06] G. Klingler, A. Kößler, and W. Elmenreich. The Smart Car - a distributed controlled autonomous robot. In *Proceedings of the Junior Scientist Conference 2006*, pages 33–34, Vienna, Austria, April 2006.

[Kop97] H. Kopetz. *Real Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academis Publishers, Boston, Dordrecht, London, 1997.

[Kop02] H. Kopetz and R. Obermaisser. Temporal Composability. *Computing & Control Engineering Journal*, 13(4):156–162, August 2002.

[Kop03] H. Kopetz and G. Bauer. The Time-Triggered Architecture. In *Proceedings of the IEEE*, volume 91, pages 112–126, January 2003.

[Köß06] A. Kößler, G. Klingler, Albert Dias, and Jörg Irran. Documentation of the Smart Car Demonstrator. Research Report 45/2002, Technische Universität Wien, Institut für Technische Informatik, Vienna, Austria, May 2006.

[Kou03] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli. Online fault detection of sensor measurements. In *Second IEEE International Conference on Sensors*, volume 2, pages 974–979, October 2003.

[Kri04] B. Krishnamachari. Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks. *IEEE Transactions On Computers*, 53(3):241–250, March 2004.

[Kün05] P. Künzle. Building topological maps for robot navigation using neural networks. Master's Thesis, McGill University, Montreal, Department of Computer Science, February 2005.

[Lab05] R. Labayrade, C. Royere, D. Gruyer, and D. Aubert. Cooperative Fusion for Multi-Obstacles Detection With Use of Stereovision and Laser Scanner. *Autonomous Robots*, 19(2):117–140, September 2005.

[Lew04] T.W. Lewis and D.M.W. Powers. Sensor fusion weighting measures in audio-visual speech recognition. In *Proceedings of the 27th conference on Australasian computer science*, pages 305–314, 2004.

[Lli04] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White. Revisions and extensions to the JDL data fusion model II. In Per Svensson and Johan Schubert, Editors, *Proceedings of the Seventh International Conference on Information Fusion*, volume II, pages 1218–1230, Mountain View, CA, June 2004. International Society of Information Fusion.

[Luo90] R.C. Luo and M.G. Kay. A tutorial on multisensor integration and fusion. In *16th Annual Conference of IEEE Industrial Electronics Society*, volume 1, pages 707–722, November 1990.

[Man98] R.S. Mangoubi. *Robust estimation and failure detection*. Springer, London, 1998.

[Mar90] K. Marzullo. Tolerating Failures of Continuous-Valued Sensors. *ACM Transactions on Computer Systems*, 8(4):284–304, November 1990.

[Mar97] M. Markin, C. Harris, M. Bernhardt, J. Austin, M. Bedworth, P. Greenway, R. Johnston, A. Little, and D. Lowe. Technology foresight on data fusion and data processing. Publication, The Royal Aeronautical Society, 1997.

[Mat98] F. Matía and A. Jiménez. Multisensor Fusion: An Autonomous Mobile Robot. *Journal of Intelligent and Robotic Systems*, 22:129–141, 1998.

[McG76] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264:746–748, 1976.

[Meu95] C. Meunier, J.J. Guerin, M. Lequime, M. Rioual, E. Noel, D. Eguiazabal, D. Fleury, J. Maurin, and R. Mongin. Industrial prototype of a fiber-optic sensor network for the thermal monitoring of the turbogenerator of a nuclear power plant-design, qualification, and settlement. *Journal of Lightwave Technology*, 13(7):1354–1361, July 1995.

[Mov98] J.R. Movellan and P. Mineiro. Robust Sensor Fusion: Analysis and Application to Audio Visual Speech Recognition. *Machine Learning*, 32(2):85 – 100, August 1998.

[Mur96] R.R. Murphy. Biological and Cognitive Foundations of Intelligent Sensor Fusion. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, 26(1):42–51, January 1996.

[Mut98] A. Mutambara. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press LLC, Boca Raton, Florida, 1998.

[Nah80] P.J. Nahin and J.L. Pokoski. NCTR Plus Sensor Fusion Equals IFFN *or* Can Two Plus Two Equal Five? *IEEE Transactions on Aerospace and Electronic Systems*, 16(3):320–337, May 1980.

[Nas93] M. Nashman. The Use of Vision and Touch Sensors for Dimensional Inspection Tasks. *Manufacturing Review*, 6(2):155–162, June 1993.

[Nie02] W. Niehsen. Information fusion based on fast covariance intersection filtering. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 2, pages 901–904, July 2002.

[oIF06] ISIF International Society of Information Fusion. Terminology & Standards, 2006. Available at `http://www.inforfusion.org/terminology.htm`, May 7, 2006.

[R D05] R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005.

[Rao91] B.S. Rao and H.F. Durrant-Whyte. Fully decentralised algorithm for multisensor Kalman filtering. In *IEE Proceedings Control Theory and Applications*, volume 138, pages 413–420, September 1991.

[Rao97] N.S.V. Rao. Distributed Decision Fusion using empirical estimation. *IEEE Transactions On Aerospace And Electronic Systems*, 33(4):1106–1114, October 1997.

