Towards a Common Software/Hardware Methodology for Future Advanced Driver Assistance Systems

The DESERVE Approach

RIVER PUBLISHERS SERIES IN TRANSPORT TECHNOLOGY

Series Editors

HAIM ABRAMOVICH
Technion - Israel Institute of Technology
Israel

THILO BEIN
Fraunhofer LBF
Germany

Indexing: All books published in this series are submitted to Thomson Reuters Book Citation Index (BkCI), CrossRef and to Google Scholar.

The "River Publishers Series in Transport Technology" is a series of comprehensive academic and professional books which focus on theory and applications in the various disciplines within Transport Technology, namely Automotive and Aerospace. The series will serve as a multi-disciplinary resource linking Transport Technology with society. The book series fulfils the rapidly growing worldwide interest in these areas.

Books published in the series include research monographs, edited volumes, handbooks and textbooks. The books provide professionals, researchers, educators, and advanced students in the field with an invaluable insight into the latest research and developments.

Topics covered in the series include, but are by no means restricted to the following:

- Automotive
- Aerodynamics
- Aerospace Engineering
- Aeronautics
- Multifunctional Materials
- Structural Mechanics

For a list of other books in this series, visit www.riverpublishers.com

Towards a Common Software/Hardware Methodology for Future Advanced Driver Assistance Systems

The DESERVE Approach

Editors

Guillermo Payá-Vayá

Leibniz Universität Hannover
Germany

Holger Blume

Leibniz Universität Hannover
Germany

River Publishers

Published, sold and distributed by:
River Publishers
Alsbjergvej 10
9260 Gistrup
Denmark

River Publishers
Lange Geer 44
2611 PW Delft
The Netherlands

Tel.: +45369953197
www.riverpublishers.com

ISBN: 978-87-93519-14-5 (Hardback)
978-87-93519-13-8 (Ebook)

© The Editor(s) (if applicable) and The Author(s) 2017. This book is published open access.

Open Access

This book is distributed under the terms of the Creative Commons Attribution-Non-Commercial 4.0 International License, CC-BY-NC 4.0) (http://creativecommons.org/licenses/by/4.0/), which permits use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, a link is provided to the Creative Commons license and any changes made are indicated. The images or other third party material in this book are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt, or reproduce the material.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper.

Contents

Preface **xiii**

List of Contributors **xvii**

List of Figures **xxi**

List of Tables **xxxiii**

List of Abbreviations **xxxv**

1 The DESERVE Project: Towards Future ADAS Functions **1**
Matti Kutila and Nereo Pallaro
1.1 Project Aim 1
1.2 Project Structure 4
1.3 DESERVE Platform Design 5
1.4 The Project Innovation Summary 5
1.5 Conclusions 6

PART I: ADAS Development Platform

2 The DESERVE Platform: A Flexible Development Framework to Seemlessly Support the ADAS Development Levels **9**
Frank Badstübner, Ralf Ködel, Wilhelm Maurer, Martin Kunert, André Rolfsmeier, Joshué Pérez, Florian Giesemann, Guillermo Payá-Vayá, Holger Blume and Gideon Reade
2.1 Introduction to the DESERVE Platform Concept 9
2.2 The DESERVE Platform – A Flexible Development Framework to Seamlessly Support the ADAS Development Levels 12

2.3 DESERVE Platform Requirements 16
2.3.1 DESERVE Platform Framework 16
2.3.2 Generic DESERVE Platform Requirements (Relevant to all Development Levels) 18
2.3.3 Rapid Prototyping Framework Requirements (Development Level 2) 21
2.3.4 Additional Requirements for Embedded Multicore Platform with FPGA (Development Level 3) 22
2.4 DESERVE Platform Specification and Architecture 23
2.4.1 DESERVE Platform Architecture 23
2.4.1.1 Hardware architecture 25
2.4.1.2 Software architecture 26
2.4.2 DESERVE Platform Interface Definition 30
2.4.2.1 Definition of DESERVE interface architecture 30
2.4.2.2 Existing ADAS interfaces 32
2.4.2.3 Definition of next generation interfaces . . 33
2.5 Safety Standards and Certification Concepts 35
2.5.1 Safety Impact of DESERVE 36
2.5.2 Functional Safety of Road Vehicles (ISO 26262) . . 36
2.5.3 Guidelines Related to ISO 26262 37
2.5.4 Safety and AUTOSAR 38
2.5.5 Safety Mechanisms for DESERVE Platform 39
References 43

3 Driver Modelling **45**
Jens Klimke and Lutz Eckstein
3.1 Introduction 45
3.2 Driver Modelling 48
3.3 Requirements for DESERVE 50
3.4 Generic Structure 52
3.4.1 Model Structure 52
3.4.2 Parameter Structure 56
3.5 Implementation 59
3.6 Applications in DESERVE and Results 61
3.7 Conclusions and Outlook 62
References 63

4 Component Based Middleware for Rapid Development of Multi-Modal Applications 65
Gwenaël Dunand
4.1 Introduction 65
4.2 Using a Middleware 65
4.3 The Multisensor Problem 66
4.3.1 Knowing the Date and Time of Your Data 67
4.3.2 Component-based GUI 68
4.3.3 The Off-the-Shelf Component Library 69
4.3.4 Custom Extensions 71
4.3.5 About Performance 71
4.4 Compatibility with Other Tools 72
4.4.1 dSPACE Prototyping Systems 72
4.4.2 Simulators 73
4.4.3 Other Standards 74
4.5 Conclusion 74
References 75

