



# ReConFig'08

2008 International Conference on  
Reconfigurable Computing and FPGAs

3-5 December 2008, Cancun, Mexico

*Organized by*



Laboratoire  
d'Informatique  
de Robotique  
et de Microélectronique  
de Montpellier



*Sponsored by*



*Sponsors*



*Edited by*  
Cesar Torres  
Lionel Torres  
René Cumpulido

*ReConFig 2008 Information*

*Papers by Session*

*Trademarks*

*Getting Started*

*Papers by Author*

*Search*

Proceedings

**2008 International Conference on Reconfigurable  
Computing and FPGAs**

**ReConFig 2008**

**3-5 December 2008  
Cancun, Mexico**



**Los Alamitos, California  
Washington • Tokyo**



Proceedings

**2008 International Conference on Reconfigurable  
Computing and FPGAs**

**ReConFig 2008**

Copyright © 2008 by The Institute of Electrical and Electronics Engineers, Inc.  
All rights reserved.

*Copyright and Reprint Permissions:* Abstracting is permitted with credit to the source. Libraries may photocopy beyond the limits of US copyright law, for private use of patrons, those articles in this volume that carry a code at the bottom of the first page, provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

Other copying, reprint, or republication requests should be addressed to: IEEE Copyrights Manager, IEEE Service Center, 445 Hoes Lane, P.O. Box 133, Piscataway, NJ 08855-1331.

*The papers in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change. Their inclusion in this publication does not necessarily constitute endorsement by the editors, the IEEE Computer Society, or the Institute of Electrical and Electronics Engineers, Inc.*

IEEE Computer Society Order Number P3474  
ISBN-13: 978-0-7695-3474-9  
BMS Part # CFP08389-CDR  
Library of Congress Number 2008908297

*Additional copies may be ordered from:*

IEEE Computer Society  
Customer Service Center  
10662 Los Vaqueros Circle  
P.O. Box 3014  
Los Alamitos, CA 90720-1314  
Tel: + 1 800 272 6657  
Fax: + 1 714 821 4641  
<http://computer.org/cspress>  
[csbooks@computer.org](mailto:csbooks@computer.org)

IEEE Service Center  
445 Hoes Lane  
P.O. Box 1331  
Piscataway, NJ 08855-1331  
Tel: + 1 732 981 0060  
Fax: + 1 732 981 9667  
[http://shop.ieee.org/store/  
customer-service@ieee.org](http://shop.ieee.org/store/customer-service@ieee.org)

IEEE Computer Society  
Asia/Pacific Office  
Watanabe Bldg., 1-4-2  
Minami-Aoyama  
Minato-ku, Tokyo 107-0062  
JAPAN  
Tel: + 81 3 3408 3118  
Fax: + 81 3 3408 3553  
[tokyo.ofc@computer.org](mailto:tokyo.ofc@computer.org)

*Individual paper REPRINTS may be ordered at: <[reprints@computer.org](mailto:reprints@computer.org)>*

Editorial production by Silvia Ceballos  
Label art production by Mark Bartosik



***IEEE Computer Society  
Conference Publishing Services (CPS)***

<http://www.computer.org/cps>

# 2008 International Conference on Reconfigurable Computing and FPGAs

## ReConFig 2008 Table of Contents

|                           |      |
|---------------------------|------|
| Preface.....              | xii  |
| Organizing Committee..... | xiii |
| Program Committee.....    | xiv  |
| Reviewers.....            | xvi  |

---

### General Session 1 - Applications 1

|   |    |
|---|----|
| Reconfigurable PDA for the Visually Impaired Using FPGAs .....                      | 1  |
| <i>Xuan Zhang, Cesar Ortega-Sanchez, and Iain Murray</i>                            |    |
| Embedded Harmonic Control for Trajectory Planning in Large Environments .....       | 7  |
| <i>Cesar Torres-Huitzil, Bernard Girau, Amine Boumaza, and Bruno Scherrer</i>       |    |
| Flexible Architecture for Three Classes of Optical Flow Extraction Algorithms ..... | 13 |
| <i>José Hugo Barrón-Zambrano, Cesar Torres-Huitzil, and Mauricio Cerda</i>          |    |

### General Session 2 - Processors 1

|  |    |
|--|----|
| Automatic Synthesis of Multiprocessor Systems from Parallel Programs<br>under Preemptive Scheduling .....          | 19 |
| <i>Harold Ishebabi, Philipp Mahr, and Christophe Bobda</i>   |    |
| Design and Implementation of a Resource-Efficient Communication Architecture<br>for Multiprocessors on FPGAs ..... | 25 |
| <i>Xiaofang Wang and Swetha Thota</i>  |    |
| Automatic Instruction-Set Extensions with the Linear Complexity Spiral Search .....                                | 31 |
| <i>Carlo Galuzzi, Dimitris Theodoropoulos, Roel Meeuws, and Koen Bertels</i>                                       |    |

### General Session 3 - Applications 2

|  |    |
|--|----|
| A Real-Time Embedded System for Stereo Vision Preprocessing Using an FPGA .....                            | 37 |
| <i>Anders Kjar-Nielsen, Lars Baunegaard With Jensen, Anders Stengaard Sorensen,<br/>and Norbert Krüger</i> |    |

|   |     |
|---|-----|
| Finite Precision Analysis of the 3GPP Standard Turbo Decoder for Fixed-Point Implementation in FPGA Devices .....       | 43  |
| <i>Anabel Morales-Cortés, R. Parra-Michel, Luis F. González-Pérez, and Gabriela Cervantes T.</i>                        |     |
| Parallel Processor for 3D Recovery from Optical Flow .....  | 49  |
| <i>José Hugo Barrón-Zambrano, Fernando Martin del Campo-Ramírez, and Miguel Arias-Estrada</i>                           |     |
| A Reconfigurable Platform for Frequent Pattern Mining .....   | 55  |
| <i>Song Sun, Michael Steffen, and Joseph Zambreno</i>   |     |
| <b>General Session 4 - Reconfigurability</b>  |     |
| A Framework for the Exploration of RTOS Dedicated to the Management of Hardware Reconfigurable Resources .....          | 61  |
| <i>J.C. Prevotet, A. Benkhelifa, B. Granado, E. Huck, B. Miramond, F. Verdier, D. Chillet, and S. Pillement</i>         |     |
| Optimizing Partial Reconfiguration of Multi-context Architectures .....   | 67  |
| <i>Sven Eisenhardt, Tobias Oppold, Thomas Schweizer, and Wolfgang Rosenstiel</i>  |     |
| Automatic Construction of Large-Scale Regular Expression Matching Engines on FPGA .....                                 | 73  |
| <i>Yi-Hua E. Yang and Viktor K. Prasanna</i>  |     |
| A Hardware Task-Graph Scheduler for Reconfigurable Multi-tasking Systems .....  | 79  |
| <i>Juan Antonio Clemente, Carlos González, Javier Resano, and Daniel Mozos</i>  |     |
| <b>General Session 5 - Processors 2</b>   |     |
| Optimized Architectural Synthesis of Fixed-Point Datapaths .....  | 85  |
| <i>Gabriel Caffarena, Juan A. López, Gerardo Leyva, Carlos Carreras, and Octavio Nieto-Taladriz</i>                     |     |
| Developing an MMX Extension for the MicroBlaze Soft Processor .....   | 91  |
| <i>Manuel Hernández Calvino, Sergio Rubén. Geninatti, and José Ignacio Benavides Benítez</i>                            |     |
| <b>General Session 6 - Methodologies 1</b>  |     |
| An ILP Formulation for the Task Graph Scheduling Problem Tailored to Bi-dimensional Reconfigurable Architectures .....  | 97  |
| <i>F. Redaelli, M.D. Santambrogio, and S. Ogrenci Memik</i>   |     |
| Architectural Model and Resource Estimation for Distributed Hardware Implementation of Discrete Signal Transforms ..... | 103 |
| <i>Rafael A. Arce-Nazario, Manuel Jiménez, and Domingo Rodríguez</i>  |     |
| A Reconfiguration-Aware Floorplacer for FPGAs .....   | 109 |
| <i>A. Montone, F. Redaelli, M.D. Santambrogio, and S. Ogrenci Memik</i>   |     |
| The Effect of LUT and Cluster Size on a Tree Based FPGA Architecture .....  | 115 |
| <i>Umer Farooq, Zied Marrakchi, Hayder Mrabet, and Habib Mehrez</i>   |     |

