

Entwurfsraum-Exploration von Hardware-Architekturen zur Klassifikation von Audio-Signalen

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades
Doktor-Ingenieur
(abgekürzt: Dr.-Ing.)
genehmigte Dissertation

von
Dipl.-Ing. Ingo Schmädecke

geboren am 25. Juli 1983 in Gehrden

2014

1. Referent
2. Referent
Tag der Promotion

Prof. Dr.-Ing. Holger Blume
Prof. Dr. ir. Simon Doclo
29.09.2014

Berichte aus der Informationstechnik

Ingo Schmäddecke

**Entwurfsraum-Exploration von Hardware-Architekturen
zur Klassifikation von Audio-Signalen**

Shaker Verlag
Aachen 2014

Bibliografische Information der Deutschen Nationalbibliothek

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

Zugl.: Hannover, Leibniz Univ., Diss., 2014

Copyright Shaker Verlag 2014

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

Printed in Germany.

ISBN 978-3-8440-3139-3

ISSN 1610-9406

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

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

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

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Mikroelektronische Systeme der Gottfried Wilhelm Leibniz Universität Hannover.

Herrn Prof. Dr.-Ing. Holger Blume danke ich herzlich für die wissenschaftliche Betreuung meiner Arbeit und für die Übernahme des 1. Referats. Er gewährte alle Freiheiten und bot optimale Arbeitsbedingungen, um Projekte wie die vorliegende Dissertation realisieren zu können. Herrn Prof. Dr. ir. Simon Doclo danke ich für sein Interesse an der Arbeit und für die Übernahme des 2. Referats.

Des Weiteren möchte ich mich bei meinen ehemaligen Kollegen des Instituts für Mikroelektronische Systeme für die langjährige, sehr gute Zusammenarbeit und die ausgesprochen angenehme Freizeitgestaltung bedanken. Außerdem möchte ich mich bei Herrn Dipl.-Ing. Steffen Blume für die freundschaftliche Zusammenarbeit während der Vorlesungsbetreuung, für die Zeit im gemeinsamen Büro, und für die vielen anregenden Diskussionen bedanken.

Insbesondere möchte ich mich bei Herrn Jun.-Prof. Dr.-Ing. Guillermo Páya Váya bedanken, der durch seine Ratschläge wertvolle Beiträge für das Gelingen dieser Arbeit geliefert hat. Bei Herrn Dipl.-Ing Jan Dürre und meinem Vater und möchte ich mich für die Durchsicht des Manuskripts und für die vielen Anregungen zur Anfertigung der Dissertation bedanken.

Meinen Eltern und meiner Schwester danke ich herzlich für die Unterstützung auf dem Weg zur Promotion. Mein Dank gilt auch allen meinen Freunden, die in schwierigen Zeiten zu mir gestanden haben. Das mir entgegengebrachte Verständnis und die Geduld waren wesentliche Voraussetzungen, um die Arbeit bewältigen zu können.

Barsinghausen, im Oktober 2014

Ingo Schmäddecke

Inhaltsverzeichnis

Abkürzungsverzeichnis	vii
Notation	ix
Kurzfassungen	x
1 Einleitung	1
1.1 Anwendungsszenario der automatisierten Musikklassifikation	2
1.2 Zielsetzung der Arbeit	4
1.3 Aufbau der Arbeit	6
2 Algorithmische Grundlagen zur Audio-Signalklassifikation	7
2.1 Verarbeitungsschritte der Audio-Signalklassifikation	7
2.1.1 Merkmalsextraktion	8
2.1.2 Prozessierung	14
2.1.3 Klassifikation	15
2.2 Evaluation von Klassifikationsmethoden durch die Klassifikationsgüte . .	17
2.2.1 Optimierung von Klassifikationsmethoden	18
2.2.2 Musik-Datenbanken und Referenzen	19
2.3 Anwendungen der Audio-Signalklassifikation	20
2.4 Stand der Technik von Musikklassifikations- und Musikempfehlungssystemen	21
2.5 Stand der Forschung im Bereich der automatisierten Musikklassifikation .	23
3 Metriken zur effizienten Exploration des applikationsspezifischen Entwurfsraums	26
3.1 Kostenfaktoren und Metriken	26
3.1.1 Kostenfaktoren	26
3.1.2 Kombinierte Kosten-Metriken	27
3.2 Quantitative Erfassung der Kostenfaktoren	28
4 Klassifikationsmethoden zur Bestimmung von Durchsatzraten	30
4.1 Software-Framework für Analysen und Demonstrationen	30
4.2 Festlegung der für Musik geeigneten Klassifikationsmethoden	32
4.2.1 Klassifikationsmethode I (MCI)	33
4.2.2 Klassifikationsmethode II (MCII)	34