5 Tuning of ADAS Functions Using Design Space Exploration 77
Abhishek Ravi, Hans Michael Koegeler and Andrea Saroldi
5.1 Introduction 77
5.1.1 Parameter Tuning: An Overview 77
5.1.2 Industrial Tuning Applications: Challenges and Opportunities 78
5.1.3 Model-based Tuning 81
5.1.4 Model-based Validation 83
5.2 Demonstrative Example 84
5.2.1 Function: An Overview 84
5.2.2 Design Variables 85
5.2.3 Key Performance Indicators (KPI) 88
5.2.4 Test Maneuver 89
5.2.5 Test Run Overview 89
5.2.6 Raw Data Plausibility Check 91
5.2.7 Meta Modelling 92
5.2.8 Optimization 95
5.2.9 Verification 97
5.3 Model-based Validation 98

5.4 Conclusions 101
Acknowledgement 101
References 101

PART II: Test Case Functions

6 Deep Learning for Advanced Driver Assistance Systems 105
Florian Giesemann, Guillermo Payá-Vayá, Holger Blume,
Matthias Limmer and Werner R. Ritter
6.1 Introduction 105
6.2 Scene Labeling in Advanced Driver Assistance Systems . . . 106
6.3 Convolutional Neural Networks and Deep Learning 107
6.3.1 Introduction to Neural Networks 108
6.3.2 Supervised Learning 109
6.3.3 Convolutional Neural Networks 112
6.4 CNN for Scene Labeling 115
6.4.1 Exemplary Network for Scene Labeling 116
6.4.2 Evaluation 116
6.5 Hardware Platforms for Scene Labeling 120
6.5.1 Theoretical Performance Requirements 121
6.5.2 CPU-based Platforms 125
6.5.3 GPU-based Platforms 125
6.5.4 FPGA-based Platforms 125
6.6 Summary 127
References 127

7 Real-Time Data Preprocessing for High-Resolution MIMO Radar Sensors 133
Frank Meinl, Eugen Schubert, Martin Kunert
and Holger Blume
7.1 Introduction 133
7.2 Signal Processing for Automotive Radar Sensors 134
7.2.1 FMCW Radar System Architecture 134
7.2.2 Two-Dimensional Spectrum Analysis for Range and Velocity Estimation 138
7.2.3 Thresholding and Target Detection 139
7.2.4 Angle Estimation 143
7.3 Hardware Accelerators for MIMO Radar Systems 145

7.3.1 Basic Structure of a Streaming Hardware Accelerator 145
7.3.2 Pipelined FFT Accelerator 146
7.3.3 Rank-Only OS-CFAR Accelerator 151
7.4 Conclusion 153
References 154

8 Self-Calibration of Wide Baseline Stereo Camera Systems for Automotive Applications 157

Nico Mentzer, Guillermo Payá-Vayá, Holger Blume,
Nora von Egloffstein and Lars Krüger

8.1 Introduction 157
8.1.1 Extraction of Image Features 158
8.1.2 Matching of Image Features 161
8.1.3 Extrinsic Online Self-Calibration 161
8.2 Algorithmic Overview 162
8.2.1 Survey of Image Features Extraction 162
8.2.1.1 Detection of features 162
8.2.1.2 Description of features 167
8.2.1.3 Characteristics of features 169
8.2.2 Feature Matching 172
8.2.3 Survey of Feature-based Self-Calibration 176
8.3 Extraction of Image Features 177
8.3.1 Detection of SIFT-Feature Points 177
8.3.2 Description of SIFT-Image Features 178
8.4 Matching of Image Features 179
8.5 Extrinsic Online Self-Calibration 181
8.6 Application-Specific Algorithmic Parameterization 182
8.6.1 Decreasing Bit Depth of Input Images for Extraction of SIFT-features 182
8.6.2 Threshold-based Feature Matching 186
8.6.3 Parameterization of Matching Methods 188
8.7 Hardware Based SIFT-Feature Extraction 192
8.7.1 Challenges of SIFT-Feature Extraction 193
8.7.2 Existing Systems for Hardware Based SIFT-Feature Extraction 194
8.8 Conclusion 196
References 197

9 Arbitration and Sharing Control Strategies in the Driving Process 201

David González, Joshué Pérez, Vicente Milanés, Fawzi Nashashibi, Marga Sáez Tort and Angel Cuevas

9.1 Introduction 201
9.2 ADAS Functions Available in the Market 202
9.2.1 Longitudinal Control Systems 203
9.2.2 Lateral Control Systems 207
9.2.3 Other Control Systems 209
9.2.4 Control Solution in ADAS 211
9.2.4.1 Perception platform 212
9.2.4.2 Application platform 214
9.2.4.3 Information Warning Intervention (IWI) platform 214
9.3 Survey on Arbitration and Control Solutions in ADAS 215
9.4 Human-Vehicle Interaction 216
9.5 Driver Monitoring 217
9.5.1 Legal and Liability Aspects 219
9.6 Sharing and Arbitration Strategies: DESERVE Approach . . 220
9.7 Conclusions 221
References 222

PART III: Validation and Evaluation

10 The HMI of Preventing Warning Systems: The DESERVE Approach 227

Caterina Calefato, Chiara Ferrarini, Elisa Landini, Roberto Montanari, Fabio Tango, Marga Sáez Tort and Eva M. García Quinteiro

