

Prüfungsunterlagen zur Vorlesung

Medientechnik

Norman Hendrich

Universität Hamburg
Fachbereich Informatik
Vogt-Kölln-Str. 30
D 22 527 Hamburg
hendrich@informatik.uni-hamburg.de

Inhaltsverzeichnis

Einführung

Literatur	3
Begriffe Medium, Multimedia	4
Anforderungen, Informatikaufgaben	5
Systemintegration, SoC	7
Wearables, Miniaturisierung	9

Datenkompression

Kriterien, CODEC-Prinzip	11
RLE	12
Huffman	14
LZW	15
BWT	17
Wavelets	18
Wavelets	20

Audio

Aufbau des Gehörs	21
Hörschwelle	22
Lautheit, Maskierung	23
Frequenzgruppen, Klangfarben	24
Raumwahrnehmung, HRTF	25
Sprache	26
Sprache	27

Einschub: Digitale Signalverarbeitung

Audio-Anwendungen	29
Impulsfolgen	30
Direktform	31
Abtastung	32
A/D und D/A-Wandlung	33
Dithering	35
Dithering	36

Audio-Datenkompression

CDDA- und Wav-Format	37
ADPCM	38
Sprachkodierung; G.711	39
CELP, GSM	40
CELP, GSM	41

Psychoakustische Kodierung

MPEG-1	43
MPEG-1	44
MPEG-1 Filterbank	44
MPEG-1 Encoder und Bit-Allokation	46
MPEG-1 Bitstrom	48
MPEG-1 Bitstrom	50
MPEG-1 Pre-Echos	51
MPEG-1 Klangvergleich	53

Images

Auge, Retina	55
Auge, Retina	56
Farbmodelle	57
Rasterbilder, Dithering	59
Bildformate	61
GIF	61
GIF	62
JPEG	64
JPEG	64
JPEG-2000	69
Filter	71
Vektorgraphik	72

Video	73
Fernsehen	74
PAL-Verfahren	75
Grabbing	76
Digitale Videoformate	77
AVI, ASF	78
MPEG-Übersicht	79
MPEG-1	80
MPEG-2	82
H.26x	84
MPEG-4	85
DV	87
Multimedia Home Platform	88
Animationen	89
Haptische Interfaces	90
Tast Sinn und Wahrnehmung	91
Gerätespektrum	93
Datenhandschuh	95
Force-Feedback und DirectX	98
Immersion Studio	100
I/O-Geräte	102
CCD-Bildsensoren	102
LCD-Displays	104
Weitere Display-Technologien	106
Drucker	107
Rechnerarchitektur	108
Anforderungen	108
Andahl's Gesetz, RISC-Prinzip	110
Flynn-Klassifikation	111
Superskalare Prozessoren	112
Speicherhierarchie	113
Cache-Organisation	114
ARM-Architektur	117
ARM-Befehlssatz	118
ARM-Erweiterungen	120
ARM-Jazelle	122
StrongARM, XScale	123
SIMD-Befehlsätze	125
MMX	125
MMX-Beispiele	128
3Dnow!	129
SSE	132
SSE2	134
MAJC	135
Piranha Multiprozessor	136
iFlow Netzwerkprouzessor	137
Optische Datenspeicherung	139
Prinzip der Compact Disc	139
Reed-Solomon Fehlerkorrektur	142
CD-Varianten	143
Datenformate	144
CD-Datensysteme	146
CDR, CDRW	147
UDF	150
DVD	151
DVD-Video	153
DVD-Recordable, DVR	155

Frameworks

Grundfunktionen 157

Streaming 158

RTP, RTCP 159

Windows Media 160

AVI und ASF 161

DirectShow, FilterGraph 162

Java Media Framework 164

JMF Klassenhierarchie 166

JMF Codebeispiele 168

JMF Codebeispiele 171

MPEG-7

Konzept und Anwendungen 172

Document Description Language 173

Beispiel Audio-Datenbanken 174

Beispiel Video-Datenbanken 177

VIBE Videodatenbank 179

VIBE Szenenerkennung 181

VIBE Szenenerkennung 182

VIBE-Navigation 184

MPEG-4 Audio

Audio-Codecs, Bitraten und Qualität 185

Verfahren zur Klangsynthese 186

Structured Audio 187

SAOL 189

SASL und MIDI 190

SA-Tools 192

SA vs. VST 193

SASBF 194

SA vs. VST 195

Audio-BIFS 196

Miniaturisierung

CMOS-Technologie und Skalierung 197

CMOS-Stromverbrauch 198

Portable vs. Wearable 199

Grenzen der Miniaturisierung 200

Mikrosysteme 201

Mikromotoren und -getriebe 202

Smartdust 203

Smartdust 205

Vorlesung

Medientechnik

(Multimedia Systems)

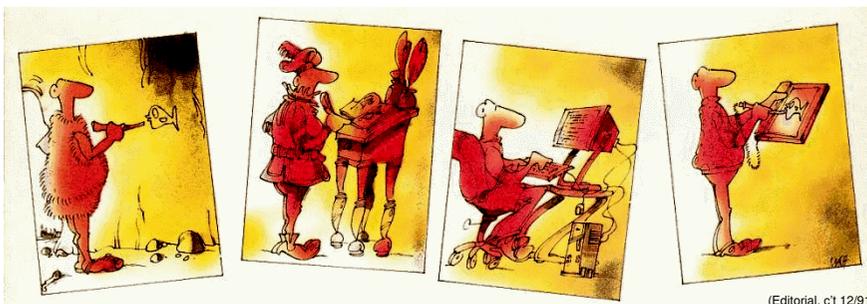
Norman Hendrich

Universität Hamburg, Fachbereich Informatik, TECH

<http://tech-www.informatik.uni-hamburg.de/lehre/ws2001/medientechnik/>

Medientechnik | WS 2001 | 18.204

digitale Medien . . .



(Editorial, c1 12/91)

Welche Vorteile haben "digitale" Medien?

- beliebig gute Qualität, keine Alterung, perfekt kopierbar
- Flexibilität, jederzeit editierbar
- Kombination aller Modalitäten: Auge, Ohr, Tastsinn, Geschmack, ...

Medientechnik | WS 2001 | 18.204

Themen und Termine

- jeweils Do, 12 - 14, Raum F334
- Einführung, Begriffe, Aktuelles
- Datenkompression und Kodierung
- Audio: Musik, Sprache, MIDI
- Bilder: Vektorgraphik, GIF, JPEG, ...
- Video: analog, MPEG-1/-2, H.26x
- Medien-Frameworks, JMF
- Medienorganisation und -suche, MPEG-4/-7/-21
- Rechnerarchitekturen, Medienbefehlsätze
- Datenspeicherung, CD/DVD/DVR
- Datenübertragung, Streaming, Modems
- I/O-Geräte: Displays, Haptik, ...
- mobile Geräte und Miniaturisierung

Termin	Termin
25.10	01.11
08.11	15.11
22.11	29.11
06.12	13.12
20.12	---
11.01	18.01
25.01	01.02
08.02	

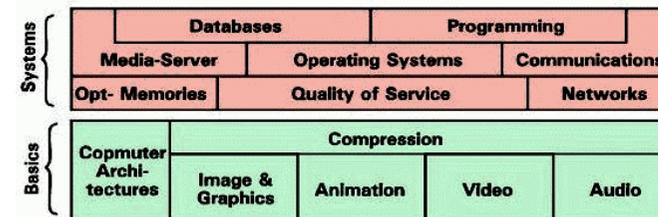
Medientechnik | WS 2001 | 18.204

"Medientechnik": bessere Titel?

im Sinne dieser Vorlesung:

Medientechnik :=

- Multimedia-Technologie
- Multimedia systems
- Studienprofil "Interaktive Medien":
- Studienprofil "Technikorientierte Informatiksysteme"



Medientechnik | WS 2001 | 18.204

Inhalt und Lernziel

- Welche Medien gibt es?
- Wie werden Mediendaten verarbeitet?



Welche Anforderungen ergeben sich daraus:

- Performance
- Speicherbedarf
- Datenkompression und Datenformate
- I/O-Geräte, neue Konzepte dafür
- Rechnerarchitektur
- usw. ...
- Kennenlernen wichtiger Konzepte und Algorithmen
- Fähigkeit zum Einschätzen zukünftiger Entwicklungen
- Chancen und Grenzen der Miniaturisierung



Medientechnik | WS 2001 | 18.204

Studienprofil: interaktive Medien

Der Rahmenstudienplan wird erweitert um ein fünftes Studienprofil "interaktive Medien" mit folgenden Angaben:

[... Ergänzungsfächer, Grundlagenfächer, Schwerpunkte, usw.]

... schlägt das Curricularteam die Einrichtung vorerst nur eines Schwerpunkts im Profil vor, um die Anlaufschwierigkeiten zu minimieren. Eine fachliche Ausdifferenzierung kann erfolgen, wenn eine gewisse Konsolidierung des Angebots erreicht wurde.

... Studienprofile dienen der Orientierung der Studierenden. Das Curricularteam sieht daher die Notwendigkeit zur Etablierung eines überschaubaren Studienangebots, das dennoch die fachliche Breite des Faches angemessen widerspiegelt. Gegenüber den derzeitigen Angeboten in den bestehenden Schwerpunkten, sind für die Zwecke des neuen Studienprofils vorrangig Lehrveranstaltungen mit großer fachlicher Breite, aber geringerer Tiefe zu konzipieren.

(Uni HH, FB Informatik, 9/2000)

Medientechnik | WS 2001 | 18.204

Studienprofil: interaktive Medien

CGB, 1 aus GBI, DOS
STE, 1 aus DIS, SNN, VSS, WBS
DKR
LOS, PNL, 1 aus AUK, STH

angewandte Informatik
praktische Informatik
technische Informatik
theoretische Informatik

Interaktionsdesign
Entwurf von Hypermediasystemen
Computergestützte kooperative Arbeit
Bildverarbeitung
Audioverarbeitung
Textanalyse
Virtuelle Realität
Verteilte Systeme für interaktive Medien
Netzsicherheit, Kryptologie
Multimedia-Datenbanken

Oberquelle
Scheffé
Oberquelle, Floyd
D.-Fischer, Neumann, Stiehl
Menzel, v.d.Heide, Wolfinger
Menzel, Habel
Mertsching, Möller
Lamersdorf
Brunnstein, Mück
NN (GRIS)
NN (TECH)

=> **Medientechnik**

Medientechnik | WS 2001 | 18.204

große Breite, geringe Tiefe :-)

oberflächlich:

- physiologische Grundlagen
- Signalverarbeitung
- Graphik- / Bild- / Sprachverarbeitung / ...
- Netzwerke / QoS / ...
- Applikationen

Themen mit einiger Tiefe:

- Datenformate, -kompression
- Rechnerarchitektur für Medienverarbeitung
- Geräte, Miniaturisierung, Ausblick auf Mikrosysteme
- Audiokodierung, MPEG-4 structured audio
- Videokodierung, MPEG bzw. H.26x

Medientechnik | WS 2001 | 18.204

Literatur: Skript

Material zur Vorlesung:

- kein ausformuliertes Skript
 - aber vollständige Folienkopien
 - Vervielfältigung via ZVV (wenn fertig)
-
- aktueller Stand jeweils auf dem TECH Webserver
 - dort auch Links und Medienbeispiele
- tech-www.informatik.uni-hamburg.de/lehre/ws2001/medientechnik/

Feedback ist erwünscht:

- Hinweise, Vorschläge, Korrekturen
- Hinweise auf gute Medienbeispiele / - samples

Medientechnik | WS 2001 | 18.204

Literatur: Bücher

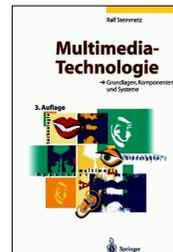
Ralf Steinmetz, Multimedia-Technologie,
3-540-67332-6, Springer Verlag, 2000
(siehe auch www.kom.e-technik.tu-darmstadt.de)

Peter A. Henning, Taschenbuch Multimedia
3-446-21751-7, Fachbuchverlag Leipzig, 2001

Arbeitsumfeld "Medien":

Martina und Wolfgang Jäger, Studienführer Neue Medien,
3-933180-62-7, Lexika Verlag

Medien-Handbuch Hamburg, über 9000 Adressen für Film, Funk, ...
3-932645-12-X



Medientechnik | WS 2001 | 18.204

Literatur: weitere Bücher



J.L.Hennessy, D.A.Patterson, Computer Organization & Design, Morgan Kaufmann, 1998

D.A.Patterson, J.L.Hennessy, Computer Architecture, Morgan Kaufmann, 1996

Steve Furber: ARM System-on-Chip Architecture, Addison Wesley, 2001

D.Salomon, Data compression, Springer 2000

R.Sedgewick, Algorithms in C++, Addison Wesley, 1998

B.Bargen,Peter Donnelly, Inside DirectX, Microsoft Press, 1998

Udo Zoelzer, Digitale Audiosignalverarbeitung, Teubner, 2000

Bob Hughes, Dust or Magic, Secrets of Multimedia Design, Addison Wesley, 2000

Medientechnik | WS 2001 | 18.204

Literatur: WWW

MIT Media Lab: www.media.mit.edu
MPEG-Infos: www.cselt.it/mpeg/
Berkeley Smart Dust: robotics.eecs.berkeley.edu/~pister/SmartDust/

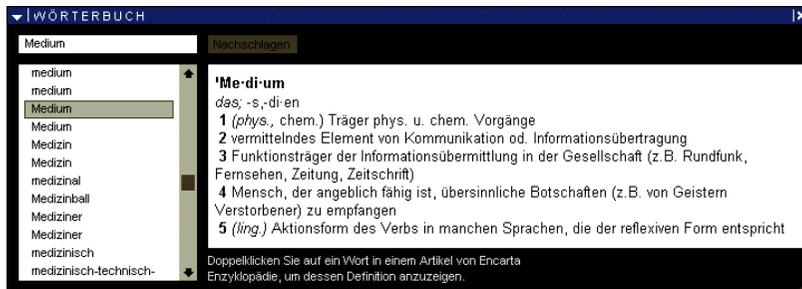
Microsoft MSDN Library msdn.microsoft.com/library/default.asp
Java Media Framework www.javasoft.com/products/java-media/jmf/

Suchmaschinen: www.google.com
Mediensuche (experimentell): www.altavista.com, www.google.com

spezielle Links zu den einzelnen Themen später

Medientechnik | WS 2001 | 18.204

Medium: Definition(en)

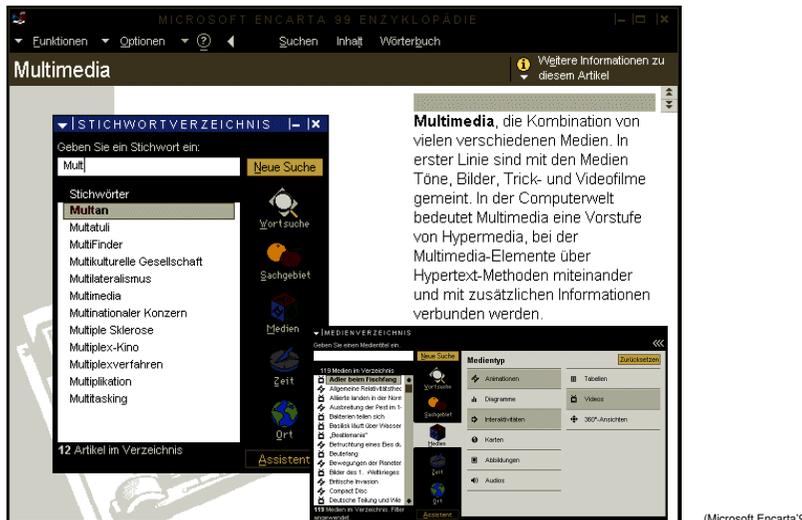


- "Ding in der Mitte"
- "anything that you can use to give somebody else an experience of some kind, in your absence" (Bob Hughes)
- also im weiteren Sinne: das "Design" für jedes Produkt

(Microsoft Encarta'99)

Medientechnik | WS 2001 | 18.204

"Multimedia"



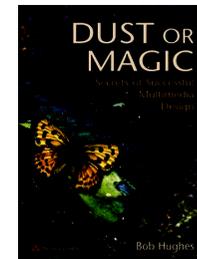
(Microsoft Encarta'99)

Medientechnik | WS 2001 | 18.204

"Multimedioricity"

"Multimedia":

- das "M-Wort"
- oft als Selbstzweck
- und nicht automatisch gut
- traditionelle Medien oft effizienter
- und benutzerfreundlicher (!)



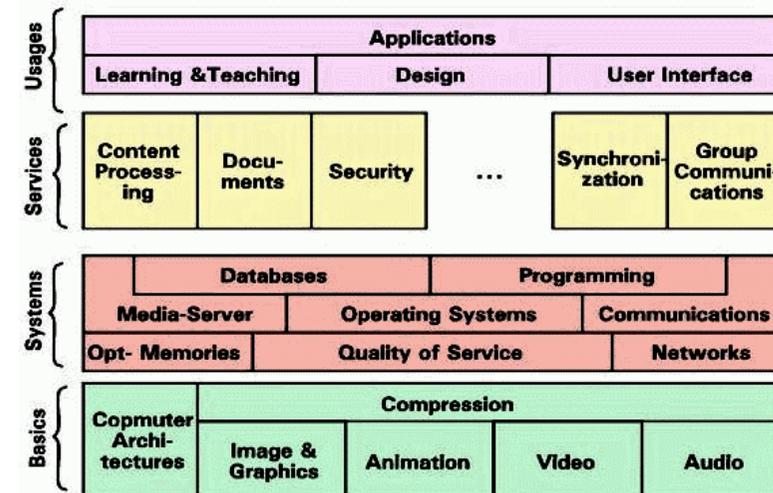
vgl. "Multimedia CD-ROMs" mit guten Büchern...
vgl. Philips/Sony CD-I Disaster...

=> Interaktionsmöglichkeiten unbedingt notwendig

- interessante Probleme und Lösungen ...
- aber nicht in dieser Vorlesung

Medientechnik | WS 2001 | 18.204

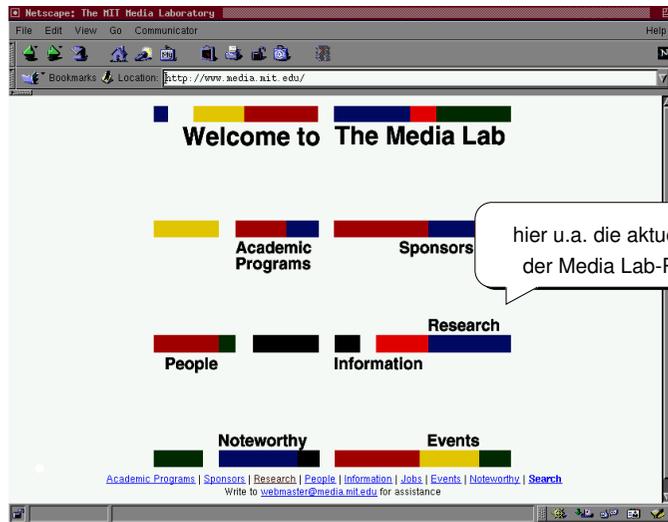
"Multimedia": Informatik-Themen



(R.Steinmetz, MM systems)

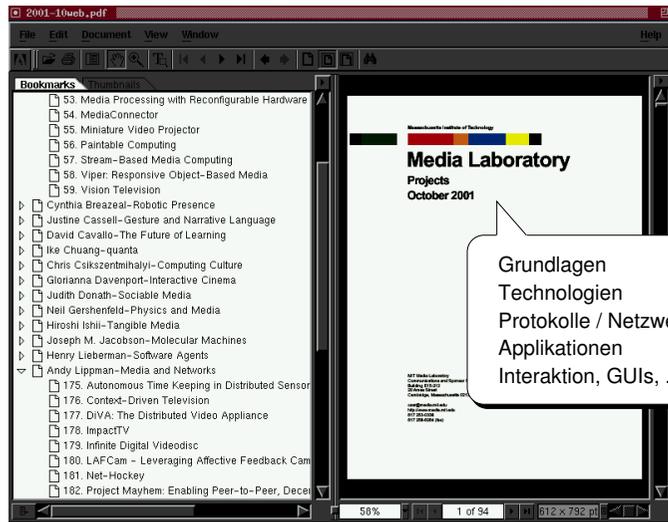
Medientechnik | WS 2001 | 18.204

MIT Media Lab



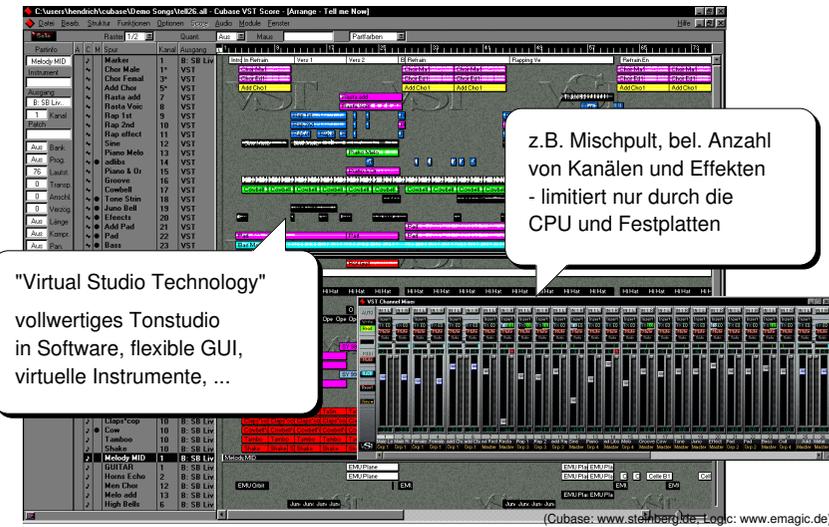
hier u.a. die aktuelle Liste der Media Lab-Projekte

MIT Media Lab: Projects



Grundlagen Technologien
Protokolle / Netzwerke
Applikationen
Interaktion, GUIs, ...

Applikationsbeispiel: Cubase VST



z.B. Mischpult, bel. Anzahl von Kanälen und Effekten - limitiert nur durch die CPU und Festplatten

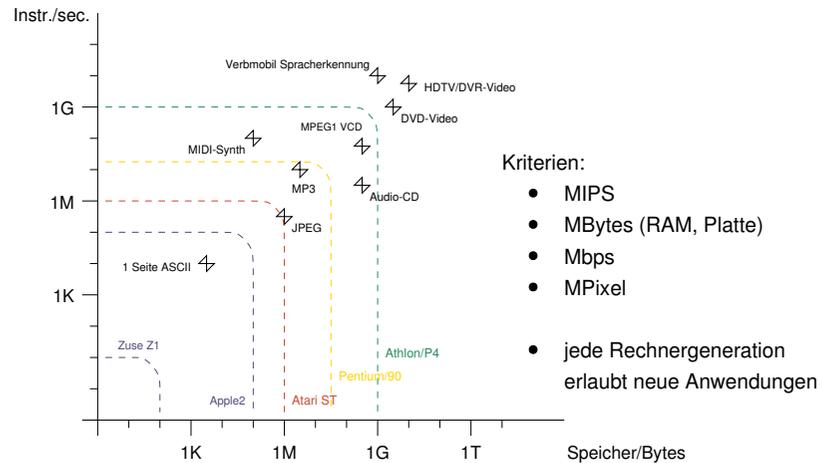
"Virtual Studio Technology" vollwertiges Tonstudio in Software, flexible GUI, virtuelle Instrumente, ...

(Cubase: www.steinberg.de; Logic: www.emagic.de)

Medien: Rechenleistung ...

- Medienverarbeitung erfordert schnelle und grosse Rechner:
- DVD-Wiedergabe, Einzelbilder 720x576 Pixel, 24 fps:
ca. 10M Pixel/sec, ca. 100 Befehle/Pixel
=> 1 GOPS
- 3D-Graphik, 100.000 Polygone, 50 fps
300.000 Ecken, je 16 MAC pro Ecke
=> 500 MFLOPS für die Koordinatentransformation
1024x768 Pixel, 50 fps, 10 Zugriffe/Texel, Overdraw 5:
=> 2 GOPS für Texturberechnung
- Stereo-Hall, Halfbuhne mit 1000 Reflexionen
2 Kanäle, 44100 Samples/sec., 10 MACs pro Reflexion:
=> 1 GOPS
- usw...

Medien: Anforderungen



Medientechnik | WS 2001 | 18.204

Informatikbezug: Algorithmen

Algorithmen zur Medienverarbeitung? in allen Bereichen:

- effiziente Datenkompression (MPEG, H.26x, GSM)
- Ausnutzen physiologischer Grenzen (MP3, JPEG)
- Multimedia-Datenformate (AVI, QT, ...)
- Beschreibungssprachen (VRML, ...)
- Medien-Indizierung, -Suche (MPEG-7, ...)

und natürlich die einzelnen Teilthemen:

- Text, Graphik (2D/3D), Audio, Video
- Datenübertragung, Netzwerke, Streaming, ...
- Sprach- / Gestenerkennung, neue Benutzerschnittstellen, ...
- kooperatives Arbeiten, Design, ...
- siehe Studienprofil!

Medientechnik | WS 2001 | 18.204

Informatik: Rechnerarchitektur

besondere Merkmale von Medienverarbeitung:

- sehr hohe Datenmengen (MB/s .. GB/s)
- einzelne Samples klein (8 bit, 16 bit)
- kaum Lokalität, Caches nutzlos (Sample einmal benutzt)
- oft hohe Parallelität
- Signalverarbeitung, MAC-Operationen
- oft besondere Arithmetiken notwendig (Sättigung, Rundung)

=> "normale" Rechner und Workstations nicht optimal

=> erfordert angepasste Rechnerarchitektur

- Medienbefehlssätze, Streaming (MMX, SSE, ...)
- Signalprozessoren, Spezialchips (z.B. MPEG-Encoder)

Medientechnik | WS 2001 | 18.204

Informatik: Systementwurf

exponentiell steigende Hochintegration (Moore's Gesetz):

- erlaubt Miniaturisierung (Mobilgeräte)
- erlaubt ganze Systeme auf einzelnen Chips
- CPU + ASICs + Speicher + I/O
- extrem kurze Marktzyklen, Time-to-Market Problem
- Softwareanteil (=Informatik) solcher Systeme nimmt extrem zu



- erfordert entsprechend kurze Entwurfszeiten
- für Hardware, Systemsoftware, Applikationen, Systemintegration
- und zwar fehlerfrei, weil nicht korrigierbar...

- Co-Design, Co-Simulation, Software für eingebettete Systeme
- Entwurf wird zunehmend zum Engpaß

Medientechnik | WS 2001 | 18.204

Multimedia-Systeme:



- einige aktuelle "MM"-Systeme...

Systeme: Eigenschaften



Prozessor	4 .. 32 bit	8 bit	-	16 .. 32 bit	32 bit	32 bit	32 bit	8 .. 64 bit	..32 bit
Speicher	1K .. 1M	< 8K	< 1K	1 .. 64M	?	< 128 M	8 .. 64M	1 K .. 10 M	< 64 M
ASICs	1 uC	1 uC	1 ASIC	1 uP	DSPs	1 uP,	1 uP,	~ 100 uC,	uP,
				ASIP		3 DSP	DSP	uP, DSP	ASIP
Netzwerk	cardIO	-	RS232	diverse	GSM	MIDI	V.90	CAN,...	I2C,...
Echtzeit	nein	nein	soft	soft	hard	soft	hard	hard	hard
Safety	keine	mittel	keine	gering	gering	gering	gering	hoch	hoch

- riesiges Spektrum: 4 bit .. 64 bit CPUs, DSPs, digitale/analoge ASICs, ...
- Sensoren/Aktoren: Tasten, Displays, Druck, Temperatur, Antennen, CCD, ...
- Echtzeit-, Sicherheits-, Zuverlässigkeitsanforderungen

Moore's Law: Lithographie, Hochintegration

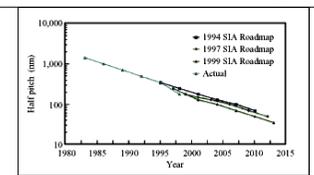


Figure 1
Historical and future trends of lithographic resolution capability. Here, half pitch is the minimum size of lithographic features on a chip. (SIA – Semiconductor Industry Association.)

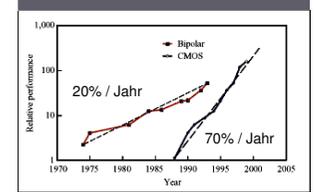


Figure 6
Historical and future server performance trends using bipolar and CMOS circuits. The straight lines represent the time-averaged exponential improvement in the performance of the technology.