4.2.3	Klassifikationsmethode III (MCIII)	37
4.2.4	Klassifikationsmethode IV (MCIV)	38
4.3	Bewertung der Klassifikationsmethoden	39
5	Konzeption der Merkmalsextraktion mit Fixpunkt-Arithmetik	43
5.1	Fixpunktzahlendarstellung	43
5.2	Algorithmen zur Approximation algebraischer Operationen	45
5.2.1	Long Division	47
5.2.2	Nonrestoring-Verfahren zur Approximation der Quadratwurzel	49
5.2.3	BKM-Algorithmus zur Approximation des natürlichen Logarithmus	50
5.3	Methodik zur Festlegung von Fixpunktattributen	52
5.3.1	Implementierung der Fixpunkt-basierten Merkmalsextraktion	54
5.3.2	Wortlängenoptimierung	55
5.3.3	Methode zur Erstellung der Fehlerspezifikation	56
5.4	Evaluation der Fixpunktattribut-Festlegung	59
6	Architekturspezifische Optimierung der Musikklassifikation	62
6.1	General Purpose Prozessoren (x86)	63
6.1.1	Bewertung der Extraktionszeiten	64
6.2	Graphics Processing Unit (GPU)	64
6.2.1	Verwendung von GPUs für allgemeine Aufgaben	64
6.2.2	GPU-spezifische Optimierung	67
6.2.3	Anwendung des CUDA-Modells auf den Extraktionsprozess	72
6.2.4	Beschleunigte Extraktion der Merkmale	73
6.2.5	Bewertung der Extraktionszeiten	75
6.2.6	Kommunikationslatenz zwischen Host und GPU	76
6.3	RISC-basierte Architekturen	77
6.3.1	ARM-Cortex A8	78
6.3.2	Laufzeitanalyse der Verarbeitungsschritte	79
6.3.3	Analyse der FFT Extraktion	80
6.3.4	Bewertung der Extraktionszeiten	82
6.4	Digitale Signalprozessoren	82
6.4.1	TI C674x DSP	83
6.4.2	Optimierung der Merkmalsextraktion	85
6.4.3	Bewertung der Extraktionszeiten	86
6.4.4	Kommunikationslatenz im Betrieb in einer heterogenen Architektur	87
6.5	Applikationsspezifische Instruktionssatz Prozessoren	88
6.5.1	ARC600 Architektur	88
6.5.2	Anwendungsspezifische Optimierung des ARC600	90
6.5.3	Bewertung der Extraktionszeiten	99
6.6	Dedizierte Hardware-Architektur	99
6.6.1	Schnittstellen-Spezifikation	101

6.6.2	Pipeline FFT-Prozessor	102
6.6.3	Funktionseinheiten für mathematische Operationen	104
6.6.4	Design der Extraktionseinheiten am Beispiel der MFCC Merkmale	108
6.6.5	Sortier-Modul zur Extraktion des OSC Merkmals	110
6.6.6	Kommunikationslatenz: Konzepte zur Übertragung extrahierter Merkmale in externen Speicher	111
6.6.7	Verifikationsumgebung	113
6.6.8	FPGA- und ASIC-spezifische Implementierung	114
7	Evaluation des quantitativ erfassten Entwurfsraums	118
7.1	Architektur-spezifische Laufzeitevaluation der vollständigen Musikklassi- fikation	118
7.2	Bewertung der Architekturansätze zur beschleunigten Merkmalsextraktion	120
7.3	Energie- und Flächeneffizienz von Architekturen zur Merkmalsextraktion	123
7.4	Bewertung heterogener Architekturansätze	126
7.4.1	Laufzeiten heterogener Architekturen	127
7.4.2	ATE-Kosten heterogener Architekturen	129
7.5	Bewertung der Ergebnisse für Endbenutzer-Geräte	131
7.5.1	Trainingslaufzeiten	132
7.5.2	Laufzeiten bei Analyse von Musikdaten auf externen Speichermedien	133
7.5.3	Laufzeiten bei komprimierten Musikdaten	134
8	Anwendung des quantitativ erfassten Entwurfsraums für die Audio-Szenen- Erkennung	136
8.1	Bewertung Architektur-spezifischer Eigenschaften für die Audio-Szenen- Erkennung	136
8.2	Klassifikationsgüten bei der Audio-Szenen Erkennung	138
9	Zusammenfassung	141
A	Software-Framework	144
A.1	Algorithmen-Übersicht	144
A.2	Anwendungsbeispiele	145
A.3	Implementierte Klassifikationsmethoden	146
B	Messergebnisse des ARC600	147
B.1	Daten-Cache abhängige Laufzeiten des ARC600	147
B.2	Instruktions-Cache abhängige Laufzeiten des ARC600	148
B.3	Effizienzwerte des ARC600 bei verschiedenen Konfigurationen	149
C	Evaluationsergebnisse	150
	Literaturverzeichnis	154