10.1 Introduction 227
10.2 Prevent Imminent Accidents: The Role of Humans, the Role of Technology 228
10.2.1 From Passive to Preventive Safety 228
10.2.2 The Role of Driver Model in ADAS Design 230
10.3 HMI Design Flow: The DESERVE Approach 233
10.3.1 Different Approaches in the HMI of the Preventing Warning Systems: A State of Art in a Glance 233

10.4 HMI Concepts Design 234
10.4.1 Concept 1: Holistic HMI 235
10.4.2 Concept 2: Immersive HMI 238
10.4.3 Concept 3: Smart HMI 239
10.5 Preliminary Testing by Focus Group 240
10.5.1 Participants 241
10.5.2 Results 241
10.5.3 List of the Winning Features and Redesign Recommendations 242
10.6 Users Test at Driving Simulator 243
10.6.1 Participants 244
10.6.2 Procedure 244
10.6.3 Results 244
10.7 Conclusions 246
Acknowledgments 247
References 247

11 Vehicle Hardware-In-the-Loop System for ADAS Virtual Testing 251

Romain Rossi, Clément Galko, Hariharan Narasimman and Xavier Savatier

11.1 Introduction 251
11.2 State of the Art 252
11.3 Proposed System 254
11.4 Hardware Implementation 256
11.4.1 Sensors Stimulation Solutions 256
11.4.2 Software Implementation 258
11.5 Experimental Setup 260
11.6 Results 262
11.7 Conclusion and Future Work 265
Acknowledgment 266
References 267

Index 269

About the Editors 271

Preface

The European research project DESERVE (DEvelopment platform for Safe and Efficient dRiVE, 2012–2015) had the aim of designing and developing a platform tool to cope with the continuously increasing complexity and the simultaneous need to reduce costs for future embedded Advanced Driver Assistance Systems (ADAS). For this purpose, the DESERVE platform profits from cross-domain software reuse, standardization of automotive software component interfaces, and easy but safety-compliant integration of heterogeneous modules. This enables the development of a new generation of ADAS applications, which challengingly combine different functions, sensors, actuators, hardware platforms, and Human Machine Interfaces (HMI).

This book provides a detailed overview of the different research activities conducted in the course of the DESERVE project. After introducing the aims of the DESERVE project in Chapter 1, selected achievements of the DESERVE project are presented in three different parts. Part I is dedicated to the ADAS development platform developed during the DESERVE project.

- Chapter 2 covers the methodology and concepts that are part of the generic DESERVE platform as the basis and key enabler for the development of new assistance systems. It describes the entire spectrum of aspects, e.g., modularity, interfaces, and standards, to be considered for the use of the DESERVE platform.
- Chapter 3 describes the development of realistic models for driver behavior as part of the DESERVE tool-chain needed for the evaluation of complex ADAS systems and driver-vehicle-environment interactions. The modelling system was used to simulate two different driving scenarios.
- Chapter 4 presents component based middleware, e.g., RTMaps and ADTF, for supporting the developer of complex systems with typical challenges like multi-sensor support, synchronization issues, and modularity. By means of different exemplary applications, in which modules like simulators or prototyping systems are connected to the middleware, the flexibility of the DESERVE tool-chain is demonstrated.

- Chapter 5 describes a model-in-the-loop approach for tuning ADAS parameters. Using the AVL CAMEO tool, model-based design space exploration and validation of a complex ADAS function is performed.

In Part II, ADAS applications used as test functions in the DESERVE project are explained.

- Chapter 6 presents an application of deep-learning techniques for semantic segmentation of camera images (i.e., Scene Labeling). After explaining the algorithmic basics, an FPGA-based implementation is presented and evaluated.
- Chapter 7 covers a system coupling an FPGA-based signal processing architecture for MIMO radar with a PC-based ADTF data post-processing. The hardware-software combination maximizes processing performance and minimizes development time of complex systems.
- Chapter 8 describes a design space exploration for online calibration of wide baseline stereo camera systems using sparse feature correspondences in stereo images. Challenges in hardware implementations of feature matching are presented and hardware-specific solutions are discussed.
- Chapter 9 presents a first approach of arbitration and sharing vehicle control between driver and assistance system based on modelling vehicles and driver behavior and intentions. Fuzzy logic techniques are used to implement the control sharing and simulations allow testing of the systems.

Part III covers the validation and evaluation of two exemplary applications of the DESERVE platform.

- Chapter 10 aims at exploring effective design of Human Machine Interface (HMI). During the DESERVE project, in-vehicle HMI solutions for different functions were developed. The HMI design process for an exemplary function is described in this chapter.
- Chapter 11 shows a prototype system for vehicle-in-the-loop testing of ADAS functions that additionally analyzes the energy efficiency of the prototyped system. Combined with multi-sensor simulation, a virtual environment for testing ADAS functions is provided.

Further detailed information about the contributions of DESERVE can be found in the list of project deliverables referenced in each chapter.

This work was supported by the European Commission under the Artemis Joint Undertaking in the scope of the DESERVE project. We would like to

thank all authors and co-authors for their excellent contributions. Special thanks to Matti Kutila for the efficient managing of the complete DESERVE project over three years. Further thanks to Martin Kunert who well-coordinated subprojects and who actively supported our work. Furthermore we want to thank the River Publishers Team, in particularly Mr. Mark de Jongh and Ms. Junko Nakajima for their great support.

We hope that you will enjoy reading this book.

