



Automatische MPSoC-Architektursynthese für Bildverarbeitungsalgorithmen

Raphael Kemmler
Betreuer:
Prof. Dr.-Ing. Bernd Schwarz





Inhalt

1. Automatische Architektursynthese auf Basis von C/C++
2. Anwendung auf Bildverarbeitungsalgorithmen
3. Roadmap zur Masterarbeit
4. Risiken
5. Zusammenfassung
6. Ausblick auf Funktionserweiterungen



Inhalt

1. **Automatische Architektursynthese auf Basis von C/C++**
 1. Zynq-MPSoC mit ARM A9-Dualcore
 2. RTL-Codegenerierung mit High-Level-Synthese (HLS)
 3. Steuerung der HLS: Strukturvariationen
2. Anwendung auf Bildverarbeitungsalgorithmen
3. Roadmap zur Masterarbeit
4. Risiken
5. Zusammenfassung
6. Ausblick auf Funktionserweiterungen

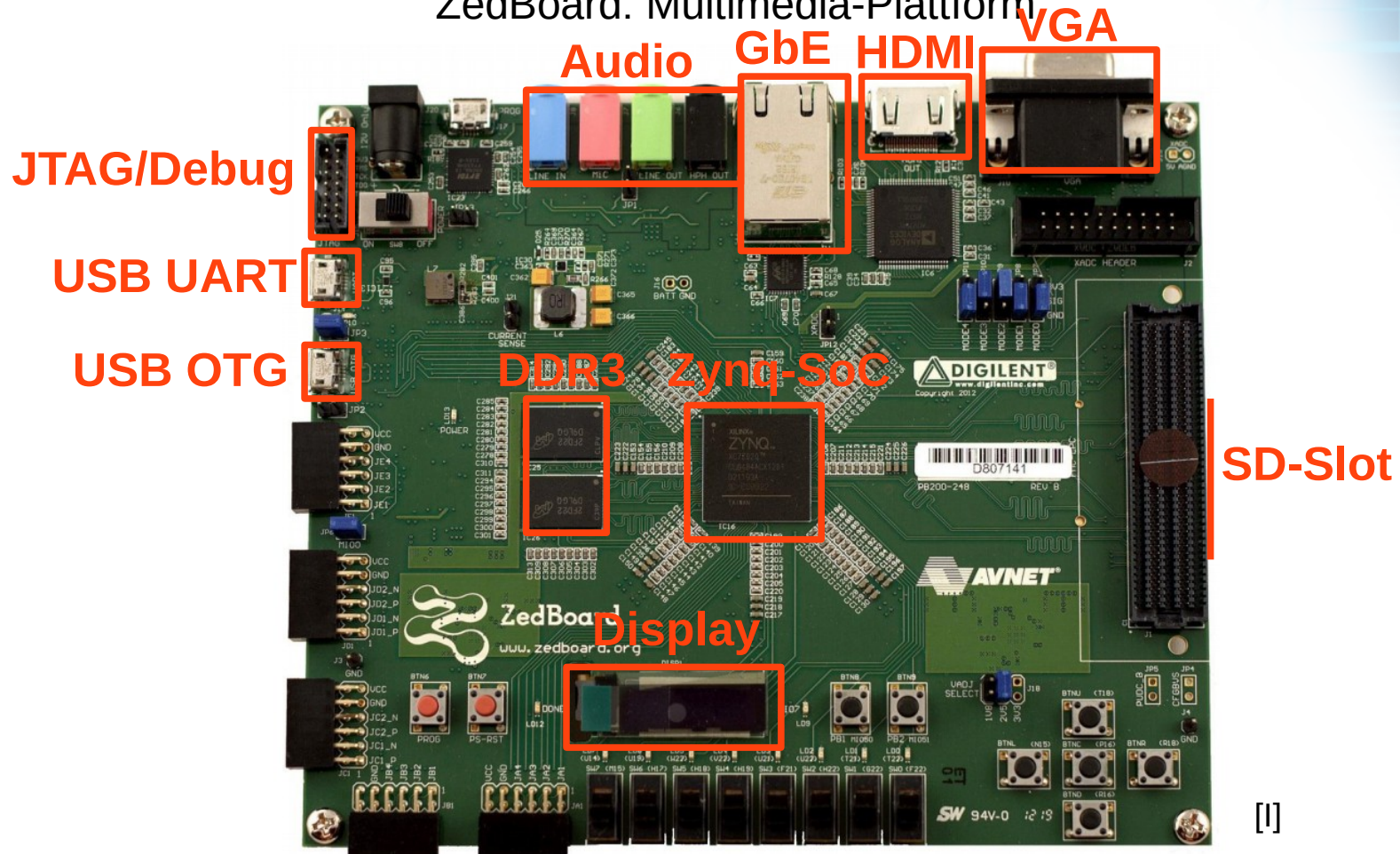


1. Automatische Architektursynthese

- High Performance Embedded Computing
- FPGA (Field Programmable Gate Array)
- Parallelität: Arithmetikressourcen - Speicher
- Hardwarebeschreibungssprachen (HDL): VHDL/Verilog
- MPSoC: ARM A9 Prozessoren (866 MHz) + FPGA
- HW-SW-Partitionierung:
Algorithmik ausgeführt auf Arithmetikressourcen
und/oder auf Prozessoren
- Leistungsvergleich durch Prototypenanalyse