Abkürzungsverzeichnis

Architektur-spezifische Abkürzungen:

AGU	<u>A</u> ddress <u>G</u> eneration <u>U</u> nit
ALU	<u>A</u> rithmetic <u>L</u> ogic <u>U</u> nit
AVX	<u>A</u> dvanced <u>V</u> ector <u>E</u> xtentions
ASIC	<u>A</u> pplication <u>S</u> pecific <u>I</u> ntegrated <u>C</u> ircuit
ASIP	<u>A</u> pplication <u>S</u> pecific <u>I</u> nstruction <u>S</u> et <u>P</u> rocessor
BUS	<u>B</u> inary <u>U</u> nit <u>S</u> ystem
CPU	<u>C</u> entral <u>P</u> rocessing <u>U</u> nit
DSP	<u>D</u> igitaler <u>S</u> ignal <u>p</u> rozessor
FPGA	<u>F</u> ield <u>P</u> rogrammable <u>G</u> ate <u>A</u> rray
FSM	<u>F</u> inite <u>S</u> tate <u>M</u> achine
GPP	<u>G</u> eneral <u>P</u> urpose <u>P</u> rozessor
GPU	<u>G</u> raphics <u>P</u> rocessing <u>U</u> nit
IBUS	<u>I</u> nter <u>n</u> et <u>B</u> US
MIB	<u>M</u> odule <u>I</u> nterconnect <u>B</u> us
RISC	<u>R</u> educed <u>I</u> nstruction <u>S</u> et <u>C</u> omputer
ROM	<u>R</u> ead <u>O</u> nly <u>M</u> emory
SIMD	<u>S</u> ingle <u>I</u> nstruction <u>M</u> ultiple <u>D</u> ata
SIMT	<u>S</u> ingle <u>I</u> nstruction <u>M</u> ultiple <u>T</u> hreads
SoC	<u>S</u> ystem-on- <u>C</u> hip
SM	<u>S</u> treaming <u>M</u> ultiprocessor
SPLOOP	<u>S</u> oftware <u>P</u> ipelined <u>L</u> OOP
TDP	<u>T</u> hermal <u>D</u> esign <u>P</u> ower
VLIW	<u>V</u> ery <u>L</u> ong <u>I</u> nstruction <u>W</u> ord

Audio-Signalklassifikationsspezifische Abkürzungen:

AMC	<u>A</u> mplitude of <u>M</u> aximum in <u>C</u> hromagram
FP	<u>F</u> rame <u>P</u> arallelisierung
FSET	<u>F</u> eature <u>S</u> ET
GMM	<u>G</u> aussian <u>M</u> ixture <u>M</u> odel
HAM	<u>H</u> AMming Fenster
HMM	<u>H</u> idden <u>M</u> arkov <u>M</u> odel
LEW	<u>L</u> ow <u>E</u> nergy <u>W</u> indow
MAG	<u>M</u> AGnitude of Spektrum
MC	<u>M</u> usic <u>C</u> lassification method (Klassifikationsmethode)
MFCC	<u>M</u> el <u>F</u> requency <u>C</u> epstral <u>C</u> oefficients
MP	<u>M</u> erkmal <u>P</u> arallelisierung

MSA	<u>M</u> odulation <u>S</u> pectral <u>A</u> nalysis
NASE	<u>N</u> ormalized <u>A</u> udio <u>S</u> pectrum <u>E</u> nvelope
OSC	<u>O</u> ctave <u>S</u> pectral <u>C</u> ontrast
RMD	<u>R</u> unning <u>M</u> ean and Running <u>D</u> eviation
RMS	<u>R</u> oot <u>M</u> ean <u>S</u> quare
SBER	<u>S</u> ub <u>B</u> and <u>E</u> nergy <u>R</u> atio
SCF	<u>S</u> pectral <u>C</u> rest <u>F</u> actor
SPC	<u>S</u> Pectral <u>C</u> entroid
SPF	<u>S</u> Pectral <u>F</u> lux
SPR	<u>S</u> Pectral <u>R</u> olloff
SV	<u>S</u> upport <u>V</u> ector
SVM	<u>S</u> upport <u>V</u> ector <u>M</u> achine
ZCR	<u>Z</u> ero <u>C</u> rossing <u>R</u> ate