## General Session 7 - Methodologies 2

|   |     |
|---|-----|
| Enhanced Methodology and Tools for Exploring Domain-Specific Coarse-Grained FPGAs .....                                   | 121 |
| <i>Husain Parvez, Zied Marrakchi, and Habib Mehrez</i>  |     |
| Integrating Logic Analyzer Functionality into VHDL Designs .....  | 127 |
| <i>G. Knittel, S. Mayer, and C. Rothlaender</i>   |     |
| Loop Transformations to Reduce the Dynamic FPGA Reconfiguration Overhead .....  | 133 |
| <i>Tom Degryse, Karel Bruneel, Harald Devos, and Dirk Stroobandt</i>  |     |
| Leveraging Firmware in Multichip Systems to Maximize FPGA Resources: An Application of Self-Partial Reconfiguration ..... | 139 |
| <i>Juan Galindo, Eric Peskin, Brad Larson, and Gene Royslance</i>   |     |
| Finite Domain Constraints Based Delay Aware Placement Tool for FPOAs .....  | 145 |
| <i>Rohit Saraswat and Brandon Eames</i>   |     |

## General Session 8 - Processors 3 and Applications 3

|  |     |
|--|-----|
| Arithmetic Operations and Their Energy Consumption in the Nios II Embedded Processor ..... | 151 |
| <i>David M. Cambre, Eduardo Boemo, and Elías Todorovich</i>                                |     |
| Operating System for Symmetric Multiprocessors on FPGA .....                               | 157 |
| <i>Pablo Huerta, Javier Castillo, Carlos Sánchez, and Jose Ignacio Martínez</i>            |     |
| Dynamically Reconfigurable Split Cache Architecture .....                                  | 163 |
| <i>Luiza M.N. Coutinho, Jose L.D. Mendes, and Carlos A.P.S. Martins</i>                    |     |
| Configurable-System-on-Programmable-Chip for Power Electronics Control Applications .....  | 169 |
| <i>Armando Astarloa, Unai Bidarte, Jesús Lázaro, Jon Andreu, and Jose Luis Martín</i>      |     |
| Parallel Backprojector for Cone-Beam Computer Tomography .....                             | 175 |
| <i>Nikolay Sorokin</i>   |     |

## General Session 9 - Systems on Chip

|  |     |
|--|-----|
| Key Research Issues for Reconfigurable Network-on-Chip .....   | 181 |
| <i>R. Dafali, J.-Ph. Diguët, and M. Sevaux</i>   |     |
| SoC-MPI: A Flexible Message Passing Library for Multiprocessor Systems-on-Chips .....                          | 187 |
| <i>Philipp Mahr, Christian Lörchner, Harold Ishebabi, and Christophe Bobda</i>                                 |     |
| Design Space Exploration and Performance Analysis for the Modular Design of CVS in a Heterogeneous MPSoC ..... | 193 |
| <i>Z. Jian Jia, Tomás Bautista, Antonio Núñez, Cayetano Guerra, and Mario Hernández</i>                        |     |

## General Session 10 - Reconfiguration and Methods

|   |     |
|---|-----|
| Part-E - A Tool for Reconfigurable System Design .....                | 199 |
| <i>Elmar Weber, Florian Dittmann, and Norma Montealegre</i>           |     |
| VIS2SOUND on Reconfigurable Hardware .....                            | 205 |
| <i>C. Morillas, J.P. Cobos, F.J. Pelayo, A. Prieto, and S. Romero</i> |     |

|   |     |
|---|-----|
| A Fast Emulation-Based NoC Prototyping Framework .....  | 211 |
| <i>Yana E. Krasteva, Francisco Criado, Eduardo de la Torre, and Teresa Riesgo</i>   |     |
| Power-Efficient High Throughput Reconfigurable Datapath Design for Portable<br>Multimedia Devices .....                                 | 217 |
| <i>Sohan Purohit, Sai Rahul Chalamalasetti, Martin Margala, and Pasquale Corsonello</i>   |     |
| <b>Track on Bioinspired Reconfigurable Computing Systems and Self-Adaptive<br/>Computing 1</b>  |     |
| Reconfigurability-Aware Structural Mapping for LUT-Based FPGAs .....  | 223 |
| <i>Karel Bruneel and Dirk Stroobandt</i>  |     |
| Fast Implementation of a Bio-inspired Model for Decentralized Gathering .....   | 229 |
| <i>Bernard Girau and Cesar Torres-Huitzel</i>   |     |
| Game-Theoretic Approach for Temperature-Aware Frequency Assignment with Task<br>Synchronization on MP-SoC .....                         | 235 |
| <i>Diego Puschini, Fabien Clermidy, Pascal Benoit, Gilles Sassatelli, and Lionel Torres</i>   |     |
| <b>Track on Bioinspired Reconfigurable Computing Systems and Self-Adaptive<br/>Computing 2</b>  |     |
| A Hybrid FPGA/Coarse Parallel Processing Architecture for Multi-modal Visual<br>Feature Descriptors .....                               | 241 |
| <i>Lars Baunegaard With Jensen, Anders Kjar-Nielsen, Javier Diaz Alonso, Eduardo Ros,<br/>and Norbert Krüger</i>                        |     |
| Design and Implementation of Adaptive Viterbi Decoder for Using A Dynamic<br>Reconfigurable Processor .....                             | 247 |
| <i>Yuken Kishimoto, Shinichiro Haruyama, and Hideharu Amano</i>   |     |
| Dynamic Self-Rescheduling of Tasks over a Heterogeneous Platform .....  | 253 |
| <i>Alécio P.D. Binotto, Edison P. Freitas, Marcelo Götz, Carlos E. Pereira, André Stork,<br/>and Tony Larsson</i>                       |     |
| <b>Track on High Performance Reconfigurable Computing 1</b>   |     |
| Sequence Alignment with Traceback on Reconfigurable Hardware .....  | 259 |
| <i>Scott Lloyd and Quinn O. Snell</i>   |     |
| A Message-Passing Hardware/Software Co-simulation Environment to Aid<br>in Reconfigurable Computing Design Using TMD-MPI .....          | 265 |
| <i>Manuel Saldana, Emanuel Ramalho, and Paul Chow</i>   |     |
| FPGA Implementation of Pseudo Random Number Generators for Monte Carlo<br>Methods in Quantitative Finance .....                         | 271 |
| <i>Simon Banks, Philip Beadling, and Andras Ferencz</i>   |     |
| A Pthreads-Based MPI-1 Implementation for MMU-Less Machines .....   | 277 |
| <i>Juan A. Rico-Gallego, Jesus M. Alvarez-Llorente, Francisco J. Perogil-Duque,<br/>Pedro P. Antúnez-Gómez, and Juan C. Díaz-Martín</i> |     |
| A Hardware Filesystem Implementation for High-Speed Secondary Storage .....   | 283 |
| <i>Ashwin A. Mendon and Ron Sass</i>  |     |