Guillermo Payá Vayá
Holger Blume

March 22th, 2017
Hannover (Germany)

List of Contributors

Abhishek Ravi, *AVL List Gmbh, Austria*

André Rolfsmeier, *dSpace GmbH, Germany*

Andrea Saroldi, *C.R.F. S.C.p.A , Italy*

Angel Cuevas, *CTAG – Centro Tecnológico de Automoción de Galicia, Spain*

Caterina Calefato, *Unimore – University of Modena and Reggio Emilia – Italy*

Chiara Ferrarini, *Unimore – University of Modena and Reggio Emilia – Italy*

Clément Galko, *Univ. Rouen, UNIROUEN, ESIGELEC, IRSEEM 76000 Rouen, France*

David González, *INRIA, France*

Elisa Landini, *RE:Lab srl, Italy*

Eugen Schubert, *Advanced Engineering Sensor Systems, Robert Bosch GmbH, Leonberg, Germany*

Eva M. García Quinteiro, *CTAG – Centro Tecnológico de Automoción de Galicia, Spain*

Fabio Tango, *CRF – Centro Ricerche Fiat, Italy*

Fawzi Nashashibi, *INRIA, France*

Florian Giesemann, *Institute of Microelectronic Systems, Leibniz Universität Hannover, Hannover, Germany*

Frank Badstübner, *Infineon Technologies AG, Germany*

Frank Meinl, *Advanced Engineering Sensor Systems, Robert Bosch GmbH, Leonberg, Germany*

Gideon Reade, *ASL, U.K.*

Guillermo Payá-Vayá, *Institute of Microelectronic Systems, Leibniz Universität Hannover, Hannover, Germany*

Gwenaël Dunand, *Intempora, France*

Hans Michael Koegeler, *AVL List Gmbh, Austria*

Hariharan Narasimman, *Univ. Rouen, UNIROUEN, ESIGELEC, IRSEEM 76000 Rouen, France*

Holger Blume, *Institute of Microelectronic Systems, Leibniz Universität Hannover, Hannover, Germany*

Jens Klimke, *Institute for Automotive Engineering, RWTH Aachen University, Steinbachstraße 7, 52074 Aachen, Germany*

Joshué Pérez, *INRIA, France*

Lars Krüger, *Daimler AG, Vision Enhancement, Ulm, Germany*

Lutz Eckstein, *Institute for Automotive Engineering, RWTH Aachen University, Steinbachstraße 7, 52074 Aachen, Germany*

Marga Sáez Tort, *CTAG – Centro Tecnológico de Automoción de Galicia, Spain*

Martin Kunert, *Advanced Engineering Sensor Systems, Robert Bosch GmbH, Leonberg, Germany*

Matthias Limmer, *Vision Enhancement, Daimler AG, Germany*

Matti Kutila, *VTT Technical Research Center of Finland Ltd., Finland*

Nereo Pallaro, *Centro Ricerche Fiat, Italy*

Nico Mentzer, *Institute of Microelectronic Systems, Leibniz Universität Hannover, Hannover, Germany*

Nora von Egloffstein, *Daimler AG, Vision Enhancement, Ulm, Germany*

Ralf Ködel, *Infineon Technologies AG, Germany*

Roberto Montanari, *RE:Lab srl, Italy*

Romain Rossi, *Univ. Rouen, UNIROUEN, ESIGELEC, IRSEEM 76000 Rouen, France*

Vicente Milanés, *INRIA, France*

Werner R. Ritter, *Vision Enhancement, Daimler AG, Germany*

Wilhelm Maurer, *Infineon Technologies AG, Germany*

Xavier Savatier, *Univ. Rouen, UNIROUEN, ESIGELEC, IRSEEM 76000 Rouen, France*

List of Figures

Figure 1.1 The DESERVE V-shape development process. . . . 4
Figure 1.2 DESERVE platform concept for speeding up the ADAS function development time. 5
Figure 2.1 The DESERVE Platform – the enabler for next generation ADAS systems. 10
Figure 2.2 DESERVE platform enabled design and development process. 12
Figure 2.3 ADAS development process. 13
Figure 2.4 DESERVE platform framework. 17
Figure 2.5 Perception platform functional architecture. 19
Figure 2.6 Application platform functional architecture. 21
Figure 2.7 DESERVE IWI platform. 22
Figure 2.8 DESERVE platform (e.g. for development Level 2 – rapid prototyping system based on mixed PC and embedded controller framework). 25
Figure 2.9 DESERVE approach – use of common platform for all ADAS modules. 27
Figure 2.10 DESERVE platform architecture. 28
Figure 2.11 Overview on the principles of virtual interaction using the AUTOSAR. 29
Figure 2.12 Message box principle for intra-unit communication. 31
Figure 2.13 AUTOSAR application software concept. 32
Figure 2.14 Camera Interface (CIF) overview. 34
Figure 2.15 Module interaction implies changes in system behavior. 36
Figure 2.16 SEooC safety mechanisms. 40
Figure 2.17 Top level safety requirements. 41
Figure 2.18 Fault tolerant time interval (FTTI) definition. . . . 41
Figure 2.19 Generic elements of safe computation hardware platform. 42