[Rao01] N.S.V. Rao. On Fusers that Perform Better than Best Sensor. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 23, pages 904–909, August 2001.

[Rao04] N.S.V. Rao. A Generic Sensor Fusion Problem: Classification and Function Estimation. In T.Windeatt F.Roli, J.Kitler, Editor, *Multiple Classifier Systems: 5th International Workshop, MCS 2004. Proceedings*, volume 3077 of *Lecture Notes in Computer Science*, pages 16–30. Springer-Verlag GmbH, January 2004.

[Rus02] J. Rushby. Formal Verification of Marzullo's Sensor Fusion Interval. Technical report, Computer Science Laboratory, SRI International, Menlo Park, California, USA, January 2002.

[Sar91] V.V.S. Sarma and S. Raju. Multisensor data fusion and Decision support for airborne target identification. *IEEE Transactions On Systems, Man, And Cybernetics*, 21(5):1224–1230, September/October 1991.

[Sch01] U. Schmid and K. Schossmaier. How to reconcile fault-tolerant interval intersection with the Lipschitz condition. *Distributed Computing*, 14(2):101–111, April 2001.

[Sch05] J. Schmalzel, F. Figueroa, J. Morris, S. Mandayam, and R. Polikar. An architecture for intelligent systems based on smart sensors. *IEEE Transactions on Instrumentation and Measurement*, 54(4):1612–1616, August 2005.

[Sch06] T.B. Schön, A. Eidehall, and F. Gustafsson. Lane departure detection for improved road geometry estimation. to be published in IEEE Intelligent Vehicle Symposium, Tokyo, Japan, available at `http://www.control.isy.liu.se/~fredrik/reports/06SchonEG.pdf`, June 2006.

[Sen04] SensComp, Inc., 36704 Commerce Rd., Livonia, MI 48150. *6500 Series Ranging Modules*, September 2004. Available at `http://www.senscomp.com`.

[Sha76] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ, 1976.

[Sha92] U. Shaked and Y. Theodor. $H_\infty$-optimal estimation: a tutorial. In *Proceedings of the 31st IEEE Conference on Decision and Control*, volume 2, pages 2278–2286, 1992.

[She05] D. Sheperd and S. Kumar. In S.S. Iyengar and R.R. Brooks, Editors, *Distributed Sensor Networks*, chapter Microsensor Applications, pages 11–27. Chapman & Hall/CRC, Boca Raton, Florida, 2005.

[Ste98] A.N. Steinberg, C.L. Bowman, and F.E. White. Revisions to the JDL Data Fusion Model. In *The Joint NATO/IRIS Conference, Quebec*, 1998.

[Ste04] A.N. Steinberg, C.L. Bowman, and F.E. White. Rethinking the JDL Data Fusion Levels, 2004. Available at `http://www.infofusion.buffalo.edu/tutorial.htm`.

[Ste05] P. Stepan, M. Kulich, and L. Preucil. Robust data fusion with occupancy grid. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, 35(1):106–115, Februarz 2005.

[Sub05] A. Subramanya, L. Deng, Z. Liu, and Z. Zhang. Multi-Sensory Speech Processing: Incorporating Automatically Extracted Hidden Dynamic Information. In *IEEE International Conference on Multimedia and Expo*, pages 1074–1077, July 2005.

[Swa05] D.C. Swanson. In S.S. Iyengar and R.R. Brooks, Editors, *Distributed Sensor Networks*, chapter Environmental Effects, pages 201–211. Chapman & Hall/CRC, Boca Raton, Florida, 2005.

[Tai02] Y. Tai, K. Yoshizawa, and H. Okubo. Robust fault detection of smart structure control systems with model uncertainty. In *Proceedings of the 41st SICE Annual Conference*, volume 2, pages 1135–1138, August 2002.

[Ten81] R.R. Tenney and N.R. Sandell. Detection with Distributed Sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 17(4):501–510, July 1981.

[The00] A. Theil, L. Kester, and É. Bossé. On measures of performance to assess sensor fusion effectiveness. In *Proceedings of the Third*

*International Conference on Information Fusion*, volume 2, pages THB/3–7, July 2000.

[Var00] P.K. Varshney. Multisensor Data Fusion. In *Lecture Notes in Computer Science*, volume 1821, pages 1–3. Springer, January 2000.

[Wal90] E. Waltz and J. Llinas. *Multisensor Data Fusion*. Artech House, Inc., Norwood, Massachusetts, 1990.

[Wal95] E. Waltz. The Principles and Practice of Image and Spatial Data Fusion. In *Proceedings of the 8th National Data Fusion Conference*, pages 257–278, Dallas, Texas, March 1995.

[Zad65] L.A. Zadeh. Fuzzy Sets. *Information and Control*, 8:338–353, 1965.

[Zha04] K. Zhang and J. Su. A New Method for Asynchronous Multisensor Information Fusion. In *Advances in Artificial Intelligence: 27th Annual German Conference on AI*, volume 3238, pages 410–423, September 2004.

[Zha05] Y. Zhao, L. Zhang, and P. Li. Texture feature fusion for high resolution satellite image classification. In *International Conference on Computer Graphics, Imaging and Vision: New Trends*, pages 19–23, July 2005.