## Track on High Performance Reconfigurable Computing 2

|   |     |
|---|-----|
| A Real-Time FPGA Based Platform for Applications in Mechatronics .....  | 289 |
| <i>Roque A. Osornio-Rios, Rene de J. Romero-Troncoso, Luis Morales-Velazquez,<br/>J. Jesus de Santiago-Perez, Rooney de J. Rivera-Guillen,<br/>and J. de Jesus Rangel-Magdaleno</i> |     |
| Disparity Map Hardware Accelerator .....  | 295 |
| <i>Humberto Calderón, Jesús Ortiz, and Jean-Guy Fontaine</i>  |     |
| A Novel Rekeying Message Authentication Procedure Based on Winternitz OTS<br>and Reconfigurable Hardware Architectures .....  | 301 |
| <i>Abdulhadi Shoufan, Sorin A. Huss, Oliver Kelm, and Sebastian Schipp</i>  |     |
| A Temporal Partitioning Methodology for Reconfigurable High Performance<br>Computers .....  | 307 |
| <i>Paulo S. Brandao do Nascimento, Victor W.C. de Medeiros, Viviane L.S. Souza,<br/>Abner C. Barros, and Manoel Eusebio de Lima</i>   |     |
| Power Consumption Reduction Explorations in Processors by Enhancing Performance<br>Using Small ESL Reprogrammable eFPGAs .....  | 313 |
| <i>Syed Zahid Ahmed, Julien Eydoux, Michael Fernandez, Laurent Rougé,<br/>Gilles Sassatelli, and Lionel Torres</i>  |     |

## Track on Reconfigurable Computing for DSP and Communications 1

|   |     |
|---|-----|
| Reconfigurable Cell Architecture for Systolic and Pipelined Computing Datapaths ..... | 319 |
| <i>Abdulrahman Hanoun, Friedrich Mayer-Lindenberg, and Bassel Soudan</i>              |     |
| Implementations and Optimizations of Pipeline FFTs on Xilinx FPGAs .....              | 325 |
| <i>Bin Zhou and David Hwang</i>   |     |
| Generalised Parallel Bilinear Interpolation Architecture for Vision Systems .....     | 331 |
| <i>Suhaib A. Fahmy</i>  |     |

## Track on Reconfigurable Computing for DSP and Communications 2

|  |     |
|--|-----|
| An FFT/IFFT Design versus Altera and Xilinx Cores .....  | 337 |
| <i>C. Gonzalez-Concejero, V. Rodellar, A. Alvarez-Marquina, E. Martinez de Icaya,<br/>and P. Gomez-Vilda</i> |     |
| Using a CSP Based Programming Model for Reconfigurable Processor Arrays .....                                | 343 |
| <i>Zain-ul-Abdin and Bertil Svensson</i>   |     |
| A Novel FPGA Implementation of a Wideband Sonar System for Target Motion<br>Estimation .....                 | 349 |
| <i>Sheng Cheng, Chien-Hsun Tseng, and Marina Cole</i>  |     |
| Hybrid Architecture for Data-Dependent Superimposed Training in Digital Receivers .....                      | 355 |
| <i>Fernando Martín del Campo, René Cumplido, Roberto Perez-Andrade,<br/>and A.G. Orozco-Lugo</i>             |     |

### Track on Reconfigurable Computing for DSP and Communications 3

|   |     |
|---|-----|
| A Comparison of Approaches for High-Level Power Estimation of LUT-Based DSP Components .....                  | 361 |
| <i>Ruzica Jevtic, Carlos Carreras, and Domenik Helms</i>  |     |
| FPGA Implementation of a Modulated Complex Lapped Transform for Watermarking Systems .....                    | 367 |
| <i>Jose Juan Garcia-Hernandez, Claudia Feregrino-Uribe, and Rene Cumplido</i>                                 |     |
| Universal Wavelet Kernel Implementation Using Reconfigurable Hardware .....                                   | 373 |
| <i>Christophe Desmouliers, Erdal Oruklu, and Jafar Saniie</i>   |     |
| Design and Implementation of a Multi-standard Interleaver for 802.11a, 802.11n, 802.16e & DVB Standards ..... | 379 |
| <i>Carlos R. Sánchez-Ortiz, R. Parra-Michel, and M.E. Guzmán-Rentería</i>                                     |     |

### Track on Reconfigurable Computing for Security and Cryptography 1

|   |     |
|---|-----|
| Analysis and Enhancement of Random Number Generator in FPGA Based on Oscillator Rings ..... | 385 |
| <i>Knut Wold and Chik How Tan</i>   |     |
| Parametric, Secure and Compact Implementation of RSA on FPGA .....                          | 391 |
| <i>Ersin Öksüzoglu and Erkay Savaş</i>  |     |
| FPGA Implementation of an Elliptic Curve Cryptosystem over $GF(3^m)$ .....                  | 397 |
| <i>Ilker Yavuz, Siddika Berna Örs Yalçın, and Çetin Kaya Koç</i>                            |     |

### Track on Reconfigurable Computing for Security and Cryptography 2

|  |     |
|--|-----|
| Enhanced Correlation Power Analysis Using Key Screening Technique .....  | 403 |
| <i>Toshihiro Katashita, Akashi Satoh, Takeshi Sugawara, Naofumi Homma, and Takafumi Aoki</i>                       |     |
| Enhancing an Embedded Processor Core with a Cryptographic Unit for Speed and Security .....                        | 409 |
| <i>Övünç Kocabaş, Erkay Savaş, and Johann Großschädl</i>   |     |
| Triple Rail Logic Robustness against DPA .....   | 415 |
| <i>Victor Lomné, Thomas Ordas, Philippe Maurine, Lionel Torres, Michel Robert, Rafael Soares, and Ney Calazans</i> |     |
| FPGA Implementation and Performance Evaluation of AES-CCM Cores for Wireless Networks .....                        | 421 |
| <i>Ignacio Algreto-Badillo, Claudia Feregrino-Uribe, René Cumplido, and Miguel Morales-Sandoval</i>                |     |
| High Performance Implementation of a Public Key Block Cipher - MQQ, for FPGA Platforms .....                       | 427 |
| <i>Mohamed El-Hadedy, Danilo Gligoroski, and Svein J. Knapskog</i>   |     |
| Power Consumption Estimations vs Measurements for FPGA-Based Security Cores .....                                  | 433 |
| <i>Dimitrios Meintanis and Ioannis Papaefstathiou</i>  |     |

