

Berichte aus der Elektrotechnik

Sebastian Ohl

**Fusion von Umfeld wahrnehmenden Sensoren
in städtischer Umgebung**

Shaker Verlag
Aachen 2014

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Zugl.: Braunschweig, Techn. Univ., Diss., 2014

Copyright Shaker Verlag 2014

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8440-3034-1

ISSN 0945-0718

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: www.shaker.de • E-Mail: info@shaker.de

Kurzfassung:

Fusion von Umfeld wahrnehmenden Sensoren in städtischer Umgebung

Sebastian Ohl

In den letzten Jahren haben sich moderne innovative Fahrerassistenzsysteme (z. B. Night-Vision oder Aktivlenkung) immer mehr zu Alleinstellungsmerkmalen und Kaufargumenten von Automobilen entwickelt (BMW Group München, 2009, Seite VI). Zunächst wurden in den 1990er Jahren Systeme wie ABS, ASR und ESC, die sich auf fahrdynamische Eigenschaften konzentrierten, eingeführt. Inzwischen sind sie als Serienausstattung in vielen Fahrzeugen zu finden und seit November 2011 verpflichtend für neue Fahrzeugtypen in der Europäischen Union (Europäisches Parlament, 2009). Mitte der 1990er Jahre waren die ersten komplexeren Systeme in Fahrzeugen der Oberklasse zu finden. Diese nahmen den Längsverkehr im Fahrzeugumfeld wahr, um ihre Aufgabe erfüllen zu können. Im Jahr 1995 wurde beispielsweise von Mitsubishi im Modell Diamante ein adaptiver Tempomat eingeführt, der mithilfe eines Lasersensors den Abstand und die Geschwindigkeit der vorausfahrenden Fahrzeuge maß und die eigene Geschwindigkeit entsprechend anpasste (Watanabe u. a. (1995) zitiert nach Winner u. a. (2009, Seite 479)). Moderne Fahrerassistenzsysteme wie Notbremsassistent, ACC-Stop&Go oder Einparkassistent bis hin zur vollkommenen Entlastung des Fahrers durch eine vollständig automatische Fahrzeugführung sind ohne Sensoren zur Umfeldwahrnehmung nicht vorstellbar.

In der Vergangenheit wurden Fahrerassistenzsysteme und die notwendigen Sensoren oft als Paket entwickelt. Dies führte dazu, dass bei der Ausrüstung eines Fahrzeugs mit mehreren Systemen keine Synergien zwischen den Sensorsystemen genutzt werden konnten. Denkbar wäre hier beispielsweise, dass ein Notbremsassistent die Kamera einer Fahrstreifenwahrnehmung nutzt, um seine Daten zu verifizieren, ohne dass eine weitere Kamera ins Fahrzeug eingerüstet werden muss. Diese Zentrierung des Informationsflusses, und so die Nutzung eines Sensors durch mehrere Assistenzfunktionen, kann mithilfe eines zentralen Umfeldwahrnehmungssystems geschehen. Es wird mit allen im Fahrzeug verbauten Sensoren verbunden und stellt anschließend allen Assistenzsystemen im Fahrzeug die aufbereiteten und miteinander fusionierten Daten zur Verfügung.

Um eine effektive Entwicklung von Umfeldwahrnehmungssystemen zu gewährleisten, sollten große Teile von vorherigen Entwicklungen einfach wiederverwendet und neue sowie bereits in früheren Projekten genutzte Sensoren ohne großen Aufwand an ein System angebunden werden können. Die Eigenschaften des Systems definieren sich dabei durch die Anforderungen der jeweiligen Assistenzfunktionen.

Im Rahmen dieser Arbeit wurde eine Softwarearchitektur zur Entwicklung von Umfeldwahrnehmungssystemen entwickelt (Ohl u. Maurer, 2011b; Ohl u. a., 2011). Basierend auf einer allgemeinen

Beschreibung der Arbeitsdomäne und einer Kontextdefinition werden Anforderungen an die Softwarearchitektur aufgestellt und anschließend die statische sowie die dynamische Architektursicht behandelt.