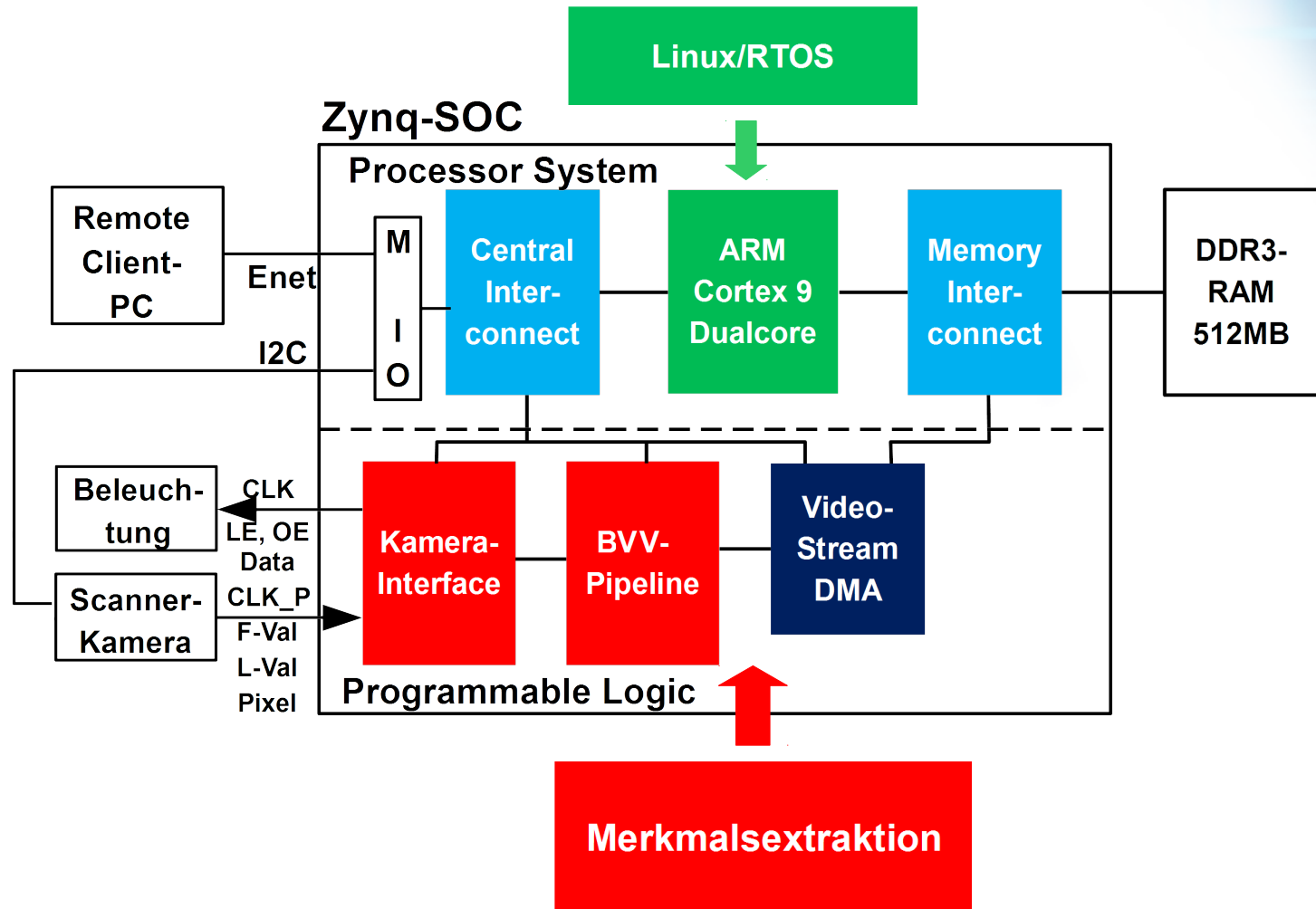
1.1 Zynq-MPSoC

ZedBoard: Multimedia-Plattform



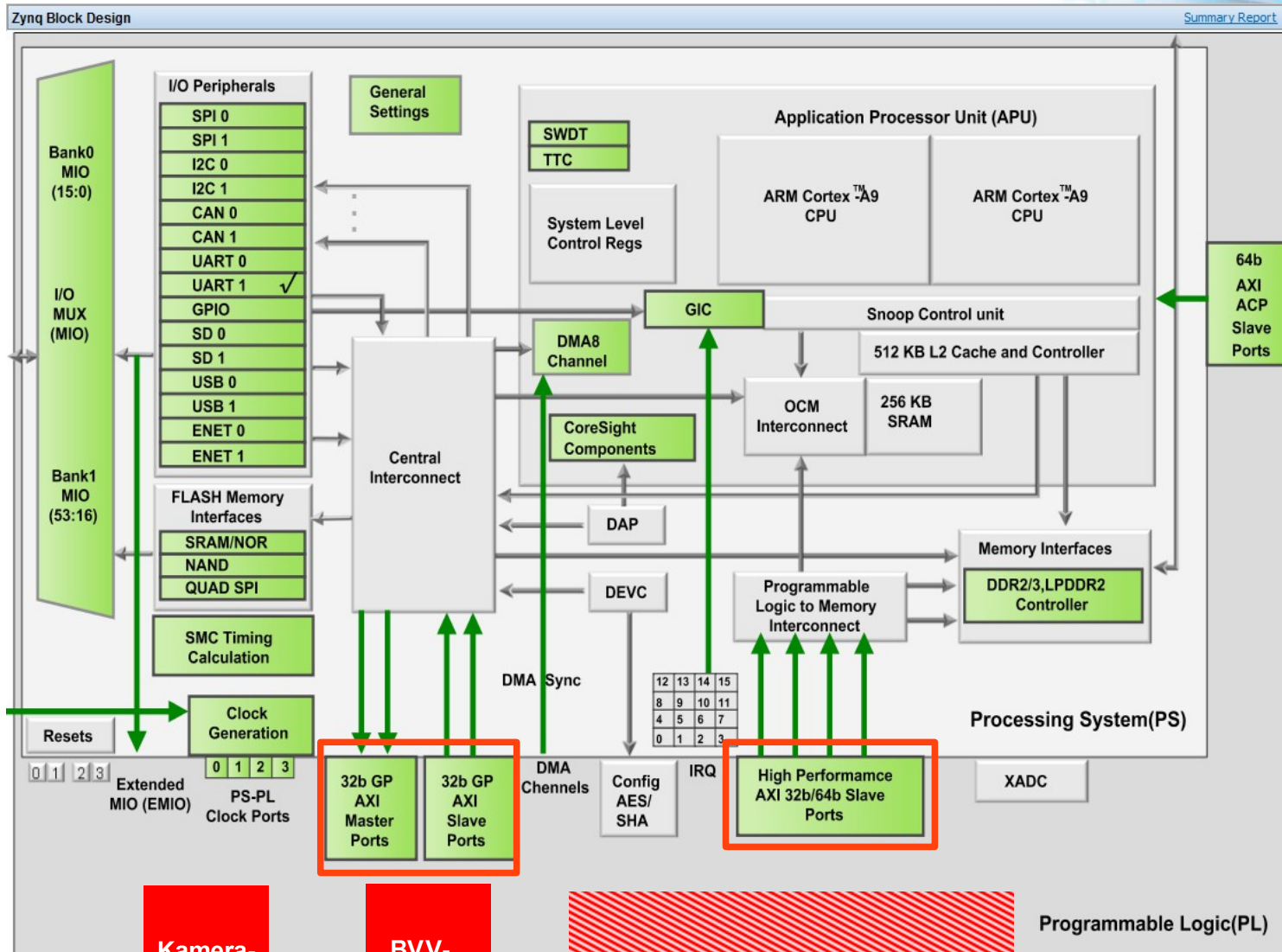
[1]