## Track on Reconfigurable Computing for Security and Cryptography 3

|   |     |
|---|-----|
| Celator: A Multi-algorithm Cryptographic Co-processor .....   | 438 |
| <i>Daniele Fronte, Annie Perez, and Eric Payrat</i>   |     |
| A Reversible Data Hiding Algorithm for Radiological Medical Images and Its<br>Hardware Implementation ..... | 444 |
| <i>Z. Jezabel Guzmán Zavaleta, Claudia Feregrino Uribe, and René Cumplido</i>                               |     |
| Forward-Secure Content Distribution to Reconfigurable Hardware .....  | 450 |
| <i>David Champagne, Reouven Elbaz, Catherine Gebotys, Lionel Torres, and Ruby B. Lee</i>                    |     |
| <b>Author Index</b> .....   | 456 |

# ReConFig08 Program committee

## General Sessions

|                      |                       |                       |
|----------------------|-----------------------|-----------------------|
| Alastair Smith       | Guy Lemieux           | Neil Bergman          |
| Andres García        | Heitor Lopes          | Oliver Diessel        |
| Andres Upegui        | João Cardoso          | Oswaldo Cadenas       |
| Bernard Girau        | José Ignacio Martinez | Paris Kitsos          |
| Carl Ebeling         | Juergen Becker        | Pascal Benoit         |
| Carlos Llanos        | Koji Nakano           | Peter Athanas         |
| Carlos Valderrama    | Lionel Torres         | Rene Cumplido         |
| Cesar Torres         | Marco Domenico        | René Romero-Troncoso  |
| Christophe Bobda     | Santambrogio          | Reouven Elbaz         |
| Claudia Feregrino    | Marco Platzner        | Roger Woods           |
| Debatosh Debnath     | Marek Gorgon          | Scott Hauck           |
| Dominique Lavenier   | Martin Langhammer     | Sergio Lopez-Buedo    |
| Eduardo Boemo        | Masahiro Iida         | Shawki Areibi         |
| Eduardo Marques      | Masanari Hariyama     | Suhaib Fahmy          |
| Fernando Gehm Moraes | Mazen Saghir          | Tulika Mitra          |
| Fernando Pardo       | Michael Huebner       | Volodymyr Kindratenko |
| Gustavo Sutter       | Miguel Arias          | William Marnane       |
| Guy Gogniat          | Miriam Leeser         | Yahir Hernandez       |

## High Performance Reconfigurable Computing

|                      |                         |                        |
|----------------------|-------------------------|------------------------|
| Aravind Dasu         | Jean-Pierre Schoellkopf | Matthias Fouquet-Lapar |
| Dan Poznanovic       | Leandro Indrusiak       | Maya Gokhale           |
| Daniel Denning       | Loic Lagadec            | Melissa Smith          |
| Duncan Buell         | Luigi Carro             | Nicolas Navet          |
| Eric Stahlberg       | Luis Gomes              | Norbert Wehn           |
| Gilles Sassatelli    | Manuel Moreno           | Volodymyr Kindratenko  |
| Herman Lam           | Martin Herbordt         | Yann Thoma             |
| Jean-Philippe Diguët | Martin Langhammer       |                        |

## Reconfigurable Computing for Security and Cryptography

|                   |                        |                     |
|-------------------|------------------------|---------------------|
| Akashi Satoh      | Jean-Pierre Seifert    | Odyseas Koufopavlou |
| Axel Poschmann    | Joan Daemen            | Paris Kitsos        |
| Catherine Gebotys | Johannes Wolkerstorfer | Pasin Israsena      |
| David Champagne   | Jorge Guajardo         | Patrick Schaumont   |
| Elisabeth Oswald  | Kris Gaj               | Ramesh Karri        |
| Emanuel Popovici  | Lejla Batina           | Reouven Elbaz       |
| Francois-Xavier   | Máire McLoone          | Saar Drimer         |
| Standaert         | Mohammed Benaissa      | Sylvain Guilley     |
| Guy Gogniat       | Nachiketh Potlapally   | Viktor Fischer      |

## **Reconfigurable Computing for DSP and Communications**

Alastair Smith  
Brad Quinton  
Christophe Moy  
Christos-Savvas Bouganis  
Gabriel Caffarena  
Irwin Kennedy  
Jim Hwang  
Juanjo Noguera  
Laurence Turner  
Leonel Sousa  
Martin Herbordt  
Ricardo Reis  
Suhaib Fahmy

## **Bioinspired Reconfigurable Computing Systems and Self-Adaptive Computing**

Andres Upegui  
Arnaud Tisserand  
Benjamin Schrauwen  
Bernard Girau  
Camel tanougast  
Cesar Torres  
Eduardo Sanchez  
Erika Cota  
Fabien Clermidy  
Fernanda Kastensmidt  
Fernando Gehm Moraes  
Gilles Sassatelli  
Guy Gogniat  
Hartmut Schmeck  
Hélène Paugam-Moisy  
Hugo Barron  
Hugues Berry  
J. Manuel Moreno  
Arostegui  
Jean-Philippe Diguët  
Lionel Torres  
Marco Nuno-Maganda  
Mauricio Cerda  
Michael Hübner  
Michiel DHaene  
Mounir Benabdenbi  
Ney Calazans  
Nicolas Ventroux  
Olivier Heron  
Pascal Benoit

# ReConFig08 Additional Reviewers

Aaron Wood  
Aerssens Matthieu  
Ahmed Elhossini  
Ali Akbar Zarezadeh  
Ali Al Ghouwayel  
Antony Savich  
Benjamin Ylvisaker  
Brian Van Essen  
Carlos Erig Lima  
Christian Plessl  
Christian Schuck  
Damien Picard  
Diana Goehringer  
Enno Lubbers  
Farhad Mehdipour  
Fernando Matin Del  
Campo  
Francesco Bruschi  
Frederico Pratas  
Gerardo Leyva  
Heiner Giefers  
Hidetomo Shibamura  
Hongzhi Wang  
Javier Gonzalez-Bayon  
Javier Hormigo  
José Arnaldo de  
Holanda  
Julien Delorme  
Junji Kitamichi  
Kazutoshi Kobayashi  
Kentaro Sano  
Kevin Cheng  
Loig Godard  
Martin Herbordt  
Masaru Fukushi  
Mathieu Giraud  
Matthias Alles  
Michael Haselman  
Mingda Huang  
Minoru Watanabe  
Morihiro Kuga  
Motoki Amagasaki  
Nele Mentens

Nozomu Togawa  
Nuno Sebastiao  
Philipp Mahr  
Ricardo Chessini Bose  
Ricardo Menotti  
Romain Vaslin  
Rosemary Francis  
Samuel Antao  
Sandeep S. Kumar  
Scott Chin  
Stephen Friedman  
Thomas Inseher  
Tiago Dias  
Timo Lehnigk  
Timo Stripf  
Toshiaki Miyazaki  
Tsutomu Maruyama  
Vanderlei Bonato  
Vincenzo Rana  
Yasushi Inoguchi  
Yohei Hori  
Yuichiro Shibata  
Zoltan Rakosi

# Preface

This book presents the proceedings of the 2008 International Conference on Reconfigurable Computing and FPGAs (ReConFig08) held in Cancun, Mexico, in December 3-5, 2008. This conference seeks to promote the use of reconfigurable computing and FPGA technology for research, education, and applications, covering from hardware architectures and devices to custom computers and high performance systems. ReConFig08 was organized by the National Institute for Astrophysics, Optics and Electronics (INAOE) of Mexico in collaboration with the Laboratory of Informatics, Robotics and Microelectronics from Montpellier (LIRMM), France.