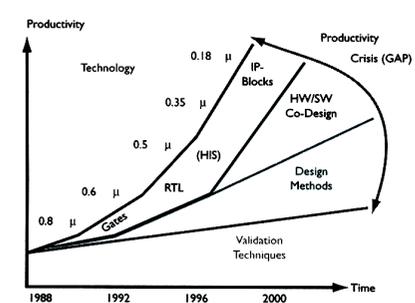
- exponentielles Wachstum
- seit 1970, bis > 2015
- durch bessere Lithographie

CMOS gewinnt über ECL:

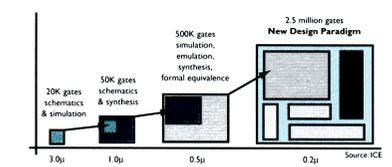
- langsame Xtors, Hochintegration
- schnelle Xtors, Abwärmeproblem
- auch die IBM-Grossrechner jetzt mit CMOS-Mikroprozessoren:
- bessere Performance
- geringer Leistungsverbrauch
- deutlich weniger Platzbedarf
- "Revolution": < 5 Jahre (!)

(R.D.Isaac, Future of CMOS technology, IBM JRD&D 44-3, 2000)

SoC: systems on a chip



Annual microelectronics productivity increase is only 21% compared with an annual complexity increase of 58%

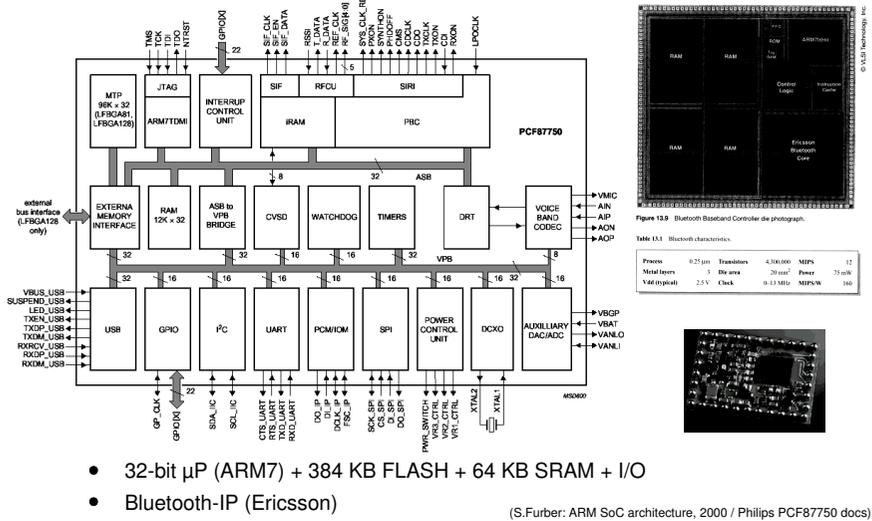


From VLSI design to SoC design – System Level Integration. Micron (μ) indicates the minimum geometry that can be implemented (1.0μ corresponds to $10^{-6}m$)

- zunehmend Einsatz von IP/Cores notwendig
- Entwurf wird zum limitierenden Faktor

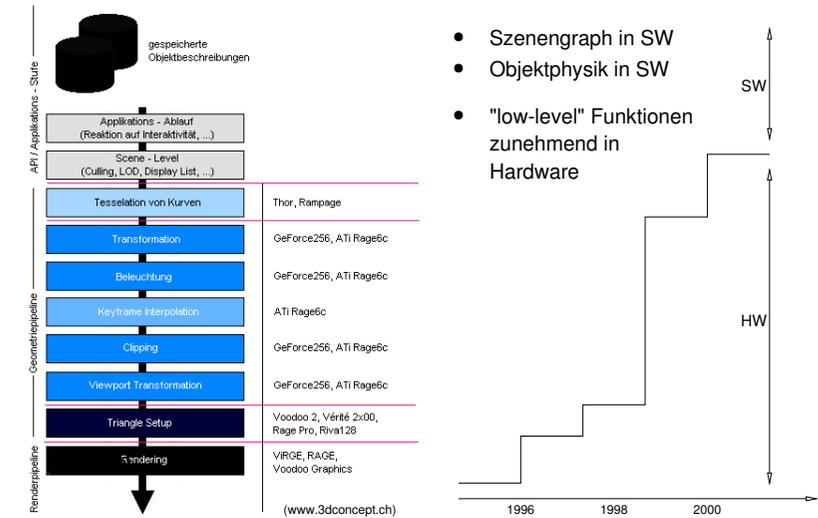
(Abbildungen: TIMA EUROCHIP newsletter, 04/2001)

SoC Beispiel: Philips Bluetooth IC



- 32-bit μ P (ARM7) + 384 KB FLASH + 64 KB SRAM + I/O
 - Bluetooth-IP (Ericsson)
- (S.Furber: ARM SoC architecture, 2000 / Philips PCF87750 docs)

Hardware vs. Software: 3D-Graphik



Softwareentwicklung: POSE



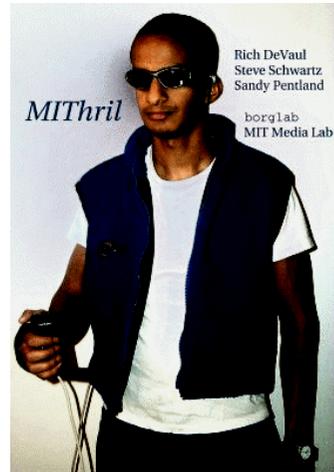
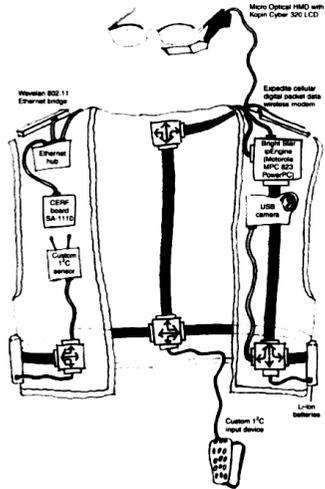
- Palm PDAs: Motorola "Dragonball"-CPU: 32 MHz 68000, div. I/O
 - Programmentwicklung mit Crosscompiler
 - Programmtest auf dem Emulator: schneller als das Original...
 - "high-level" Systemmodell auf OS-Ebene, ohne Timing
- (www.palmos.com/developer/)

Miniaturisierung . . .



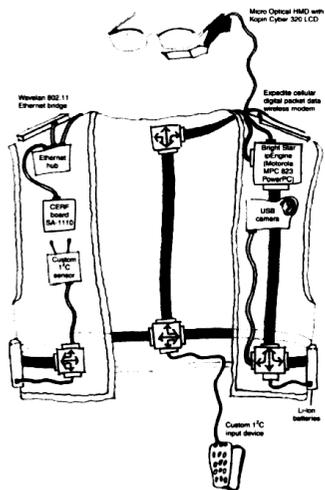
- Sprachqualität, Komfort
- Gewicht, Grösse, Laufzeit, ...
- MIPS / Watt

Wearable Computers: MITHril



(IEEE Micro 06/2001, www.media.mit.edu/wearables/mithril)

MITHril: Komponenten



StrongARM CPU, 206 MHz
 mit Chipsatz (Intel SA1110/1111)
 128 MByte DRAM
 2 MB Flash, 1 GB Microdrive
 TMS 320VC33 DSP

Schnittstellen:
 "MicroOptical" Displaybrille
 Mikrophon, Lautsprecher
 2 Kameras, 256x256 Pixel
 IrDA, Netzwerk, Modem, I2C
 Tastatur in die Weste integriert
 oder separat

(IEEE Micro 06/2001, www.media.mit.edu/wearables/mithril)

Wearables: Generator im Schuh . . .

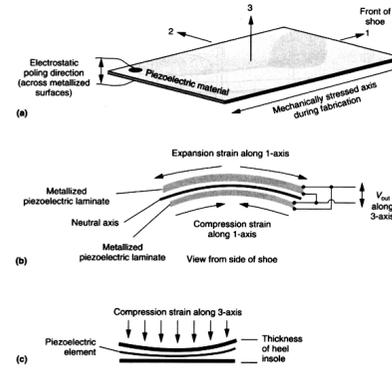


Figure 1. Conventional axis definition for a piezoelectric material (a). Our applications rely on 31-mode (b) in bending. The 33-mode (c)—when the heel motion in the 3-axis direction compresses the shoe's sole and induces an electric field along the same axis—is intuitively appealing but inefficient because of the small integrated strain across the thickness dimension (3-axis).

- liefert beim Gehen ca. 1 Watt
- Entwicklung noch in den "Kinderschuhen"

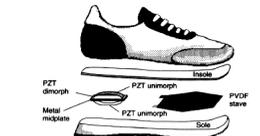


Figure 2. Two approaches to unobtrusive 31-mode piezoelectric energy scavenging in shoes: a PVDF stave under the ball of the foot and a PZT dimorph under the heel.

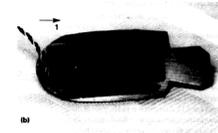
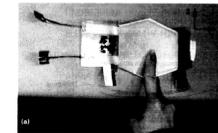


Figure 3. View looking down at the PVDF insole stave (a) and the PZT dimorph heel insert (b). In both views, the right side of the device is pointing to the shoe's toe.

(IEEE Micro, 6/2001)

Wearables: Spannungsversorgung . . .

Beispielschaltung des Generators:

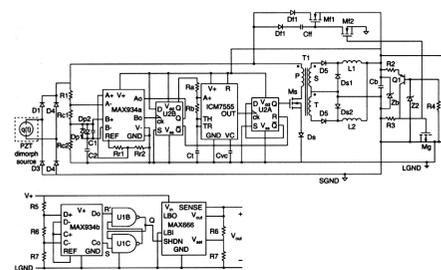


Figure 8. Forward-switching converter system.

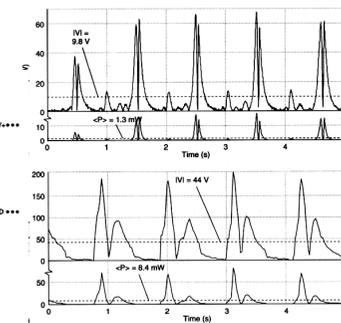
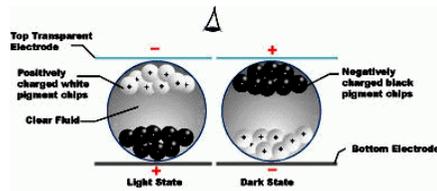
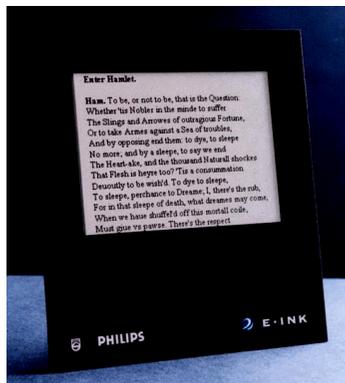


Figure 4. Power and rectified voltage waveforms from brisk-walking tests of optimally loaded PVDF stave (a) and PZT dimorph (b).

- ungünstige Spannungsspitzen
- viel System-Know-How erforderlich
- nur "digital" reicht nicht (!)

(IEEE Micro, 06/2001)

Miniaturisierung: eInk / ePaper:



(Philips/eInk Prototyp, 80 dpi, Juni 2001)

- schwarz/weiß gefüllte/gefärbte Kugeln auf Trägermaterial
- Ansteuerung wie LCD (passiv oder TFT)
- aber metastabil: daher stromsparend

(www.gyriconmedia.com, www.eink.com)

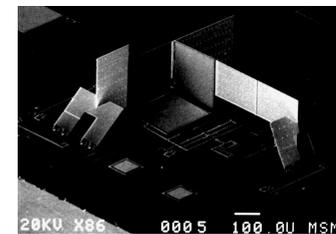
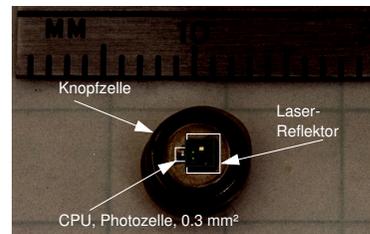
Miniaturisierung: ePaper



- oder einfach ausdrucken...

(www.parc.xerox.com/dhl/projects/gyricon/)

Miniaturisierung: Smart Dust



Berkeley "Smart Dust" Projekt:

- autonome Rechensysteme in 1mm³
- automone Vernetzung - via Funk oder Laser
- integrierte Sensoren
- Grenzen der Miniaturisierung ?!



(robotics.eecs.berkeley.edu/~pister/SmartDust/)

- Leerseite

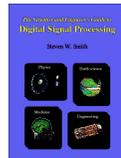
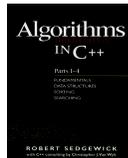
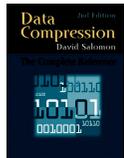
Datenkompression

- Übersicht
- Literatur
- Klassifikation:

ausgewählte Verfahren:

- Lauflängenkodierung
- Huffman Kodierung, arithmetische Kodierung
- Lempel-Ziv Kodierung und Varianten
- Burrows-Wheeler Transformation
- Wavelets
- weitere Verfahren später (Audio, Image, Video)

Datenkompression: Literatur



David Salomon, Data Compression - The Complete Reference, Springer 2000

Robert Sedgewick, Algorithms in C++, Addison-Wesley, 1998

Stephen W. Smith, Guide to Digital Signal Processing, California Technical Publ., 1997

online als PDF verfügbar: www.dspguide.com

Mark Nelson, Data Compression with the Burrows-Wheeler Transform,

Dr. Dobbs Journal, 9/1996

F. Bauernöppel, Verfahren und Techniken zur Datenkompression, c't 10/1991

M. Tamm, Packen wie noch nie - BWT-Algorithmus, c't 16/2000, 194

Ad-Hoc Kodierung: Morse, Braille, ...

A	..-.	N	-. -.	1	.-.-.-	Period	.-.-.-
B	-...-	O	---	2	..-.-	Comma	..-.-
C	-.-.-	P	-. -. 3	3	..-.-	Colon	..-.-
Ch	-.--	Q	-. -. 4	4	..-.-	Question mark	..-.-
D	..-.-	R	-. -. 5	5	..-.-	Apostrophe	..-.-
E	..-.-	S	...-	6	..-.-	Hyphen	..-.-
F	..-.-	T	-. -. 7	7	..-.-	Dash	..-.-
G	..-.-	U	..--	8	..-.-	Parentheses	..-.-
H	V	..-. 8	8	..-.-	Quotation marks	..-.-
I	..--	W	..-. 0	0	..-.-		
J	..-.-	X	-. -. 1	1	..-.-		
K	-.-.-	Y	-. -. 2	2	..-.-		
L	-.-.-	Z	..-- 3	3	..-.-		
M	--..						

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Table 1.1: The 26 Braille Letters.

and	for	of	the	with	ch	gh	sh	th
-----	-----	----	-----	------	----	----	----	----

Table 1.2: Some Words and Strings in Braille.

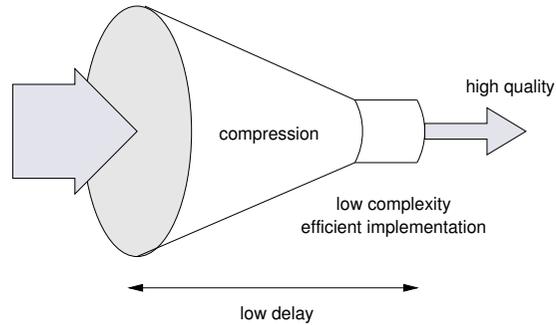
If the duration of a dot is taken to be one unit, then that of a dash is three units. The space between the dots and dashes of one character is one unit, between characters is three units, and between words six units (five for automatic transmission). To indicate that a mistake has been made and for the receiver to delete the last word, send "." (eight dots).

- zwei wichtige Beispiele für "frühe" Codes
- Morse: häufige Symbole ("e") mit kurzen Codes
- Braille: alle 64 Codes verwendet, keine Fehlererkennung

Multimedia: Datenmengen

Medium:	Datenmenge:	Komprimierung:	
Text	80 x 60 Zeichen 4.8 KB	Huffman 1:2 PKZIP 1:3	
Bild	640x480x3 = 900 KByte	Fax 1:10 JPEG 1:15	
Grafik	500 Linien		
Sprache	1 Kanal, 8Khz 64 Kbps, 480 KB/min	GSM 1:8	
MIDI	15 KB/min		
Musik	stereo, 44.1 KHz 1.44 Mbps, 10 MB/min	MP3 1:10	
Video	640x480x3x25 1.3 GB/min	MPEG2 1:60	

Datenkompression: Kriterien



- gute Kompressionsrate, gute Qualität
- geringe Latenz (Verzögerung)
- geringer Bedarf an Rechenleistung und Speicherbedarf

Klassifikation:

"direct coding"		PCM
"entropy coding"	repetitive sequence suppression	zero suppression run-length encoding
	statistical encoding	pattern substitution Huffman encoding
"source coding" (e.g. speech)	transform encoding	FFT
		DCT
		...
	differential encoding	DPCM delta modulation ADPCM
	vector quantization	general / fractal / ...
"channel coding"		HDLC, ...

Entropie- vs. Quellenkodierung

Entropiekodierung:

- Eigenschaften der Datenquelle werden ignoriert
- Signalwiederholungen entfernen
- statistische Verfahren, z.B. Huffman-Kodierung
- verlustfrei und reversibel
- geringe Kompressionsfaktoren (z.B. ca. 2 für Audiodaten)

Quellenkodierung (source encoding):

- Eigenheiten der Datenquelle / senke berücksichtigen
- z.B. Frequenzgang / Maskierung / Rauschschwellen des Ohrs
- verlustfrei
- verlustbehaftet für bessere Kompression (z.B. MP3 bis ca 10:1)

Codec

- verlustlose Datenkompression:

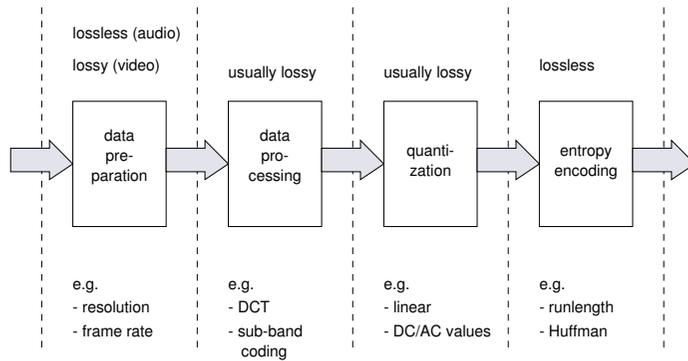
$$\text{decode}(\text{encode}(x)) = x$$

- jedes Verfahren benötigt Paar aus Coder und Decoder
:= "CODEC"

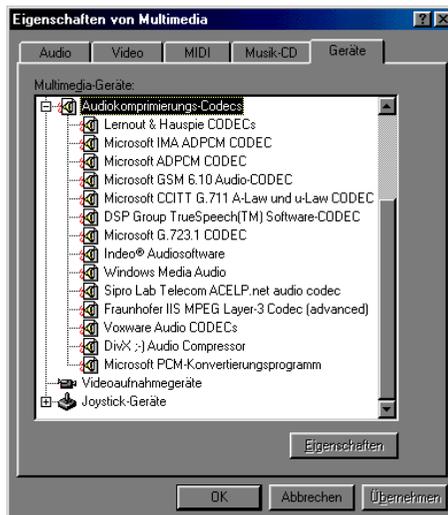
vollständige Abstraktion und Kapselung möglich:

- $\text{decode1}(\text{decode2}(\text{kanal}(\text{encode2}(\text{encode1}(x)))) = x$
- äußere / innere Schichten brauchen nicht vom Codec zu wissen
- beliebig tiefe Schachtelung
- siehe Windows / Java Media Framework

CODECs: Verkettung



Codecs: Beispiel Windows 9x



Windows-Systemsteuerung:

- sammelt CODECS
- Audio, Video, MIDI
- je nach installierter SW
- hier: 14 Audio-Codecs
- Sprache vs. Musik

Datenkompression: mehrfach?

- komprimierte Daten beliebig weiter komprimierbar?
 - diverse Patentanmeldungen dazu :-)
 - aber nicht möglich (sonst jede Datei auf 1 Bit komprimierbar...)
 - nur mit "Tricks" (z.B. Daten in Dateinamen speichern)
- => jeder Algorithmus expandiert bestimmte Dateien

Achtung: Verschlüsselung beseitigt Redundanzen der Daten:

- Verschlüsselung nur als letzter Schritt
- nach allen Kodierungsschritten
- verschlüsselte Daten nicht komprimierbar

Transformationen



- Sinuston, Darstellung im Zeitbereich und Frequenzbereich
- Darstellung im Frequenzbereich offenbar weit effizienter

wichtige Beispiele für Transformationen:

- Burrows-Wheeler (für Textdateien)
- Fouriertransformation (Audio, Bilder)
- Cosinustransformation
- Wavelet-Transformation

Datenkompression: Szenarien

Backup, Archivierung:

- möglichst gute Kompression, schneller Encoder
- Komplexität und Geschwindigkeit des Decoders egal

Software-Archive (JAR, Zip):

- möglichst schneller Decoder, gute Kompression
- Encoder wird selten benutzt,

Datenübertragung (Modem):

- Echtzeitanforderungen, möglichst gute Performance
- Notwendigkeit zum Umgang mit bereits komprimierten Daten

CD/DVD, weak-signal Kommunikation:

- möglichst gute Fehlererkennung und -korrektur

Informationsgehalt, Redundanz

The entropy of the data depends on the individual probabilities P_i , and is smallest when all n probabilities are equal. This fact is used to define the redundancy R in the data. It is defined as the difference between the entropy and the smallest entropy. Thus

$$R = \left(- \sum_1^n P_i \log_2 P_i \right) - \log_2(1/P) = - \sum_1^n P_i \log_2 P_i + \log_2 n.$$

The test for fully compressed data (no redundancy) is thus $\sum_1^n P_i \log_2 P_i = \log_2 n$.

- kann bei bekannter Nachricht leicht berechnet werden
- oder aus bisherigen Symbolen schätzen
- gute Kodierung erreicht $R \sim= 0$

(Shannon)

Run-Length Kodierung

2. all is too well
2. a2l is t2o we2l
2. a@2l is t@2l we@l

"Laufängenkodierung"

- Ersetzen von wiederholten Symbolen durch Marke+Anzahl+Symbol
- lohnt erst ab Lauflänge 3
- diverse Varianten zur Kodierung des Zählers
- sehr einfach, sehr schnell
- nur für bestimmte Daten geeignet
- z.B. für S/W-Bilder
- aber nicht für normalen Text, Audio, Grauwert- oder Farbbilder

Laufängenkodierung: FAX (group 3)

Run length	White code-word	Black code-word	Run length	White code-word	Black code-word
0	00110101	0000110111	32	0011011	0000110110
1	00111	010	33	0011010	0000110111
2	0111	11	34	0001001	0000110100
3	000	10	35	0001000	0000110101
4	1011	011	36	0001010	0000110100
5	1100	001	37	0001011	0000110101
6	1110	0010	38	0001111	0000110110
7	1111	0001	39	0010100	0000110111
8	1001	00010	40	0101001	0000110100
9	1010	000100	41	0101010	0000110101
10	0111	0000100	42	0101011	0000110110
11	0100	0000101	43	0010100	0000110111
12	01000	000011	44	0010101	0000110100
13	00101	0000100	45	0010100	0000110101
14	10100	0000111	46	0000101	0000110110
15	10101	00001100	47	0000101	0000110111
16	10100	00001011	48	0000101	0000110010
17	10101	00001010	49	0101010	0000110011
18	010011	00001000	50	0101011	0000110010
19	001100	000010001	51	0101010	0000110011
20	001100	000010100	52	0101011	0000110010
21	001011	000010101	53	0101010	0000110011
22	000011	000011011	54	0100101	0000111000
23	0000100	000011010	55	0101000	0000111001
24	010100	000011011	56	0101001	0000111000
25	010101	000011100	57	0101010	0000111001
26	001001	00001101010	58	0101011	0000111000
27	000100	00001101011	59	0101010	0000111001
28	001000	00001101100	60	0101011	0000111000
29	0000010	00001101101	61	0110101	0000111001
30	0000011	00001101100	62	0110101	0000111000
31	0011010	00001101001	63	0110100	0000111001

(a)

Run length	White code-word	Black code-word	Run length	White code-word	Black code-word
64	11011	000001111	134	01101010	00000110011
128	10010	0000100000	135	01101011	00000110010
152	01011	000011001001	1472	01001000	00000110011
224	010110	0000011011	1536	0101001	00000110010
320	0110110	00000110011	1600	01001010	00000110011
384	0110111	00000110100	1664	011000	00000110010
448	0110100	00000110101	1728	01011011	00000110011
512	0110101	00000110100	1792	00000100	white
576	0110100	00000110101	1856	000001100	white
640	0110011	000001100101	1920	000001101	run-1/4
704	0110010	000001100101	1984	00000110010	point
768	0110101	000001100100	2048	00000110011	run-3/4
832	0110100	000001100101	2112	00000110100	point
896	0110101	000001100100	2176	00000110101	run-1/4
960	0110100	000001100101	2240	00000110110	point
1024	0110101	000001100100	2304	00000110111	run-3/4
1088	0110100	000001100101	2368	00000111100	point
1152	0110101	000001100100	2432	00000111101	run-1/4
1216	0110100	000001100101	2496	00000111110	point
1280	01101001	000001100100	2560	00000111111	run-3/4

(b)

Table 2.30: Group 3 and 4 Fax Codes: (a) Termination Codes, (b) Make-Up Codes.

Auflösung:

- horizontal: 8.5 Pixel/mm
- vertikal: 3.85 / 7.7 / 15.4 Zeilen/mm
- (etwa 2200x1700 Pixel pro Seite A4)

Analyse typischer Briefe / Dokumente:

- häufig 2, 3, 4 schwarze Pixel
- oder 2 .. 7 weisse Pixel
- oder ganz weisse Zeilen ...
- ungeeignet für Bilder / Graustufen
- dafür besserer Code ("group 4")

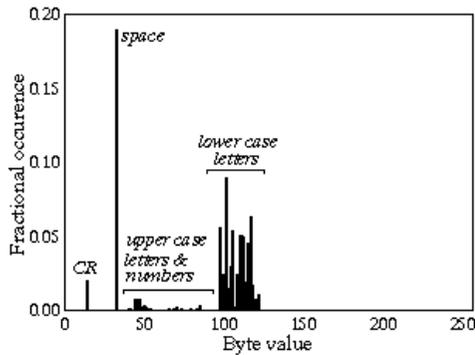
Buchstabenhäufigkeiten . . .

53++!305)6*;4826)4+.)4+);806*;48!8'60)85;]8*:+*8!83(88)5*!;
46(;88*96*?;8)*+(;485);5*!2:*+(;4956*2(5*)8'8*;4069285);)6
!8)4++;1(+9;48081;8:8+1;48!85;4)485!528806*81(+9;48;(88;4(+?3
4;48)4+;161;:188;+?;

"Here, then, we have, in the very beginning, the groundwork for something more than a mere guess. The general use which may be made of the table is obvious - but, in this particular cipher, we shall only very partially require its aid. As our predominant character is 8, we will commence by assuming it as the "e" of the natural alphabet. To verify the supposition, let us observe if the 8 be seen often in couples - for "e" is doubled with great frequency in English - in such works, for example, as "meet", "fleet", "speed", "seen", "been", "agree", etc. In the present instance we see it double no less than five times, although the cyrptograph is brief.