Die Softwarearchitektur teilt sich in drei Ebenen auf. Die Sensorebene beschreibt die Dekodierung von sensorspezifischen Protokollen und die Konvertierung in eine einheitliche Containerdatenstruktur. Darauf aufbauend schließt sich die Fusionsebene an. Als Sukzessor der Sensorebene konsumiert sie die Containerdatenstruktur und verarbeitet sie in einer gitter- bzw. objekthypothesenbasierten Sensordatenfusion. Als dritte Ebene wird die Applikationsebene eingeführt. Sie dient als Datensinke für die Fusionsebene und enthält die verarbeitenden Assistenzsysteme.

Aus Effizienzgründen werden innerhalb der Fusionsebene für die gitter- und objekthypothesenbasierte Fusion je getrennte Architekturen aufgestellt. Diese lassen sich jedoch auf die Architektur der Arbeitsdomäne zurückführen. Innerhalb der einzelnen Architekturen wird auf ein Höchstmaß an Unabhängigkeit zwischen den Verarbeitungsstufen sowie auf eindeutig spezifizierte Schnittstellen geachtet. Dies ermöglicht eine Wiederverwendung von Algorithmen und Komponenten in späteren Projekten und verkürzt so die Entwicklungszeit.

An die Beschreibung der Softwarearchitektur schließt sich deren Realisierung an. Sie wurde im Rahmen des Projekts „Stadtpilot“ durchgeführt. Das Projekt „Stadtpilot“ widmet sich dem teilautomatischen Fahren im innerstädtischen Bereich (Nothdurft u. a., 2011). Als Projektziel wird hierbei die automatische Umrundung des Stadtkerns von Braunschweig im öffentlichen Straßenverkehr angestrebt. Um dies zu erreichen, wurde der Versuchsträger mit Radar- und Lasersensoren, einer Trägheitsplattform, einem Datenverarbeitungssystem sowie einer elektronischen Ansteuerung der Fahrzeugaktoren ausgerüstet.

Zur Verarbeitung der Umfeld wahrnehmenden Sensoren des Versuchsträgers wurde sowohl ein gitter- als auch ein objekthypothesenbasiertes Fusionssystem auf Basis der Softwarearchitektur entwickelt. Diese werden je für unterschiedliche Aufgaben eingesetzt. Das objekthypothesenbasierte System teilt sich dabei in eine Hauptfusion sowie eine Vorfusion auf. Die Hauptfusion enthält ein konturschätzendes Kalman Filter (Ohl u. Maurer, 2011a). Dieses ist in der Lage, sein Objekthypothesenmodell den von den Sensoren wahrgenommenen Messungen anzupassen. Auf diese Weise beschreibt das Filter unterschiedliche Objekthypothesen mit der jeweils minimalen Anzahl an Stützpunkten. Um eine verlässliche Bestimmung des Konturtyps zu ermöglichen, wird mithilfe einer Ähnlichkeitsfunktion und eines Dempster-Shafer-Filters eine Stabilisierung der Konturschätzung durchgeführt. Ergänzt wird die Hauptfusion durch eine Fusion zur Stabilitätsanalyse von Objekthypothesen. Diese zweite Fusion hat zum Ziel, Objekthypothesen, die sich nicht modellkonform verhalten, zu klassifizieren und auszusortieren. Darüber hinaus werden fehlende Messdaten der Sensoren (z. B. Geschwindigkeitsvektor) aus einem Modell abgeleitet und so verbesserte Initialwerte für die Hauptfusion bereitgestellt. Zusätzlich werden redundante Erfassungsbereiche der Sensoren ausgenutzt, um Fehldetektionen von Objekten zu reduzieren. Dabei wird ein System von logischen Ausdrücken und Bereichen um den Versuchsträger verwendet, um zu entscheiden, ob und wann eine Objekthypothese von der Vorfusion in die Hauptfusion übergeht.