For ReConFig08, submissions were invited for general sessions as well as the following special tracks on reconfigurable computing applications: High Performance Reconfigurable Computing, Reconfigurable Computing for Security and Cryptography, Reconfigurable Computing for DSP and Communications, Bioinspired Reconfigurable Computing Systems and Self-Adaptive Computing.

Overall, including general sessions and special tracks, 125 submissions were received from 32 countries: Argentina, Australia, Belgium, Bolivia, Brazil, Canada, Colombia, Cuba, Denmark, Egypt, France, Germany, Greece, India, Ireland, Israel, Italy, Japan, Mexico, Netherlands, New Zealand, Norway, Pakistan, Puerto Rico, Russia, Spain, Sweden, Turkey, United Arab Emirates, United Kingdom, United States, and Uruguay. After a peer-review process, 76 papers were accepted for inclusion in these proceedings.

On behalf of the program and organizing committees, it is our pleasure to express our deep gratitude to everyone who contributed in any way to the success of the conference and the conclusion into these proceedings. We would like to thank all the members of the program committee and reviewers for their invaluable time and effort in the review process and to provide constructive feedback to authors. We are very grateful to the members of the organizing committee whose hard work made the vision of the ReConFig08 a reality. Also we acknowledge the cooperation provided by the IEEE Computer Society to produce these proceedings.

Last but not least, we thank all authors who contributed to this conference for sharing their novel ideas, methods and results with the research community in the growing and developing field of Reconfigurable Computing and FPGA technology. We hope that you enjoy the conference, the reading of these proceedings and find them as a source of inspiration for further developments.

**Cesar Torres**  
**Lionel Torres**  
**René Cumplido**

# A Hybrid FPGA/Coarse Parallel Processing Architecture for Multi-modal Visual Feature Descriptors

Lars Baunegaard With Jensen<sup>†</sup>, Anders Kjær-Nielsen<sup>†</sup>, Javier Díaz Alonso<sup>‡</sup>, Eduardo Ros<sup>‡</sup>  
and Norbert Krüger<sup>†</sup>

<sup>†</sup> The Mærsk Mc-Kinney Møller Institute,  
University of Southern Denmark,  
Campusvej 55, DK-5230 Odense, Denmark  
Email: {lbwj, akn, norbert}@mmmi.sdu.dk

<sup>‡</sup> Dept. Arquitectura y Tecnología de Computadores  
E.T.S.I. Informática, C/ Periodista Daniel Saucedo, s/n  
E-18071 Granada, Spain  
Email: {jdiaz, eduardo}@atc.ugr.es

## Abstract

*This paper describes the hybrid architecture developed for speeding up the processing of so-called multi-modal visual primitives which are sparse image descriptors extracted along contours. In the system, the first stages of visual processing are implemented on FPGAs due to their highly parallel nature whereas the higher stages are implemented in a coarse parallel way on a multicore PC. A significant increase in processing speed could be achieved (factor 11.5) as well as in terms of latency (factor 3.3). These factors can be further increased by optimizing the processes implemented on the multicore PC.*

## 1. Introduction

Different levels of visual processing contain a different amount of complexity and regularity. It is known that at early stages of visual processing (in the following called 'early vision') in humans, local feature extraction processes are performed at each retinal location along the sensory processing path [7, 8]. Consequently also in many artificial vision systems such regular processes take place at the first levels (see, e.g., [14]). Such processes can be easily programmed and computed efficiently on FPGAs or other highly parallel platforms such as GPUs.

In contrast to such early operations, it is known that higher level visual processes (in the following called 'early cognitive vision', see also [17]) are much less regular and

are based on discrete entities (often even of symbolic nature) that even vary in their shape and parameterization depending on the local signal (see, e.g., [15]). Moreover, while processes on the level of 'early vision' can be represented in general as matrices (close to the original image structure) on higher levels 'lists' are usually the more appropriate structure. Hence, such processes can not be as easily implemented on highly parallel computational devices.

Moreover, at higher levels of visual processing important information is coded as second- and higher order relations of such discrete entities, leading to an exponential increase in size of a highly structured but also largely inhomogeneous state space. Hence condensation of the data is required [12] that makes reasoning operations in this relational space feasible. Furthermore these processes are difficult to implement on parallel hardware and are in general preferred to be programmed on more sequential architectures due to their inherent complexity.

In this work, we present a system in which real-time processing on these two different levels becomes realised by making use of a hybrid computer architecture containing highly parallel hardware (FPGAs) for the computation of information at early vision stages as well as coarse parallel computation for higher levels (see figure 1).

In our system, stereo information is processed. As a front end, our system makes use of an additional 'small' FPGA that performs basic image operations such as cutting regions of interest as well as undistortion and rectification that can be seen as simple extensions of the camera signal [10] (representing a smart image acquisition system with embedded

processing on the FPGA device). Our vision system is a general purpose system that is currently used in two European projects addressing robotics [1] and driver assistance [6] (for some publications based on our representations, see e.g., [11, 17]).

The paper is structured as follows: In section 2, an overview of the the system architecture is given. In section 3, the preprocessing and early vision processing on FPGAs is described. In section 4, the higher levels of our visual representation are briefly described as well as our real-time implementation on a 4-core PC. In section 5, we give measurements of computation time of the subcomponents and compare it to a purely sequential processing. Section 6 concludes and suggests future work.

## 2. System architecture

The system architecture realises different stages of visual processing on different hardware components. Camera associated functions are performed on a small FPGA in a tight loop with the camera. Linear and non-linear filtering processes (mainly ordinary convolutions and pixel-wise normalisations) compute pixel-wise information and are realised on a larger FPGA. This pixel-wise information then becomes condensed into local descriptors (called primitives) that represent discrete and sparse positions in the image in a list structure that is then compared in the stereo matching phase. Due to the inherent complexity this is done on a (coarse parallel) PC.