Signalverarbeitungsspezifische Abkürzungen:

BM	<u>B</u> itonic <u>M</u> erge
CORDIC	<u>C</u> Oordinate <u>R</u> otation <u>D</u> igital <u>C</u> omputer
CUFFT	<u>C</u> UDA <u>F</u> ast <u>F</u> ourier <u>T</u> ransformation
DCT	<u>D</u> iskrete <u>C</u> osinus <u>T</u> ransformation
DFT	<u>D</u> iskrete <u>F</u> ourier- <u>T</u> ransformation
DIT	<u>D</u> ecimation- <u>I</u> n- <u>T</u> ime
DIF	<u>D</u> ecimation- <u>I</u> n- <u>F</u> requency
FFT	<u>F</u> ast <u>F</u> ourier- <u>T</u> ransformation (schnelle Fourier-Transformation)
KV	<u>K</u> reuz <u>V</u> alidierung
LSB	<u>L</u> east <u>S</u> ignificant <u>B</u> it
MSB	<u>M</u> ost <u>S</u> ignificant <u>B</u> it
R2MDC	<u>R</u> adix-2 <u>M</u> ulti-path <u>D</u> elay <u>C</u> ommutator
R2SDF	<u>R</u> adix-2 <u>S</u> ingle-path <u>D</u> elay <u>F</u> eedback

Sonstige Abkürzungen:

GUI	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface
HDD	<u>H</u> ard <u>D</u> isk <u>D</u> rive
ISMIR	<u>I</u> nternational <u>S</u> ociety on <u>M</u> usic <u>I</u> nformation <u>R</u> etrieval
MAD	<u>M</u> PEG <u>A</u> udio <u>D</u> ecoder
MIREX	<u>M</u> usic <u>I</u> nformation <u>R</u> etrieval <u>E</u> valuation e <u>X</u> change
SD	<u>S</u> ecure <u>D</u> igital
SDD	<u>S</u> olid <u>S</u> tate <u>D</u> rive
TI	<u>T</u> exas <u>I</u> nstruments

Notation

N	Audio-Samples pro Frame
K	Spektralkomponenten pro Frame
L_w	zeitliche Länge eines Frames
F_s	Sampling-Rate
X	Fourierspektrum
A	Betragsspektrum
w	Hamming-Fenster
M	Melbänder
A_{chroma}	Chroma Vektor
R	Klassifikationsgüte
A_S	Siliziumfläche
P	Verlustleistungsaufnahme
T	Zeit zur Berechnung eines Ergebnisses
$E_{Ergebnis}$	Energieverbrauch pro Ergebnis
$C_{A,Eff}$	Flächeneffizienz einer Architektur
$C_{E,Eff}$	Energieeffizienz einer Architektur
C_{ATE}	ATE-Produkt/-Kosten einer Architektur
$T_{heterogen}$	Berechnungszeit pro Ergebnis einer heterogenen Architektur
$C_{ATE,heterogen}$	ATE-Produkt/-Kosten einer heterogenen Architektur
R_E	Extraktionsrate
x_{spc}	SPC Wert eines Frames
x_{spr}	SPR Wert eines Frames
x_{spf}	SPF Wert eines Frames
x_{mfcc}	MFCC Werte eines Frames
x_{nase}	NASE Werte eines Frames
x_{osc}	OSC Werte eines Frames
x_{zcr}	ZCR Wert eines Frames
x_{rms}	RMS Wert eines Frames
x_{lew}	LEW Wert eines Frames
x_{scf}	SCF Werte eines Frames
x_{sber}	SBER Werte eines Frames
x_{amc}	AMC Wert eines Frames