Die gitterbasierte Fusion enthält ein binäres Bayes-Filter. Die Daten dieser Fusion werden durch drei Verfahren mit unterschiedlichen Zielsetzungen ausgewertet. Das erste Verfahren bestimmt Bereiche, die von den Sensoren nicht vermessen werden können. Diese Daten werden genutzt, um bei Abbiege- manövern Informationen über Verdeckungen zu gewinnen. Das zweite Verfahren nimmt Verengungen im Fahrstreifen aufgrund von abgestellten Fahrzeugen wahr, wie sie im innerstädtischen Bereich oft durch abgestellte Fahrzeuge vorkommen. Auf Grundlage des von Weiss u. a. vorgestellten Algorithmus

lassen sich diese leichten Verengungen detektieren und so eine Verschiebung der Trajektorie veranlassen (Weiss u. a., 2007; Weiss, 2011). Die hier eingesetzte gitterbasierte Fusion ermöglicht vor allem eine akkurate Abbildung von sich nicht bewegenden Objekten. Aus diesem Grund lassen sich mit dem dritten Verfahren aus diesen Daten Objekthypothesen statischer Objekte gewinnen, die wiederum von der objekthypothesenbasierten Fusion verarbeitet werden können. Hierzu wird ein neu entwickelter Algorithmus der Familie der Split&Merge-Algorithmen aus dem Bereich der Robotik eingesetzt. Ihr Hauptanwendungsbereich findet sich in der Auswertung von 1D-Laserscannern. Aufgrund dieser Ausrichtung können Split&Merge-Algorithmen nicht direkt auf Gitterdatenstrukturen angewendet werden. Deshalb wird im Rahmen dieser Arbeit ein 2D-Split&Merge-Algorithmus vorgestellt, der auf der Basis von Gitterdaten Objekthypothesen bildet.

Ergänzt wird die Arbeit durch eine Bewertung der aufgestellten Anforderungen und vorgestellten Algorithmen. Hierbei werden sowohl simulierte Daten, Daten von Messfahrten auf einem Testgelände als auch Aufzeichnungen aus dem öffentlichen Straßenverkehr genutzt.

Stichworte:

- Umfeldwahrnehmung
- Urbanes Umfeld
- Sensordatenfusion
- Erweitertes Kalman-Filter
- Konturtracking
- Softwarearchitektur

Abstract:

Environmental perception using Multi Sensor Data Fusion in Urban Environments

Sebastian Ohl

Recently, the development of modern innovative driver assistance systems, like night-vision or active steering, gained an emerging interest in the automotive industry (BMW Group München, 2009, p. VI). In an effort to improve vehicle dynamics, in the beginning of the 1990s, systems like ABS, ASC, and ESC, are developed. In the meantime, recently developed vehicles are equipped with these systems, which have become a obligatory requirement in the European Union since November 2011 (Europäisches Parlament, 2009). In the mid-1990s, the first advanced driver assistance system could be found in upper class vehicles. These systems perceive the vehicle's environment to fulfill their task. For example, in 1995 Mitsubishi introduced an adaptive cruise control system in its model Diamante. The core idea is to measure the distance to the preceding vehicle by a laser sensor and adapt the vehicle's velocity accordingly (according to Watanabe u. a. (1995) as cited in Winner u. a. (2009, p. 479)). Modern driver assistance systems, e.g. emergency breaking, ACC-Stop&Go, parking assistance, up to a fully automatic driving system are of no practical use without the aid of environmental perception sensors.

In the past, driver assistance systems and their necessary sensors are developed together as a sole integrated packet. As a result, the usage of synergies between sensors is practically not possible. This consequently leads to further complication like the non-ability to employ an emergency breaking system that uses the camera of a lane detection system to verify its targets without the need to install a second camera in the vehicle. Having the information centrally available would render the usage of one sensor by several driver assistance systems possible. Briefly, the idea behind this is to develop a central perception system that is connected to every sensor to make the fused and aligned data available to other driver assistance systems.