Figure 3.1 Primary driving tasks which are implemented in the driver model within the DESERVE project separated by longitudinal and lateral control. 52
Figure 3.2 Manoeuvres which are implemented in the driver model within the DESERVE project. 52
Figure 3.3 Driver model structure in the context of environment and vehicle: the structure includes perception, processing and action blocks including its functional modules and the regarded dynamic information flow. 53
Figure 3.4 Process variables for the four basic driving motivations *free moving*, *following*, *lane keeping* and *standing*. 55
Figure 3.5 Process variables for the three manoeuvres *lane change*, *stopping* and *Safe Passing*. 56
Figure 3.6 Sketch of the parameter blocks (brown) and model blocks (blue) of the driver model. 57
Figure 3.7 Distribution of lower following time gaps for real drivers (blue bars) and the modelled distribution dependent on a normal distributed *need for safety* parameter (red line). 58
Figure 3.8 Stateflow model for a two-phase lane change including decision (A), progress control (B) and sequence control (C). 60
Figure 3.9 Trajectories (velocity over x- and y-position) for left turn including the simulation results for different parameter sets. The real driver data is measured on one intersection with 136 different drivers during day time. 61
Figure 4.1 ADAS function requires many different type of sensor. 66
Figure 4.2 Synchronisation issues. 67
Figure 4.3 The RTMaps Studio. 68
Figure 4.4 Components and interfaces. 69
Figure 4.5 Inspecting data with the data viewer. 70
Figure 4.6 Developing a new component. 71
Figure 4.7 dSPACE MicroAutobox and RTMaps Bridge. . . . 72
Figure 4.8 ProSivic working together with RTMaps. 73

Figure 5.1 Separation of software and tuning parameters in a control unit. 78
Figure 5.2 History of powertrain tuning (calibration). 78
Figure 5.3 Illustration of a generalized development environment and manual tuning process. 80
Figure 5.4 Model-based tuning task illustrated. 81
Figure 5.5 Velocity profiles for a sample test run using the control function. 84
Figure 5.6 Function developed using IPG carmaker and MATLAB simulink. 85
Figure 5.7 Function overview. 86
Figure 5.8 Illustration of the kinematic variables A_MAX and J_MAX. 86
Figure 5.9 Illustration of the design variable (variation) J_HOR. 88
Figure 5.10 Key performance indicators. 89
Figure 5.11 IPG Carmaker test environment. 90
Figure 5.12 Test run overview illustrating the work flow. 90
Figure 5.13 Left image illustrates the test preparation window while the right image illustrates the test run window. 91
Figure 5.14 Checking for outliers in the measured variables. . . 92
Figure 5.15 Check of DoE design and the boundaries of variation parameters. 93
Figure 5.16 Figure depicting the quality of empirical modeling. 93
Figure 5.17 Intersection plot highlighting the influence of each variation on the output variables and their interaction. 94
Figure 5.18 Optimization setting window in AVL CAMEO. . . 95
Figure 5.19 Trade-off plot between comfort and speed. 96
Figure 5.20 Sporty mode vs comfort mode. 97
Figure 5.21 Verification plot to see how well the measured results from the verification run fit the model results. 98
Figure 5.22 Digitized road used for the validation run. 99
Figure 5.23 Measurements comparison when run on comfort mode (in blue) and sporty mode (in red). 100
Figure 6.1 Model of an artificial neuron. 108

Figure 6.2 Exemplary activation functions used in neural networks. 109

Figure 6.3 Example of a fragmentation after a 2 × 2 pooling. The naïve approach would only produce the bright pixels, while an overlapping pooling produces all other possible pixels (purple, green, and blue). These pixels must be reordered to be able to correctly continue with the forward propagation of the neural network. 114

Figure 6.4 The complete processing chain from input image to a scene labeled image is displayed. After building an image pyramid of 3 layers and the local normalization every scale is fed to its own processing chain. This produces 6 class membership probability maps. They can be interpreted and augmented as seen in the output image. 116

Figure 6.5 The image pyramid construction layer produces 3 scales that are locally normalized in 15 × 15 windows. Every scale is propagated independently. There are in total 2 convolution layers with 16 × 7 × 7 filter kernels using the ReLU activation function. After activation a 2 × 2 max-pooling is performed followed by a fragmentation in the first pooling layer. A second fragmentation is not necessary since the second pooling layer is followed by a defragmentation. The small scaled feature maps are sampled up and fed to a classification layer, being a 6 × 1 × 1 convolution layer. Finally, a pixel wise softmax is applied. 117

Figure 6.6 Displayed are the learn curves of three different network topologies. Each topology was trained three times and the learn curves were averaged. The averaged learn curves are displayed as solid lines while the standard deviation for 50 epochs is displayed as the area around the lines. 118

Figure 7.1 FMCW ramp waveform shown as frequency over time f(t). The solid line represents the transmitted signal (TX) while the dashed line is the received signal (RX). 135

Figure 7.2 Chirp-sequence modulation. 136
Figure 7.3 Possible MIMO antenna array design: The physical receiver array (blue) is extended by several virtual antennas (red squares) due to the second transmitter TX 2. 137
Figure 7.4 CA-CFAR sliding window implementation. 139
Figure 7.5 Rank-only OS-CFAR implementation. 141
Figure 7.6 Additive white Gaussian noise model. 143
Figure 7.7 Histogram of a noise measurement showing the chi-squared distribution before and after NCI. . 143
Figure 7.8 Uniform linear antenna array with spacing d and resulting steering vector $\boldsymbol{v}(\alpha)$. 144
Figure 7.9 Architecture of a streaming hardware accelerator. . 145
Figure 7.10 Radix-2 FFT implementation based on a multi-path delay commutator (MDC) pipeline. 146
Figure 7.11 Radix-2 FFT implementation based on a SDF pipeline. 147
Figure 7.12 Effects of different word lengths on the amount of quantization noise. 150
Figure 7.13 Architecture of the rank-only OS-CFAR accelerator. 152
Figure 7.14 Resource usage against number of channels for a constant window size (128 cells). 152
Figure 7.15 Resource usage against window size for different number of channels. 153
Figure 8.1 Algorithmic overview. Input of the processing chain is a stereo image pair, in which sparse pixel correspondences are extracted for online camera calibration. After the calibration, rectification is performed as a preprocessing step for disparity estimation. 158
Figure 8.2 Left (*top*) and right (*bottom*) image from a stereo camera system showing detected SIFT-image features. Detected feature points of the left/right image are displayed in red/green, matches are displayed in blue. Scale and rotation of the SIFT-features are illustrated by the circle properties. 159