In more technical terms, the hybrid system consists of two Pike F-421C cameras, two FPGAs and two PCs as sketched in figure 1. On the first stage — preprocessing — the cameras are connected to an FPGA using IEEE 1394a. This FPGA handles undistortion and rectification of the images (see, e.g., [5]) in addition to controlling the camera, e.g. synchronized triggering. At the next stage — early vision — the undistorted and rectified images are passed on to another FPGA using a direct custom connection. This FPGA is part of a PCI board placed in a PC running Windows. This PC reads out the results from the FPGA — the original image and different filter responses for each camera — and sends the images using gigabit Ethernet to last stage — early cognitive vision. At this last stage the condensation of multi-modal primitives (explained in section 4) is taking place on a multi core PC running Ubuntu.

## 3. Preprocessing and early vision on FPGAs

This section describes the hardware and algorithms used and developed for the first two stages (as shown in Figure 1). First a brief description of the preprocessing stage is given followed by a more elaborated description of the processes taking place on the early vision stage.

### 3.1. Preprocessing

The preprocessing is handled by a Spartan 3 FPGA using the IEEE 1394a bus for communication with the cameras. It handles all camera control including synchronous triggering. The processes performed on the FPGA include the following: Bayer pattern demosaicing, region of interest control and rectification and undistortion.

The FPGA handles two modes of operation: the down-sampling mode where the original 2048x2048 image is downsampled to 512x512, or region of interest where a 512x512 part of the original image is extracted. The hardware and processes for this stage are explained in detail in [10].

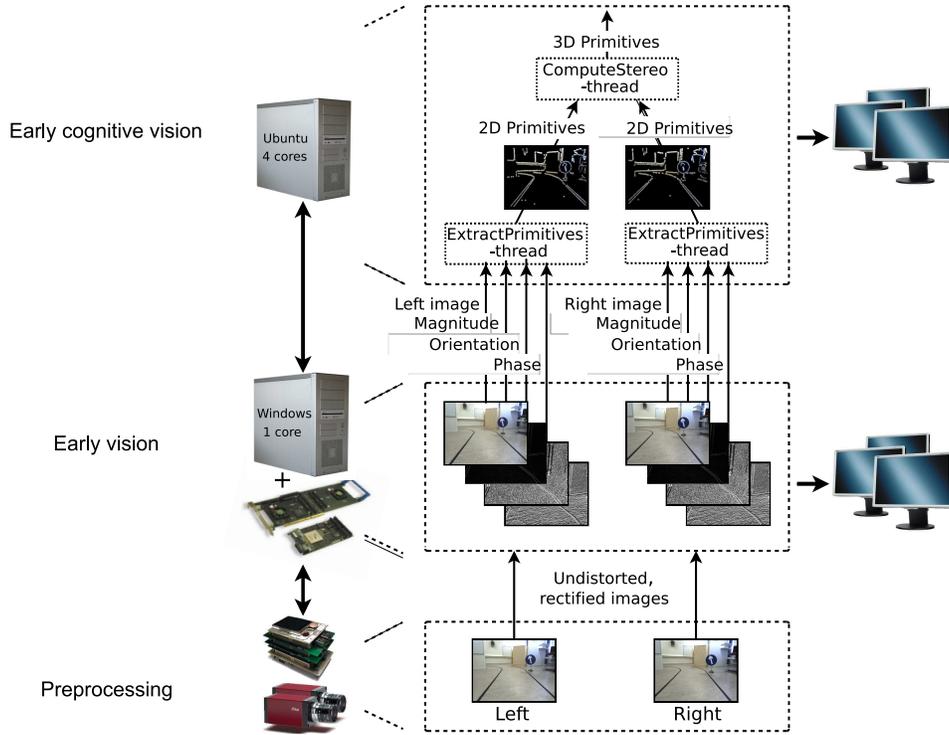
### 3.2. Local features computation

Local image features, magnitude, phase and orientation can be extracted using Quadrature filters. In our system, these filters are computed using the Hilbert transform of the Gaussian derivative as described in [3]. Concerning the constraints of real-time embedded systems, we will focus on second order Gaussian derivatives because they represent a good trade-off between accuracy and resource consumption. In our architecture, we use 9 tap kernels with peak frequency of  $f_0 = 0.21 \text{ pixels}^{-1}$  and bandwidth  $\beta = 0.1 \text{ pixels}^{-1}$  as utilized in [3]. This derivative order is able to provide high accuracy at a reasonable system cost.

The details about the different computation stages required could be found in [3] and can be summarized as follows:

1. Separable convolution using Second order Gaussian derivatives  $G_{xx}$ ,  $G_{xy}$ , and  $G_{yy}$  and their Hilbert transforms  $H_{xx}$ ,  $H_{xy}$ ,  $H_{yx}$ , and  $H_{yy}$ .
2. Linear filter combination and steering at 8 orientations.

After these two stages, we have 8 complex numbers for each pixel corresponding to the different orientations and even-odd filter output pairs. The next stage is the computation of the local features (magnitude, phase and orientation). In order to do that, we need to interpolate the feature values computed from this set of outputs. We will focus on the tensor-based Haglund's approach [4]. Based on a local tensor that projects the different orientations, information can be computed as described in equations (1-3). In these equations,  $j$  stands for the complex unit and  $i$  for the oriented filter number (1-8). The even and odd filter convolution outputs are represented by  $c$  and  $s$ . Finally,  $M_{local}$ , stands for the local magnitude,  $\theta_{local}$  for the orientation and  $\phi_{local}$  for



**Figure 1. System architecture.** From bottom up, the data stream is processed by different platforms. After the images are captured from the cameras they are rectified in an FPGA platform (this can be considered a specific purpose embedded system that delivers undistorted and rectified images). This data stream is then transferred to a FPGA based PCI co-processing board that extracts the low level vision cues (local magnitude, orientation and phase). High performance computing engines embedded in the FPGA devices allow efficient handling of regular processing path. Finally, software modules perform the condensation and stereo computation using coarse parallelism on multicore PCs.

the phase.

$$M_{local} = \frac{\sum_i \sqrt{c_i^2 + s_i^2}}{N} \quad (1)$$

$$\theta_{local} = \frac{1}{2} \arg[\sum_i (c_i^2 + s_i^2) \exp(j2\theta_i)] \quad (2)$$

$$\phi_{local} = \arctan\left(\frac{\tilde{S}}{\tilde{C}}\right) \quad (3)$$

$$\tilde{C} = \sum_i c_i |\cos(\theta_i - \theta_{local})|$$

$$\tilde{S} = \sum_i s_i \cos(\theta_i - \theta_{local})$$

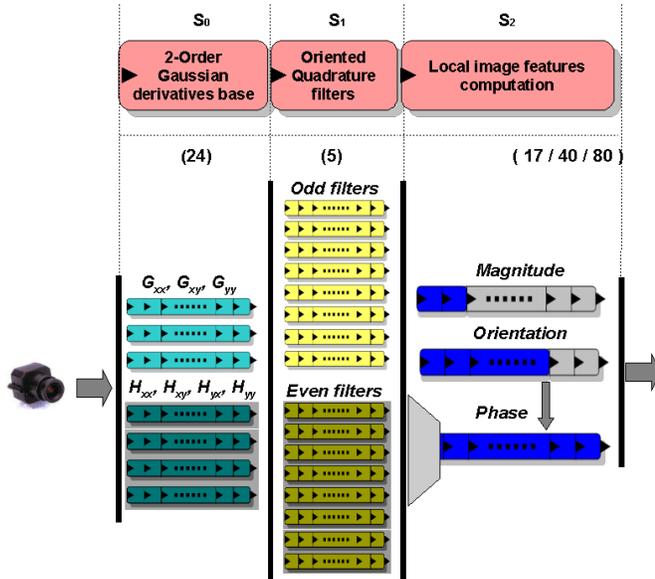
For an example of the outputs, see Figure 3B. Please note that equation (2) uses the value of the energy instead of magnitude as weighting factor for orientation computation (square root operation avoided). The effect on the orientation accuracy is indiscernible but significantly reduces

hardware delays as shown in [2].