- Edgar Allen Poe, The Gold Bug

Beispiel: Buchstabenhäufigkeiten

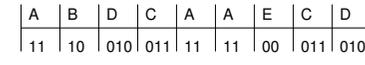


- Einzelhäufigkeiten in English: "ENATOIN SHRDLU ..."
- entsprechende Modelle auch für Paare ("qu"), Tripets, ...
- Grundidee: häufige Symbole mit kurzen Bitstrings kodieren

(Histogramm: DSP Guide)

Huffman-Kodierung

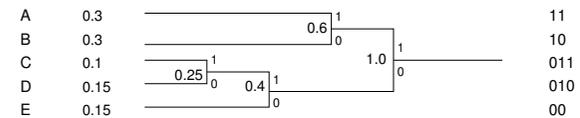
Symbole	Häufigkeit	Codewort
A	0.3	11
B	0.3	10
C	0.1	011
D	0.15	010
E	0.15	00



Ausnutzen der statistischen Eigenschaften der Daten:

- häufige Zeichen mit kurzen Bitfolgen kodieren
- seltene Symbole mit längeren (evtl. sehr langen)

Huffman: Konstruktion

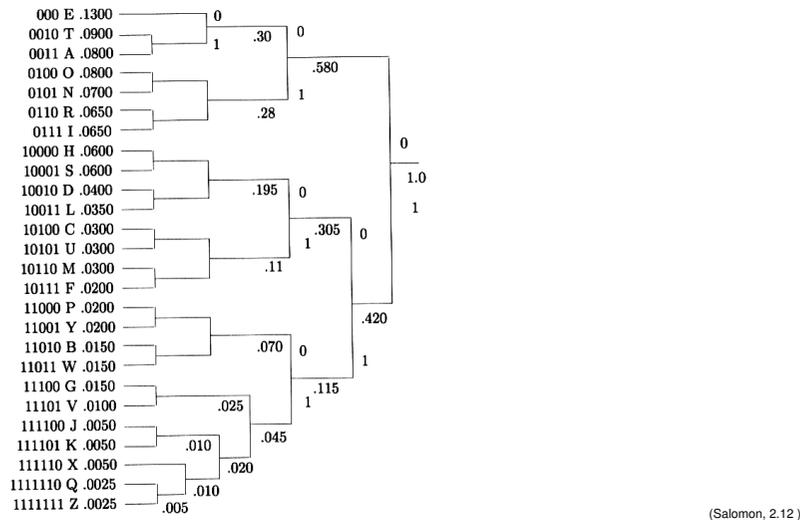


1. sortiere die Symbole nach ihrer Häufigkeit
2. kombiniere die zwei Symbole mit geringster Häufigkeit (bei gleichen Wahrscheinlichkeiten: zufällige Auswahl)
3. wiederhole die Konstruktion, bis alle Symbole kombiniert sind

- "bottom-up" Verfahren
- erfordert a-priori Kenntnis der Häufigkeiten
- "semi-adaptive": 2-pass Verfahren, sehr langsam
- "adaptive": Anpassen des Baums an geänderte Häufigkeiten

(siehe Salomon 2.9)

Huffman: Beispiel



Shannon-Fano Kodierung

Prob.	Steps	Final
1. 0.25	1 1	:11
2. 0.20	1 0	:10
3. 0.15	0 1 1	:011
4. 0.15	0 1 0	:010
5. 0.10	0 0 1	:001
6. 0.10	0 0 0 1	:0001
7. 0.05	0 0 0 0	:0000

Table 2.9: Shannon-Fano Example.

- Symbole in möglichst gleichgroße Mengen einteilen
- linke (im Bild: obere) Menge bekommt Bit 1, andere Menge Bit 0
- für alle Teilmengen mit mehr als 2 Elementen wiederholen

- Verfahren ähnlich zur Huffman-Konstruktion
- aber nur optimal, wenn in jedem Schritt gleichgroße Mengen

Prob.	Steps	Final
1. 0.25	1 1	:11
2. 0.25	1 0	:10
3. 0.125	0 1 1	:011
4. 0.125	0 1 0	:010
5. 0.125	0 0 1	:001
6. 0.125	0 0 0	:000

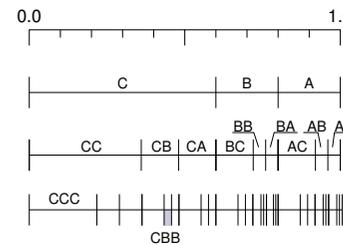
2.10: Shannon-Fano Balanced Example.

Arithmetic Coding

- Huffman-Kodierung ist nur in Spezialfällen optimal
- weil jedem einzelnen Symbol ein Code zugeordnet wird

arithmetische Kodierung:

- Zuordnung von Zeichen zu arith. Intervallen
- entsprechend der Zeichenhäufigkeiten
- rekursiv für Zeichenfolgen
- Verfahren patentiert und lizenzpflichtig



$p(C) = 0.6$ $p(B) = 0.2$ $p(A) = 0.2$
 Beispiel-Intervall für Folge "cbb":
 $0.6 \cdot 0.6 + 0.8 \cdot (0.48 - 0.36) = 0.456$
 $0.6 \cdot 0.6 + 0.6 \cdot (0.48 - 0.36) = 0.432$
 $0.432 = 0.0110$ $0.456 = 0.0111$
 Binärcode für "cbb" = 0111

Dictionary-Methoden

- warum Beschränkung auf Einzelsymbole?
- Verzeichnis mit häufigen Worten (Symbolfolgen) aufbauen
- Index in dieses Verzeichnis kodieren
- unbekannte Symbolfolgen in das Verzeichnis aufnehmen

- Lempel-Ziv Verfahren, Dutzende Varianten
- universell anwendbar: Texte, Programme, Bilder, ...
- sehr gute Kompressionsraten (für verlustfreie Algorithmen)

LZW

```

sir sid eastman easily teases sea sick seals
      sir sid eastman ...      (0,0,"s")
s   ir sid eastman           (0,0,"r")
si  r sid eastman           (0,0," ")
sir  sid eastman eas...     (0,0," ")
sir  sid eastman           (4,2,"d")

sir sid eastman easily t eases sea sick seals      (16,3,"e")

```

"sliding window" Methode:

- "search buffer" (links) als Dictionary, einige KByte
- Tokens mit Index ins Dictionary und nächstes Symbol schreiben

Medientechnik | WS 2001 | 18.204

LZW-Verfahren

- Welsh, 1984
- Dictionary mit Einzelsymbolen (z.B. 256 Bytes) initialisieren
- unbekannte Strings in Dictionary aufnehmen
- dabei Strings maximaler Länge versuchen
- was passiert bei vollem Dictionary?
- diverse Varianten: Dictionary löschen, LRU, ...
- Einsatz u.a. für ZIP, gzip, GIF, TIFF; ...

Medientechnik | WS 2001 | 18.204

LZW-Algorithmus

```

for (i=0; i < 255; i++) {
    append i as a 1-symbol to the dictionary
}
append null to the dictionary;
di = dictionary index of null;
repeat
    read(ch);
    if <<<di,ch>> is in the dictionary then
        di = dictionary index of <<<di,ch>>;
    else
        output(di);
        append <<di,ch>> to the dictionary;
        di = dictionary index of ch;
    endif;
until end-of-input;

```

Medientechnik | WS 2001 | 18.204

LZW: Beispiel

0	NULL	256	si
1	SOH	257	ir
	...	258	r_
32	space	259	_s
	...	260	sid
97	a	261	d_
98	b	262	_e
99	c	263	ea
			...
255	255	4095	

- Strings wachsen nur langsam (max. 1 Zeichen)

Medientechnik | WS 2001 | 18.204

BWT: Burrows-Wheeler Transformation

- Eingangsdaten in (großen) Blöcken bearbeiten
- z.B. bzip2: 100 .. 900 KByte Blockgröße
- Daten so umordnen daß ähnliche Strings entstehen
- anschließend normale statistische Kodierung
- völlig neuartiges Prinzip, ursprüngliche Idee (Wheeler 1983)
- adaptiert automatisch an statistische Eigenschaften der Daten
- Algorithmus funktioniert für Texte, Programme, beliebige Daten

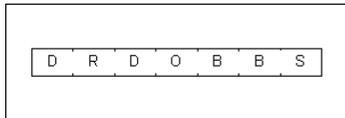
gute Beschreibungen:

M. Nelson, Data compression with the BWT, Dr. Dobbs. 9/1996

M. Tamm, Packen wie noch nie, c't 16/2000

(Burrows, Wheeler: A block sorting lossless data compression algorithm, DEC report 124, 1994)

BWT: Aufbau der Matrix

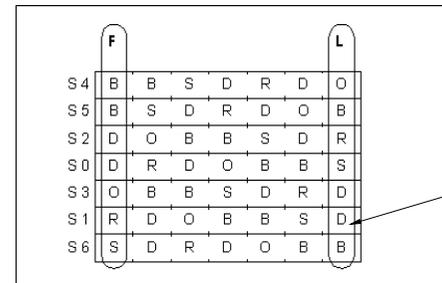


- Eingabedaten zu einer Matrix anordnen
- jede Zeile der Matrix um ein Zeichen rotiert

String 0	D	R	D	O	B	B	S
S 1	R	D	O	B	B	S	D
S 2	D	O	B	B	S	D	R
S 3	O	B	B	S	D	R	D
S 4	B	B	S	D	R	D	O
S 5	B	S	D	R	D	O	B
S 6	S	D	R	D	O	B	B

(aber 100000 x 100000 Matrix ist unhandlich)

BWT: Puffer, Rotation



- Sortieren der Matrix, Aufbau eines Indexvektors
- geeignete Vergleichsfunktion wie strcmp() mit wrap-around
- markierte Spalten F (first) und L (last) der Originaldaten
- Zeichen in L sind jeweils die Vorgänger der Zeichen in F
- Ausgabe der BWT: Spalte L und "Primärindex" (hier 5)

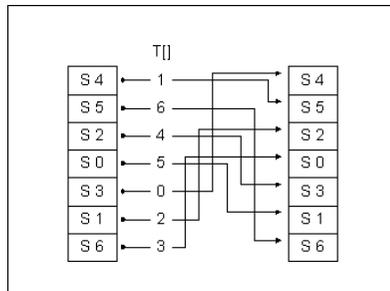
BWT: Puffer, sortiert

L: F.....

```
t: hat redistributors of a free
t: hat refer to this License an
t: hat system in reliance on co
w: hat the Program does.\n\n 1
t: hat there is no warranty (or
t: hat there is no warranty for
w: hat they have is not the ori
t: hat they, too, receive or ca
t: hat users may redistribute t
t: hat version or of any later
t: hat what they have is not th
t: hat work are not derived fro
```

- (BWT eines Ausschnitts aus der GPL)

BWT: Transformationsvektor



- $T[i]$ enthält die Zeile, die $S[i+1]$ enthält
- Zeile 3 enthält S0, Zeile 5 enthält S1, also $T[3] = 5$
- Zeile 2 enthält S2, also $T[5] = 2$, usw.
- im Beispiel also $T = \{ 1, 6, 4, 5, 0, 2, 3 \}$
- aber: Berechnung von T aus L ?

BWT: Rekonstruktion von T

L							
O	?	?	?	?	?	?	?
B	?	?	?	?	?	?	?
R	?	?	?	?	?	?	?
S	?	?	?	?	?	?	?
D	?	?	?	?	?	?	?
I	?	?	?	?	?	?	?
B	?	?	?	?	?	?	?

L	F						
O	B	?	?	?	?	?	?
B	B	?	?	?	?	?	?
R	D	?	?	?	?	?	?
S	D	?	?	?	?	?	?
D	O	?	?	?	?	?	?
D	R	?	?	?	?	?	?
B	S	?	?	?	?	?	?

- Wiederherstellung von F:
- L ist gegeben, F war sortiert: einfach die Zeichen in L sortieren
- Zeichen "O": offenbar gilt $T[4] = 0$
- F sortiert: $T[1] = 0$, $T[6] = 1$, usw.

BWT: Move-to-Front Kodierung

- verwaltet Liste mit 256 Werten
- Ausgabe eines Zeichens:
Index des Zeichens in der Liste ausgeben
und Zeichen nach vorne (Index 0) verschieben
- Beispiel: "tttWtttt" (aus sortierter GPL)
Ausgabe: { 116, 0, 0, 88, 1, 119, 1, 0, 0 }
- anschließend noch Huffman- oder arithmetische Kodierung

BWT: Gesamttablauf

Compressing a file using the demo programs:

```
RLE input-file | BWT | MTF | RLE | ARI > output-file
```

A brief description of each of the programs follows:

RLE.CPP

This program implements a simple run-length encoder. If the input file has many long runs of identical characters, the sorting procedure in the BWT can be degraded dramatically.

BWT.CPP

The standard Burrows-Wheeler transform is done here. This program outputs repeated blocks consisting of a block size integer, a copy of L, the primary index, and a special last character index. This is repeated until BWT.EXE runs out of input data.

MTF.CPP

The Move to Front encoder.

RLE.CPP

The fact that the output file is top-heavy with runs containing zeros means that applying another RLE pass to the output can improve overall compression.

ARI.CPP

This is an order-0 adaptive arithmetic encoder, directly derived from the code published by Witten and Cleary in their 1987 CACM article.

BWT: Vergleich mit anderen Verfahren

Performance Figures

The table below shows obtained bit/byte for some different universal coding utilities and files. I have chosen pretty large files to lessen the boundary-effects at the start.

File × Utility → bit/byte	bzip	dmc	gzip	zip	arj	lha	lharc	compress
bible.tar (5140480 bytes)	1.80	1.83	2.53	2.53	2.60	2.80	2.97	2.89
netscape (4501956 bytes)	3.29	3.45	3.51	3.51	3.51	3.58	3.72	4.89
gcc-2.7.2.1.tar (7090289 bytes)	1.49	1.61	2.02	2.01	2.02	N/A	N/A	2.99

The file *bible.tar* consists of selected texts from the bible (In swedish). See [Project Runeberg](#). Notice the superior performance of bzip and dmc for this kind of source! The perhaps not so well-known coder *dmc* uses dynamic markov modelling and arithmetic coding and was written by [Gordon V. Cormack](#). It performs almost as good as bzip but is slower.

The file *netscape* is a stripped binary. bzip performs best also here but the difference is not so big, although note the lousy performance of the standard UNIX utility compress. (Uses LZW technique).

The file *gcc-2.7.2.1.tar* consists of lot of C-code. Also here bzip is outstanding. UNIX compress has to use double as many bits as bzip...

The following programs and flags were used to generate the different codes:

- *bzip -9* ver. 0.21
- *dmc 5000000* (That's a large buffer in case you wondered) ver 0.0.0
- *gzip -9* ver 1.2.4
- *zip -9* ver 2.0.1 for UNIX (performs better than *pkzip -ex*)
- *arj -em* ver 2.30
- *lha* LHa ver. 1.00 for UNIX
- *lharc* LHarc ver. 1.02 for UNIX
- *compress* UNIX (N)compress ver. 4.2.4

Wavelet-Transformation

- Wavelets := "kleine Wellen"
- "lokalisierte", skalierbare Basisfunktionen

$$\Phi_{(s,l)}(x) = 2^{-\frac{s}{2}} \Phi(2^{-s}x - l)$$

- diverse Basisfunktionen möglich
- Eigenschaften können "maßgeschneidert" werden:
- lineare Transformation:

$$W(x) = \sum_{k=-1}^{N-2} (-1)^k c_{k+1} \Phi(2x + k)$$

- Normierungsbedingung: $\sum_{k=0}^{N-1} c_k = 2, \sum_{k=0}^{N-1} c_k c_l = 2\delta_{l,0}$

Wavelets: Beispielfunktionen

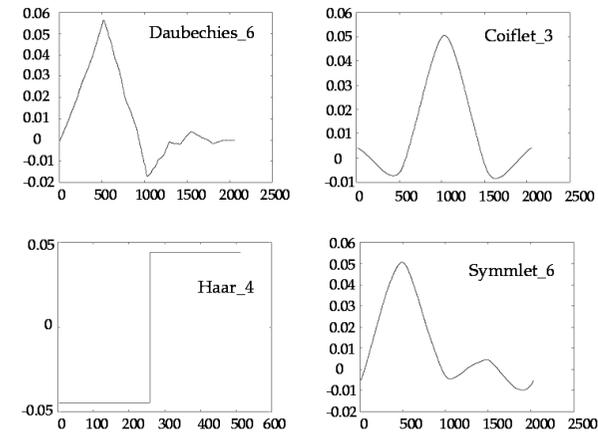
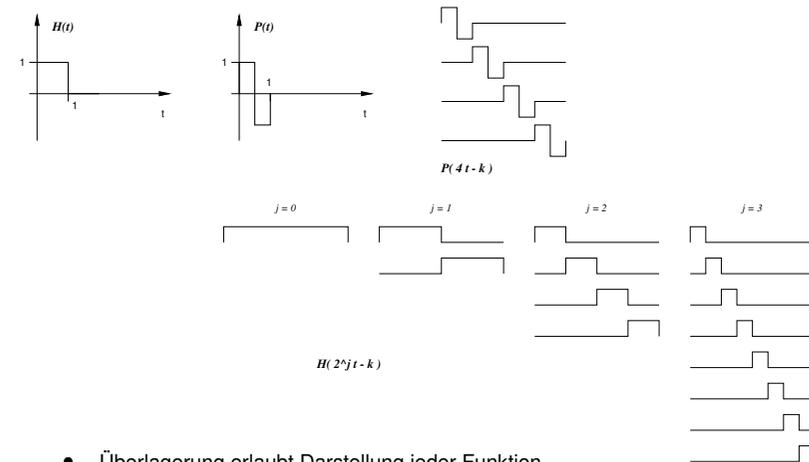


Fig. 4. Several different families of wavelets. The number next to the wavelet name represents the number of vanishing moments (A stringent mathematical definition related to the number of wavelet coefficients) for the subclass of wavelet. These figures were generated using WaveLab.

(Amara Graph, Introduction to Wavelets)

Wavelets: Haar-Wavelet

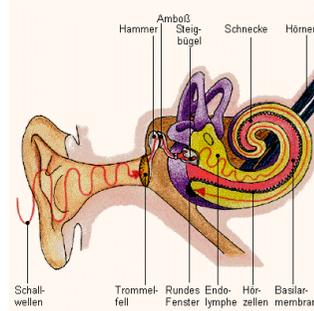


- Überlagerung erlaubt Darstellung jeder Funktion

(s. Salomon 5.6)

Audio: Agenda

- Schall und Schallausbreitung
 - Aufbau des Gehörs
 - Hörwahrnehmung
 - Sprache, Vokalartikulation
-
- Signalverarbeitung
 - Abtastung, Abtasttheorem
 - A/D- und D/A-Wandlung
 - Filter, elementare Algorithmen
-
- Kodierung von Audiodaten
 - psychoakustische Verfahren, MP3



Audio: Literatur

K.v.d.Heide, Skripte zu "Signalverarbeitung" und "Nachrichtentechnik", FB Informatik, Uni HH, tech-www.informatik.uni-hamburg.de/lehre/ (Matlab)

Mathworks, Inc., Matlab 5.3 User and Toolbox Manuals, www.matlab.com

B. Kollmeier, Vorlesungsskripte Audiologie, www.physik.uni-oldenburg.de/Docs/medi/download/

U. Zölzer, Digitale Audiosignalverarbeitung, Teubner 1996

P.Gerdson, P.Kröger, Digitale Signalverarbeitung in der Nachrichtenübertragung, Springer 1997

R.W.Hamming, Digital Filters, Prentice Hall, 1983

W.H.Press, B.P.Flannery, S.A.Teukolsky, W.T. Vetterling, Numerical Recipes, Cambridge Univ. Press

U. Tietze, Ch. Schenk, Halbleiter-Schaltungstechnik, Springer 1993 (analoge Schaltungen)

IEEE Journal Signal Processing

diverse Konferenzen zum Thema, Audio u.a. Audio Engineering Convention

DSP-Datenbücher (www.motorola.com, www.analog.com)

Motivation

wozu die Gehörwahrnehmung untersuchen?

- Forschung / Erkenntnisgewinn
möglichst optimale Verfahren zur:
 - Aufzeichnung / Speicherung / Wiedergabe von Audio
 - so gut wie nötig - aber nicht besser (!)
- psychoakustische Verfahren, etwa MP3-Kompression
- bessere Algorithmen für Musik und Musikproduktion
- Raumsimulation für Spiele oder virtuelle Realität
- möglichst effiziente (low-level) Sprachverarbeitung
- medizinische Anwendungen, vor allem Hörgeräte

Schall

physikalischer Schallbegriff:

- alle elastischen Schwingungen eines Materials
- im allg. Fall longitudinal und transversal
- aber in Gasen (Luft) nur longitudinal (=Druckschwingungen)

• Schallgeschwindigkeit	ca. 340 m/s in Luft	
• Schallintensität	lokale Druckänderung	
Luftdruck auf Meereshöhe	1 bar	= 10.000 Pa
Schmerzschwelle	1 mbar	= 10 Pa
wahrnehmbare Druckänderung	10E-9 bar	= 0.00001 Pa

- Ohr hat optimale Empfindlichkeit
- weitere Steigerung: Wahrnehmen der Molekularbewegung

Schall: Leistung, dB

- Kenngrößen des Schallfelds: Druckamplitude P bzw. Intensität I
- Gesamtleistung: Integral der Intensität über eine Oberfläche
- Lautstärke (Intensität): dB

$$L / \text{dB} := 10 \log_{10}(I / I_{\text{ref}})$$

$$:= 20 \log_{10}(P / P_{\text{ref}})$$

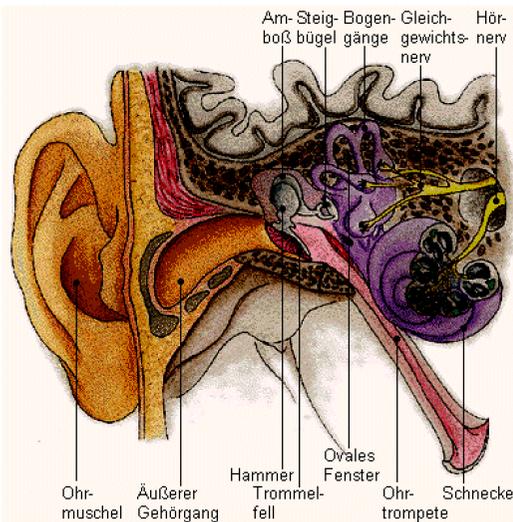
$$P_{\text{ref}} := 0.00002 \text{ Pa}$$

- alternativ: dB als Maß für das Verhältnis (p_1/p_2)

20 dB	zehnfacher Schalldruck
6 dB	doppelter Schalldruck
-6 dB	halber Schalldruck

Quelle	Leistung (typ.)
Sprache	10 μ W
Geige	1 mW
Lautsprecher	100 W

Ohr: Schema



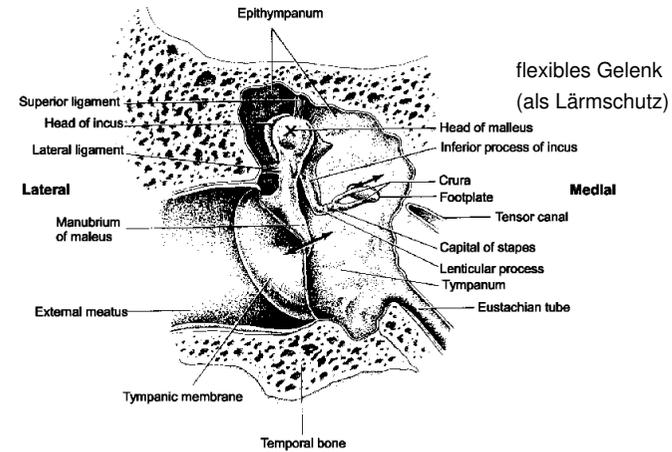
drei Untereinheiten:

- Außenohr
- Mittelohr
- Innenohr

bzw.:

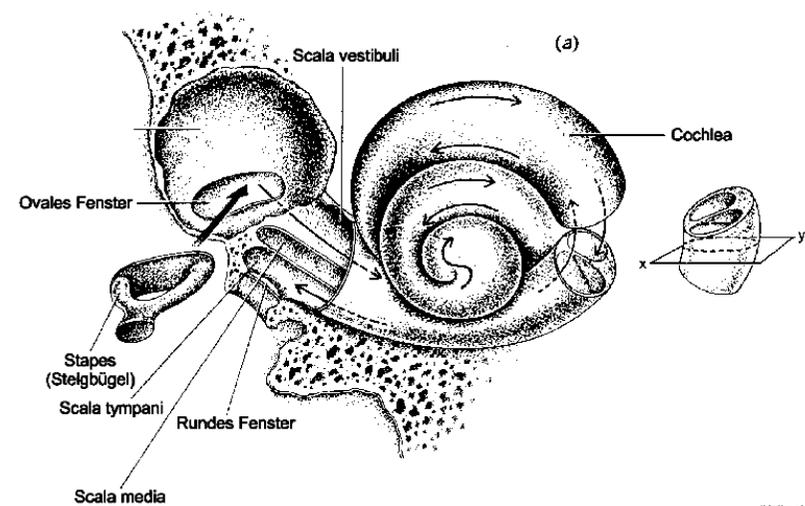
- Verstärkung, Delay
- Impedanzwandlung
- Frequenzanalyse

Mittelohr



- Verstärkung (Impedanzwandlung), Übergang Luft nach Wasser
- Trommelfeld ca. 20fache Fläche des ovalen Fensters

Innenohr: Aufbau und Cochlea



Innenohr: cortisches Organ

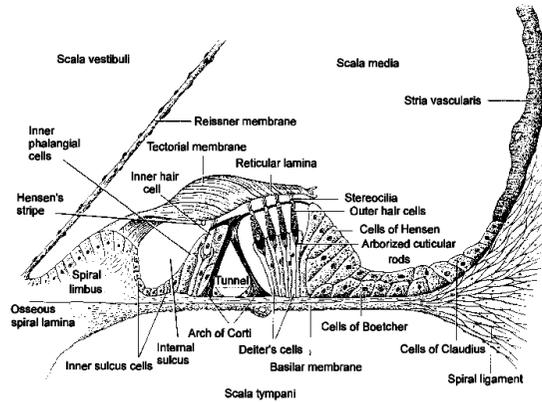
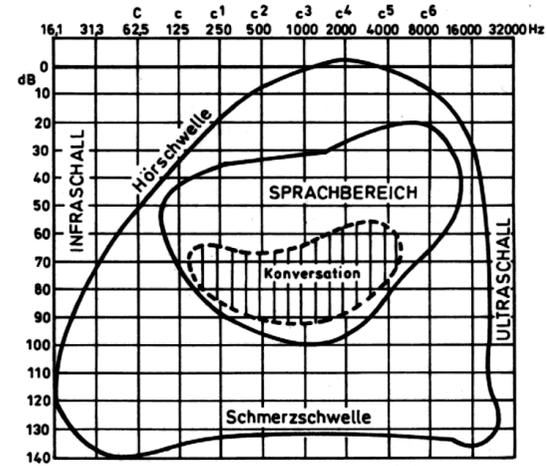


Abbildung 3.7: Schematische Darstellung des Cortischen Organs (aus Gulick, W. L., Gescheider, G. A., Frisina, R. D.: Hearing. Oxford University Press, Oxford 1989)

(Kollmeier)

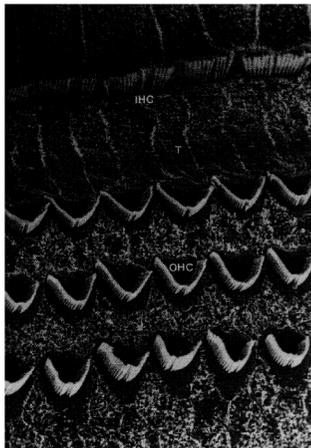
Hörbereich und Sprachbereich



gute Sprachverständlichkeit im "Hauptsprachbereich" (ca. 300-3500 Hz)

(c'110/89)

Innenohr: Haarzellen



innere Haarzellen
"Sensoren"

äußere Haarzellen
"Aktoren"

Abbildung 3.8: Detailsicht der Innenohr-Anatomie, IHC = Innere Haarzellen, OHC = Äußere Haarzellen (aus Dallos, P., Popper, A. N., Fay, R. R. (Hrsg.): The Cochlea. Springer Verlag, New York 1996)

(Kollmeier)

absolute Hörschwelle

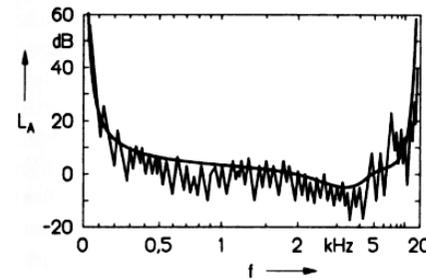


Abb. 9.11. Monaurale Absoluthörschwelle für Sinustöne. Glatte Kurve: Mittelwert des Schwellenpegels L_A von normalhörenden Personen. Gezackte Kurve: Mit der Methode des pendelnden Regels (Békésy tracking) gemessen (25jähriger Mann, rechtes Ohr). Abszisse: Tonfrequenz, SPINC-skaliert; Ordinate: Schallpegel

- geringe Empfindlichkeit bei tiefen und hohen Tönen
 - maximale Empfindlichkeit zwischen 2.5 KHz
- => Loudness-Kurven

(Terhardt)

Lautheit

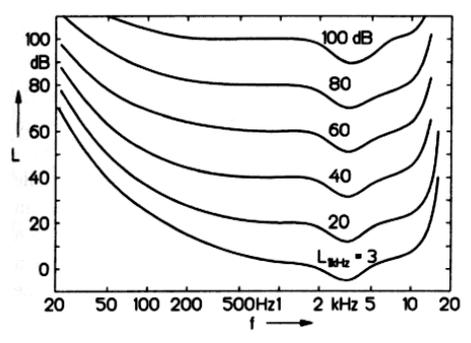


Abb. 10.8. Kurven gleicher Lautheit von Sinustönen im ebenen Schallfeld: Schallpegel L eines Sinustones (Ordinate), welcher die gleiche Lautheit hervorruft wie ein 1 kHz-Ton mit dem angegebenen Pegel, in Abhängigkeit von der Tonfrequenz f (Abszisse)

- bei leisen Tönen schwache Wahrnehmung von Höhen/Bässen
- Empfindlichkeitsmaximum bei ca. 3 KHz
- "loudness"-Regler am Hifi-Verstärker zur Kompensation

(Terhardt)

Tuningkurven: Katzen

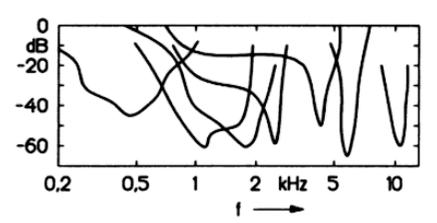


Abb. 9.10. Tuningkurven, gemessen an anästhesierten Katzen. Jede Kurve gehört zu einer anderen Faser des akustischen Nerven. Die Ordinate gibt denjenigen Schallpegel eines Sinustones an, welcher zur Erregung einer festen Anzahl von Aktionspotentialen pro Zeiteinheit erforderlich ist. Abszisse: Frequenz des Sinustones. Nach Kiang [501]

- hohe Güte der Frequenzselektion
- "aktive" Frequenzbereiche der Fasern überlappen
- menschliches Gehör entsprechend

(Terhardt)

Maskierung

Mithörschwelle eines Sinustones bei Verdeckung durch Schmalbandrauschen. In Abb.11.4 sind drei Mithörschwellen von Sinustönen verdeckt durch Schmalbandrauschen der angegebenen Mittenfrequenzen dargestellt [1125]. Die Fourier-Spektren der maskierenden Schmalbandgeräusche sind durch Verwendung von Bandpaßfiltern hoher Flankensteilheit auf Bandbreiten von ungefähr 100, 160 und 700 Hz beschränkt [CD 2].