1.1 Zynq BV-System



MPSoC - System Assembly View

uC-
Peripherie



Kamera-
Interface

BVV-
Pipeline

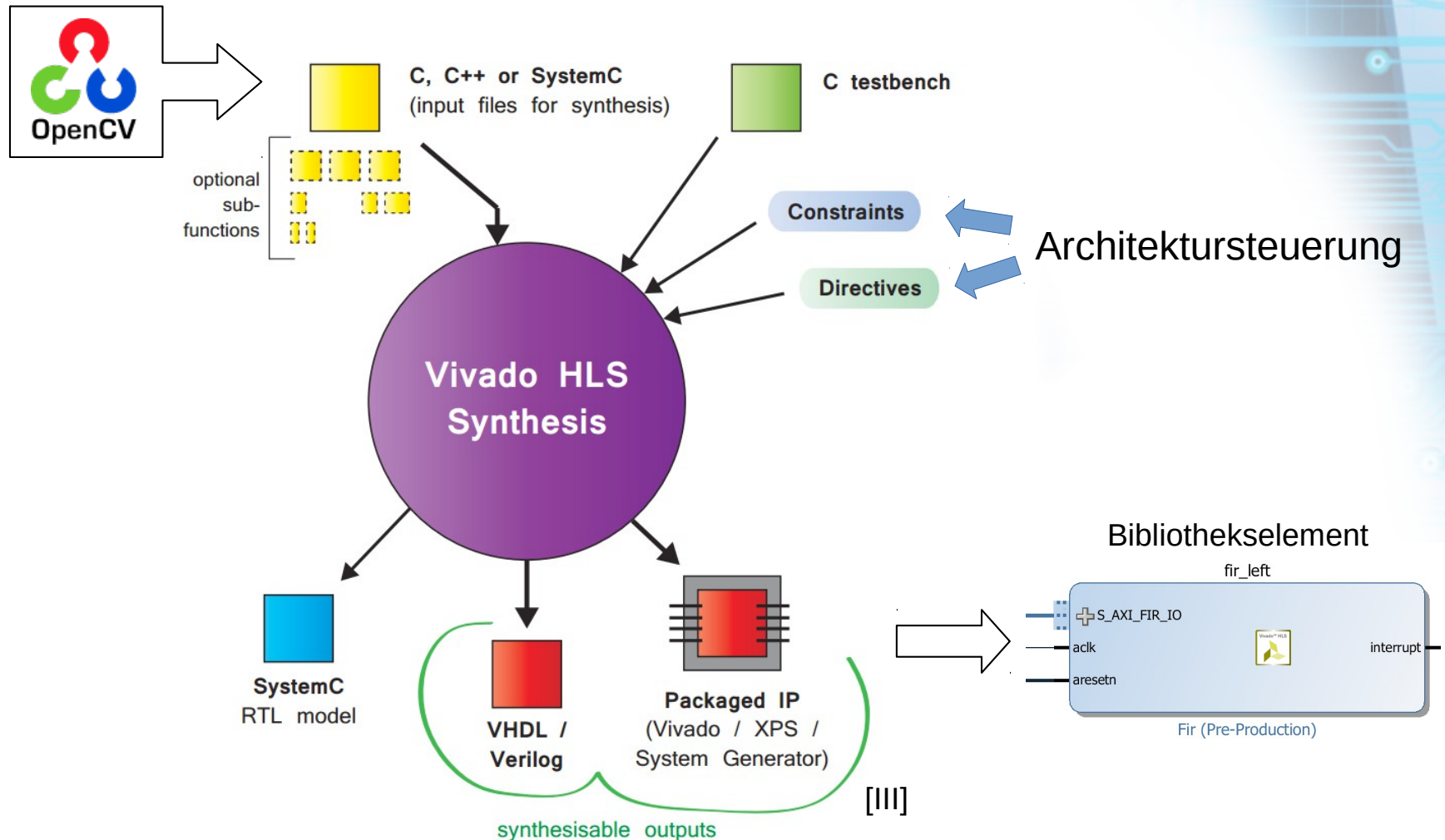
Merkmalsextraktion



1.2 High Level Synthese

- Synthesefähige HDL-Codegenerierung aus Hochsprachen-Code
 - C/C++, SystemC
- Ziel: das Beste aus zwei Welten
 - Verringerte Entwurfskomplexität → einfachere und schnellere Entwicklung
 - hoher Datendurchsatz durch Parallelität