### 3.3. Image processing core for real-time computation of magnitude, phase and orientation

The equations (1-3) have been used for designing our real-time architecture with reconfigurable hardware. The processing core is a long-datapath pipelined architecture whose parallelism grows along the stages to keep our throughput goal of one pixel output per clock cycle. This is a key factor to achieve high performance in our system.

The proposed architecture is based on the system described in [2]. Nevertheless, we have increased the bit-width of the different variables as well as modified the way of computing the phase to significantly increase the system accuracy. The new circuit architecture is schematically shown in Figure 2.



**Figure 2. Image feature processing core. Coarse pipeline stages are represented at the top, and long-datapath scalar units, at the bottom. The number of parallel datapaths increases according to the algorithm structure. The entire system has more than 109 pipelined stages (without counting other interfacing hardware controllers such as memory or video input/output interfaces). This allows for the computation of the three image features at one estimation per clock cycle. The number of substages for each coarse-pipeline stage is indicated in brackets in the upper part of the figure. Note that long buffers are required in the last stage to account for the data dependencies that are required at this point.**

The resulting core resource consumption and performance is shown in table 1. Note that, taking advantage of the highly pipelined architecture, for each clock cycle the core is able to process one pixel, resulting in a very high performance circuit.

The comparison with our previous results of [2] indicate that the new version consumes significantly more resources. This is mainly due to two factors. First, the bit-width of the variables has been increased to make them similar to a 16-bit software implementation. Second, the new phase computation technique requires to delay the 16 quadrature filters values until the orientation value is ready. In order to avoid degrading the system throughput, this is done by using 16 memory buffers of distributed memory. Although

this consumes a high quantity of resources, it is justified because the new phase estimations are much more accurate.

## 4. Early cognitive vision in a coarse parallel architecture

At the stage of early cognitive vision, the processing is parallelized in a more coarsely fashion. The computer used for this stage has two AMD Opteron 285 2.6GHz dual core processors giving a total of four cores. Two processes are computed at this stage: 2D-primitive extraction for each camera and stereo computation generating 3D-primitives<sup>1</sup>.

The primitives can be seen as a functional abstraction of so called 'hypercolumns' which are a cortical structure in the visual area V1 (see [7]) in which different aspects (or modalities) of visual information such as orientation, colour, local motion, and disparity are processed at a certain retinal position. The primitives cover the very same modalities that are known to be processed in a hypercolumn and condense this information to be accessible and usable for higher level processes (for a more detailed discussion of the biological analogy see [13]).

### 4.1. 2D-primitive extraction

2D-primitives are sparse image descriptor, that are extracted at contours. The primitives form a feature vector containing the edge position with sub-pixel accuracy, the local orientation, phase (contrast transition) and colour on both sides of the edge. The extraction is performed on the information from the filter responses — magnitude, orientation and phase — provided by the early vision stage (see figure 3A+B). Figure 3C shows the extracted primitives from the left image in figure 3A. For details see [13].

### 4.2. Stereo computation

2D-primitives are extracted on stereo pairs of images and are matched using the epipolar line (the images were rectified in the preprocessing stage) and similarity constraints. 3D-primitives are then reconstructed from the matched pairs of 2D-primitives (see figure 3D). For further information, see [13, 16].

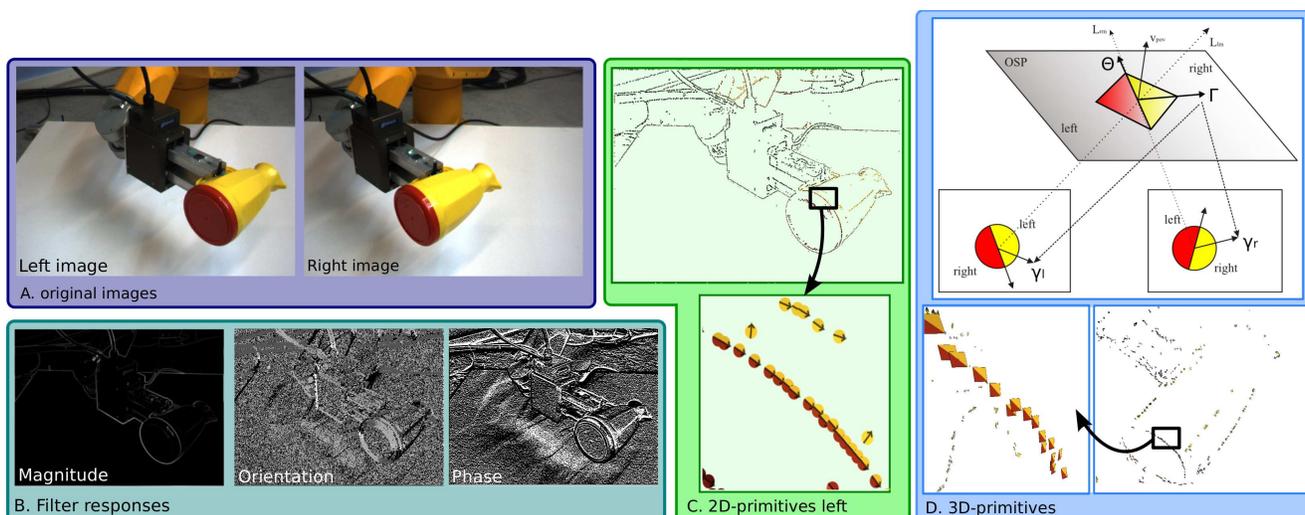
Figure 4 shows the pipelined structure of the parallelization scheme and how the different cores are utilized.

As displayed in figure 4, three cores are used for computation and the last core is used to visualize the results of the different processes. The 2D-primitive extraction need to be finished before stereo computation can begin. Consequently, there is a delay of two time steps before the 3D-primitives are ready for visualization and other processes.