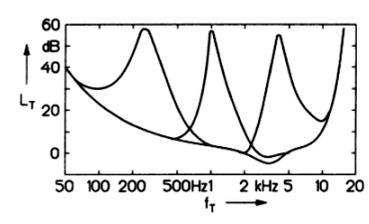
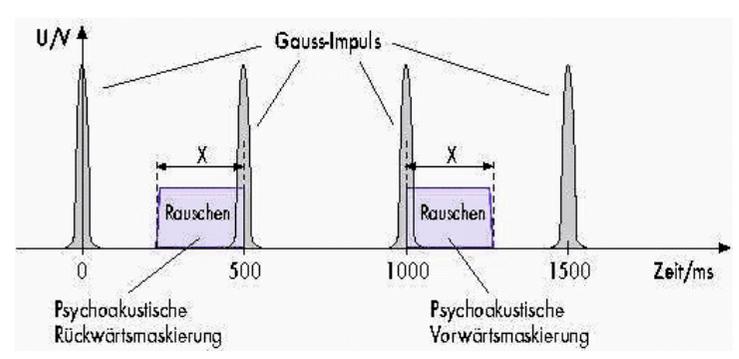


Abb. 11.4. Mithörschwellen von Sinustönen bei Verdeckung durch frequenzgruppenbreite Schmalbandrauschen mit den Mittenfrequenzen 250, 1000, 4000 Hz und dem Maskierpegel $L_M = 60$ dB. Nach [1125]

(Terhardt)

- Frequenzanalyse in der Cochlea nicht perfekt
- laute Töne verdecken benachbarte leisere
- Ausnutzung in psychoakustischen Modellen (MiniDisc, MP3, usw)

Maskierung: temporal



- laute Töne verdecken spätere leisere Töne
- und frühere leisere Töne (!!)
- Ausnutzung in psychoakustischen Modellen (MiniDisc, MP3, usw)

Frequenzgruppen

Tabelle 9.1: Frequenzgruppen nach Zwicker 1982

z/Bark	f_u/Hz	f_o/Hz	$\Delta f_G/\text{Hz}$	f_m/Hz
0	0	100	100	50
1	100	200	100	150
2	200	300	100	250
3	300	400	100	350
4	400	510	110	450
5	510	630	120	570
6	630	770	140	700
7	770	920	150	840
8	920	1080	160	1000
9	1080	1270	190	1170
10	1270	1480	210	1370
11	1480	1720	240	1600
12	1720	2000	280	1850
13	2000	2320	320	2150
14	2320	2700	380	2500
15	2700	3150	450	2900
16	3150	3700	550	3400
17	3700	4400	700	4000
18	4400	5300	900	4800
19	5300	6400	1100	5800
20	6400	7700	1300	7000
21	7700	9500	1800	8500
22	9500	12000	2500	10500
23	12000	15500	3500	13500
24	15500			

Tonhöhenwahrnehmung?

- Messungen
- vgl. Tuningkurven

- tiefe Frequenzen:
ca. 100 Hz Bandbreite

- hohe Frequenzen:
log. Breite der Gruppen

=> "Bark" Frequenzskala

- Ausgangspunkt für
psychoakustische Modelle

(Tabelle aus Zoelzer)

Medientechnik | WS 2001 | 18.204

Klangfarben

akustisches Signal mit vielen Teiltönen:

- keine Wahrnehmung aller Einzeltöne
- sondern Zuordnung zu Quellen/Instrumenten
- tiefste Frequenz bestimmt die wahrgenommene Tonhöhe
- Obertonspektrum identifiziert die Quelle
- mit Eigenschaft "Klangfarbe"
- Klangfarbe auch stark vom "attack" des Klangs abhängig
(=> Synthesizer mit kurzen Samples plus Schleifen)
- entsprechend automatische "Sprecheridentifikation"
- komplexe Zusammenhänge für Akkordwahrnehmung

Medientechnik | WS 2001 | 18.204

Phasen: Hilbertfilter

original



Sprachsample "Oberseminar"

phasengedreht



(vdHeide)

- Ohr unempfindlich gegen Phaseninformation
- verändertes Signal klingt einzeln (vollkommen) gleich
- erst bei Interferenzen mehrerer Signale hörbar
- Phasengang vieler Systeme (Telefon!) extrem schlecht

Medientechnik | WS 2001 | 18.204

Klangfarben: Spektrogramm

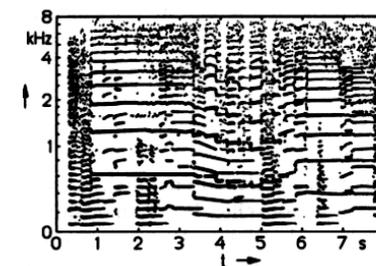


Abb. 1.9. Beispiel eines Teiltonzeitmusters. Ausschnitt einer mehrstimmigen Blasmusikdarbietung. Die hauptsächlich beteiligten Instrumente sind Trompete, Posaune, Klarinette und Tuba. Die instantane Teilton-Amplitude wird durch die Strichstärke angedeutet. Ordinate Bark-skaliert. Nach [969]



Abb. 1.10. Transkription der Trompeten- und der Tuba-Stimme des Musikbeispiels, dessen Teiltonzeitmuster in Abb. 1.9 dargestellt ist

(Terhardt)

Medientechnik | WS 2001 | 18.204

Klangfarben: Beispiele

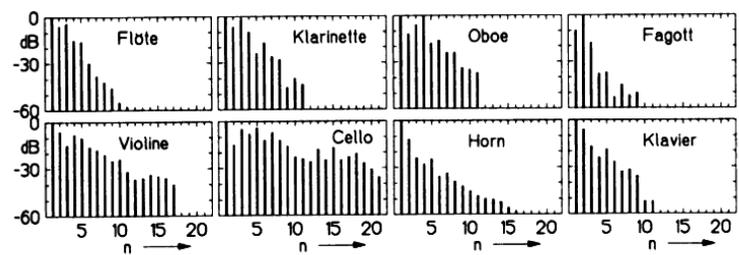
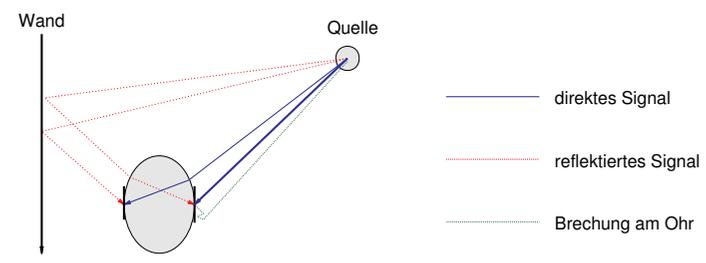


Abb. 8.12. Teiltonspektren verschiedener Musikinstrumente schematisch

(Terhardt)

- Streicher mit reichem Obertonspektrum
- Flöten mit sehr reinen Tönen (insb. Blockflöte)
- Klavier stark anschlag- und tonhöhenabhängig
- aber auch "Attack-Phase" maßgeblich für Klangwahrnehmung

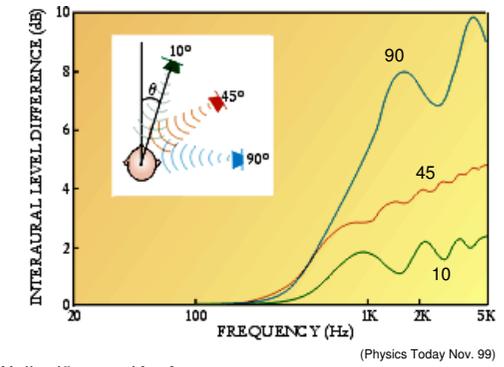
Akustische Raumwahrnehmung



- direktes Signal
- reflektierte Signale
- jeweils mit Beugung um den Kopf herum
- jeweils mit Brechung an Ohr, Schultern, ...
- zusätzlich "obstruction" und "occlusion" für verdeckte Quellen ...

ILD: Lautstärkedifferenz

interaural level difference:



(Physics Today Nov. 99)

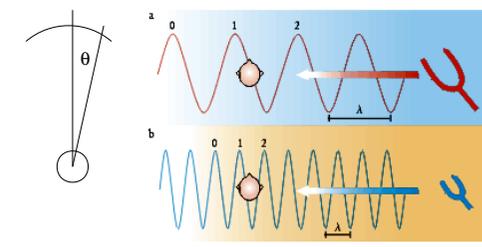
- Welle kaum gestört, wenn Wellenlänge > Kopf
- daher kaum Effekt bei niedrigen Frequenzen
- Bässe schlecht ortbar

ITD: Zeit- und Phasendifferenz

interaural time difference:

$$dt = \frac{3a}{c} \sin \theta$$

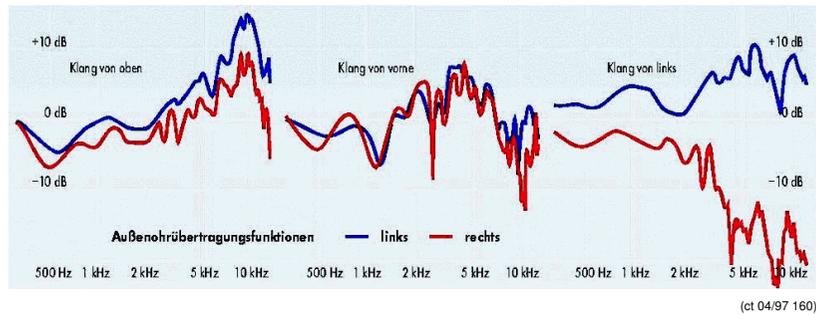
a = Kopradius ~ 8.75 cm
c = Schallgeschw ~ 340 m/s



(Physics Today Nov. 99)

- 500 Hz Sinuston von vorne ortbar mit ca. 1 Grad Auflösung
=> ca. 13 ms Zeitauflösung
- mehrdeutig, wenn n* Wellenlänge ~ = Kopfdurchmesser
- Ohr unempfindlich für Phaseninformation oberhalb ca. 1 KHz
- vgl. Hilbertfilter-Experiment

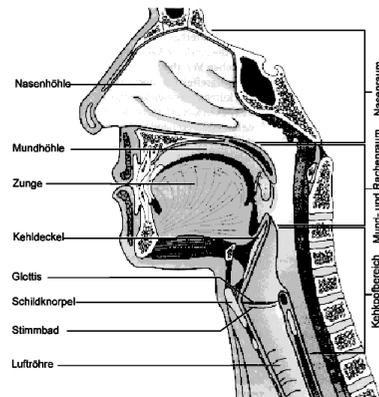
HRTF



"head related transfer function": Außenohr/Kopf-Übertragungsfunktion

- Einfluß von Kopf, Schultern, Außenohr auf Schallsignale
- sehr komplexe Winkel- und Frequenzabhängigkeit
- Verbesserung der Ortung durch (kleine) Kopfbewegungen

Sprache: Vokaltrakt



- Glottis, Stimmband
- Klang zusätzlich durch Zungen-/Lippenstellung
- typische Spektren: Mann / Frau / Kind normal / erregt

Abbildung 6.1: Vokaltrakt im Querschnitt (schematisch) (aus Kießling, J., Kollmeier, B., Diller, G.: Versorgung und Rehabilitation mit Hörgeräten. Thieme Verlag, Stuttgart 1997)

(Kollmeier)

Sprache: Spektrogramm

Formanten, d. h. die Frequenzbereiche, bei denen der Vokaltrakt eine besonders hohe Verstärkung des akustischen Signals bewirkt. Die Formanten stellen sich damit als Spitzen im Spektrum dar und sind charakteristisch für den jeweils artikulierten Vokal (s. unten).

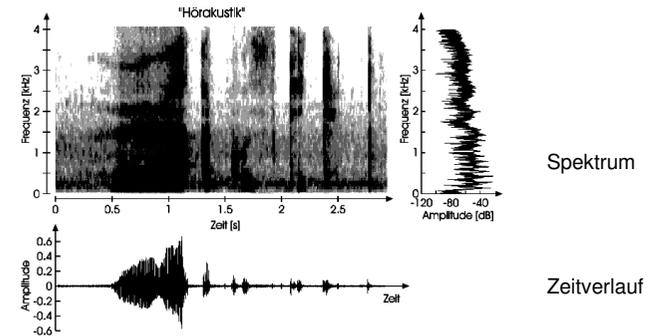


Abbildung 6.2: Zeit-Signal (unten), Spektrum (rechts) und Spektrogramm (links oben) des Wortes „Hörakustik“

(Kollmeier)

Sprache: Artikulation

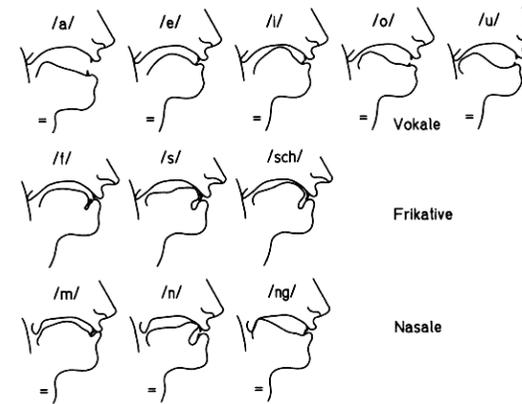


Abb. 7.1. Schematische Darstellung der Artikulation einiger Sprachlaute

- Zungen- und Lippenstellung maßgeblich

(Terhardt)

Sprache: Vokaltrakt

- Stimmband + Resonanzrohr
- Lage der Formanten:
Wellenlänge: $L/4 = 17\text{cm}$

$F_1 = c/L = 340\text{m/s} / 68\text{cm} = 500\text{ Hz}$
 $F_2 = 1500\text{ Hz}$
 $F_3 = 2500\text{ Hz}$
 $F_4 = 3500\text{ Hz}$

Gehör: Zusammenfassung

- Gehör außerordentlich leistungsfähig:
- überdeckt 13 Größenordnungen Schalldruck
 - fast optimale Empfindlichkeit
 - Hörbereich etwa 20Hz .. 20 KHz
 - Frequenzanalyse "mechanisch" in der Cochlea
 - komplexe Verarbeitung/Erkennung im Gehirn
 - Wahrnehmung Klang vs. Geräusch
- insbesondere zur Raumwahrnehmung:
- Zeit- und Intensitätsdifferenzen der Signale
 - Kopfübertragungsfunktion
 - Auswertung von Echos
- => entsprechende Modelle für effiziente Signalverarbeitung

Sprache: Vokaldreieck

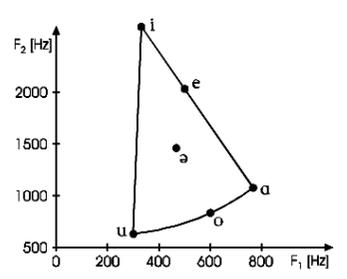


Abbildung 6.1.4: Vokaldreieck. Aufgetragen ist die Frequenz des ersten Formanten F_1 (Abzisse) und des zweiten Formanten F_2 (Ordinate) für unterschiedliche Vokale (schematisch) (Kollmeier)

- typische Frequenzen / Formanten der Vokale
- verhältnismäßig einfache Erkennung von stimmhaften Lauten

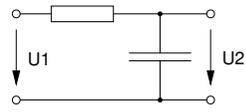
Digitale Signalverarbeitung

- Motivation
- analoge vs. digitale Signalverarbeitung
- Zahlenfolgen, LTI-Systeme
- Abtasttheorem
- Quantisierung, AD/DA Konverter, Dithering
- Beispiele für Audio-Algorithmen
- digitale Filter

DSP: Motivation

analoge Signalverarbeitung:

- Modellierung mit Differentialgleichungen
- direkte Umsetzung mit diskreten Bauteilen
- hauptsächlich mit "analoger" Elektronik
- Widerstände, Kondensatoren, Spulen, Transistoren



aber:

- entweder geringe Genauigkeit (z.B. 10%) oder sehr teuer
- Alterung, Exemplarstreuungen, aufwendiger Abgleich
- wirtschaftlich nur für geringe Anzahl der Bauelemente
- Datenspeicherung (analog) problematisch

=> komplexe Algorithmen nicht umsetzbar
=> digitale Verarbeitung verspricht Abhilfe

DSP: Definition

"digital signal processing" (digitale Signalverarbeitung):

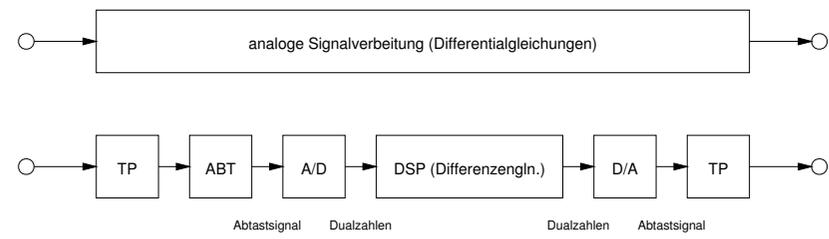
- | | |
|---|--|
| <p>Verbeitung von</p> <ul style="list-style-type: none"> • zeitdiskreten • wertdiskreten • Zahlenfolgen <p>auf Digitalrechnern</p> | <p>für:</p> <p>Nachrichtenübertragung</p> <p>Bildverarbeitung</p> <p>Audioverarbeitung</p> <p>usw.</p> |
|---|--|
- als Ersatz analoger Verfahren
 - Differenzen- statt Differentialgleichungen
 - oft mit Echtzeitanforderungen
 - im Prinzip beliebige Genauigkeit
 - oft mit Spezialhardware, etwa Signalprozessoren / ASICs

DSP: Eigenschaften

Verarbeitung zeit- und wertdiskreter Zahlenfolgen:

- als Ersatz für zeit/wert-kontinuierliche Verarbeitung
- wenn Abtasttheorem erfüllt (genügend hohe Abtastrate)
- beliebige Genauigkeit (Wortbreite anpassen)
- optimale Stabilität (z.B. keine Temperaturabhängigkeit)
- kein Abgleich notwendig (z.B. Nullpunktseinstellung)
- keine Alterung, keine Toleranzen
- auch für sehr komplexe Algorithmen
- die analog nicht (wirtschaftlich) realisierbar sind
- insbesondere: Datenspeicherung
- VLSI-Technologie erlaubt sehr hohe Performance

DSP: vs. analoge Verarbeitung



- | | |
|---|--|
| <p>"analoge" Signalverarbeitung</p> <ul style="list-style-type: none"> • zeitkontinuierlich • wertkontinuierlich • Differentialgleichungen • Drift, Rauschen, ... | <p>"digitale"</p> <ul style="list-style-type: none"> • zeitdiskret (Abtastung) • wertdiskret (Quantisierung) • Differenzengleichungen • Überlauf, Rundungsfehler |
|---|--|

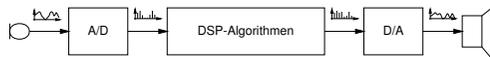
DSP: Audio-Anwendungen

Signalverarbeitung im Audio-Bereich:

- Datenspeicherung (CD, DVD, ...)
 - Datenkompression (MP3, AC3, MiniDisc, ...)
 - Datenübertragung (DSR, GSM, ...)
- jeweils für Musik und Sprache
- Musikproduktion (digitales Tonstudio)
 - Musiksynthese (z.B. virtuell analog)
 - Audioeffekte (z.B. Hall)
 - Surround-Verfahren (z.B. DirectX mit HRTF)
- Samplefrequenz typ. 44.1 / 48 KHz
 - hohe Genauigkeit erforderlich: 16 .. 24 bit

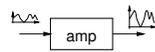
Medientechnik | WS 2001 | 18.204

DSP: Verstärker, Mixer



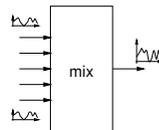
- digitaler Verstärker

```
output[t] = input[t] * gain;
```



- digitaler Mixer:

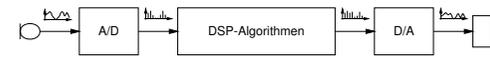
```
output[t] = 0;
for( int i=0; i < n_inputs; i++) {
    output[t] += input[i][t] * gain[i];
}
```



- viele MAC-Operationen (multiply-accumulate)
- Problem Aussteuerung vs. Überlauf => saturation arithmetic

Medientechnik | WS 2001 | 18.204

DSP: Echo, Hall, und mehr



- Verzögerung: auf alte Abtastwerte zurückgreifen

```
delay[t] = input[t]*gain + input[t-t2]*gain2;
```

- Rekursion möglich: Echo, Hall, Raumsimulation

```
hall[t] = input[t]*gain + hall[t-t2]*gain2;
```

- Algorithmen für viele Anwendungen:

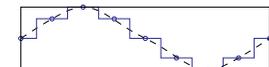
Verzögerung, Echo, Hall, Overdrive, Verzerrung, ...
Filter, Formatfilter, Tonhöhenänderung, Tempoänderung, ...

Medientechnik | WS 2001 | 18.204

DSP: Sampling und Synthese

- Sampling: Abtastwerte abspeichern

```
sample[t] = input[t];
```

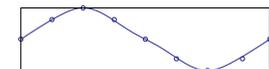


- Samples direkt abspielbar (CD, Spiele)



- "Wavetable"-Synthesizer:

```
output[t] = sample[t*pitch]
            interpolate( sample[] )
            filter( interpolate( sample[] ) )
            effects( filter( interpolate( sample[] ) ) )
```



keine/lineare/Spline-Interpolation

- Interpolation: nearest / linear / splines / ...

Medientechnik | WS 2001 | 18.204

Audiosysteme: Anforderungen

Grundanforderungen:

- CD/DAT Qualität 16 bit linear, 44.1 / 48 KHz
- höhere Genauigkeit 24 bit linear PCM
- oder Gleitkomma 32 bit IEEE

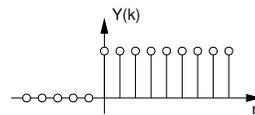
Beispiele:

	# MAC/s	(48 KHz)
• 10 Kanal Mixer	480.000	
• 100-tap FIR-Filter	4.800.000	
• 1000-tap FIR-Filter	48.000.000	
• 10 Kanal Mixer, je 3 100-tap FIR	144.480.000	
• ...		
• Pentium-200 (10 Takte/Mult.)	20.000.000	
• Pentium-III, 1 GHz, ISSE (max:)	4.000.000.000	

=> Spezialhardware oder modern(st)e Prozessoren

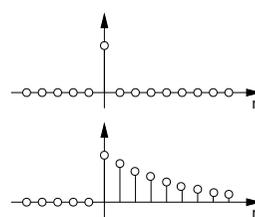
Zahlenfolgen, Elementarfolgen

Wertemenge {Y(k)}
Zuordnung: Index k -> Y(k)



wichtige "Elementarfolgen":

- Impulsfolge
- Sprungfolge
- Exponentialfolge
- Sinusfolge



lineare Systeme:

- Signale aus Elementarfolgen zusammensetzen
- Systemverhalten entsprechend berechenbar

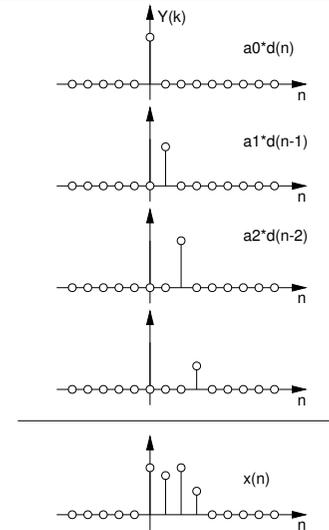
Impulsfolge:

Konstruktion beliebiger Zahlenfolgen:

- Summe
- skaliertes
- zeitverschobener Impulsfolgen

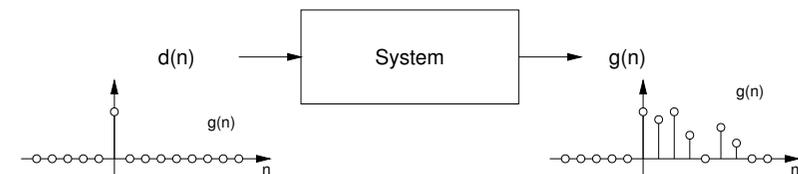
$$x(n) = a_0 * d(n) + a_1 * d(n-1) + \dots$$

- gleichwertig: Konstruktion mit
- skalierten, zeitverschobenen
- Sprungfolgen



Impulsantwort

"Impulsantwort" := Ausgangsfolge als Reaktion auf die Impulsfolge



=> liefert vollständige Beschreibung von LTI-Systemen
"linear, time-invariant"

z.B. Verstärker, Filter, usw.

=> auch als lineare Approximation nichtlinearer Systeme

Differenzgleichungen

Berechnung des Ausgangswerts $Y(n)$

- aus gegenwärtigem Eingangswert $X(n)$
- aus früheren Eingangswerten $X(n-k)$
- aus früheren Ausgangswerten $Y(n-k)$ (Rekursion)
- besonders wichtig: linear, konstante Koeffizienten, 2. Ordnung

$$y(n) = A_0 x(n) + A_1 x(n-1) + A_2 x(n-2) - B_1 y(n-1) - B_2 y(n-2)$$

Ordnung N: $B_1 \dots B_N$ kommen vor, $A_1 \dots A_M$, $M < N$

- Übertragungsfunktion $H(z)$ durch Z-Transformation:

$$Y(z) = A_0 X(z) + A_1 X(z)/z + A_2 X(z)/z^2 - B_1 Y(z)/z + B_2 Y(z)/z^2$$

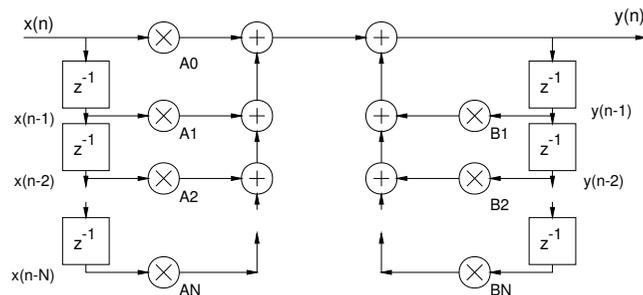
DGL: Direktform

Differenzgleichungen, linear, konstante Koeffizienten

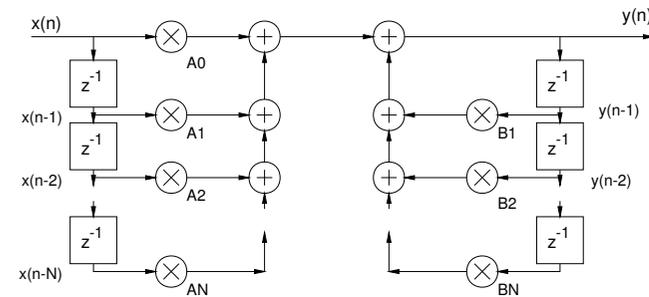
$$y(n) = A_0 x(n) + A_1 x(n-1) + A_2 x(n-2) - B_1 y(n-1) - B_2 y(n-2)$$

Umsetzung in "Direktform":

- Addition, Multiplikation, Verzögerung (=Register)



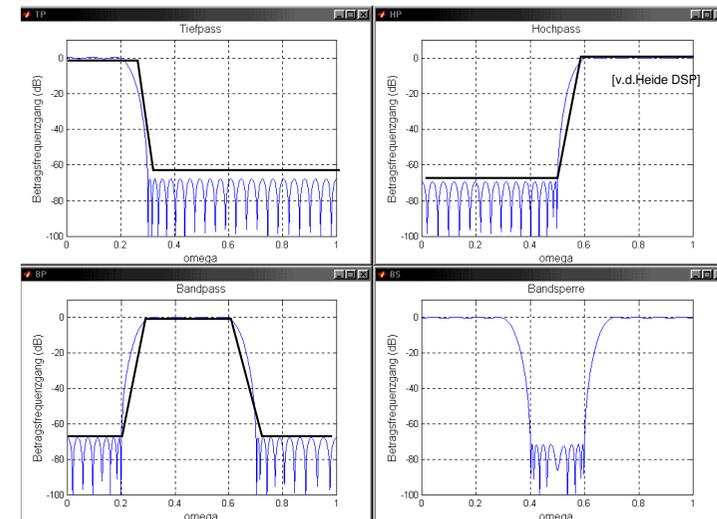
DGLs: Blockschaltbilder



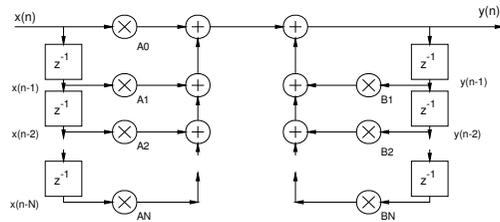
- Direktform erlaubt sofortige Realisierung von DSP-Algorithmen
- Varianten möglich (Vertauschen von Registern/Operatoren)
- bei Bedarf zusätzlich weitere (z.B. nicht-lineare) Operatoren

=> Darstellung mit Blockschaltbildern

Filter: Tiefpaß, Hochpaß, etc.