1.2 High Level Synthese

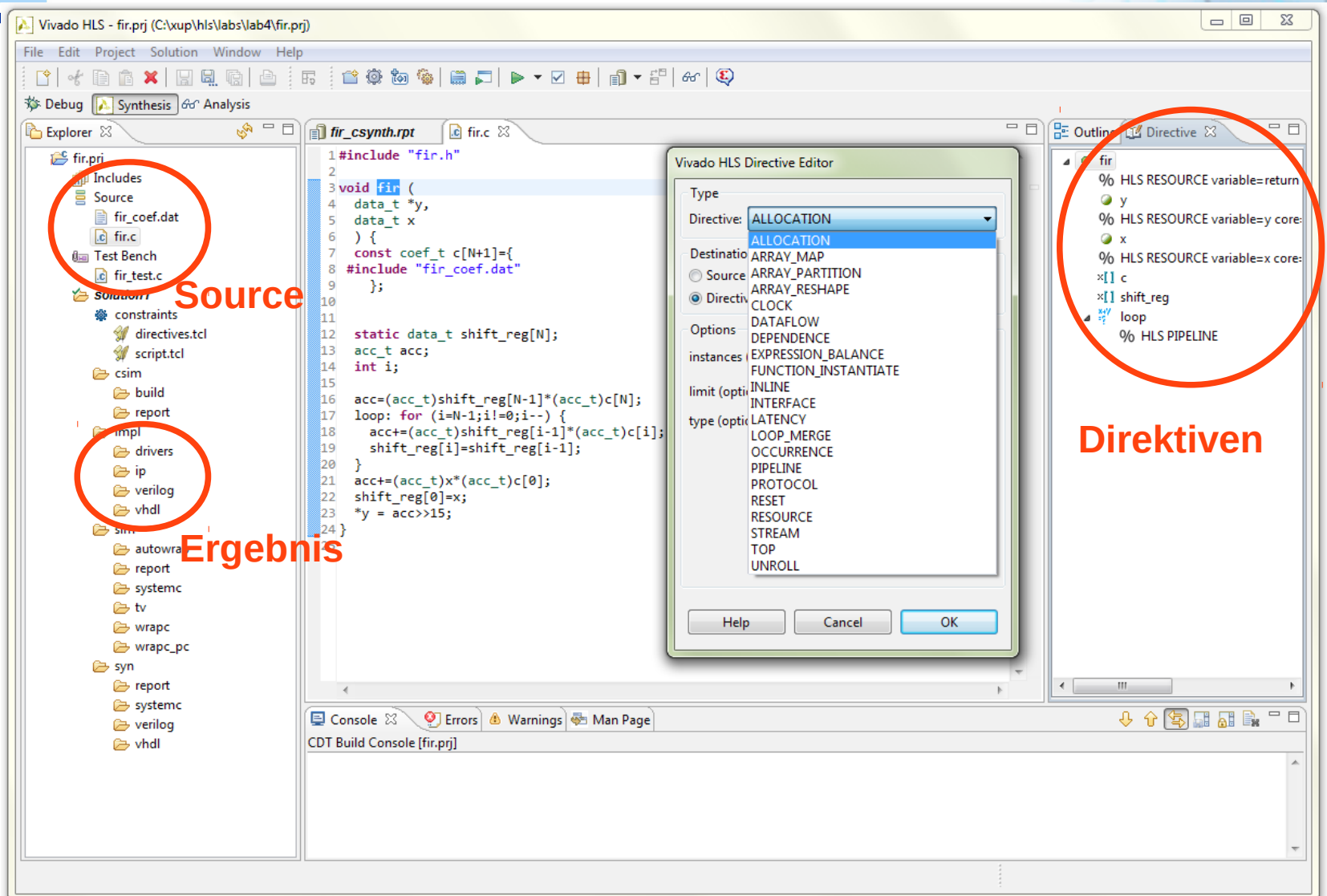




1.3 Steuerung der automatischen Synthese

- Direktiven
 - Pipelining: Durchsatzserhöhung
 - Allocation: Resource-Sharing
 - Inline: Funktionshierarchie wird entfernt
 - Dataflow: Erlaubt Funktionen/Loops parallele Ausführung
 - Resource: Konfiguration des Speichers von Variablen (z.B. Bram/DDR)
- Bitgenaue Datentypen: Ressourcen- und Genauigkeitsplanung
- Floating-Point-Operationen durch Fix-Point-Operationen ersetzen

Vivado HLS





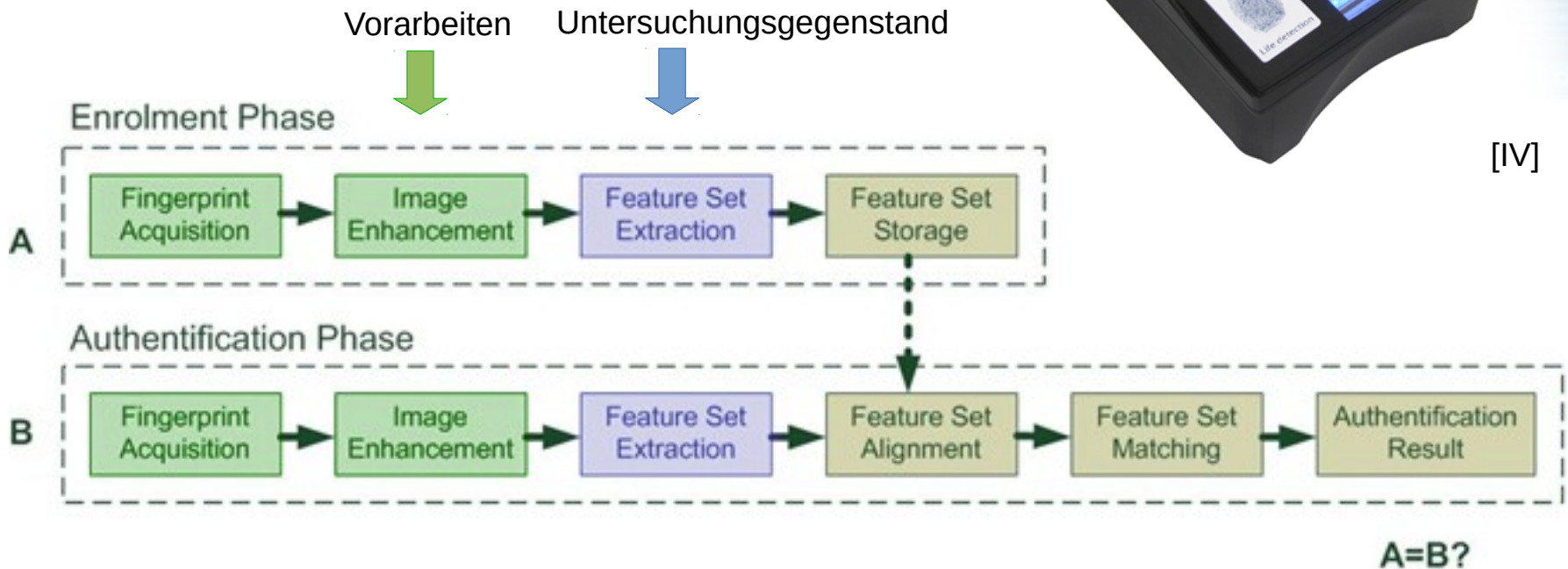
Inhalt

1. Automatische Architektursynthese auf Basis von C/C++
- 2. Anwendung auf Bildverarbeitungsalgorithmen**
 1. Fingerabdruckscanner
 2. Bildvorverarbeitungskette: Bildkorrekturen
 3. Merkmalsextraktion: Bifurcations, Ridge Endings
3. Roadmap zur Masterarbeit
4. Risiken
5. Zusammenfassung
6. Ausblick