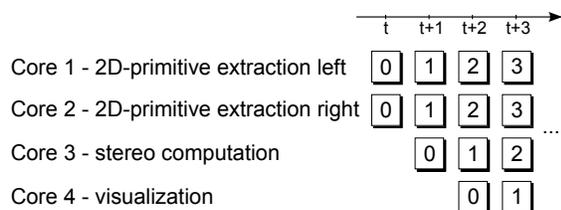
<sup>1</sup>These processes are programmed in C++ and compiled with O3 optimization

**Table 1. Complete system resources required for the local image features computing core. The circuits have been synthesized on a Xilinx FPGA Virtex II XC2V6000-4. Results obtained using the Xilinx ISE Foundation software. (EMBS stands for embedded memory blocks). 'Max image resolution' states the maximum image resolution that the core can handle. In this setup, the core processes 512x512 image pairs.**

| Slices/(%)  | EMBS / (%) | Embedded multipliers /(%) | Max clock frequency | Max image resolution | Fps  |
|-------------|------------|---------------------------|---------------------|----------------------|------|
| 20512 (60%) | 9 (6%)     | 121 (84%)                 | 40.5                | 1024x1024            | 38.6 |



**Figure 3. Illustration of the primitive extraction stage. A) original images (left and right); B) filter responses for the left image; C) extraction of 2D-primitives in the left image; and D) stereo reconstruction of 3D-primitives.**



**Figure 4. Pipelined structure of the parallelization scheme demonstrating what is computed on the four different cores at different times. The numbers in the boxes illustrate the index of the image data.**

## 5. Results

Table 2 shows the latency measurements of the different stages of the hybrid system and a corresponding sequential system running on a single core processing a pair of

512x512 images.

The time used by the two processes at the early cognitive vision stage vary slightly depending on the number of primitives which depend on the number of edge structures in the scene.

The measurements add up to a latency of 0.69s through the hybrid system and 2.31s through a sequential equivalent. Some of the latencies of the processes in the sequential system are multiplied by two since they need to be performed on both the left and right image.

Even though the latencies of the processes at early vision and early cognitive vision stages in the hybrid system are equal, there is a huge difference in the throughput of the different stages due to the massively pipelined FPGA implementations. The FPGA implementation handling the preprocessing can execute around 40fps ([10]), the early vision core handles 38.6fps (table 1), whereas each of the processes at the early cognitive vision stage can handle  $\frac{1}{0.2s/frame} = 5fps$ .

**Table 2. Latency measurements. The latency adds up to 0.69s for the hybrid system and 2.31s for the sequential system. The frequency of the hybrid system is limited by the processes at the early cognitive vision stage to  $\frac{1}{0.2s/frame} = 5fps$ , whereas the frequency of the sequential system is  $\frac{1}{2.31s/frame} \approx 0.43fps$ .**

| Stage                  | Process  | Hybrid | Sequential |
|------------------------|--|--------|------------|
| Preprocessing          | Pike image acquisition and IEEE 1394a transfer | 0.06s  | 0.06s      |
|                        | Bayer demosaic, ROI/DS, and undist+rect        | 0.03s  | 2×0.45s    |
| Early Vision           | Magnitude, orientation and phase               | 0.20s  | 2×0.43s    |
| Early Cognitive Vision | 2D-primitive extraction                        | 0.20s  | 2×0.20s    |
|                        | Stereo computation                             | 0.20s  | 0.20s      |
| Total                  |  | 0.69s  | 2.31s      |

## 6. Conclusion

In this paper, we have presented a hybrid architecture consisting of FPGAs and a multicore PC for the processing of multi-modal visual primitives. The processing speed has been improved significantly compared to a sequential system — from 0.43fps to 5fps — but the system has potential for speeding up the overall processing even further (see below). Also the latency of the system has been decreased from 2.31s to 0.69s. This increase of performance is crucial for the two application domains we are working in, i.e., robotics and driver assistance.

The bottleneck of the system is the last stage — early cognitive vision — limiting the system to perform at 5fps. In the future, we will investigate ways of improving this, e.g. using newer and faster processors with more cores or putting parts of the processes on FPGAs/GPUs. The latency is still a factor 3 larger than in the the human visual system [9]. However, we plan that the FPGA PCI board for the early vision stage will be placed directly in the multicore Ubuntu-PC, thus removing the latency caused by the Windows-PC.

## 7. Acknowledgement

This work has been supported by the European Commission - FP6 Project DRIVSCO (IST-016276-2). Thank you to Nicolas Pugeault for providing material for figure 3. Thank you to Kai Welke for his support concerning computing on the coarse parallel machine.

## References

- [1] Pacoplus: Perception, action and cognition through learning of object-action complexes. *IST-FP6-IP-027657, Integrated Project*, 2006-2010.
- [2] J. Díaz, E. Ros, and R. Carrillo. Local image phase, energy, and orientation extraction using FPGAs. *International Journal of Electronics*, 95(7):743–760, 2008.
- [3] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.
- [4] L. Haglund. *Adaptive Multidimensional Filtering*. PhD thesis, n° 284, Linköping University, SE-581 83 Linköping, Sweden, 1992.
- [5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [6] <http://www.pspc.dibe.unige.it/drivscot/>, editor. *DRIVSCO: Learning to Emulate Perception-Action Cycles in a Driving School Scenario (FP6-IST-FET, contract 016276-2)*. 2006-2009.
- [7] D. Hubel and T. Wiesel. Brain mechanisms of vision. *Scientific American*, 241:130–144, 1979.
- [8] J. Jones and L. Palmer. An evaluation of the two dimensional Gabor filter model of simple receptive fields in striate cortex. *Journal of Neurophysiology*, 58(6):1223–1258, 1987.
- [9] E. Kandell, J. Schwartz, and T. Messel. *Principles of Neural Science (4th edition)*. McGraw Hill, 2000.
- [10] A. Kjær-Nielsen, L. B. W. Jensen, and N. Krüger. A real-time embedded system for stereo vision preprocessing using an FPGA. *ReConFig'08*, (Submitted), 2008.
- [11] D. Kraft, N. Pugeault, E. Başeski, M. Popović, D. Kragic, S. Kalkan, F. Wörgötter, and N. Krüger. Birth of the Object: Detection of Objectness and Extraction of Object Shape through Object Action Complexes. *Special Issue on "Cognitive Humanoid Robots" of the International Journal of Humanoid Robotics*, 2008. (accepted).
- [12] N. Krüger. Signal- and symbol-based representations in computer vision. *Proceedings of the 1st International Symposium on Brain, Vision and Artificial Intelligence 19-21 October, 2005, Naples, Italy, Lecture Notes in Computer Science, Springer, LNCS 3704*, pages 167–176, 2005.
- [13] N. Krüger, M. Lappe, and F. Wörgötter. Biologically Motivated Multi-modal Processing of Visual Primitives. *Interdisciplinary Journal of Artificial Intelligence & the Simulation of Behaviour, AISB Journal*, 1(5):417–427, 2004.
- [14] M. Lades, J. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Würtz, and W. Konen. Distortion invariant object recognition in the dynamik link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993.
- [15] H.-H. Nagel. Image sequence evaluation: 30 years and still going strong. *ICPR*, pages 149–158, 2000.
- [16] N. Pugeault. *Early Cognitive Vision: Feedback Mechanisms for the Disambiguation of Early Visual Representation*. PhD thesis, Informatics Institute, University of Göttingen, 2008.
- [17] F. Wörgötter, N. Krüger, N. Pugeault, D. Calow, M. Lappe, K. Pauwels, M. V. Hulle, S. Tan, and A. Johnston. Early cognitive vision: Using gestalt-laws for task-dependent, active image-processing. *Natural Computing*, 3(3):293–321, 2004.