Figure 8.3 Detection of edges and corners by image gradients. The blue circle shows a possible feature point, surrounded by a local neighborhood. (a) Low image gradients in two spatial directions represent texture free image areas. (b) A high image gradient in one spatial direction indicates a possible edge, (c) in two spatial directions a possible corner. . . . 163

Figure 8.4 Intensity comparisons of pixel, which are located on a Bresenham Circle. The central pixel is determined as a corner if a certain number of continuous pixel intensities is brighter or darker than the central pixel. This is combined with an adoptable threshold to avoid instabilities. 164

Figure 8.5 Detection of corners of different image scales. With strongly different object sizes in the image, a corresponding corner is not detectable (red circle), but by a repeated image scaling. 164

Figure 8.6 Blob detector. The detected blobs are displayed as red circles. The blob's size is displayed as the diameter of the circle. 165

Figure 8.7 Blob detection based on circular image region for a scene with a large viewpoint change. The region on which the blob feature extraction is based only partially covers the corresponding region and thus, will lead to non-matching image features. 165

Figure 8.8 Affine-Invariant Interest Point Detection. The circular point neighborhood is replaced with an ellipse in order to achieve independent orthogonal varying detection scales for interest point detection. Before applying a detection algorithm, the local neighborhood is affine normalized, which results in a circular neighborhood and a transformed image patch. 166

Figure 8.9 Sampling grids for generating different descriptors: (a) SIFT, (b) Shape Context, (c) DAISY. 167

Figure 8.10 Sampling pattern. (a) BRISK descriptor, (b) FREAK descriptor. Sampling patterns define a set of sampling locations (blue circles), of whose image information is smoothed with spatial-dependent filter kernels (red circles). Out of the sampling pattern the sampling pairs for the binary tests for the descriptor generation are selected. 168

Figure 8.11 Two variations of sampling pairs of the FREAK descriptor. A fixed combination of sampling locations is selected as descriptor specific sampling pairs, with which the binary tests for the descriptor generation is performed. 169

Figure 8.12 Rotation invariance is achieved by rotating the sampling grid by the main orientation before extracting the descriptor. 170

Figure 8.13 Scale-space. An input image is down sampled to achieve multiple scales of the image. On each scale, feature candidates are found, whereas repeated candidates are removed. The scale with the highest information content for the feature candidate is selected as the feature scale. 170

Figure 8.14 Multi-scale approach for blob detection. The same blob with differing scales in two images and the related response (normalized Laplacian of Gaussian) over scales is shown. The scale with the highest information content is chosen as a blob. 171

Figure 8.15 Image pyramid. The scale-space is constructed by different octaves, which consists of multiple intervals. Each interval indicates a specific variant of the used Gaussian kernel. In order to approximate the Laplace scale-space, the Difference of Gaussian is determined. 177

Figure 8.16 Generation of feature descriptor. The local neighborhood is subdivided into independent subregions, which are combined into individual histograms. After a weighting and smoothing, the feature descriptor is generated by concatenating the single histograms to as a resulting feature vector. 178

Figure 8.17 Extracted SIFT-features with exemplary geometry-based restriction of matching candidates. By restricting possible matching candidates geometrically, the problem size is significantly reduced. 179

Figure 8.18 Exemplary results of feature matching. The left and right stereo images are overlaid; features of the left/right image are displayed in red/green. Correct matches are depicted in yellow; false matches are shown in blue. The upper image shows the results of the initial brute force matching, whereas the lower image shows the results of the enhanced matching process. 180

Figure 8.19 Verification of match positions with disparity maps. For rectified images, the horizontal difference of feature positions of a corresponding pixel pair equals the related value of the disparity map. With this technique, it is possible to validate resulting matching lists for datasets with ground truth disparity maps. 183

Figure 8.20 Comparison of the resulting SIFT-features of the left input image for 12 bpp images and 8 bpp images. In the 12 bpp input image, an overall number of 1,069 features have been detected, whereas in the 8 bpp input image 1,056 features have been determined. A subset of 1,045 features (97.8%) is identical in both images (blue). There are 14 (1.3%) exclusive 8 bpp feature positions (red) detected and 24 (2.2%) exclusive 12 bpp feature positions (orange). 185

Figure 8.21 Comparison of the resulting pixel correspondences for the 8 bpp and 12 bpp input images. In the 12 bpp input image, an overall number of 611 pixel pairs has been detected, whereas in the 8 bpp input image 608 correspondences have been determined. A subset of 587 pairs (96.1%)

is identical in both images (blue lines). Furthermore, there are 23 (3.8%) exclusive 8 bpp pairs (red lines) and 24 (3.9%) exclusive 12 bpp pixel correspondences (orange lines). 185

Figure 8.22 Histogram of random generated SIFT-descriptor distances of an idealized NNB feature matching. The right distribution with mean μ_2 displays the distances of wrong matches, whereas the left distribution with mean μ_1 illustrates the correct matches. 186