Kurzfassung

Die inhaltsbasierte, automatisierte Klassifikation von Audio-Signalen gewinnt zunehmend an Bedeutung für den Einsatz in multimedialen Produkten und intelligenten Hörgeräten. Entsprechende Anwendungen führen eine Verknüpfung von digital gespeicherten Audio-Daten oder von Audio-Datenströmen mit passenden Bezeichnern durch. Dies wird zum Beispiel zur automatisierten Strukturierung von Musik-Datenbanken in Genres (z.B. Klassik, Rock, ...) genutzt. Außerdem wird mit denselben Methoden eine automatisierte Anpassung von Hörgeräte-Einstellungen an gegebene Szenarien und somit eine Verbesserung der Personensicherheit realisiert. Typische Ansätze der Audio-Signalklassifikation sind in bis zu drei Phasen unterteilt, die die Extraktion von charakteristischen Audio-Merkmalen und die Klassifikation von erstellten Merkmalsvektoren mit einbeziehen. Die Anforderungen derartiger Verfahren an zugrunde liegende Hardware-Architekturen sind vielfältig und besitzen einen komplexen Zusammenhang zueinander. Ein entsprechender Entwurfsraum von Hardware-Architekturen ist bisher nicht erfasst, so dass die Design-Phase geeigneter Systeme erschwert ist. Hieraus folgen erhöhte Produktionskosten, welche schließlich die Attraktivität dieser Verfahren senkt.

In dieser Arbeit wird eine quantitative Entwurfsraum-Exploration von Hardware-Architekturen für die Klassifikation von Audio-Signalen durchgeführt. Auf Basis der Musikklassifikation werden unterschiedliche, algorithmische Verfahren aus der Literatur und eigene Ansätze gewählt, die den Entwurfsraum im Bezug auf die Klassifikationsqualität und die Rechenkomplexität aufspannen. Für eine umfangreiche Untersuchung von Architektur-spezifischen Kostenmetriken wie der Verarbeitungszeit, der Energieeffizienz, der Flächeneffizienz und dem Fläche-Zeit-Energie-Produkt werden die algorithmischen Methoden für GPP, DSP, GPU, ASIP, FPGA und ASIC Architekturen optimiert. Außerdem werden heterogene Architekturansätze betrachtet, welche in Bezug auf die festgelegten Metriken besonders geringere Kosten aufweisen können.

Die Forschungsergebnisse unterstützen zukünftige Design-Phasen geeigneter Systeme zur Audio-Signalklassifikation. So wird gezeigt, dass aktuelle Architekturen ausreichende Rechenkapazitäten zur Auswertung von Musikdaten bieten, da die gemessenen Verarbeitungszeiten deutlich kürzer als die Abspiellängen der Musikinhalte sind. Anhand des explorierten Entwurfsraums wird außerdem deutlich, dass sowohl für mobile als auch für stationäre Systeme geeignete Hardware-Architekturen existieren, die den heutigen Anforderungen an Energiekosten gerecht werden. Schließlich wird nachgewiesen, dass der untersuchte Entwurfsraum auch Architekturen inkludiert, die die Echtzeitanforderungen von Hörgeräten bei möglichst geringem Leistungsverbrauch erfüllen.

Schlagnworte: Entwurfsraum-Exploration, Hardware-Architekturen, Endbenutzer-Geräte, Inhaltsbasierte Musikklassifikation, Audio-Szenen-Detektion

Abstract

An increasing interest in the content-based classification of audio signals exists. Such methods improve the usage of multimedia devices and hearing aids as they assign suitable labels to digitally stored audio data or to audio data streams. As application examples, the audio signal classification allows an automated clustering of music databases into genres (e.g. classical, rock, ...) and enables an automated adjustment of parameter settings for hearing devices in order to increase an user's safety.

Audio signal classification methods are typically subdivided into three phases that include the extraction of audio features and the classification of generated feature vectors. For a reasonable product integration, various hardware architecture requirements must be respected that possess complex relations to each other. A corresponding design space of hardware architectures is not determined up to now. Thus, the design phase of suitable devices, that support audio signal classification methods, implies high production costs which decreases the attractiveness of such applications.

In this work, a design space exploration of hardware architectures for the classification of audio signals is performed. Based on the music classification as case of study, different algorithmic approaches, that are selected from literature or are self-defined, are respected to cover the design space in terms of classification quality and runtime complexity. The computation time, the energy and area efficiency, and the area-time-energy product are applied as cost metrics for the evaluation of application optimized GPP, DSP, GPU, ASIP, FPGA, and ASIC architectures. Furthermore, heterogeneous architectures are respected as such approaches can correspond to especially cost attractive designs.

The research results enhance the design phase of future devices that support the classification of audio signals. Thereby, recent hardware architectures are demonstrated to provide sufficient computation power for rapid music classifications. Moreover, the covered design space includes several architecture designs that comply the demands of today's energy costs of both static and mobile devices. Finally, the explored design space additionally allows the identification of suitable architectures that can perform auditory scene recognition methods at the design restrictions of typical hearing aids.

Keywords: design space exploration, hardware architectures, end consumer devices, content-based music classification, auditory scene recognition