To ensure a effective driver assistance system development, the majority of the previously developed systems should be reused. In addition, sensors from previous projects should be integrated into new systems without a great effort. In this context, it is worthwhile to mention that the characteristics of a new system are defined by the requirements of the particular assistance functions.

In this PhD thesis, a software architecture for defining environmental perception systems is described (Ohl u. Maurer, 2011b; Ohl u. a., 2011). Based on the description of the architecture's domain and its context, requirements are defined. Afterward, the static and the dynamic views of the software architecture are discussed.

The software architecture is divided into three layers. The sensor layer describes the decoding of sensor specific protocols and the conversion of sensor data into a common container structure. The next layer is the fusion layer that, as a successor to the sensor layer, consumes the container data

structure. Afterward, this data is processed in a grid-based or in a object hypotheses-based sensor data fusion. As a third layer, the application layer is introduced. This layer works as a data sink for the fusion layer and contains the driver assistance systems.

For efficiency reasons, there are different architectures for the grid-based and the object hypotheses-based sensor data fusions within the fusion layer. Both architectures can be reduced to the domain's architecture. Within the scope of each architecture individually, the processing algorithms and interfaces are defined independently and clearly. As a natural result, this allows the reuse of algorithms and components in other projects and, therefore, reduces the developmental time and cost.

Next, the realization of the above-described architecture is addressed. The architecture is realized within the project "Stadtpilot". The Stadtpilot-project is dedicated to semi-autonomous driving in urban environments (Nothdurft u. a., 2011). The project's goal is to drive automatically on the inner-city ring of the city Braunschweig among public traffic. To accomplish this task, the test vehicle is equipped with radar and laser sensors, an inertial measurement unit, a computer system as well as an electronic vehicle control system.

The processing of the vehicle's environmental perception systems consists of a grid-based as well as an object hypotheses-based sensor data fusion system. These fusion systems are used simultaneously for different tasks. The main object hypotheses-based data fusion contains a contour classifying Kalman-filter (Ohl u. Maurer, 2011a). This filter can adapt its object hypothesis model to the data of the perception sensors. This way, the filter describes different object hypotheses with the minimal necessary contour point count. To achieve a reliable contour classification, a similarity metric and a Dempster-Shafer-filter is used to stabilize the contour classification. The second data fusion detects object hypotheses, which do not comply with a certain object hypothesis model. Furthermore, missing measurement values (e.g. velocity vector) are derived from a physical model to improve the initial tracking of the main fusion. In addition, redundant detection areas of the perception sensors are used to reduce measurement errors and ghost object hypotheses. Thereby, a system of logical expression is used to decide whether and when an object hypothesis moves from the preceding to the main fusion system.

The grid-based data fusion contains a binary Bayes-filter. Three different algorithms with different goals process the grid-based fusion's data. The first algorithm detects areas that can currently not be measured by the sensors. For example, this data can be used during turn maneuvers. The second algorithm sense lane constrictions by parking vehicles, as they occur in inner-city environments very often. Based on the algorithm of Weiss u. a., these constrictions can be detected easily and the test vehicle's trajectory can, therefore, be accordingly adapted (Weiss u. a., 2007; Weiss, 2011). Due to its configuration, the grid-based sensor data fusion of project Stadtpilot represents especially static objects. Therefore, the third algorithm is developed to create object hypotheses of static objects from this data. Afterward, these object hypotheses could be sent to the object hypotheses-based fusion system. The newly developed algorithm is based on the split&merge-algorithm family from the robotics domain. In this domain, the algorithms are mainly used to process 1D-laserscanner data. Consequently, these algorithms cannot be deployed to 2D-grid-based data structures. Therefore, a 2D-split&merge-algorithm is developed. It can process grid-based data structures and create object hypotheses of the represented objects.

In addition to the research work done in this thesis, it is enriched by an assessment of the defined requirements and presented algorithms. To do so, simulated data as well as real measured data from a closed test track and Braunschweig's inner-city ring are used.

Keywords:

- Environmental perception
- Urban environment
- Sensor data fusion
- Extend Kalman-filter
- Contour tracking
- Software architecture