Figure 8.23 Histogram of descriptor distances for a NNB SIFT-feature matching with the extracted threshold according to Otsu. Distances of correct/wrong matches are displayed in blue/orange. The complete distribution is shown in purple. 187

Figure 8.24 Histograms of descriptor distances for different NNB feature matching case studies with the extracted threshold according to Otsu. Distances of correct/wrong matches are displayed in blue/orange. The complete distribution is shown in purple. Due to different descriptors and resulting matching distances, various axis scales for clear presentation are used. 189

Figure 8.25 Exemplary histogram for the distribution of matching candidates for the geometry-based feature matching (see Table 8.4). The average number of candidates is 7 candidates per matching event. 190

Figure 8.26 Rates of disparity verified pixel correspondences for different offsets ε and three matching methods. For all methods, the rate of correct matches runs into saturation. The NNB matching method performs best over all offsets ε. (TB: Threshold-Based Matching; NNB: Nearest-Neighbor-Based Matching; NNDR: Nearest-Neighbor Distance Ratio Matching). 192

Figure 8.27 Break down of SIFT-feature extraction into four algorithmic steps and relating qualitatively quota of control complexity and complexity (i.e., regular arithmetic). 193
Figure 9.1 ACC Systems. 204
Figure 9.2 Stages on the longitudinal control of the vehicle. . . 205
Figure 9.3 CSW system. 206
Figure 9.4 TSR system. 206
Figure 9.5 LDW system. 208
Figure 9.6 BSD/LCA system. 208
Figure 9.7 Top view of a parking assistance system. 210
Figure 9.8 Aided park system. 210
Figure 9.9 Automatic park systems. 211
Figure 9.10 DESERVE platform. 212
Figure 9.11 DESERVE platform framework. 213
Figure 9.12 SAE J3016 standards of driving automation levels for on-road vehicles. 215
Figure 9.13 Arbitration and control sharing application: General diagram. 220
Figure 10.1 Total number of fatalities in road traffic accidents in Europe. 229
Figure 10.2 Holistic HMI concept, that shows: IPC display 12”; SW commands; left stalk commands; buttons; knobs. 236
Figure 10.3 Holistic HMI layout. 236
Figure 10.4 Holistic HMI layout with the user menu in the central area. 237
Figure 10.5 Holistic HMI layout with the lane change assist in the central area. 237
Figure 10.6 Holistic HMI layout with the rear view camera in the central area. 237
Figure 10.7 Holistic HMI layout with the night vision system in the central area. 238
Figure 10.8 (A-B-C-D) Holistic HMI left area with: lane departure warning, collision warning, Rear approaching vehicle system, pedestrian safety system. 238

Figure 10.9 Immersive HMI concept shows: 3,5” IPC display; touch display 8,5” in the dashboard; head-up display for the windscreen; SW commands; left stalk commands; buttons; knobs. 239
Figure 10.10 Immersive HMI concept: instrument panel cluster display. 239
Figure 10.11 Immersive HMI concept: dashboard display. 240
Figure 10.12 Immersive HMI concept: head-up display details. 240
Figure 10.13 Smart HMI concept. 241
Figure 10.14 Smart HMI concept: Nomadic device with night vision system. 241
Figure 10.15 Radar chart summarizing HMI evaluation for the 6 HMI concepts. Bis concepts are concept 1, 2, 3 with implicit drowsiness. 243
Figure 10.16 Proposed change to create the final DESERVE HMI concept. 245
Figure 10.17 Final DESERVE HMI concept: warning area. . . . 245
Figure 10.18 Final DESERVE HMI concept: rear view camera. 245
Figure 10.19 Final DESERVE HMI concept: navigation. 246
Figure 11.1 Overview of the SERBER VeHIL system. 254
Figure 11.2 Block diagram of the SERBER system. 258
Figure 11.3 Sample video output of Pro-Sivic. 259
Figure 11.4 RTMAPS diagram of the system (extract). 260
Figure 11.5 Mobileye 560 aftermarket vision-based ADAS. . . 261
Figure 11.6 RTMAPS diagram of the V2V task. 261
Figure 11.7 The Biocar test vehicle on the Horiba chassis dynamometer. 262
Figure 11.8 Overview of the urban environment in Pro-Sivic. 263
Figure 11.9 Inner view of the vehicle. 264
Figure 11.10 Lane departure warning triggered. 264
Figure 11.11 V2V Communication HMI. 265

List of Tables

Table 1.1 Scientific and technical objectives 2

Table 5.1 Range of variation parameters used in the tuning task 88

Table 5.2 Variations values for comfort and sporty mode . . . 96

Table 6.1 The confusion matrix of topology 3-2-32 and the respective FNR, FPR and IU for each class. The classes are background (Bg), road (Rd), vehicle (Veh), sky, vulnerable road users (VRU) and infrastructure (Inf). Each cell shows the percentage (from all pixels in the dataset) of actual class (row) predicted as class (column) . . 118

Table 6.2 Displayed are the measures Accuracy (ACC), mean Intersection over Union (mIU), Matthews Correlation Coefficient (MCC) and mean False Negative Rate (mFNR) for 3 topologies 120

Table 6.3 Input image sizes for three different scales in the exemplary convolutional neural network . . . 123

Table 6.4 Number of operations for the exemplary convolutional neural network 123

Table 6.5 Comparison of different implementations of convolutional neural networks on different platforms 124