Filter: Frequenzgang



- normierte Frequenz $F = f/FA$
- direkte Berechnung des Frequenzgangs via $H(z)$:

$$|\underline{A}(j\omega)| = \sqrt{\frac{\left[\sum_{k=0}^n \alpha_k \cos 2\pi k F \right]^2 + \left[\sum_{k=0}^n \alpha_k \sin 2\pi k F \right]^2}{\left[\sum_{k=0}^n \beta_k \cos 2\pi k F \right]^2 + \left[\sum_{k=0}^n \beta_k \sin 2\pi k F \right]^2}}$$

Abtasttheorem

macht die Verarbeitung von Zahlenfolgen überhaupt Sinn ?

Abtasttheorem:

ein auf die halbe Abtastfrequenz (=Nyquistfrequenz)
bandbegrenzte Signal kann nach der Abtastung
perfekt rekonstruiert werden

- Grundlage aller digitalen Signalverarbeitung
- Beweis mit Fouriertransformation
- siehe [Shannon 48]
- siehe Matlab-Skript Signalverarbeitung

Abtasttheorem

7.4 Abtasttheorem für Zeitfunktionen

Ein reelles bandbegrenzte Signal $u(t)$ hat eine Signalgrenzfrequenz f_{gs} . Für die Fourier-Transformierte $\underline{U}(f)$ dieses Signals gilt:

$$\underline{U}(f) = 0 \text{ für } |f| > f_{gs} \quad (7.5)$$

Durch periodische Wiederholung von $\underline{U}(f)$ mit der Periode $2 f_{gs}$ entsteht eine periodische Frequenzfunktion, die in eine Fourier-Reihe

$$\underline{U}_{per}(f) = \sum_{n=-\infty}^{\infty} c_n \exp(-j 2\pi n \frac{f}{f_{gs}}) \quad (7.6)$$

entwickelt werden kann. Die Zeitfunktion

$$u(t) = \int_{-\infty}^{\infty} \underline{U}(f) \exp(j 2\pi f t) df \quad (7.7)$$

ergibt sich als inverse Fourier-Transformierte der Frequenzfunktion $\underline{U}(f)$. Da diese

(Gerdsen u. Kröger)

Abtasttheorem

nur im Bereich

$$-f_{gs} < f < +f_{gs} \quad (7.8)$$

von null verschieden ist und innerhalb dieses Bereiches mit $\underline{U}_{per}(f)$ übereinstimmt, kann für die Zeitfunktion auch

$$u(t) = \int_{-f_{gs}}^{f_{gs}} \left\{ \sum_{n=-\infty}^{\infty} c_n \exp(-j 2\pi n \frac{f}{f_{gs}}) \right\} \exp(j 2\pi f t) df \quad (7.9)$$

geschrieben werden. Man fasst die e-Funktionen zusammen und bildet die Summe der Integrale. So erhält man:

$$u(t) = \sum_{n=-\infty}^{\infty} c_n \int_{-f_{gs}}^{f_{gs}} \exp(-j 2\pi f t + \frac{j 2\pi n f t}{f_{gs}}) df \quad (7.10)$$

Die Lösung des Integrals in dieser Summe ergibt:

$$\int_{-f_{gs}}^{f_{gs}} \exp(-j 2\pi f t + \frac{j 2\pi n f t}{f_{gs}}) df = 2 f_{gs} \frac{\sin(2\pi f_{gs} t (\frac{n}{2f_{gs}} - t))}{2\pi f_{gs} t (\frac{n}{2f_{gs}} - t)} \quad (7.11)$$

Zur Bestimmung der Fourier-Koeffizienten c_n wird

$$t = k \frac{1}{2 f_{gs}} \quad k \text{ ganze Zahl} \quad (7.12)$$

gesetzt. Dann sind in der Summe (7.10) alle Terme mit $n \neq k$ null. Damit gilt für

Abtasttheorem

die Koeffizienten:

$$s_n = \frac{1}{f_{gs}} u\left(n \frac{1}{2f_{gs}}\right) \quad (7.13)$$

So entsteht aus (7.10) die Interpolationsformel

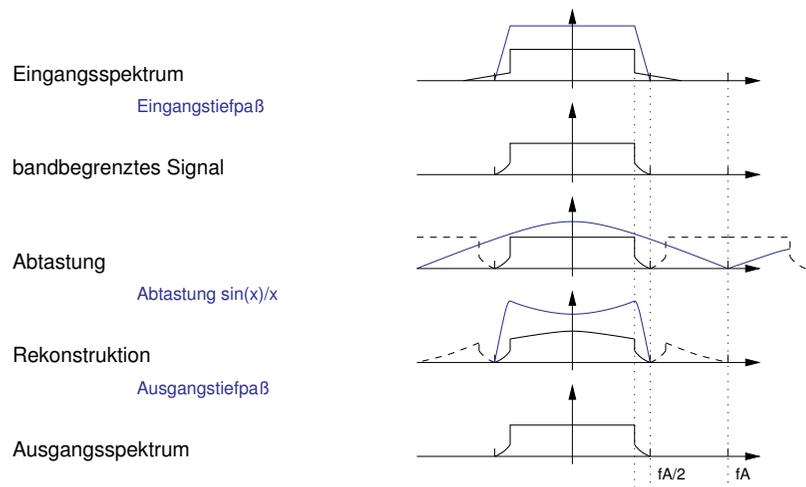
$$u(t) = \sum_{n=-\infty}^{\infty} u\left(n \frac{1}{2f_{gs}}\right) \frac{\sin\left(2\pi f_{gs}\left[t - \frac{n}{2f_{gs}}\right]\right)}{2\pi f_{gs}\left[t - \frac{n}{2f_{gs}}\right]} \quad (7.14)$$

die aussagt, daß ein bandbegrenzt Signal $u(t)$ mit der Signalgrenzfrequenz f_{gs} vollständig beschrieben ist durch seine Abtastwerte zu den Zeiten

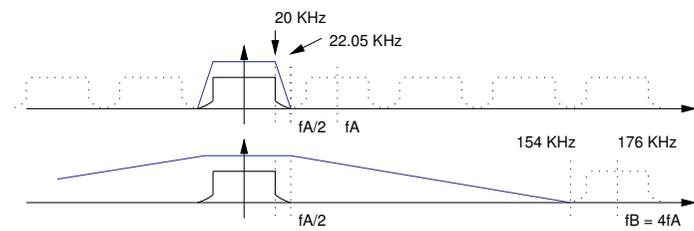
$$t_k = k \frac{1}{2f_{gs}} \quad k \text{ ganze Zahl} \quad (7.15)$$

Dieses Abtasttheorem für Zeitfunktionen ist von grundlegender Bedeutung für die digitale Signalverarbeitung.

Abtasttheorem: Schema



Abtasttheorem: Überabtastung



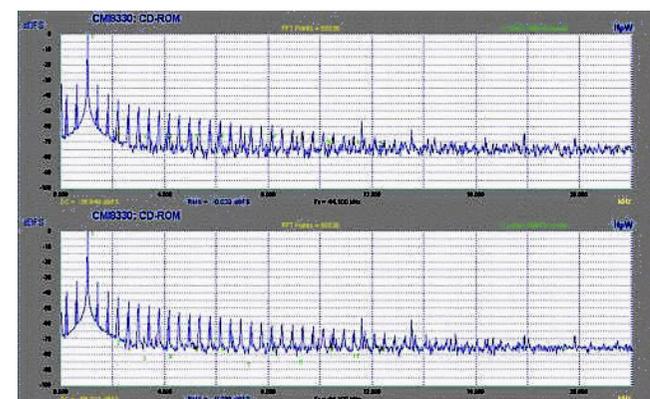
Abtasttheorem:

- vollständige Rekonstruktion des Signals, aber:
- erfordert analogen Tiefpaß sehr hoher Qualität
- Beispiel CD: Dämpfung 96 dB von 20 KHz bis 22.05 KHz
- typische analoge Filter: 6 .. 24 dB / Oktave

Oversampling:

- digitale Neu-Abtastung mit höherer Rate (z.B. 4x, 8x)
- digitale Tiefpaßfilterung beseitigt die Spiegelfrequenzen
- einfacher analoger Tiefpaß genügt

Abtasttheorem: Aliasing-Fehler . . .



(c) 10/98

- Meßwerte der CMI 8330 Soundkarte
- Sinussignal und massive Anti-Aliasing Fehler (Spiegelfrequenzen)
- massiver Qualitätsverlust durch Stümperei

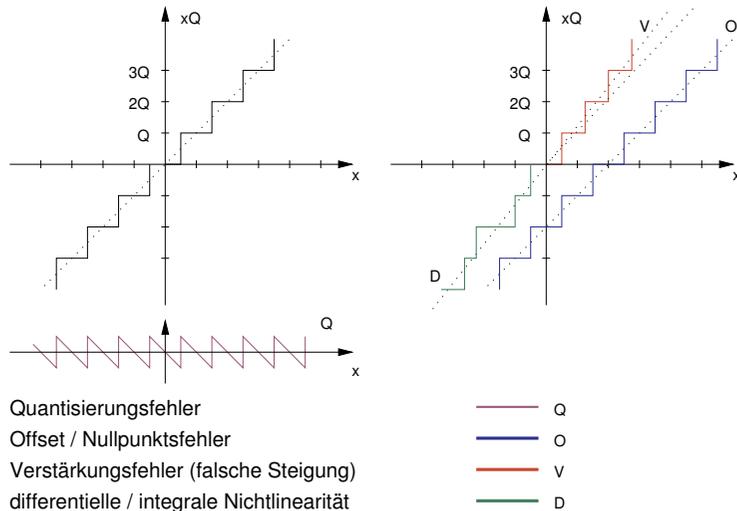
A/D-Wandlung

Digitalisierung eines Eingangssignals:

- Zeitquantisierung (Abtaster, Sample&Hold)
 - Amplitudenquantisierung (A/D-Konverter)
 - Codierung (meistens 2-Komplement)
-
- nur für bandbegrenzte Eingangssignale
 - analoger Tiefpaß vor dem Eingang erforderlich
-
- diverse Verfahren
 - Trade-off zwischen Performance / Genauigkeit / Kosten
 - für Audio: 24-bit Wandler bereits erhältlich

Medientechnik | WS 2001 | 18.204

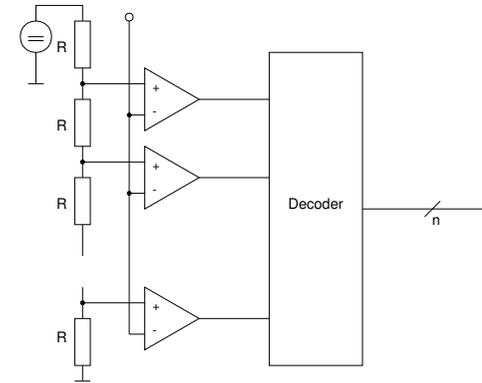
A/D: Parameter



- Quantisierungsfehler
- Offset / Nullpunktsfehler
- Verstärkungsfehler (falsche Steigung)
- differentielle / integrale Nichtlinearität

Medientechnik | WS 2001 | 18.204

A/D: Parallel-Umsetzer



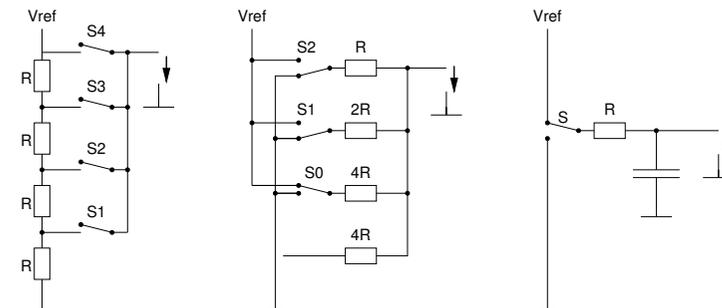
- erfordert $2 \cdot n$ Komparatoren
- sehr schnell
- aber nur für geringe Auflösungen (z.B. 8-bit)

Medientechnik | WS 2001 | 18.204

D/A-Wandlung

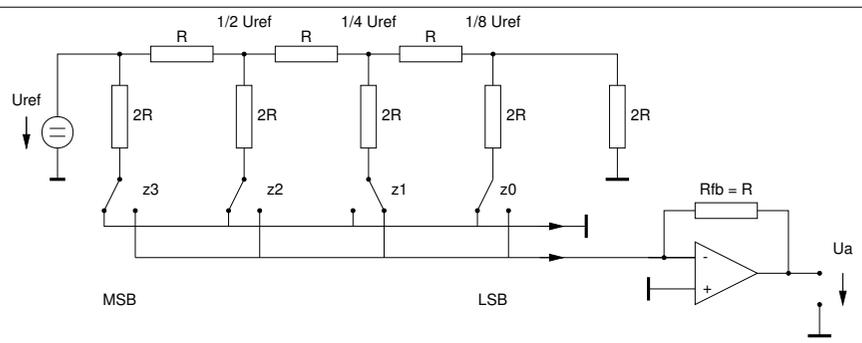
D/A: Umwandlung numerischer Werte in analoge Werte (Spannungen)

- Parallelverfahren
- Wägeverfahren
- Zählverfahren



Medientechnik | WS 2001 | 18.204

D/A: R-2R Netzwerk



- VLSI: Herstellung unterschiedlicher Widerstände schwierig
- aber gleiche Widerstände mit hoher Genauigkeit
- => fortgesetzte Spannungsteilung mit R-2R Netzwerk
- $$U_a = -U_{ref} \cdot (R_{fb}/16R) \cdot (8z_3 + 4z_2 + 2z_1 + z_0)$$
- in CMOS-Technologie oft verwendet

Quantisierungsprobleme

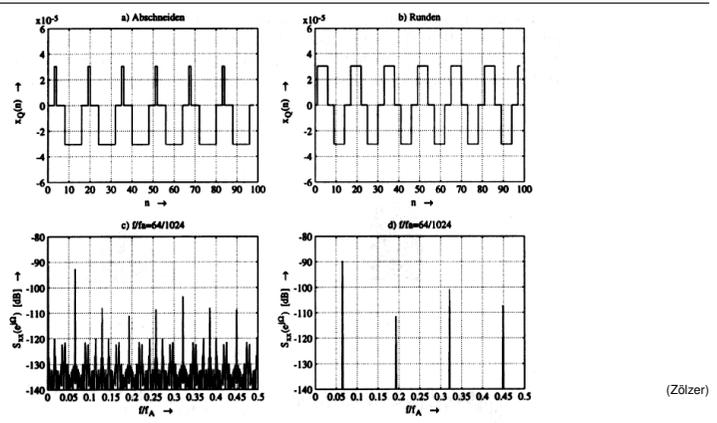
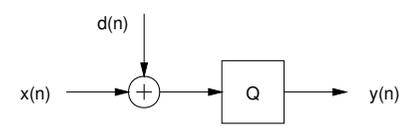


Bild 2.19: 1 Bit Aussteuerung des Quantisierers bei Abschneiden a/c und Runden b/d

- Quantisierung erzeugt störende "Nebensignale"
- Verbesserung durch Hinzufügen von Rauschen: "Dithering"

Dither: Prinzip



		bits
x(n)	Eingangssignal	w+r
d(n)	Zufallssignal (dither)	s < r
y(n)	Ausgangssignal	w

- Q: Re-Quantisierung des Signals, oder D/A-Wandler
- Amplitude des Dithers kleiner als LSB des Eingangssignals
- Verteilung (Statistik) des Dithers frei wählbar: noise shaping
- Mathematik siehe Zölzer S.38 ff

Dither: bei 0.25 bit Aussteuerung

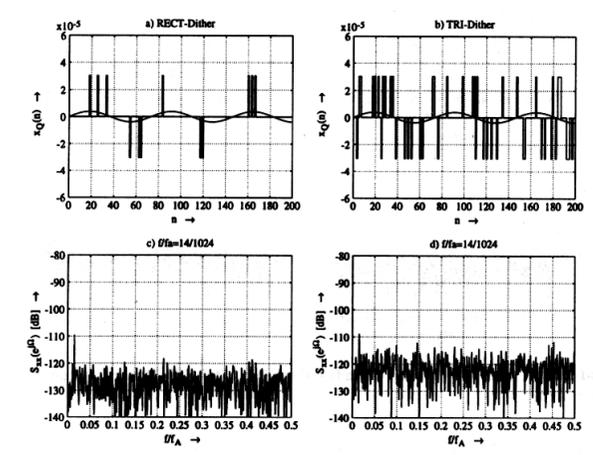


Bild 2.21: 0.25 Bit Aussteuerung - Rundungskennlinie mit RECT-Dither a/c und TRI-Dither b/d

Datenformate

effiziente Speicherung und Übertragung von Audiodaten?

- unkomprimierte Darstellung, PCM
- WAV-Format
- ADPCM
- Sprachcodecs, Kompandierung

spätere Themen:

- psychoakustische Verfahren
- Streaming
- "Meta"-Kodierung: MIDI, MPEG4 Structured Audio

Datenformate: Zahldarstellung

geeignete Zahldarstellung für Audiodaten ?!

- 8/16/24-bit Integer, Offset oder Zweierkomplement
- 32-bit Gleitkomma
- Bitströme

1000 0000

 8-bit mit Offset 128

0000 0000	0000 0000
-----------	-----------

 Zweierkomplement

0000 0000	0000 0000	0000 0000
-----------	-----------	-----------

 Zweierkomplement

0000 0000	0000 0000	0000 0000	0000 0000
-----------	-----------	-----------	-----------

 Zweierkomplement

V	EXP	MANTISSE
---	-----	----------

 IEEE 754 32-bit FP
Konvention: $-1 \leq X \leq +1$

31 24 23 16 15 8 7 0

Datenformate: Kodierung

"direct coding"		PCM
"entropy coding"	repetitive sequence suppression	zero suppression run-length encoding
	statistical encoding	pattern substitution Huffman encoding
"source coding" (e.g. speech)	transform encoding	FFT
		DCT
		...
	differential encoding	DPCM delta modulation ADPCM
vector quantization	general / fractal / ...	

Datenformate: Entropie/Quellenkodierung

Entropiekodierung:

- Eigenschaften der Datenquelle werden ignoriert
- Signalwiederholungen entfernen
- statistische Verfahren, z.B. Huffman-Kodierung
- verlustfrei und reversibel
- für Audiodaten: ca. Kompressionsfaktor 2 erreichbar

Quellenkodierung (source encoding):

- Eigenheiten der Datenquelle / senke berücksichtigen
- z.B. Frequenzgang / Maskierung / Rauschschwellen des Ohrs
- verlustfrei
- verlustbehaftet für bessere Kompression, z.B. MP3 bis ca 10:1

Datenformate: RAW

Audio-CD "raw" Format:

- direkte PCM-Kodierung
- 16-bit Samplewerte, Zweierkomplement
- 44.1 KHz Abtastrate
- Stereo
- Datenrate $44.100 \cdot 2 \cdot 16 \text{ bps} = 1411200 \text{ bps} = 176 \text{ KB/s}$
- bzw. 10.5 MB/min
- großer Speicherbedarf
- aber leicht zu bearbeiten (schneiden, skalieren, ...)
- Details später (CD/DVD)

Medientechnik | WS 2001 | 18.204

Datenformate: SND

```
typedef struct {
    int magic;           /* 0x2e736e64 = ".snd" */
    int dataLocation;   /* offset to the data */
    int dataSize;       /* number of bytes of data */
    int dataFormat;     /* 1=μ-law, 2= linear8, ... */
    int samplingRate;   /* samples per second */
    int channelCount;   /* 1=mono, 2=stereo, ... */
    char info[4];       /* optional text info */
} SNDSoundStruct;
```

- erstes Audioformat auf NeXT und Sun
- einfache Dateistruktur mit Kopf (SNDSoundStruct) und Daten
- übliche Datenrate: 8-bit mono, 8 KHz Samplerate
- diverse Datenformate von 8-bit linear bis G.723
- Zugriff über entsprechende API (NeXT Sound Kit)

Medientechnik | WS 2001 | 18.204

Datenformate: WAV

- Standard-Dateiformat für Audiodaten unter Windows
- Abkömmling des EA IFF85 Formats

"Chunk"-Format:

- Datei besteht aus einzelnen "Häppchen"
- jeder Chunk enthält eigene Headerinformation
- und optional Daten
- Format kann nachträglich um neue Chunks erweitert werden

Zugriff auf hintere Chunks:

- durch Verketteten der Länge der vorherigen Chunks
- erfordert Kenntnis aller vorangegangenen Chunks
- ungeeignet für Streaming / verlustbehaftete Kanäle

Medientechnik | WS 2001 | 18.204

Datenformate: WAV

Hierarchie mit Unzahl von Chunk-Typen:

RIFF Chunk	(Wave File Header)
Format Chunk	(Struktur des Data Chunks)
Data Chunk	(Daten, z.B. PCM Samples)
Fact Chunk	(Info über komprimierte Daten)
Cue Chunk	(Offset zu wichtigen Zeitpunkten)
Playlist Chunk	(Anspielfolge von Cuepunkten)
Associated Data Chunk	(z.B. Songtitel)
Label Chunk	(eigentlicher Titel)
...	

- oft nur drei Chunks: Header/Format/Data
- alle Chunks "word-aligned", evtl. ein Füllbyte 0 ergänzen

Medientechnik | WS 2001 | 18.204

Datenformate: WAV header

```
typedef struct {
    ID ckID; /* 0x52494646 = "RIFF" */
    long ckSize; /* file size -8 */
    ID formType; /* 0x57415645 = "WAVE" */
    char pad[]; /* padding, if ckSize odd */
} WaveChunk;
```

```
typedef struct {
    ID ckID; /* 0x666D7420 = ".fmt" */
    long ckSize; /* 16 + extra format bytes */
    short wFormatTag; /* e.g. WAVE_FORMAT_PCM */
    ushort nChannels;
    ushort nSamplesPerSec;
    ushort nAvgBytesPerSec;
    ushort nBlockAlign;
    ushort nBitsPerSample;
} FormatChunk;
```

Medientechnik | WS 2001 | 18.204

Datenformate: WAV data

```
typedef struct {
    ID ckID; /* 0x64617461 = "data" */
    long ckSize; /* in bytes */
    char[] data;
}
```

- WAVE_FORMAT_PCM: 16-bit Zweierkomplementdaten
- aber auch G.711 / G. 721 / GSM / MPEG implementiert
- Stereo/Mehrkanaldaten als Frames
links / rechts, bzw. Kanal 1, 2, 3, ...
- erlaubt das Abspielen, ohne die Datei komplett laden zu müssen

Medientechnik | WS 2001 | 18.204

AIFF: Beispiel

FORM AIFF file	ckID	'F','O','R','M'
	ckSize	99764
	formType	'A','I','F','F'
Version	ckID	'V','E','R','S'
Chunk	ckDataSize	4
	timestamp	2776318400
Common	ckID	'C','O','M','M'
Chunk	ckDataSize	38
	numChannels	1
	numSampleFrames	99611
	sampleSize	8
	sampleRate	22254.54
	compressionType	'N','O','N','E'
	compressionName	'M','A','R','K','E','T','C','O','M','P','R','E','S','S','E','D'
Marker	ckID	'M','A','R','K'
Chunk	ckDataSize	66
	numMarkers	4
	id	101
	position	318
	markerName	'9','B','E','G','I','D','I','N','M'
	id	115
	position	47829
	markerName	'9','B','E','G','I','D','I','N','M'
	id	108
	position	97127
	markerName	'9','E','N','D','I','N','M'
	id	103
	position	45233
	markerName	'9','B','E','G','I','D','I','N','M'
Sound Data	ckID	'S','I','G','N'
Chunk	ckDataSize	99619
	offset	0
	blockSize	0
	soundData	...

sample frames 99611th pad byte
sample frame

Medientechnik | WS 2001 | 18.204

DPCM, ADPCM:

DPCM, "differential PCM":

- Differenz zum vorhergehenden Samplewert abspeichern
- lohnt sich, wenn aufeinanderfolgende Werte ähnlich
- anschließend noch RLE- oder Huffman-Kodierung möglich
- für Audiodaten wenig effektiv

ADPCM, "adaptive differential PCM":

- internes Modell zur Vorhersage von Samplewerten
- Abspeichern der Differenz von Vorhersage und Samplewert
- Kompressionsraten bis ca. Faktor 2..4 erreichbar
- Rauschen / rauschartige Klänge verhindern höhere Werte
- Mathematik und Beispiel: siehe Zölzer, Kap. 9.1

Medientechnik | WS 2001 | 18.204

ADPCM: Beispiel

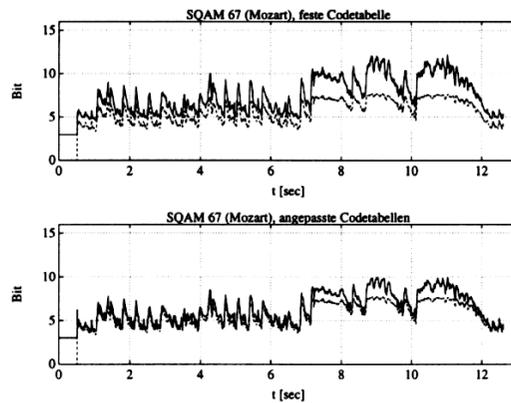


Bild 9.2: Verlustlose Datenkompression (Mozart): Wortbreite [Bit] über der Zeit (Entropie - - , lineare Prädiktion mit Huffman-Codierung —)

(Zölzer)

Medientechnik | WS 2001 | 18.204

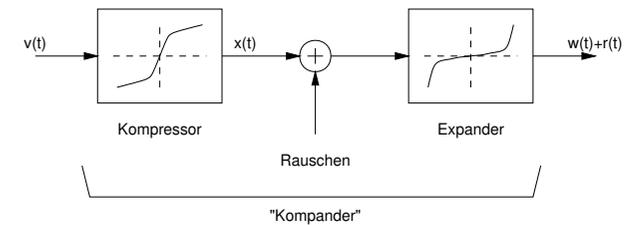
Sprach-Codecs:

Sprachkodierung vs. Audio/Musik allgemein:

- möglichst gute Verständlichkeit
- Klangverluste sind durchaus akzeptabel
- Bandbreite von < 4KHz reicht aus
- enorme Bedeutung für die Telefonie
- diverse internationale Standards etabliert
- G.711 8-bit, 8 KHz, μ -Law / a-Law (ISDN)
- G.721 32 kbps ADPCM (4bit/sample)
- G.723 24/40 Kbps ADPCM (3bit/5bit/sample)
- GSM Varianten: full/enhanced full/half-rate
- viele weitere

Medientechnik | WS 2001 | 18.204

G.711: Kompander



"Überdeckungseffekt" des Gehörs:

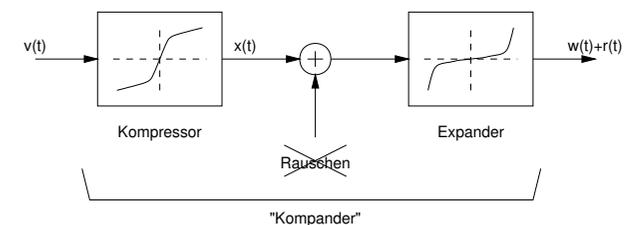
- Rauschen stört bei leisen Signalen stärker als bei lauten
- in leisen Signalen möglichst wenig Rauschen

=> leise Signale vor der Übertragung anheben, später absenken

- Herleitung der Kennlinien: siehe Kammeyer

Medientechnik | WS 2001 | 18.204

G.711: Kompander



- digitale Übertragung (ISDN) ist rauschfrei

trotzdem Einsatz der Kompanderung:

- Erhöhung des Dynamikbereichs für die Sprachübertragung
- bzw. Reduzierung des Quantisierungsrauschens (leise Signale) durch Anwendung der Kompressor-Kennlinie

Medientechnik | WS 2001 | 18.204

μ -Law, a -Law

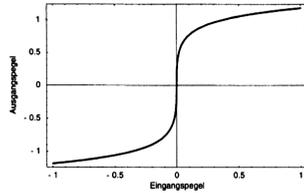


Bild 5.3: μ -Law-Kompressionskennlinie für normierte Signalpegel

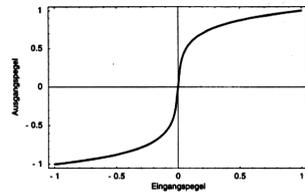


Bild 5.2: A-Law-Kompressionskennlinie für normierte Signalpegel

μ -Gesetz (μ -Law)

Diese Art der Dynamikkompression ist im angelsächsischen Raum verbreitet, sie wird manchmal irreführend als mu-Law bezeichnet. Der zu digitalisierende Wert ergibt sich aus dem Eingangspegel S (auf 1 normierter Signalpegel) als

$$S' = \text{sign}(S) \cdot \frac{1 + \ln(1 + \mu \cdot \text{abs}(S))}{\ln(1 + \mu)}, \quad \mu = 255$$

A-Gesetz (A-Law)

Der zu digitalisierende Wert ergibt sich aus dem Eingangspegel S (auf 1 normierter Signalpegel) als logarithmische Funktion, die für praktische Zwecke meist in linearen Stücken approximiert wird (siehe Tabelle 5.2). Mit einem konstanten Wert $A = 87.6$ lautet die Transferfunktion

$$S' = \frac{\text{sign}(S)}{1 + \ln A} \begin{cases} A \cdot \text{abs}(S) & \text{wenn } \text{abs}(S) \leq 1/A \\ 1 + \ln(A \cdot \text{abs}(S)) & \text{sonst} \end{cases}$$

- Kompression von 12..16 bit linear auf 8 bit
- Idee: logarithmische statt linearer Kodierung
- Berechnung: stückweise lineare Approximation / Tabellen

a -Law: lineare Approximation

```

/*
 * linear2alaw() - Convert a 16-bit linear PCM value to 8-bit A-law
 *
 * Linear Input Code      Compressed Code
 * -----
 * 0000000wxyz          000wxyz
 * 0000001wxyz          001wxyz
 * 000001wxyzab         010wxyz
 * 00001wxyzabc         011wxyz
 * 0001wxyzabcd         100wxyz
 * 001wxyzabcde         101wxyz
 * 01wxyzabcdef         110wxyz
 * 1wxyzabcdefg         111wxyz
 *
 * For further information see John C. Bellamy's Digital Telephony, 1982,
 * John Wiley & Sons, pps 98-111 and 472-476.
 */

```

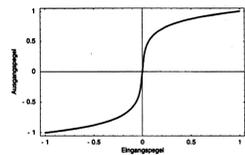


Bild 5.2: A-Law-Kompressionskennlinie für normierte Signalpegel

a -Law: Umrechnung

```

unsigned char _a2u[128] = {          /* A- to u-law conversions */
    1,  3,  5,  7,  9, 11, 13, 15,
    16, 17, 18, 19, 20, 21, 22, 23,
    24, 25, 26, 27, 28, 29, 30, 31,
    32, 32, 33, 33, 34, 34, 35, 35,
    36, 37, 38, 39, 40, 41, 42, 43,
    44, 45, 46, 47, 48, 48, 49, 49,
    50, 51, 52, 53, 54, 55, 56, 57,
    58, 59, 60, 61, 62, 63, 64, 64,
    65, 66, 67, 68, 69, 70, 71, 72,
    73, 74, 75, 76, 77, 78, 79, 79,
    80, 81, 82, 83, 84, 85, 86, 87,
    88, 89, 90, 91, 92, 93, 94, 95,
    96, 97, 98, 99, 100, 101, 102, 103,
    104, 105, 106, 107, 108, 109, 110, 111,
    112, 113, 114, 115, 116, 117, 118, 119,
    120, 121, 122, 123, 124, 125, 126, 127};

```

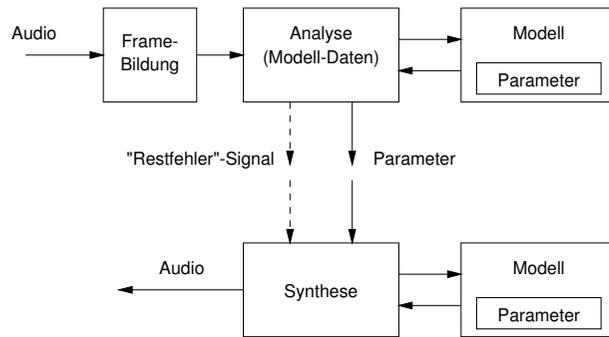
- Konvertierung zwischen a-Law und μ -Law mit Tabellen
- beträchtliche Rundungs/Quantisierungsfehler

CELP

"Code(book) Excited Linear Prediction"

- spezieller Codec für Sprachkodierung bei 4.8 kbps
- verwendet Modell für Sprache (Vokaltrakt)
- Modellparameter sind zeitabhängig
- Anpassung der Modellparameter an die Eingabedaten
- LPC: Übertragung der Modellparameter
- Empfänger liest Parameter, Modell erzeugt die Ausgabedaten
- Sprachverständlichkeit ist ok, aber "synthetischer" Klang
- CELP: Berechnung eines Fehlersignals (Modell - Samples)
- Übertragung der Modellparameter und des Fehlersignals

CELP: Blockschaltbild



LPC: einfaches, lineares Modell

CELP: Suche nach besten Parametern in (fester) Codetabelle

GSM

"Global Standard for Mobile Communication"

- spezieller Codec für Sprachkodierung
- gute Sprachverständlichkeit
- ungeeignet für Musiksignale
- möglichst einfacher Dekoder (Mobilgeräte!)
- trotzdem sehr rechenaufwendig
- mehrere Datenraten: full / enhanced-full / half-rate
- z.B. full-rate mit 13.3 kb/s
13-bit Samples, 8 KHz, 160 Samples -> 260 bits
- Dokumentation und Demo-Code:
<http://kbs.cs.tu-berlin.de/~jutta/toast.html>

Sprach-Codex: Sprachqualität?

Gauging the speech quality is an important but also very difficult task. The signal-to-noise ratio (SNR) is one of the most common objective measures for evaluating the performance of a compression algorithm. This is given by:

$$SNR = 10 \log_{10} \left\{ \frac{\sum_{n=0}^M s^2(n)}{\sum_{n=0}^M (s(n) - \hat{s}(n))^2} \right\} \quad (1)$$

where $s(n)$ is the original speech data while $\hat{s}(n)$ is the coded speech data. The SNR is a long term measure for the accuracy of speech reconstruction and as such it tends to "hide" temporal reconstruction noise particularly for low level signals. Temporal variations of the performance can be better detected and evaluated using a short-time signal-to-noise ratio, i.e., by computing the SNR for each N-point segment of speech. A performance measure that exposes weak signal performance, is the segmental SNR (SEGSNR) which is given by

$$SEGSNR = \frac{10}{L} \sum_{i=0}^{L-1} \log_{10} \left\{ \frac{\sum_{n=0}^{N-1} s^2(iN+n)}{\sum_{n=0}^{N-1} (s(iN+n) - \hat{s}(iN+n))^2} \right\} \quad (2)$$

Sprach-Codex: Bitrate vs. MIPS

Algorithm	Bit Rate (bits/sec)	MOS/DRT/DAM	MIPS*	References
PCM (G.711)	64k	4.3/95/73	0.01	[150][152]
ADPCM (G.721)	32k	4.1/94/68	~2	[22][32][150]
LD-CELP (G.728)	16k	4.0 ⁺ /-/	~19	[35][38]
RPE-LTP (GSM)	13k	3.47 ⁺ /-/	6	[119][307]
Skyphone-MPLP	9.6k	3.4/-/	11	[25]
VSELP (IS-54)	8k	3.45 ⁺ /-/	13.5	[70][100]
CELP (FS1016)	4.8k	3.2/93.7/62.2	16	[30][78]
STC-1	4.8k	3.52/92.7/63.	13	[210][212][213]
IMBE	4.15k	3.4/-/	3	[26][121][141]
STC-2	2.4k	2.9/90.1/56	13	[210][212][213]
LPC-10e (FS 1015)	2.4k	2.3/89.9/52.3	~7	[77][301]
LPC-LSP	800	-/91.2/-	~20	[166]

~ estimated, + low score reported, * processor dependent

The above complexity and performance figures were obtained from different sources and correspond to different implementation platforms and test environments. Therefore the performance and complexity figures do not always constitute an absolute measure for comparison.

Perceptual Coding: Agenda

- Motivation für psychoakustische Kodierung
- Übersicht der wichtigsten Verfahren
- Literatur

MPEG1 - Audio:

- Konzept
 - Encoder / Decoder
 - Layer 1 / 2 / 3
 - Einige Details der Algorithmen
 - Beispiele zur Klangqualität
- Ausblick auf MPEG2 und MPEG4
 - Ausblick auf MS-Audio (Windows Media) und RealAudio



Medientechnik | WS 2001 | 18.204

Perceptual Coding: Motivation

perceptual coding = psychoakustische Kodierung :=

- verlustbehaftete Audiokodierung
- minimale Dateigröße bei möglichst hoher Klangqualität
- Kompression bis ca. 10:1 bei CD-Qualität: 2 bits / sample (!)
- durch Ausnutzen der Gehörwahrnehmung:
 - Ruhegehörschwelle
 - Maskierungseffekte
- erfordert psychoakustische Modelle des Gehörs
- erfordert (Frequenz-) Analyse der Eingangsdaten:
 - "subband coders"
 - "transform coders"

Medientechnik | WS 2001 | 18.204

Perceptual Coding: Verfahren

Verfahren	Methode	kHz	Kanäle	kb/s	Qualität
MPEG-1 layer 1	hybrid	32, 44.1, 48	1, 2	32 .. 448	AM / FM / (CD)
MPEG-1 layer 2	hybrid	32, 44.1, 48	1, 2	32 .. 384	AM / FM / (CD)
MPEG-1 layer 3	hybrid	32, 44.1, 48	1, 2	32 .. 320	AM / FM / CD
MPEG-2 AAC	hybrid	16 .. 48	1 .. 5.1	16 ..	AM / FM / CD
ATRAC	subband	44.1	2	256/ch	CD
Dolby AC-2	trans.	44.1	2	256/ch	CD
Dolby AC-3	trans.	44.1	1 .. 5.1	32 .. 384	CD
AT&T PAC	subband	44.1	1 .. 5.1	128/stereo	FM / CD
MS Audio ASF	?	44.1	2	10 ..	AM / FM / CD

- viele weitere Verfahren vorgeschlagen / erforscht
- detaillierte Übersicht siehe Spanias Review

Medientechnik | WS 2001 | 18.204

Perceptual Coding: Literatur

- T. Painter, A. Spanias: Perceptual Coding of Digital Audio, Proc. IEEE 88-4, p.451ff (2000)
- T. Painter, A. Spanias: Review of Algorithms for Perceptual Coding of Audio Signals
www.eas.asu.edu/~spanias/papers.html
- D. Pan, An Overview of the MPEG/Audio Compression Algorithm, Proc. SPIE 260 (1997)
- S. Shlien, Guide to MPEG-1 Audio Standard, IEEE Trans. Broadcasting 40-4, p.206 (1994)
- Microsoft / RealAudio: msdn.microsoft.com/windowsmedia/, www.realaudio.com
- Fraunhofer IIS Dokumentation, www.iis.fhg.de
- LAME MP3 Encoder, www.sulaco.org/mp3/
- ISO Reference Software, ftp.tnt.uni-hannover.de/mpeg2/software/
- ISO/IEC Standards:
- MPEG-1, part 3: audio, ISO/IEC 11172-3 (1993)
 - MPEG-2, part 3: audio, ISO/IEC 13818-3 (1994) (Mehrkanal)
 - MPEG-4, part 3: audio
- c't, MP3-Feature (Algorithmus, Tools, Klangtest), c't 03/2000 p.122-151
- c't, MP3 vs MSAudio vs. RealAudio (mit Klangtest), c't 23/2000 p.152-161

Medientechnik | WS 2001 | 18.204

MPEG-1 Audio

- erster internationaler Standard für (Hifi-) Audio-Kompression
- entwickelt im Rahmen von MPEG-1
- bei Bitraten bis 1.5 Mb/s für Video mit Audio (Video-CD)
- verabschiedet Ende 1992 als ISO/IEC 11172-3 (1993)
- definiert das Format des Audio-Bitstroms
- und die Funktion des Decoders
- aber nicht den Encoder: Raum für spätere Verbesserungen
- Format geeignet für Sprache und Musik
- keine Quellenkodierung / keine Annahmen über Herkunft des Signals
- statt dessen psychoakustisches Modell
- Ausnutzen von Maskierungseffekten
- drei Varianten (layers), jeweils mono / stereo / joint stereo
- Sampleraten 32, 44.1, 48 KHz, Bitraten 32 .. 224 kb/s/channel

Medientechnik | WS 2001 | 18.204

MPEG-1 Audio: Varianten

- drei Varianten, Bandbreite gesamt bis 1.5 Mb/s für Video+Audio
- zwei psychoakustische Modelle
- Encoder dürfen eigene psychoakustische Modelle verwenden

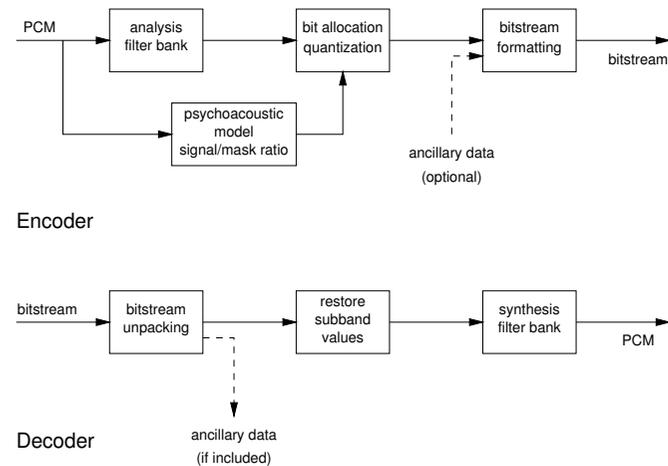
Layer 1: einfachstes Verfahren
für Bitraten höher als 128 kb/s pro Kanal
eingesetzt für Philips DCC: 384 kb/s stereo

Layer 2: mittlere Komplexität
für Bitraten von 128 kb/s pro Kanal
eingesetzt für CD-I und Video-CD

Layer 3: bessere Qualität, aber deutlich komplexer
ab 64 kb/s pro Kanal, gute Qualität ab 128 kb/s
"MP3", ISDN-Übertragungen

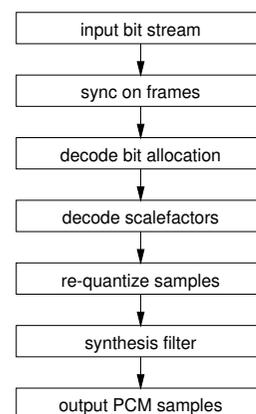
Medientechnik | WS 2001 | 18.204

MPEG-1: Blockschaltbild (grob)



Medientechnik | WS 2001 | 18.204

MPEG-1: Decoder für Layer 1 und 2



- Layer 3 Decoder deutlich komplexer (siehe ISO 11172-3 S.40)

Medientechnik | WS 2001 | 18.204

MPEG-1: Bitstrom-Header

```

audio sequence()
{
  while (true)
  {
    frame()
  }
}

header()
{
  syncword      12 bits  bslbf
  ID            1 bit   bslbf
  layer        2 bits  bslbf
  protection_bit 1 bit  bslbf
  bitrate_index 4 bits  bslbf
  sampling_frequency 2 bits bslbf
  padding_bit   1 bit  bslbf
  private_bit   1 bit  bslbf
  mode         2 bits  bslbf
  mode_extension 2 bits bslbf
  copyright    1 bit  bslbf
  original/home 1 bit  bslbf
  emphasis     2 bits  bslbf
}

frame()
{
  header()
  error_check()
  audio_data()
  ancillary_data()
}

```

- bslbf: bit string, left bit first
- Synchronisation über Bitmuster '1111 1111 1111'

Medientechnik | WS 2001 | 18.204

MPEG-1: Bitstrom-Daten, Layer 1

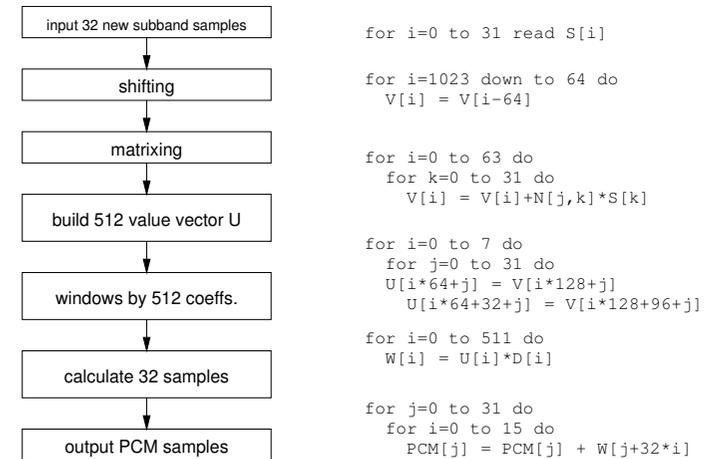
```

audio_data()
{
  if (mode==single_channel)
  {
    ...
  }
  if (mode==stereo) || (mode==dual_channel)
  {
    for (sb=0; sb<32; sb++)
    for (ch=0; ch<2; ch++)
      allocation[ch][sb] 4 bits  bsmsbf
    for (sb=0; sb<32; sb++)
    for (ch=0; ch<2; ch++)
      if (allocation[ch][sb]!=0)
        scalefactor[ch][sb] 6 bits  uimsbf
    for (s=0; s<12; s++)
    for (sb=0; sb<32; sb++)
    for (ch=0; ch<2; ch++)
      if (allocation[ch][sb]!=0)
        sample[ch][sb][s] 2..15 bits  uimsbf
  }
  if (mode==intensity_stereo)
  {
    ...
  }
}

```

Medientechnik | WS 2001 | 18.204

MPEG-1: Synthesis Subband Filter



Medientechnik | WS 2001 | 18.204

MPEG-1: Zahldarstellung

- möglichst kompakte Zahldarstellung gesucht
- also mit datenabhängiger Genauigkeit
- aber mit hohem Dynamikbereich

=> MPEG-1 Bitstrom verwendet spezielles Gleitkommaformat:

- gemeinsamer Skalenfaktor (Exponent) für eine Anzahl von Samples
- variable Anzahl nb von Nachkommastellen
- Auswahl der jeweiligen Stellenanzahl über Codetabellen
Mantisse hat nb Bits, Scalefactor aus Tabelle
- Layer 3 zusätzlich Huffman-Codierung

wert = scalefactor * (2**nb / (2**(nb-1))) * (mantisse + 2**(-nb+1))

Medientechnik | WS 2001 | 18.204

MPEG-1: Subband-Kodierung, Filterbank

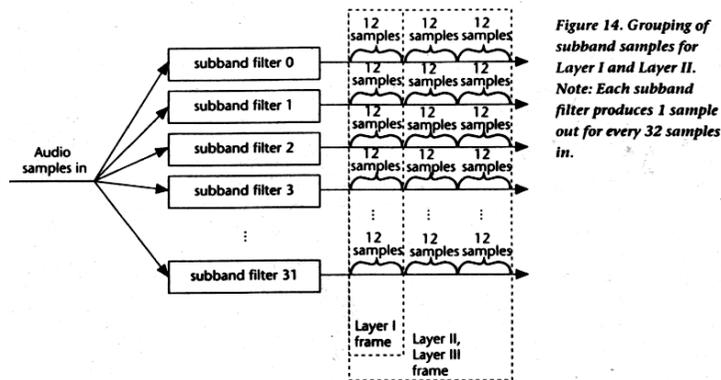
"subband-Kodierung":

- Filterbank zerlegt das Eingangssignal
- in mehrere (gleich breite) Frequenzbänder

Idee:

- Gesamtenergie des Signals verteilt sich auf die Bänder
- und zwar meistens ungleichmäßig
- wenige Bänder mit viel Energie: präzise kodieren
- viele Bänder mit wenig Energie: mit wenig Bits kodieren
- Maskierung und Hörschwelle: kleine Koeffizienten auf Null setzen
- inverse Filterbank setzt das Signal (fast) perfekt wieder zusammen

MPEG-1: Filterbank, Layer 1 und 2



- 32 Filterbänder, gleiche Frequenzbreite (je 20/32 KHz)
- Layer 1: 384 samples/frame input, 32x12 samples output
- Layer 2: 1152 samples/frame, 32x 3groups x 12 samples

MPEG-1: Filterbank, Layer 1 und 2

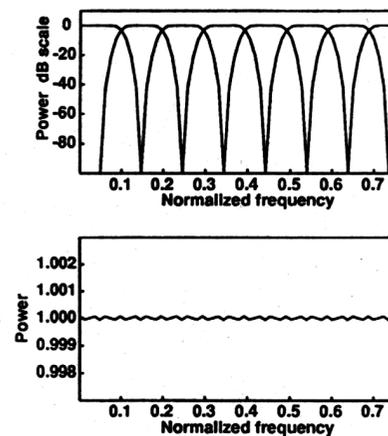
$$s_t[i] = \sum_{k=0}^{63} \sum_{j=0}^7 M[i][k] * (C[k+64j] * x[k+64j]) \quad (1)$$

where:

i is the subband index and ranges from 0 to 31,
 $s_t[i]$ is the filter output sample for subband i at time t , where t is an integer multiple of 32 audio sample intervals,
 $C[n]$ is one of 512 coefficients of the analysis window defined in the standard,
 $x[n]$ is an audio input sample read from a 512 sample buffer, and
 $M[i][k] = \cos\left(\frac{(2^i+1) * (k-16) * \pi}{64}\right)$ are the analysis matrix coefficients.

- Berechnung der Subband-Samples $St[i]$
- $M[i][k]$ ist um k bzw. j verschoben: "polyphase" Filterbank
- effizientere Berechnung via FFT möglich
- Layer 3 verwendet aufwendigere Filterbank

MPEG-1: Filterbank



Filter überlappen stark

- aber Überlagerung führt zu fast-idealem Frequenzgang

Fig. 1. Top: frequency response of the first 8 subband filters. Bottom: frequency response of the analysis filter bank. The frequency has been normalized to the Nyquist frequency 2π .

MPEG-1: Filterbank Aliasing

- einzelne Filterbänder überlappen
- Signale nahe der Übergänge werden in zwei Bänder kodiert
- Alias-Effekte möglich
- Layer-3 verwendet nachgeschaltete MDCT-Transformation
- zur Verbesserung der Frequenzauflösung
- mit Versuch der Alias-Reduzierung

Frequenzgruppen, Bark-Skala

Tabelle 9.1: Frequenzgruppen nach Zwicker 1982

z/Bark	f_u /Hz	f_o /Hz	Δf_G /Hz	f_m /Hz
0	0	100	100	50
1	100	200	100	150
2	200	300	100	250
3	300	400	100	350
4	400	510	110	450
5	510	630	120	570
6	630	770	140	700
7	770	920	150	840
8	920	1080	160	1000
9	1080	1270	190	1170
10	1270	1480	210	1370
11	1480	1720	240	1600
12	1720	2000	280	1850
13	2000	2320	320	2150
14	2320	2700	380	2500
15	2700	3150	450	2900
16	3150	3700	550	3400
17	3700	4400	700	4000
18	4400	5300	900	4800
19	5300	6400	1100	5800
20	6400	7700	1300	7000
21	7700	9500	1800	8500
22	9500	12000	2500	10500
23	12000	15500	3500	13500
24	15500			

Tonhöhenwahrnehmung?

- Messungen
- vgl. Tuningkurven
- tiefe Frequenzen: ca. 100 Hz Bandbreite
- hohe Frequenzen: log. Breite der Gruppen
- => "Bark" Frequenzskala
- Ausgangspunkt für psychoakustische Modelle

MPEG-1: Maskierung

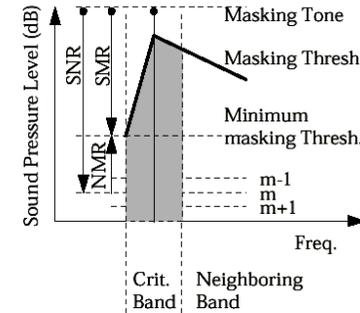
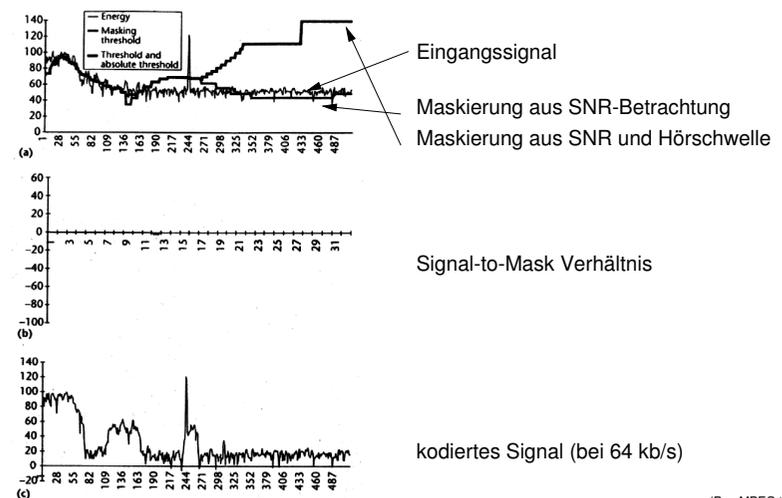


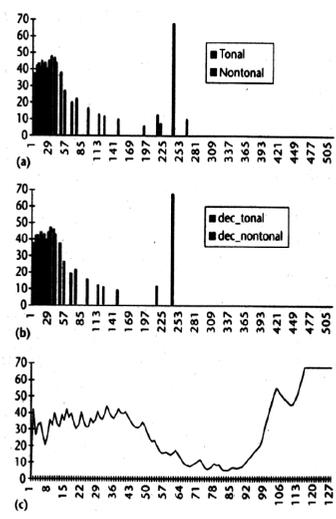
Fig. 6. Schematic Representation of Simultaneous Masking (after [30])

- lautester Ton im Frequenzband maskiert benachbarte
- Maskierung linear darstellbar, wenn Frequenz in Bark gemessen wird
- Töne leiser als der Masking-Threshold nicht wahrnehmbar
- SNR / SMR / NMR Verhältnisse

MPEG-1: psychoakustisches Modell 2



MPEG-1: psychoakustisches Modell 1



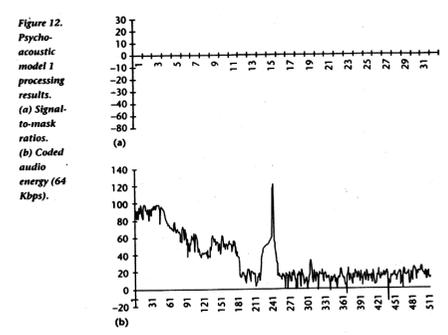
tonale und nicht-tonale Komponenten

nach der Dezimierung

masking thresholds

(Pan MPEG tutorial)

MPEG-1: psychoakustisches Modell 1



- Ergebnis der Kodierung nach Modell 1
- teilweise deutlich schlechter als Modell 2

(Pan MPEG tutorial)

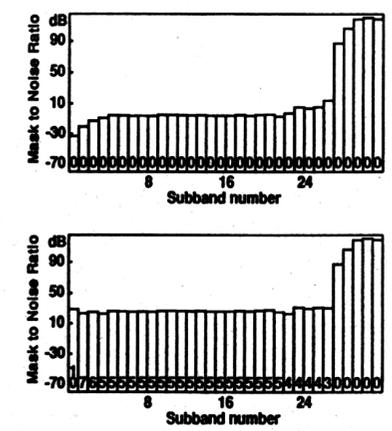
MPEG-1: bit allocation

- psychoakustisches Modell liefert "signal-to-mask" Verhältnis
- für jedes einzelne Subband

Bit-Allokation Algorithmus:

- Berechnung von MNR und SNR (mask-noise, signal-noise-ratio)
- wieviele Bits stehen zur Verfügung? Bitrate minus Sync. etc.
- Aufgabe: Bitzuweisung, so daß das minimale MNR maximal
 - für die Skalierungsfaktoren
 - und die Koeffizienten aller einzelnen Subbänder
 - Details für die Layer 1,2,3 leicht unterschiedlich
 - iterativ: berechne MNR und SNR für jedes Subband
 - weise dem Subband mit der schlechtesten MNR Bits zu
 - solange noch Bits zur Verfügung stehen

MPEG-1: bit allocation



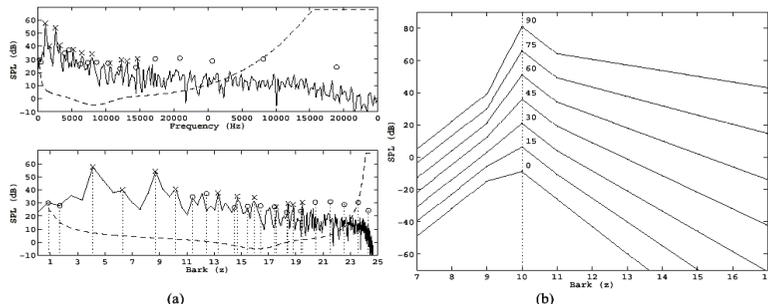
Mask-to-noise ratio

- beim Start der bit-allokation
- nach der bit-allokation
- Anzahl bits / Band

Fig. 8. Mask to noise ratio at the beginning (top) and end (bottom) of the bit allocation algorithm. The number of bits assigned to each subband sample appears at the base of each bar.