2. Anwendung

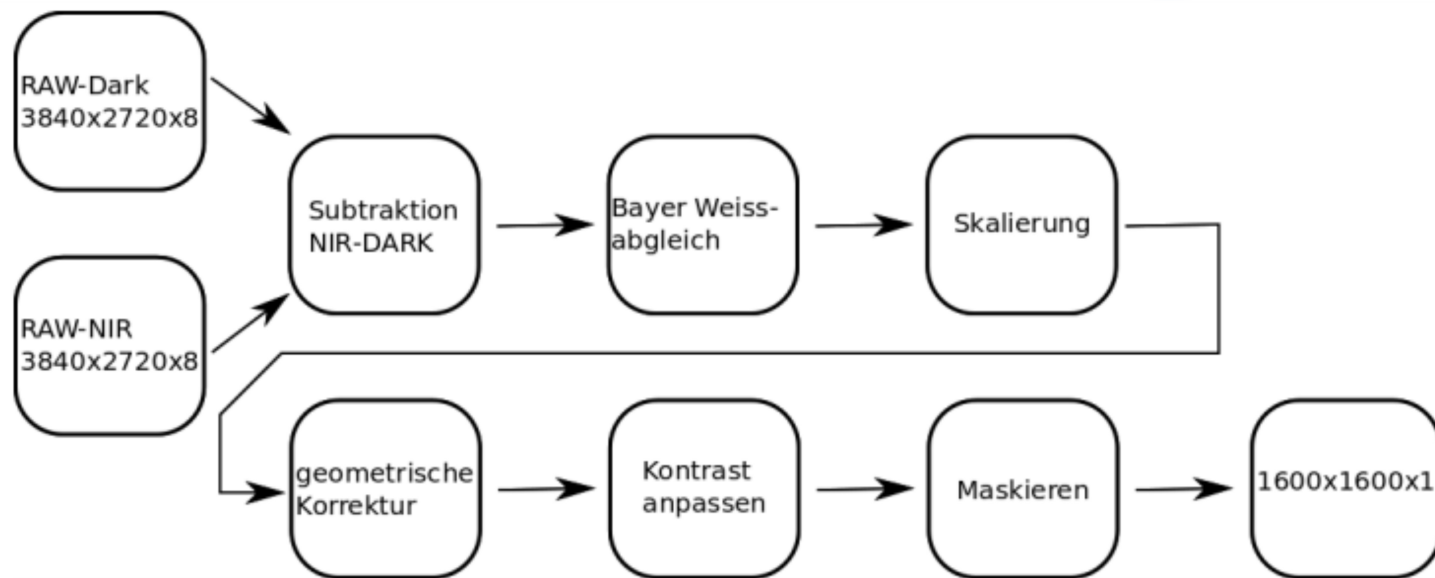
2.1 Fingerabdruckscanner

- Biometrisches Identifikationssystem
- Scanner LF10 von Dermalog



2.2 Bildvorverarbeitungskette

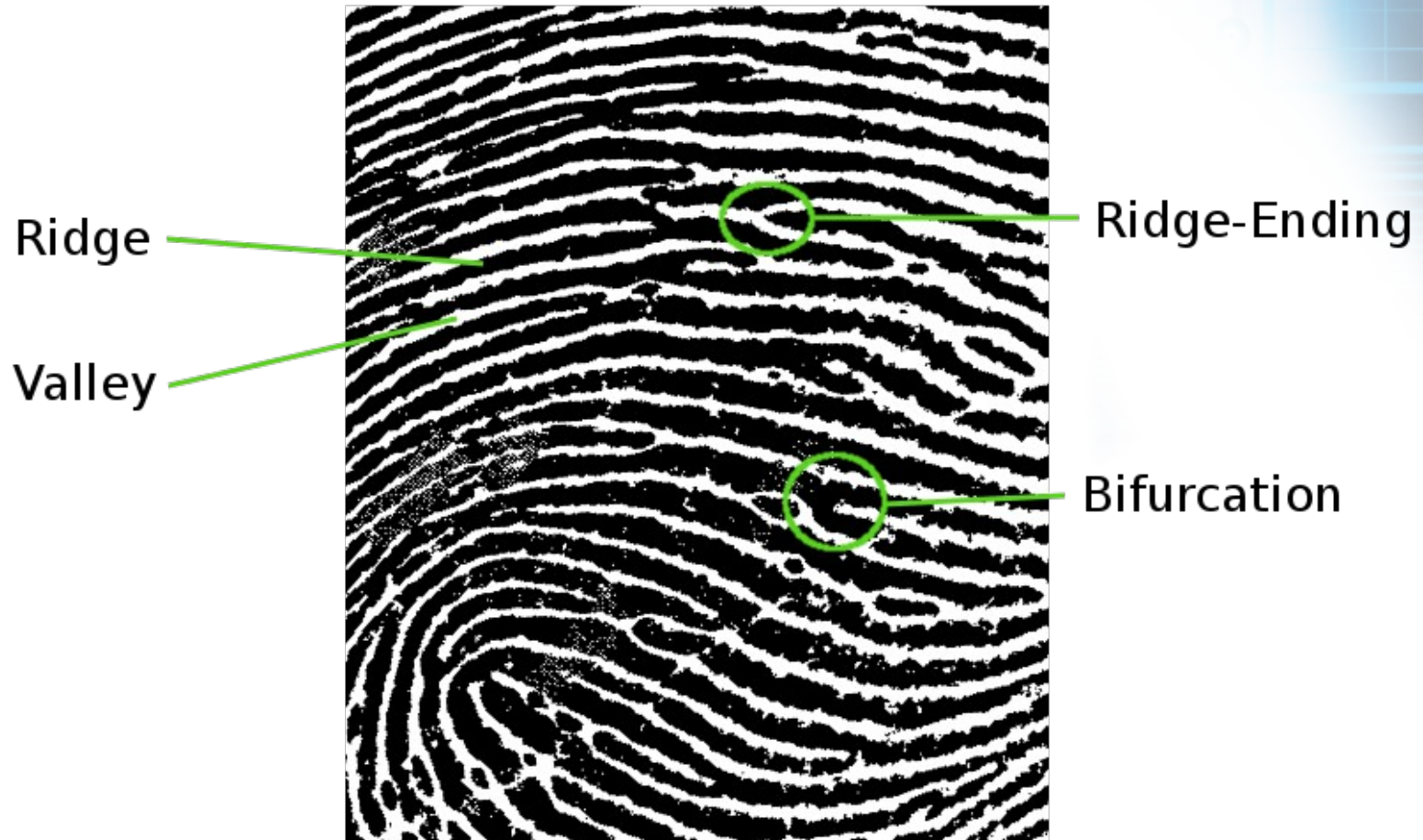
- Interface: CCD-Sensor ↔ Zynq-MPSoC
- Größenskalierung, geometrische Entzerrung, Kontrastverstärkung
- Bachelor- und Masterarbeiten: HDL-Modelle