Table 7.1 Resource usage of different pipelined FFT implementations 147

Table 8.1 Overview of feature detectors 172

Table 8.2 Overview of feature descriptors 172

Table 8.3 Numbers of extracted SIFT-features and detected matches for 8 bpp input images and 12 bpp images. The number of the geometry-based (GB) nearest-neighbor distance ratio matches (NNDR) drops significantly but ensures a high explicitness of matches. The algorithmic parameters of the

SIFT-feature extraction of the two test cases are adjusted in order to extract a similar number of features, which lead to an identical number of verified matches 184

Table 8.4 Results for a SIFT-feature matching for a global matching and a geometry-based feature matching. The window size for the geometry-based feature matching is $+/-4$ pixel in y-direction and $+100/-4$ pixel in x-direction 190

Table 8.5 Results of disparity verified feature correspondences for different combinations of global and spatial restriction matching methods. In addition to a high rate of correct matches, a minimal number of pixel correspondences has to be given for a reliable subsequent image processing. The total numbers of detected matches for selected algorithmic combinations are given in brackets. The number of correct matches and wrong matches do not result in 100% because of missing values in the ground truth disparity maps. Those values are skipped for evaluation 191

Table 8.6 Overview of existing systems for SIFT-feature extraction 194

List of Abbreviations

ABS	Anti-lock Breaking System
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
ADC	Analog-to-digital converter
AEB	Automatic/Autonomous Emergency Braking
AR	Autoregressive
ASIC	Application-Specific Integrated Circuit
ASIP	Application-Specific Instruction-Set Processor
avg	Average
BASt	German Federal Highway Research Institute
bpp	Bit per pixel
BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
BSD	Blind Spot Detection
CA-CFAR	Cell-averaging constant false alarm rate
CAN Bus	Controller Area Network
CDMA	Code division multiple access
CenSurE	Center Surround Extremas
CFAR	Constant false alarm rate
CM4SL	Carmaker for simulink
CMbB	Collision Mitigation by Braking
CMOS	Complementary Metal-Oxide-Semiconductor
CNN	Convolutional Neural Network
COR	Customized Output Range
CPU	Central Processing Unit
CRF	Conditional Random Field
CUT	Cell under test
DAISY	Name of a feature descriptor
DAS	Driver assistance systems
DBC	data base CAN
DIF	Decimation-in-frequency

DMA	driving monitoring automotive
DOA	Direction of arrival
DoE	Design of Experiment
DoG	Difference of Gaussian
DRAM	Dynamic random-access memory
ECU	Electronic Control Unit
ESC	Electronic Stability Control
ESPRIT	Estimation of signal parameters via rotational invariant techniques
FAST	Features from Accelerated Segment Test
FCW	Frontal Collision Warning or Forward Collision Warning
FDM	Frequency-division multiplexing
FFT	Fast Fourier transform
FIR	Finite impulse response
FMCW	Frequency-modulated continuous-wave
FN(R)	False Negative (Rate)
FP(R)	False Positive (Rate)
FPGA	Field-Programmable Gate Array
fps	Frames per second
FREAK	Fast Retina Keypoint
GB	Geometry-based
GOPS	Billion Operations Per Second
GPGPU	General Purpose Graphics Processing Unit
GPP	General Purpose Processor
GPU	Graphics Processing Unit
HD	High-definition, 1280×720 pixel
HiL	Hardware in the Loop
HMI	Human-machine interface
HW	Hardware
I/O	input/output
I2C	Inter-Integrated Circuit
IMU	Inertial measurement unit
IU	Intersection over Union
IWI	information-warning-intervention
KD-Tree	K-dimensional tree
KPI	Key Performance Indicator
LCA	Lane Change Assistant
LDW	Lane Departure Warning

LKA	Lane Keeping Assistance
LoG	Laplacian of Gaussian
LSB	Least significant bit
LUT	Lookup table
MCC	Matthews Correlation Coefficient
MDC	Multi-path delay commutator
MiL	Model in the Loop
MIMO	Multiple-input multiple-output
MLP	Multi Layer Perceptron
MOPS	Million Operations Per Second
MUSIC	Multiple signal classification
NCI	Non-coherent integration
NHTSA	National Highway Traffic Safety Administration
NMEA	National Marine Electronics Association
NNB	Nearest-Neighbor-Based
NNDR	Nearest-Neighbor Distance Ratio
OpenCL	Open Computing Language
OpenGL	Open Graphics Library
ORB	Oriented FAST and Rotated BRIEF
OS-CFAR	Ordered-statistic constant false alarm rate
PCA	Principal Component Analysis
PID	proportional, integral, derivative controller
QVGA	Quarter Video Graphics Array, 320×240 pixel
RCS	Radar cross-section
RDE	Reak Driving Emissions
ReLU	Rectifier Linear Unit
RMS	Root Mean Square
RPM	Revolution per minute
RTSP	Real Time Streaming Protocol
SAE	Society of Automotive Engineers
SDF	Single-path delay feedback
SIFT	Scale-Invariant Feature Transform
SIP	Session Initialization Protocol
SLA	Speed Limit Assistant
SNR	Signal-to-noise ratio
SoP	Start of Production
SQNR	Signal-to-quantization-noise ratio
SRAM	Static random-access memory

SURF	Speeded Up Robust Features
SW	Software
TB	Threshold-Based
TDM	Time-division multiplexing
TP	True Positive
UUT	Unit Under Test
VGA	Video Graphics Array, 640×480 pixel