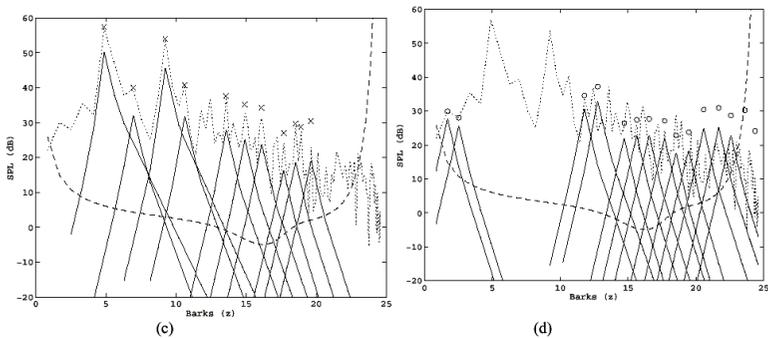
(Shlien MPEG guide)

MPEG-1: model 1, tonal / noise maskers



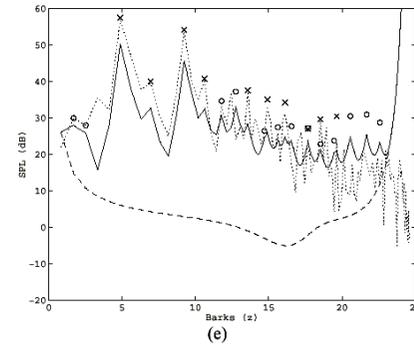
(a) *Step 1:* Obtain PSD, express in dB SPL. Top panel gives linear frequency scale, bottom panel gives Bark frequency scale. Absolute threshold superimposed. *Step 2:* Tonal maskers identified and denoted by 'X' symbol; Noise maskers identified and denoted by 'O' symbol. (b) Collection of prototype spreading functions (Eq. 28) shown with level as the parameter. These illustrate the incorporation of excitation pattern level-dependence into the model. Note that the prototype functions are defined to be piecewise linear on the Bark scale. These will be associated with maskers in steps 3, 4.

MPEG-1: model 1, masking and spreading



(c) *Steps 3,4:* Spreading functions are associated with each of the individual tonal maskers satisfying the rules outlined in the text. Note that the Signal-to-Mask Ratio (SMR) at the peak is close to the widely accepted tonal value of 14.5 dB. (d) Spreading functions are associated with each of the individual noise maskers that were extracted after the tonal maskers had been eliminated from consideration, as described in the text. Note that the peak SMR is close to the widely accepted noise-masker value of 5 dB.

MPEG-1: model 1, masking and spreading



(e) *Step 5:* A global masking threshold is obtained by combining the individual thresholds as described in the text. The maximum of the global threshold and the absolute threshold are taken at each point in frequency to be the final global threshold. The figure clearly shows that some portions of the input spectrum require SNRs of better than 20 dB to prevent audible distortion, while other spectral regions require less than 3 dB SNR. In fact, some high-frequency portions of the signal spectrum are masked and therefore perceptually irrelevant, ultimately requiring no bits for quantization without the introduction of artifacts.

MPEG-1: Bitraten für Layer 1 .. 3

MPEG-1 Bitraten (mono/stereo/joint stereo):

index	Layer I		Layer II		Layer III	
'0000'	free	format	free	format	free	format
'0001'	32	kbit/s	32	kbit/s	32	kbit/s
'0010'	64	kbit/s	48	kbit/s	40	kbit/s
'0011'	96	kbit/s	56	kbit/s	48	kbit/s
'0100'	128	kbit/s	64	kbit/s	56	kbit/s
'0101'	160	kbit/s	80	kbit/s	64	kbit/s
'0110'	192	kbit/s	96	kbit/s	80	kbit/s
'0111'	224	kbit/s	112	kbit/s	96	kbit/s
'1000'	256	kbit/s	128	kbit/s	112	kbit/s
'1001'	288	kbit/s	160	kbit/s	128	kbit/s
'1010'	320	kbit/s	192	kbit/s	160	kbit/s
'1011'	352	kbit/s	224	kbit/s	192	kbit/s
'1100'	384	kbit/s	256	kbit/s	224	kbit/s
'1101'	416	kbit/s	320	kbit/s	256	kbit/s
'1110'	448	kbit/s	384	kbit/s	320	kbit/s

- 32 .. 448 kb/s

MPEG-1: bitstream format

Header (32)	CRC (0,16)	Bit allocation (128-256)	Scale factors (0-384)	Samples	Ancillary data
-------------	------------	--------------------------	-----------------------	---------	----------------

(a)

Header (32)	CRC (0,16)	Bit allocation (26-188)	SCFSI (0-60)	Scale factors (0-1080)	Samples	Ancillary data
-------------	------------	-------------------------	--------------	------------------------	---------	----------------

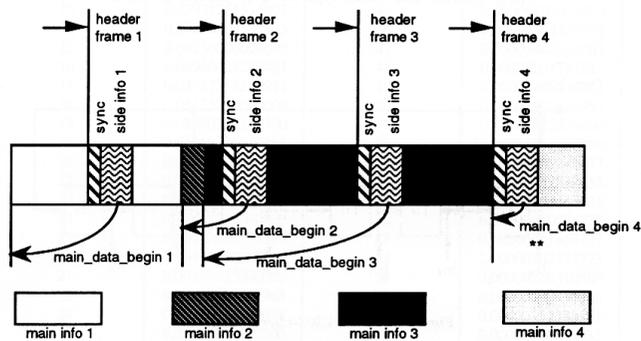
(b)

Header (32)	CRC (0,16)	Side information (136, 256)	Main data; not necessarily linked to this frame. See Figure 18.
-------------	------------	-----------------------------	---

(c)

- Header mit speziellem Sync-Muster, dann CRC
- Layer 3 erlaubt "borrowing" zwischen Frames

MPEG-1: bitstream format

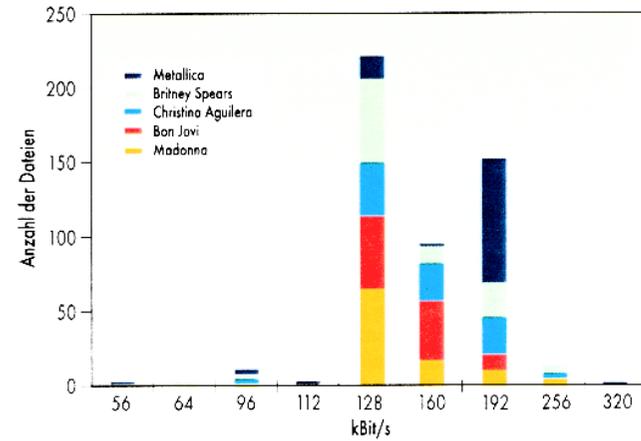


***) main_data_begin 4 == 0: This signifies that main data starts directly after the side information for frame 4. This is the lower limit for main_data_begin; main_data cannot start later than this point. Note that data bytes used by "sync" and "side info" are not counted by the main_data_begin pointer.

Note: 'info' means information

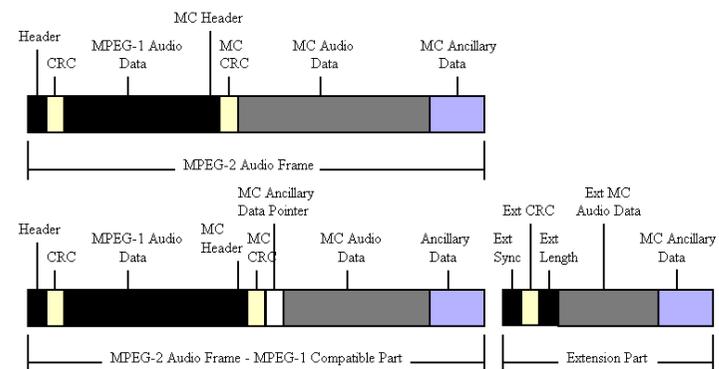
Figure A.7.b -- Layer III bitstream organization with peak demand at main info 3 and small demand at main info 2.

MPEG-1: gängige Bitraten



- Histogramm der Bitraten (MP3)
- Suche per Napster, aus (c't 2x/2000)

MPEG-2: multichannel bitstream



- MPEG-1 kompatibles Stereosignal, plus Mehrkanal-Erweiterung

Figure A.2 Structure of the ISO 13818-3 Layer II multichannel extension, using the ISO/IEC 11172-3 compatible bit stream as well as the extension bit stream

MPEG-1: pre-echos

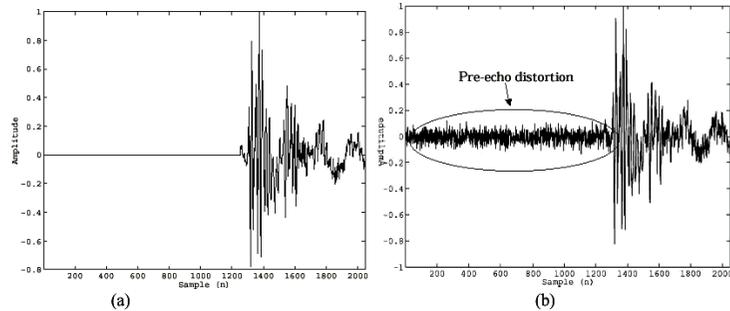
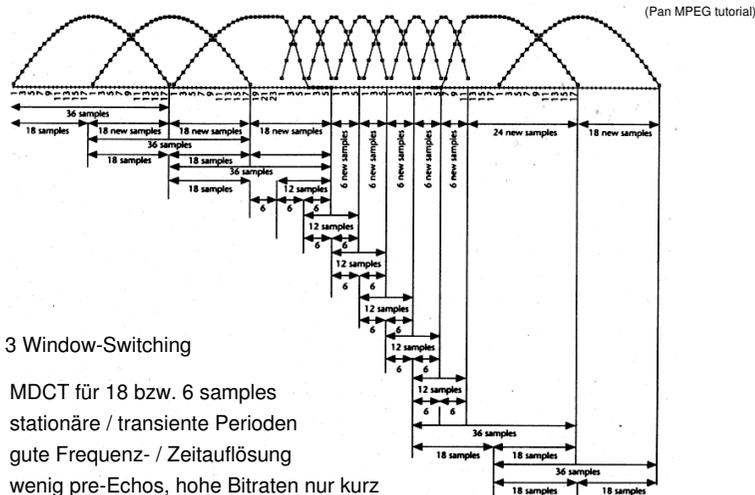


Fig. 14. Pre-Echo Example: (a) Uncoded Castanets. (b) Transform Coded Castanets, 2048-Point Block Size (Painter Spanias)

- Problem mit plötzlichen lauten Signalen nach einer Blockgrenze
 - vorheriger Block hat sehr wenig Bits kodiert
 - Rauschen des lauten Signals wird nicht mehr maskiert
- => Situation erkennen und behandeln (kürzere MDCT Blöcke)

MPEG-1: window-switching



Layer 3 Window-Switching

- MDCT für 18 bzw. 6 samples
- stationäre / transiente Perioden
- gute Frequenz- / Zeitauflösung
- wenig pre-Echos, hohe Bitraten nur kurz

MPEG-1: temporal noise-shaping

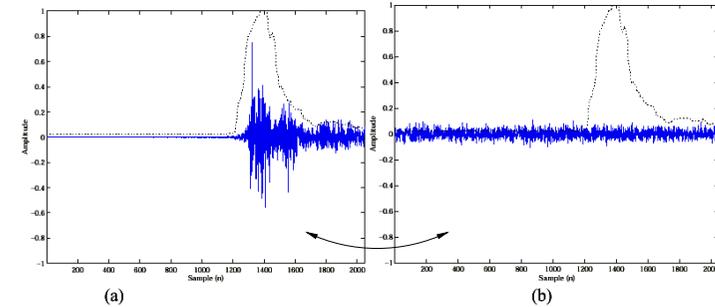


Fig. 17. Temporal Noise Shaping Example Showing Quantization Noise and the Input Signal Energy Envelope for Castanets: (a) Without TNS, and (b) With TNS

(Painter Spanias, a) b) vertauscht)

weitere Verbesserung durch TNS:

- Verlagerung des Rauschens in den Bereich mit höchster Maskierung
- wird für MPEG-2 AAC eingesetzt

MPEG-2: BC und NBC (AAC)

"backwards compatible":

- ähnliches Format wie MPEG-1 layer 3
- aber mit 3/2-Mehrkanal-Erweiterung
- abwärtskompatibel (für die 2/0 Stereosignale)
- ungeeignet für 3/2 mit Bitraten unter 640 kb/s

"non backwards compatible" / "advanced audio coding"

- gleiches Konzept wie MPEG-1 layer 3
- aber viele Verbesserungen, mit Ziel geringerer Bitrate
- viele neue Details, bessere Filterbank, bessere Zeitauflösung
- temporal noise shaping, . . .
- zusätzliche Bitraten bis herunter zu 16 kb/s/channel

MPEG-2: AAC Eigenschaften

MPEG-2 AAC (ISO 13818-7, 1997):

- Mehrkanal-Audio (3/2) bei Bitraten bis 384 kb/s
- transparente Kodierung (Hörtests) ab 320 kb/s
- mehrere "Profile": main / low / scalable sample rate profile
- bessere Filterbank als MP3:
- Frequenzauflösung 23 Hz (75Hz), Zeitauflösung 2.6 ms
- keine Hybrid-Filterbank, sondern nur MDCT
- MDCT window shape switching (stationary / transient signals)
- time prediction of transform coefficients
- more efficient Huffman code tables
- temporal noise shaping
- scalability and complexity management tools
- eingesetzt u.a. für DVD und Liquid Audio

Medientechnik | WS 2001 | 18.204

MPEG-2: AAC-Decoder Performance

"revised report on complexity of MPEG-2 AAC tools2"

"ISO/IEC JTC1/SC229/WD11/N2957 MPEG99"

welche Hardware-Anforderungen für AAC-Dekodierung?!

Decoder	Complexity
2-channel MP software decoder	40% Pentium/133
2-channel low complexity sw decoder	25% Pentium/133
5-channel MP hardware decoder	90 mm ² 0.5 µm CMOS

Huffman decode	13.3 %
inv. quantization, scaling	1.7 %
temporal noise shaping	6.6 %
prediction	43.2 %
IMDCT	19.4 %
coupling channel	15.5 %

Medientechnik | WS 2001 | 18.204

MPEG-2: und Nachfolger

- entwickelt für geringe Bitraten
- Hörtests bescheinigen (sehr) gute Qualität
- auch für kritisches Audio-Material
- aber nicht unbedingt besser als MP3
- siehe MPEG-2 quality report Hörtests / Vergleich mehrerer Algorithmen
- MPEG-4 (ISO/IEC 14496, 1998) enthält MPEG-2
- zusätzlich "objekt-orientierte" Algorithmen
- für Bitraten von 64 kb/s bis hinunter zu 200 b/s (!)
- für Sprachkodierung / -synthese
- für Musikkodierung / -synthese (a la MIDI)

Medientechnik | WS 2001 | 18.204

MPEG-1: LAME

"Lame ain't an MPEG encoder"

- MP3-Encoder unter GPL Lizenz, im Quellcode verfügbar
- ursprünglich als Patch gegen den ISO Referenzcode
- derzeit als völlig neuentwickeltes Programm
- Decoder basiert auf mpeglib (www.mpg123.de)
- mit eigenem psychoakustischem Modell
- Qualität vergleichbar mit dem FhG-Encoder
- ideal als Grundlage für Eigenentwicklungen
- unterstützt MPEG-1 layer 1 bis 3
- recht schnelle Kodierung
- www.sulaco.com/mp3 (aktuell ist Lame 3.87)

Medientechnik | WS 2001 | 18.204

LAME: psychoakustisches Modell

```

/* psymodel.c
 * Copyright (c) 1999 Mark Taylor
 *
 * compute FFTs
 * FFT data for mid and side channel is derived from L & R
 * compute energies
 * compute unpredictability of first six spectral lines
 * compute unpredictability of next 200 spectral lines
 *
 * Calculate the energy and the unpredictability in the threshold
 * calculation partitions
 *
 * convolve the partitioned energy and unpredictability
 * with the spreading function, s3_l[b][k]
 *
 * calculate the tonality of each threshold calculation partition
 * calculate the SNR in each threshold calculation partition
 * tonality = -0.299 - .43*log(ctb/ecb);
 * tonality = 0:      use NMT (lots of masking)
 * tonality = 1:      use TMN (little masking)
 *
 * at this point, tbb represents the amount the spreading function
 * will be reduced. The smaller the value, the less masking.
 *
 * long block pre-echo control.
 * compute masking thresholds for both short and long blocks
 * threshold calculation for short blocks
 * compute M/S thresholds from Johnston & Ferreira 1992 ICASSP paper
 * determine final block type
 *
 * compute side_energy / (side+mid)_energy
 * 0 = no energy in side channel
 * .5 = half of total energy in side channel
 *
 * Read long block data for converting threshold calculation
 * Read short block data for converting threshold calculation
 *
 * compute which partition bands are in which scalefactor bands
 * compute bark value and ATH of each critical band
 *
 * Now compute the spreading function, s[j][i], the value of the spread-
 * ing function, centered at band j, for band i, store for later use
 * setup stereo demasking thresholds
 */

```

LAME: Status, Performance

- Echtzeit-Kodierung ist möglich
- auch bei hohen Bitraten / guter Qualität

			vs. Echtzeit
lame --preset studio	256 kb/s	3X	
lame --preset cd	192 kb/s	3X	
lame --preset fm	112 kb/s	3X	
			auf Celeron/466

- psychoakustisches Modell liegt im Quelltext vor
- weitere Verbesserungen vorgeschlagen und geplant
- aber massive rechtliche Probleme:
- fast alle MPEG-1/2 Verfahren patentiert und lizenzpflichtig
- siehe MPEG-2 AAC Lizenzvertrag (PDF)

LAME: Presets

Presets are some shortcuts for common settings. They can be combined with -v if you want VBR MP3s.

```

=====
                phone phon+  lw   sw   fm  voice radio tape  hifi  cd  studio
--resample      8   11   11   11   32   24
--highpass     0.125 0.1
--lowpass       3.4   4   4   4   15   12   15   18   18
--lowpass-width 0   0   0   0   0   0   0   0.9  0.9
--nosshort      yes  yes  no  no  no  yes  no  no  no  no  no
                -mm  -mm  -mm  -mm  -mj  -mm  -mj  -mj  -mj  -ms  -ms
                -h   -h
-b              16   24   24   24  112  56  128  128  160  192  256
-- PLUS WITH -v -----
-V              6   4   3   3   3   4   3   3   2   1   0
-b              8  16  16  16  80  40  96  96  112  128  160
-B              24  32  56  56 256 112 256 256 320 320 320
=====

```

```

EXAMPLES:
a) --preset fm
   equal to: -mj -b112 --resample 32 --lowpass 15 --lowpass-width 0
b) -v --preset studio
   equals to: -h -ms -V0 -b160 -B320

```

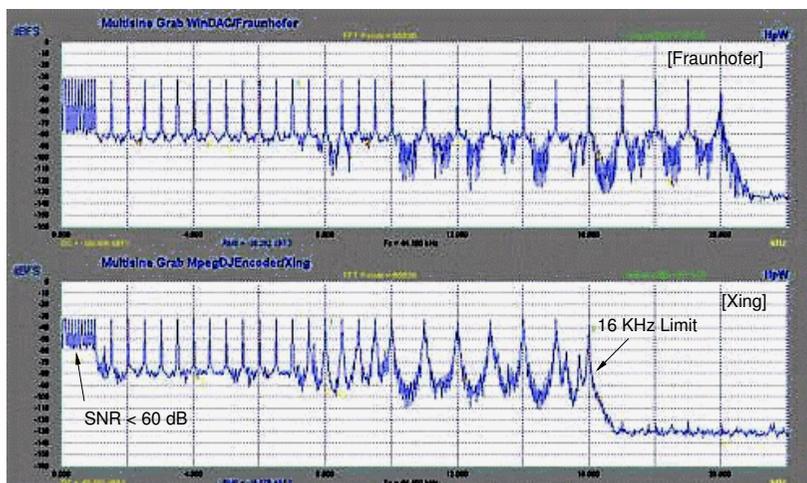
- feste Bitraten von 16 kb/s (mono) bis 256 kb/s (stereo)
- -B setzt maximal zulässigen Wert bei variabler Bitrate
- -V 0 für beste Qualität, -V 9 für minimale Dateigröße

Klangvergleich

	MP3	WMA	RA8	ePAC	AAC	MP+
Modem	o	++	+	++	-	x
Rock	+ 160	+ 160	+ 185	+ 160	o 256	++ 184
Pop	+ 160	++ 128	+ 185	++ 128	+ 192	++ 171
Klassik	+ 160	+ 160	++ 136	+ 160	+ 190	++ 158
Stresstest	+ 320	-- 192	o 360	-- 160	++ 320	++ 296

- Vergleichstest in c't 23/2000 p.152
- Klangqualität bei 128/138 kb/s / Bitrate für CD-nahe Qualität
- MP+ ist Eigenentwicklung der Autoren, "audiophil"

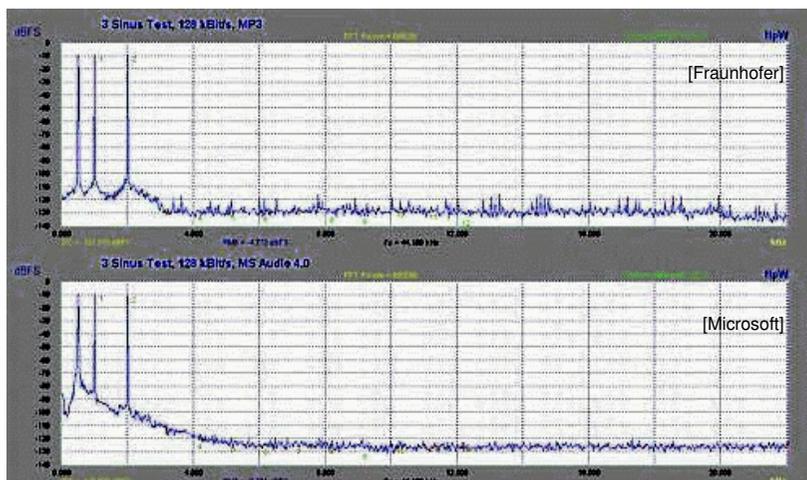
MP3: zwei Encoder . . .



- Eingangssignal mit 44 Sinustönen . . .

(c't 21/98 242)

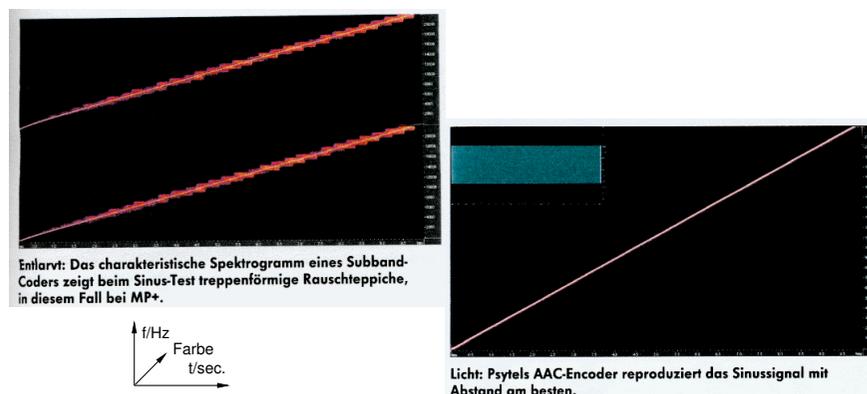
MP3: vs. MS Audio 4.0



- neuere Versionen von WMA sind besser

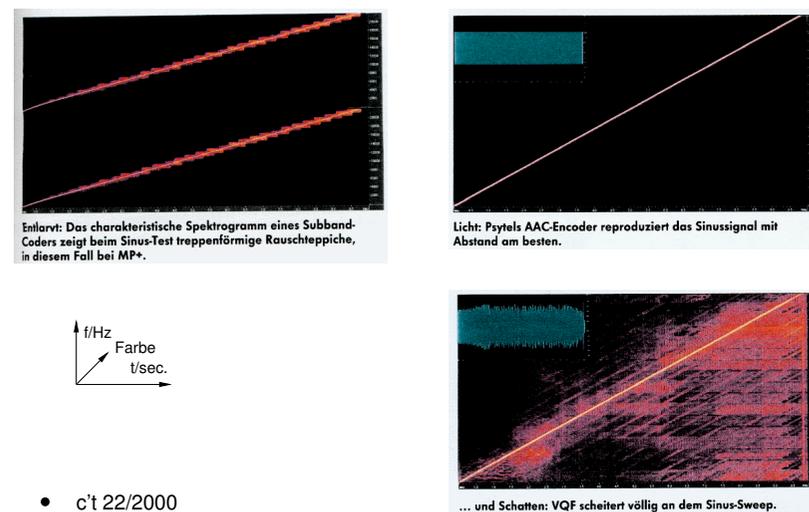
(c't 21/98 242)

MPEG-1: Sinus-Sweep



- Subband-Kodierung zerlegt das Signal in Frequenzbereiche
- Frequenz-Sweep besonders schwer zu kodieren
- Test für MP3, AAC, MP+, VQF in c't 22/2000,152

Sinus-Sweep: MP+, AAC, VCQ



- c't 22/2000

Images

- Licht
- Auge, Retina
- Farbwahrnehmung
- Farbmodelle: RGB, HSB, YUV
- Rasterbilder
- Bildformate PNM, BMP, GIF, PNG
- Filter und elementare Algorithmen
- JPEG-Kodierung
- JPEG-2000
- Wavelets
- (Vektorgraphik, Animationen)

Medientechnik | WS 2001 | 18.204

Literatur

- B. Jähne, Digitale Bildverarbeitung, Springer 1997, 3-540-61379-X
 D. Salomon, Data Compression, Springer 2000,
 Proceedings of the IEEE, Vol. 86-4, 1998, "multimedia signal processing"
 Skripte Bildverarbeitung, Farbbildverarbeitung, ...

netpbm.sourceforge.net, www.acme.com/software/pbmplus
www.jpeg.org, jj2000.epfl.ch, www.libpng.org
www.gimp.org, www.adobe.com/products/photoshop/main.html

Medientechnik | WS 2001 | 18.204

Warnung

- Thema Bildverarbeitung hier nur angerissen
- als Grundlage für Videokodierung
- zur Abschätzung der Anforderungen an Spezialrechner
- keine mathematischen Grundlagen
- keine Algorithmen, Filter, ...
- keine Computergraphik, ...
- keine Details zu Applikationen

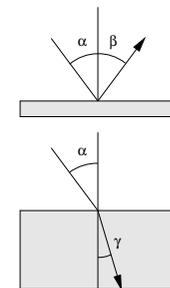
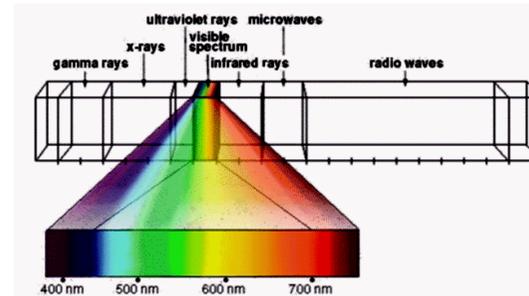
Vertiefung siehe Vorlesungen und Seminare:

- Bildverarbeitung (KOGS, IMA, ...)
- Computergraphik (TIS, ...)