[IV]

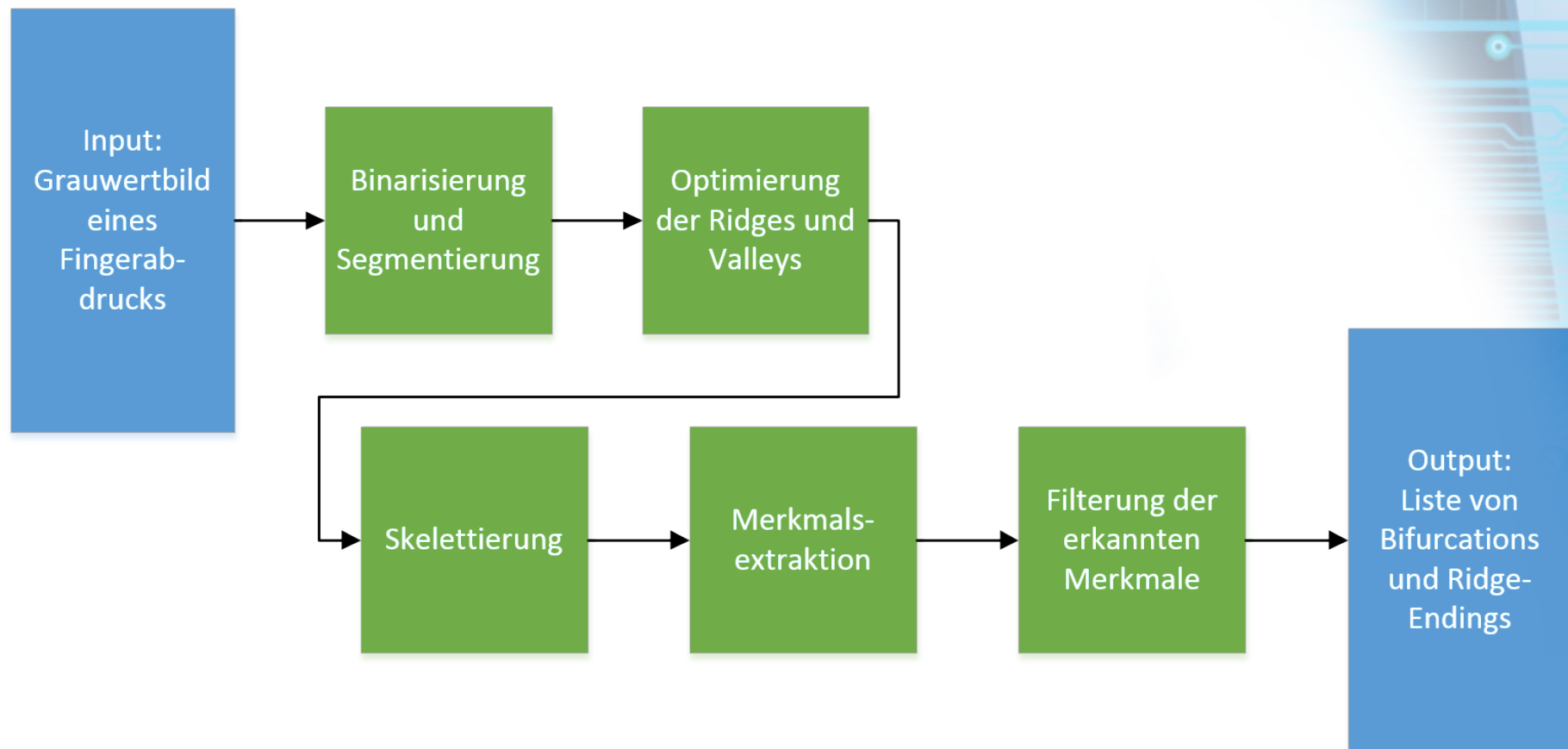


Begriffe für Minutien



Minutien = Merkmale eines Fingerabdrucks

2.3 Bildverarbeitung zur Merkmalsextraktion





2.3 Bildverarbeitung zur Merkmalsextraktion

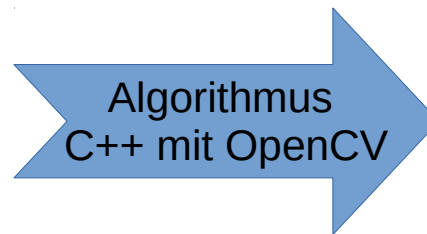
Softwaremodelle für:

- Datenmengenreduzierung
- Trennung Objekt-Hintergrund
- Mathematische Morphologie:
 - Zusammenhängende Ridges und Valleys erzeugen
 - Löcher in Ridges schließen
- Skelettierung: Ridges auf Breite von 1 Pixel reduzieren
- Merkmalsextraktion zur Identifikation von:
 - Bifurcations
 - Ridge Endings

→ in Projekt 1 implementiert: OpenCV-basiert in C++

Binarisierung und Segmentierung

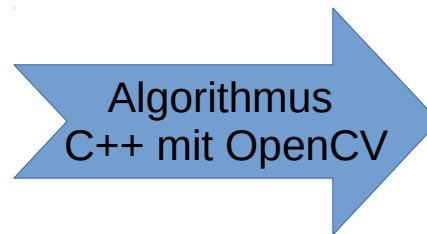
- Adaptive Local Thresholding: Grauwerte → Schwarz-Weiß
- Threshold Segmentierung: Hintergrund eliminiert





Optimierung

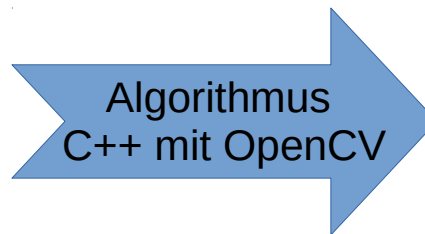
- Mathematische Morphologie [1]: Zusammenhängende Ridges und Valleys erzeugen, Löcher in Ridges schließen





Skelettierung

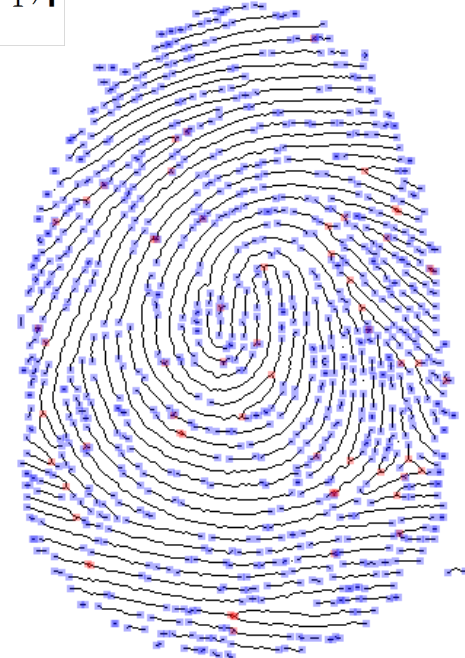
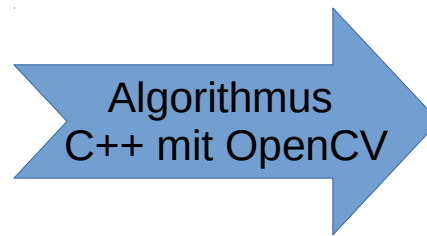
- Parallel thinning algorithm with two-subiteration Algorithms [2]:
1-Pixel Linienbreite



Merkmalsextraktion

- Crossing-Number-Verfahren [3]: Bifurcations und Ridge-Endings finden

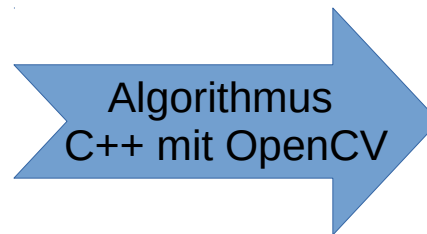
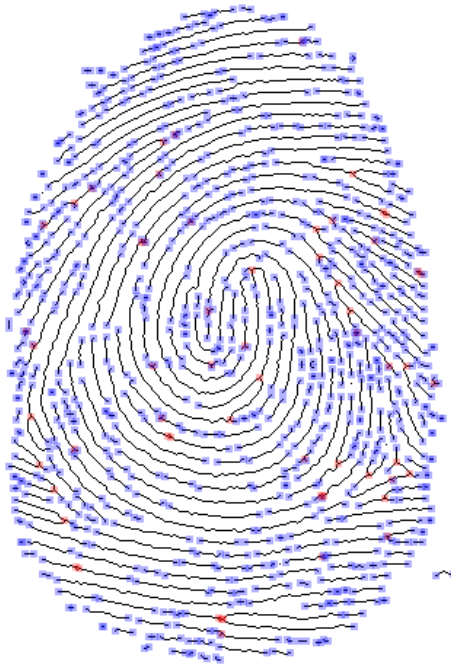
$$cn(p) = \frac{1}{2} \sum_{i=1 \dots 8} |val(p_{i \bmod 8}) - val(p_{i-1})|$$





Merkmalfilterung

- Abstandsfilterung: Minutien filtern





Inhalt

1. Automatische Architektursynthese auf Basis von C/C++
2. Anwendung auf Bildverarbeitungsalgorithmen
3. **Roadmap zur Masterarbeit**
 1. **Projekt 1 & 2: Algorithmusertprobung und Methodenanalyse**
 2. **Masterarbeit: Geräteprototyp**
4. Risiken
5. Zusammenfassung
6. Ausblick



3. Roadmap zur Masterarbeit

Projekt 1:

- Entwurf und Implementierung der Algorithmenkette für die Merkmalsextraktion
- Dokumentation bis: Mitte Januar 2015

Projekt 2:

- HLS-Entwurfsstrategien analysieren
- Algorithmenstruktur ↔ Architektur
- Kommunikationsinterfaces zwischen IP-Cores auswählen
- Ausgewählten Algorithmus mit HLS abbilden
- Dokumentation bis: 30. April 2015

3. Roadmap zur Masterarbeit

	Binarisierung/ Segmentierung	Optimierung (math. Morphologie)	Skelettierung	Merkmalsextra ktion	Filterung der Merkmale	Methodik in HLS
Dynamischer Speicher				✓	✓	
Statischer Speicher	✓	✓	✓	✓	✓	
Neighborhood- Operation (Schieberegister)		✓	✓	✓		
Bildsegmentierung	✓					
Wahlfreier Zugriff auf Bildpunkte	✓					
Logische Operationen		✓	✓			
(For-)Schleifen	✓		✓	✓	✓	
If-Bedingung	✓			✓	✓	
Addition/Subtraktion	✓		✓	✓	✓	
Multiplikation					✓	
Division	✓					

Zu klären



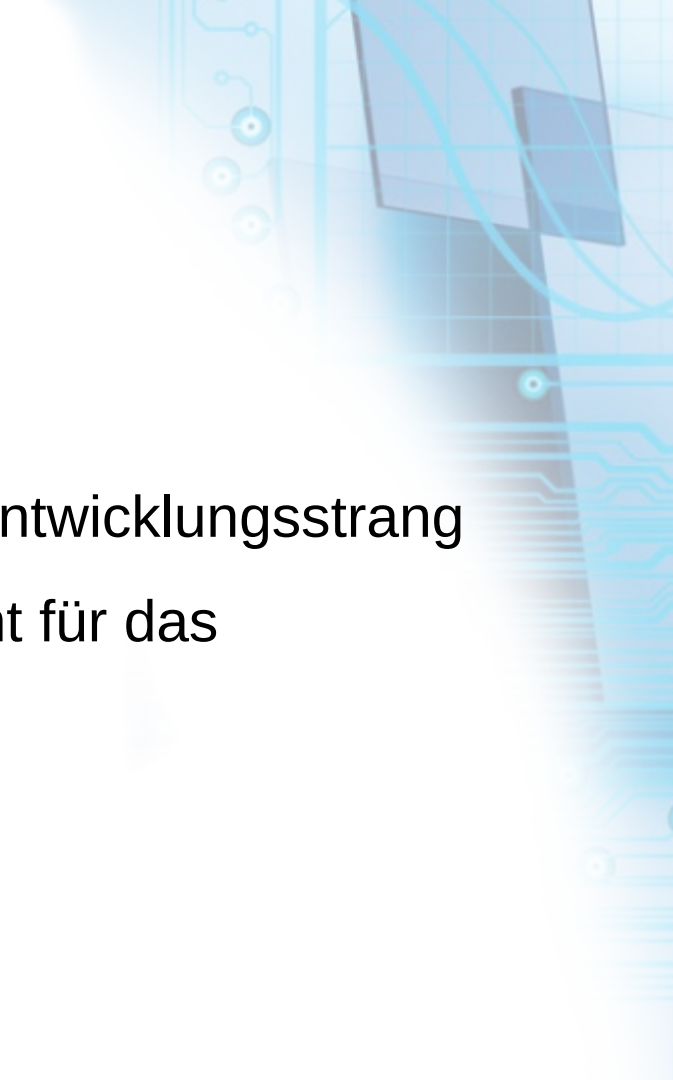
3. Roadmap zur Masterarbeit

Masterarbeit:

- IP-Cores mit HLS erstellen
- Direktiven/Constraints -> IP-Core-Optimierung
- Integration in ein System zur Merkmalsextraktion
- Test und Evaluation gegen Implementierung auf ARM-CPU und Workstation
- Gesamtsystem erstellen inkl. Vorverarbeitung
- Zieltermin: Oktober 2015



4. Risiken



- Abhängigkeit von Vorarbeiten
- Abstimmung mit dem ARM A9 Softwareentwicklungsstrang
- Ressourcen des Zynq-SoCs reichen nicht für das Gesamtsystem
 - HW-SW-Partitionierung
 - Besseren Chip
- Weiterentwicklung der HLS-Toolchain



5. Zusammenfassung

- Automatische Architektursynthese mit HLS
- Anwendung: Merkmalsextraktion innerhalb eines Systems zur Fingerabdruckerkennung
 - Binarisierung und Segmentierung
 - Optimierung
 - Skelettierung
 - Minutienextraktion
 - Filterung
- Gesamtsystemintegration
- Vergleich mit ARM A9 und Workstation



6. Ausblick auf Funktionserweiterung

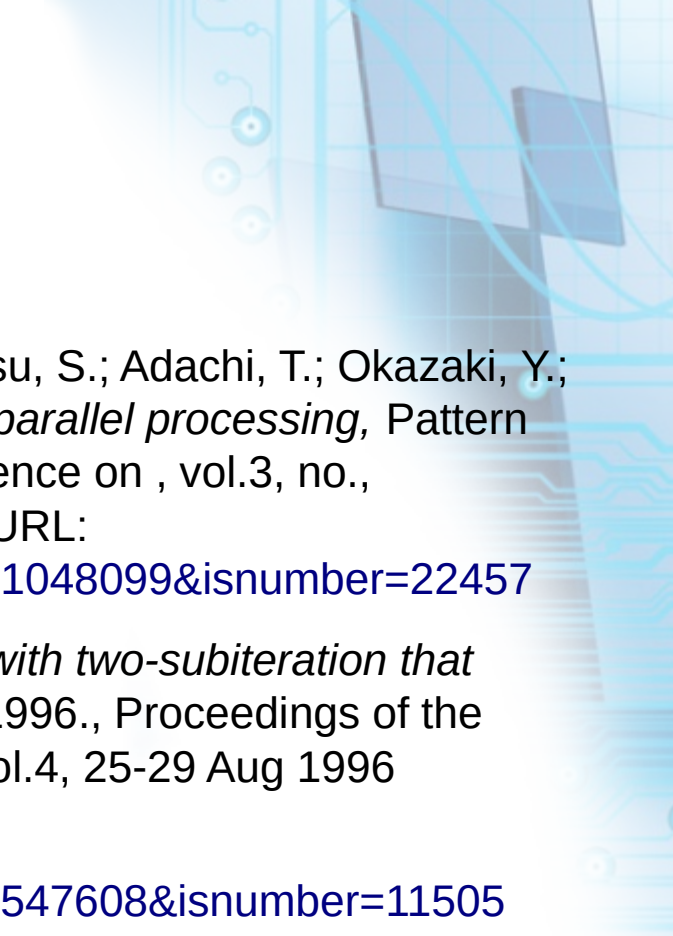
- Matching
- Optimierung der einzelnen Algorithmen
- Fingerabdruckverifikation mit on/off
- Minutien Standardkonform abspeichern
- Lebenderkennung



Fragen/Anmerkungen?



Quellen



[1] Ikeda, N.; Nakanishi, M.; Fujii, K.; Hatano, T.; Shigematsu, S.; Adachi, T.; Okazaki, Y.; Kyuragi, Hakaru. *Fingerprint image enhancement by pixel-parallel processing*, Pattern Recognition, 2002. Proceedings. 16th International Conference on , vol.3, no., pp.752,755 vol.3, 2002. doi: 10.1109/ICPR.2002.1048099 URL:

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1048099&isnumber=22457>

[2] Zhang, Y.Y.; Wang, P.S.P., *A parallel thinning algorithm with two-subiteration that generates one-pixel-wide skeletons*. Pattern Recognition, 1996., Proceedings of the 13th International Conference on , vol.4, no., pp.457,461 vol.4, 25-29 Aug 1996

doi: 10.1109/ICPR.1996.547608, URL:

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=547608&isnumber=11505>

[3] Maltoni, Davide ; Maio, Dario ; Jain, Anil K. ; Prabhakar, Salil: *Handbook of Fingerprint Recognition*. Berlin Heidelberg: Springer Science & Business Media, 2009. -ISBN 978-1-848-82254-2. S. 1-512



Bilderquellen

- [I] Avnet. *ZedBoard (Zynq™ Evaluation and Development) Hardware User's Guide*. 2012. URL:
http://zedboard.org/sites/default/files/ZedBoard_HW_UG_v1_1.pdf (besucht am 20.03.2014)
- [II] Louise H. Crockett, Ross a. Elliot, Martin a. Enderwitz. *The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc*. Strathclyde Academic Media, 2014. ISBN: 9780992978709
- [III] Xilinx. *UG998: Introduction to FPGA Design with Vivado HLS*. 2013. URL:
http://www.xilinx.com/support/documentation/sw_manuals/ug998-vivado-intro-fpga-design-hls.pdf
(besucht am 21.11.2014)
- [IV] Weber, R. *MPSoC-basierte Bildvorverarbeitung für ein biometrisches Identifikationssystem*. Masterarbeit, 2014.