Medientechnik | WS 2001 | 18.204

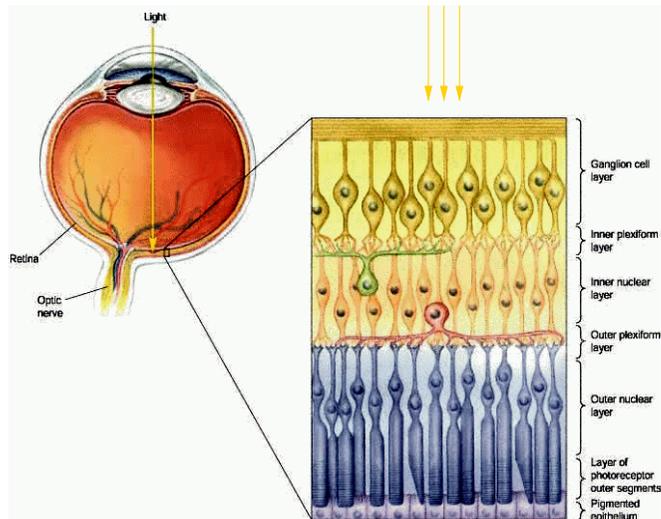
Licht:



- Licht: elektromagnetische Wellen im Bereich 400 .. 700 nm
- Brechung, Reflexion, Beugung
- Quantenmechanik: Strahlung des "schwarzen Körpers"
- Intensitätsverteilung abhängig von Temperatur: "Farbtemperatur"
 z.B. Sonnenlicht 5500K, Glühbirne 60W 2200K, Leuchtstoffröhre 4400K

Medientechnik | WS 2001 | 18.204

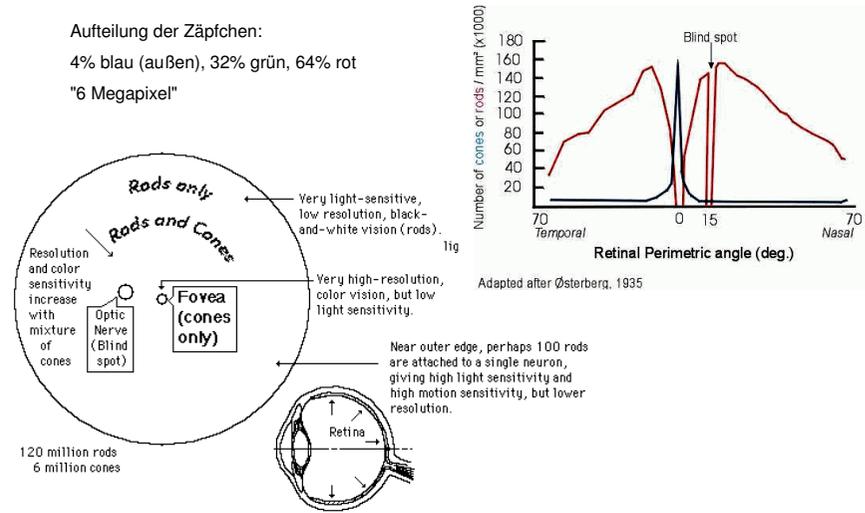
Auge: Aufbau



Medientechnik | WS 2001 | 18.204

Retina: Auflösung

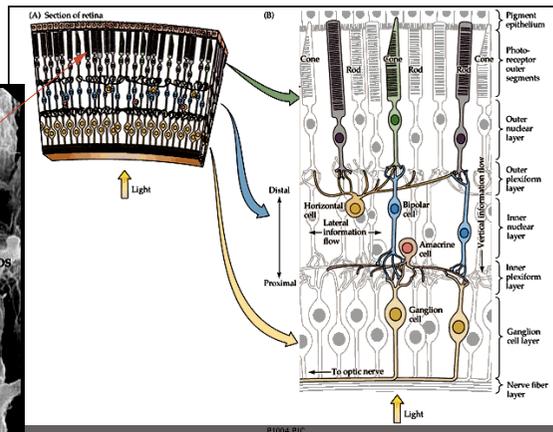
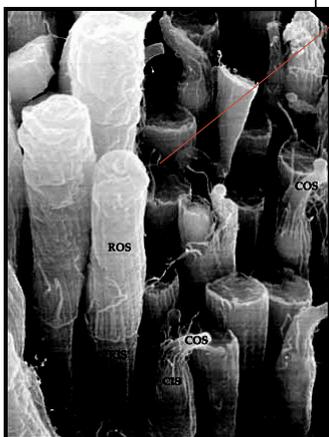
Aufteilung der Zapfen:
4% blau (außen), 32% grün, 64% rot
"6 Megapixel"



Medientechnik | WS 2001 | 18.204

Auge: Retina

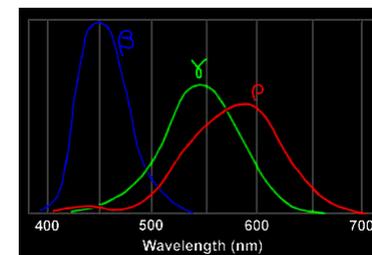
Stäbchen, "rods"
Zapfen, "cones"



- "festverdrahtete" Vorverarbeitung
- Ecken-, Kanten-, Bewegungserkennung

Medientechnik | WS 2001 | 18.204

Auge: Farbempfindung



spektrale Empfindlichkeit der blauen, grünen, roten Zapfen

Farb-Vorverarbeitung in Retina und Sehnerv:

- R - G Differenzsignal aus (rot - grün)
- Y = R + G Helligkeits- und Gelbwahrnehmung
- Y - B Differenz (gelb - blau) zur Gelb/Blau-Unterscheidung

=> ca. 128 Farbtöne, 130 Stufen Farbsättigung, 16..26 Helligkeiten

Medientechnik | WS 2001 | 18.204

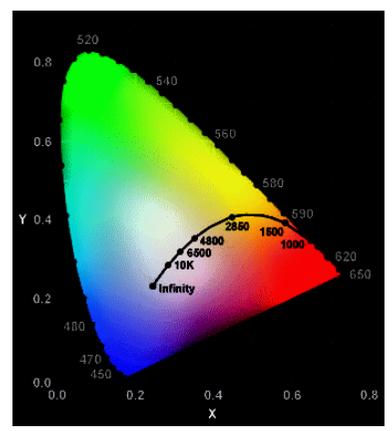
Farbe: Richtlinien

Aufbau der Retina, Vorverarbeitung: (R-G), Y=(R+G), (Y-B)

- möglichst nicht mehrere gesättigte Farben
- möglichst kein blauer Text oder feine blaue Linien:
(weil keine blauempfindlichen Zäpfchen in der Fovea)
- möglichst keine roten/grünen Details am Rand von Bildern
(weil außen an der Retina keine rot/grünempfindlichen Zäpfchen)
- keine roten Zeichen vor blauem Hintergrund
- keine benachbarten Farben, die sich nur im Blauanteil unterscheiden
- Verwendung von Farbe als einziges Mittel zur Kodierung
- usw.

(Henning 4.2.4)

Farbmodelle



CIE = comm. int. de l'elclairage
non-profit, gegründet 1913
diverse Farbmodelle / -Standards

Grundkoordinaten X, Y, Z
aufbauend auf Lichtempfindlichkeit
der blauen/grünen/roten Zäpfchen

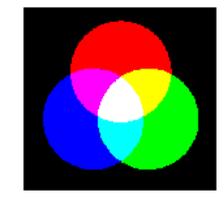
"normierte" Koordinaten (x,y,z):
 $x = X / (X+Y+Z)$
 $y = Y / (X+Y+Z)$
 $z = 1-x-y$

monochromatisches Licht:
schwarzer Körper (weiß):

liegt im CIE-Diagramm am Rand
im Zentrum, abhängig von Farbtemperatur

RGB: Additive Farbmischung

"Red Green Blue"



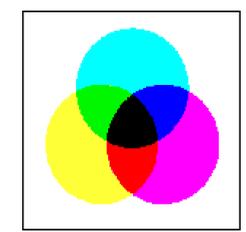
- Modell der additiven Farbmischung
- aus den Grundfarben rot, grün, blau

zyan = blau + grün
gelb = rot + grün
magenta = rot + blau
weiß = rot + blau + grün

- Anwendung beim Fernseher / Farbmonitor / Drei-Farb-LCDs
- deshalb meistens als Datenstruktur verwendet
- üblicher Wertebereich 0 .. 255 für jede Grundfarbe
- direkte Bearbeitung mit Algorithmen möglich

CMYK: Subtraktive Farbmischung

"Cyan Magenta Yellow black":



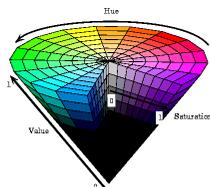
- Herausfiltern von Farben aus weissem Licht
- Grundfarben Zyan, Magenta, Gelb

- im Prinzip komplementär zu RGB
- aber unterschiedlicher Farbumfang (mit verfügbaren Farbstoffen)

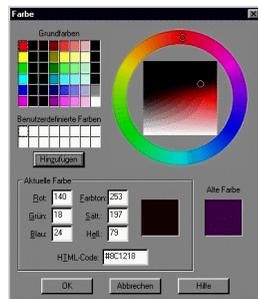
- Anwendung bei reflektierenden Medien (z.B. Papier)
- Farbdruck, "drei-Schichten" Farbfilme, ...
- weitere Farben für größeren Farbumfang
- bzw. um Farben zu sparen (insb. schwarz bei CMYK)

HSB: Hue Saturation Brightness

oder "HSV": hue saturation value



- Auswahl einer Grundfarbe (sechsstufige Pyramide)
- Sättigung
- Helligkeit
- oft als GUI zur Farbauswahl



YUV

Y = Luminanzwert der CIE-Koordinaten, Wert [0 .. 1]

UV = Farbwerte ("Chrominanz"), Werte [-0.5 .. 0.5]

$$\begin{aligned}
 Y &= 0.299 R + 0.587 G + 0.144 B \\
 U &= -0.169 R - 0.331 B + 0.500 G \\
 V &= 0.500 R - 0.419 B - 0.081 G \\
 Cb &= B - Y & U &= 0.577 Cb \\
 Cr &= R - Y & V &= 0.713 Cr
 \end{aligned}$$

- verwendet beim Farbfernsehen und für digitales Video
- abwärtskompatibel zum S/W-Fernsehen (nur Y)
- Details später, siehe Video
- andere Skalierung von UV: YCbCr-Modell

Umrechnung

$$\begin{aligned}
 C &= 1 - R \\
 M &= 1 - G \\
 Y &= 1 - B
 \end{aligned}$$

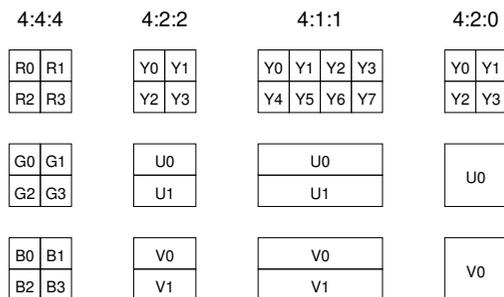
$$\begin{aligned}
 K &= \min(C, M, Y) \\
 C' &= C - K \\
 M' &= M - K \\
 Y' &= Y - K
 \end{aligned}$$

$$\begin{aligned}
 Y &= 0.299 R + 0.587 G + 0.144 B \\
 U &= -0.169 R - 0.331 B + 0.500 G \\
 V &= 0.500 R - 0.419 B - 0.081 G \\
 Cb &= B - Y & U &= 0.577 Cb \\
 Cr &= R - Y & V &= 0.713 Cr
 \end{aligned}$$

weitere Details: siehe Henning 4.4.3

Farb-Subsampling

- Stäbchen und Zäpfchen liefern Helligkeitsinformation
 - aber Farbinformation nur aus Zäpfchen
- => in der Bildkodierung und -kompression ausnutzen



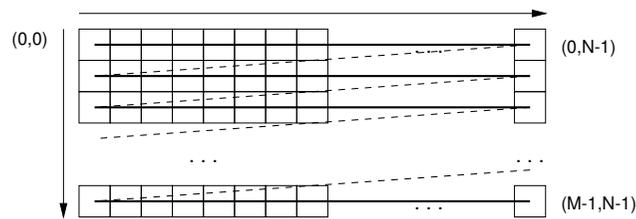


1861 stellte James Clerk Maxwell drei Diapositive eines Ordensbandes mit Streifenmuster her, fotografiert durch rote, grüne und blaue Filter. Diese drei Diapositive wurden mit drei Laterna magica-Projektoren durch die gleichen roten, grünen und blauen Filter projiziert. Sich genau überdeckend ergaben die drei Teilbilder eine beachtliche Farbrepräsentation des Originals. Dieses war die erste farbige Photographie.

Rechts oben: Ein Ives Kromogram (um 1897), direkt durch den Kromoskop-Betrachter fotografiert, ein Geröll, in dem die separaten roten, grünen und blauen Teilbilder zu einem Bild mit allen Farben kombiniert wurden.
Rechts unten: Ein 1892 mit dem Ives Photochromoscope-Projektor vorgeführtes Bild, reproduziert als Kodak Dye Transfer Vergrößerung nach den Originaldiapositiven.

(additive Farbmischung, Maxwell, 1861)

Raster-Bildformate

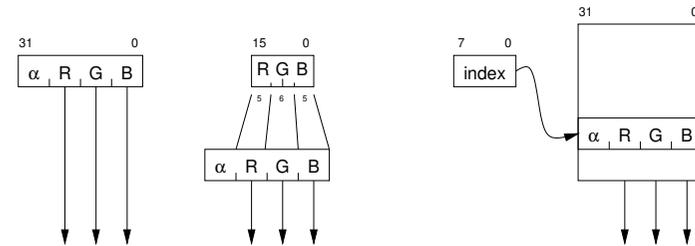


Digitalisierung von Bilddaten:

- rechteckiges Raster, "Pixel" (picture element, picture cell)
- Auflösung gemessen in dpi (dots per inch)
- Farbtiefe (bits per pixel)
- andere Rasterung möglich (z.B. hexagonal)

VGA:	640 x 480 x 8	300 KByte
XGA	1024 x 768 x 24	2.3 MByte
	1600 x 1200 x 24	5.7 MByte

Pseudo, High, True-Color



Speicherung der Helligkeits- / Farbwerte eines Pixels?

- | | |
|-------------------|-------------------------------------|
| direkt | "true color", üblich 24 bit 888 RGB |
| | "high color", z.B. 16 bit 565 RGB |
| indirekt über LUT | "indexed color" / "pseudo color" |

Gammakorrektur

- Farbmonitore / Fernsehen haben nichtlineare Kennlinien
- Helligkeit als Funktion der Eingangsspannung:

$$Y = U^\gamma \quad \gamma = 2.5 \dots 3.5$$

- Einsatz einer Lookup-Tabelle für (RGB) -> (R'G'B')
- "Gammakorrektur"



Rasterung und Dithering



Darstellung von Bildern auf "minderwertigen" Geräten?!

- Reduktion der Farbtiefe True-Color -> High-Color -> 256 Farben
- Reduktion auf S/W S/W-Monitor, Laserdrucker, ...
- Vierfarbdruck Tintenstrahldrucker, Offsetdruck, ...

"Dithering"

- höhere Farbauflösung auf Kosten der räumlichen Auflösung
- regelmässige, periodische Rasterung
- oder "Error-Diffusion", diverse Algorithmen

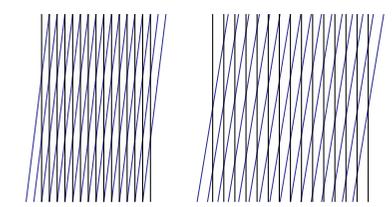
Dithering: Beispiel Grau nach SW

- "nearest neighbor"
- geordnetes, regelmässiges Raster
- Dithering, Floyd-Steinberg Algorithmus
- evtl. schwere Alias-Probleme, s.u.

Dithering: Alias-Effekte

mehrfache Rasterung:

- Überlagerung der einzelnen Raster
- period. Verstärkung/Auslöschung



Alias- bzw. "Moiré"-Effekte

- Darstellung eines gerasterten S/W-Bildes auf Monitor (Auflösung des Bildes vs. Auflösung des Monitors)
- Einscannen eines gedruckten Bildes (Druckauflösung vs. Auflösung des Scanners)
- usw.
- Verwendung "kompatibler" Raster (ganzzahliges Verhältnis)
- komplizierte Algorithmen zur "Entrasterung"

Netpbm: PBM, PGM, PPM

```
P3
# width height maxval r g b r g b r g b . . .
512 512
255
226 137 125 226 137 125 223 137 133 223 136 128 226 138 120
226 129 116 228 138 123 227 134 124 227 140 127 225 136 119
228 135 126 225 134 121 223 130 108 226 139 119 223 135 120
221 129 114 221 134 108 221 131 113 222 138 121 222 139 114
...
```

"Portable Bitmap / Graymap / Pixmap / Anymap":

- minimale, portable Dateiformate für S/W-, Grau-, Farbbilder
- "raw" oder ASCII-Format
- Format, Größe, max. Farbwert, Pixeldaten
- unkomprimiert, aber ideal für eigene Parser/Writer
- Jef Poskanzer (1991), jetzt als SourceForge-Projekt

Netpbm: Tools

Dutzende Format-Konverter:

BMP, GIF, JPEG, PNG, TIFF, usw.
 Palm Pixmap, Nokia SMF, Windows Icons, WAP bitmap, ...

Beispiele für Transformationen und Filter:

ppmlabel	Text
ppmbrighten, ppmddim, ppmgamma	Helligkeit
ppmnorm	Histogrammausgleich
ppmscale, ppmrotate, ppmsharpen	Skalierung, Rotation, ...
ppmdither, ppmquant, ...	Dithering, Quantisierung
ppmmix, ppmarith, ...	Überlagerung, Verknüpfung
ppmconvol	allg. Filter
ppmshadow, ppmrainbow, ...	diverse Spezialfunktionen

(netpbm.1 Manpage)

Netpbm: Anwendungsbeispiel

- Verkettung von Filtern / Operatoren via Unix-Pipes
- auch per Skript: Automatisierung / Batch-Verarbeitung
- Beispiel: mehrere Bilder konvertieren:

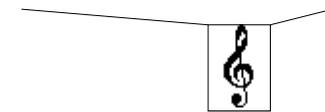
```
for i in *.png; do pngtopnm $i | ppmtotjpeg > `basename $i .png`.jpg; done
```

- Beispiel: Scans (TIFF) für dieses Skript aufbereiten:

<code>tifftoppm scan.tif </code>	von TIFF nach PPM
<code>ppmbrighten -value +10 </code>	Bild aufhellen
<code>pnmscale -width 800 </code>	neue Bildgröße
<code>ppmquant 31 </code>	Anzahl der Farben
<code>ppmtogif > scan.gif</code>	Umwandlung nach GIF

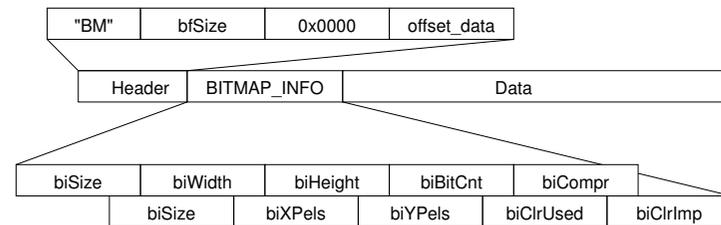
XBM-Format

```
#define key0_width 32
#define key0_height 42
static char key0_bits[] = {
0x00, 0x00, 0x03, 0x00, 0x00, 0x80, 0x07, 0x00, 0x00, 0x80, 0x07, 0x00, 0x00, 0xc0, 0x0f,
0x00, 0x00, 0xc0, 0x0d, 0x00, 0x00, 0xc0, 0x09, 0x00, 0x00, 0xc0, 0x08, 0x00, 0x00, 0xc0,
0xc0, 0x00, 0x00, 0x40, 0x0e, 0x00, 0x00, 0x40, 0x0e, 0x00, 0x00, 0x40, 0x0e, 0x00, 0x00,
0xc0, 0x07, 0x00, 0xc0, 0x07, 0x00, 0xc0, 0x03, 0x00, 0x00, 0xe0, 0x01, 0x00,
0x00, 0xf0, 0x01, 0x00, 0xc0, 0xf8, 0x01, 0x00, 0xc0, 0x7c, 0x01, 0x00, 0xc0, 0x3c, 0x01,
0x00, 0xc0, 0x1e, 0x01, 0x00, 0xc0, 0x8f, 0x1f, 0x00, 0xc0, 0xc3, 0x00, 0xc0, 0xe7,
0x3f, 0x00, 0xe7, 0x7a, 0x00, 0x67, 0x72, 0x00, 0x67, 0x62, 0x00, 0x67,
0x62, 0x00, 0x4e, 0x64, 0x00, 0x8c, 0x24, 0x00, 0x18, 0x34, 0x00,
0x00, 0x70, 0x1c, 0x00, 0xc0, 0x0f, 0x00, 0x00, 0x04, 0x00, 0x00, 0x04,
0x00, 0x00, 0x08, 0x00, 0xe0, 0x08, 0x00, 0xf0, 0x09, 0x00, 0xf0,
0x09, 0x00, 0xf0, 0x09, 0x00, 0xf0, 0x08, 0x00, 0x60, 0x04, 0x00, 0xc0,
0xc0, 0x03, 0x00};
```



- einfaches Format für S/W-Bilder
- als C-Quelltext, unkomprimiert
- ähnliches Prinzip für XPM (X11 Pixmap)

BMP: Bitmap-Format

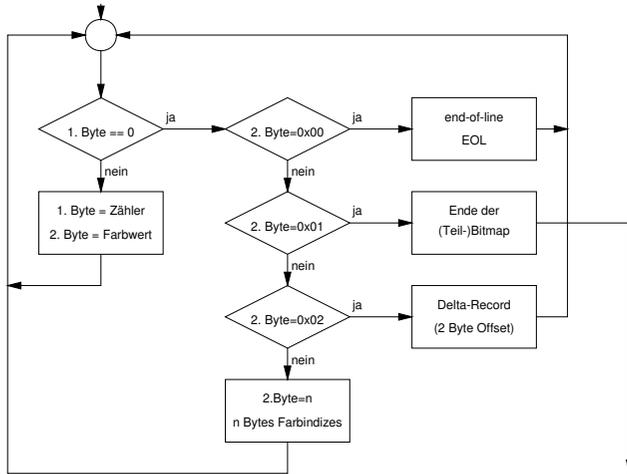


ab Offset 0x36 die Daten:



- unkomprimiert als RGB_QUAD:
- unkomprimiert mit "Maske", z.B. für 16-bit 5-6-5 Kodierung
- Lauflängenkodiert

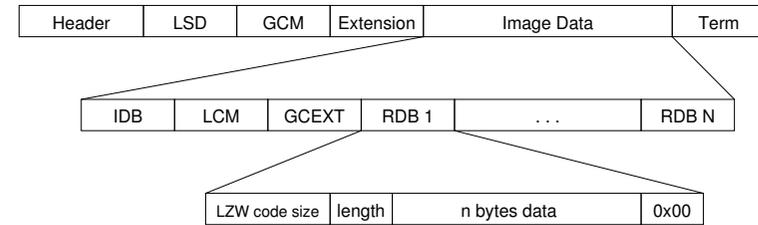
BMP: Runlength Encoding



GIF

- flexibles Bildformat für Grau- und Farbbilder
- nutzt LZW-Kompression
- weite Verbreitung dank Einsatz in HTML
- Unisys / Comuserve, 1987
- Verfahren ist patentiert, Einsatz im Web lizenzfrei
- aber Lizenz erforderlich für Programme, die GIFs erzeugen
- Index-Farbmodell, max. 256 Farben
- max. 16.000 x 16.000 Pixel
- erweiterte Version (GIF 89a):
- transparente Hintergrundfarbe
- "animated GIFs": mehrere Teilbilder in einer Datei

GIF 89a: Dateiformat



LSD	logical screen descriptor: width, height, background, ...
GCM, LCM	global/local color map, 256 entries
EXT	extension block (user defined data)
IDB	image descriptor block
RDB	raster data block: LZW-encoded image data
GCEXT (GIF89a)	graphics control extension, u.a. animation delay
CEXT (GIF89a)	connect extension block

(members.aol.com/royalef/gif89a.txt)

GIF 89a: Interlacing

The rows of an Interlaced images are arranged in the following order:

- Group 1 : Every 8th. row, starting with row 0. (Pass 1)
- Group 2 : Every 8th. row, starting with row 4. (Pass 2)
- Group 3 : Every 4th. row, starting with row 2. (Pass 3)
- Group 4 : Every 2nd. row, starting with row 1. (Pass 4)

Row Number	Interlace Pass
0	1
1	
2	
3	3
4	
5	2
6	
7	3
8	
9	1
10	
11	3
12	
13	2
14	
15	3
16	
17	1
18	
19	3
	4

(members.aol.com/royalef/gif89a.txt)

GIF: LZW-Kompression

- zeilenweise Kodierung der Bilddaten
- nur ein-dimensionale Korrelationen
- gut geeignet für Zeichnungen etc.

LZW-Variante:

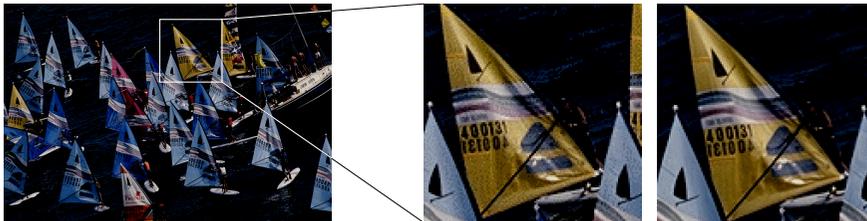
- Dictionary wird während der Kodierung gefüllt
- Startgröße ist $b=2$ (S/W) bzw. $b=8$ (Grau-/Farbbilder)
- max. 4096 Einträge
- bei Bedarf Anlegen eines neuen Dictionary

1 byte block size

data (pointers into dictionary into color map)

1 byte end marker (0x00)

GIF: Farbphotos



GIF, 256 Farben

GIF

JPEG

GIF-Format nur bedingt für Farbphotos geeignet:

- Beschränkung auf 256 Farben
- "Farbrauschen"
- selbst bei Einsatz guter Dithering-Verfahren
- andere Verfahren (PNG, JPEG, TIFF) besser

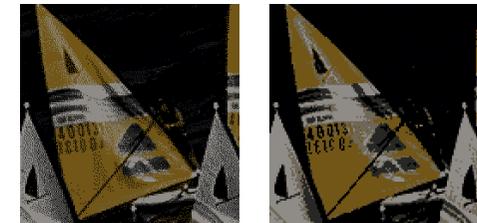
GIF: Dithering



GIF; 256 Farben

32 Farben, Floyd-Steinberg

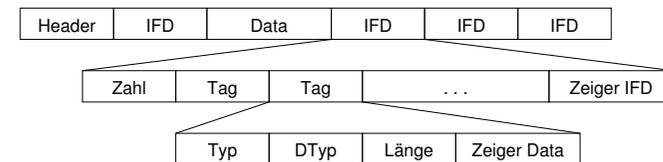
32 Farben, nearest



8 Farben, Floyd-Steinberg

8 Farben, nearest

TIFF



"Tagged Image File Format"

- Aldus Corporation, Hewlett Packard, Microsoft, 1980
- hierarchischer Aufbau, Header und var. Anzahl von Datenblöcken
- Verkettung der Blöcke über Zeiger (Image File Directories)
- Zeiger kodiert in 12-Byte "Tag"-Strukturen
- Vielzahl von Tag-Varianten definiert
- für Bildformat, -auflösung, -größe, -kompression, ...

DCT: Diskrete Kosinus-Transformation

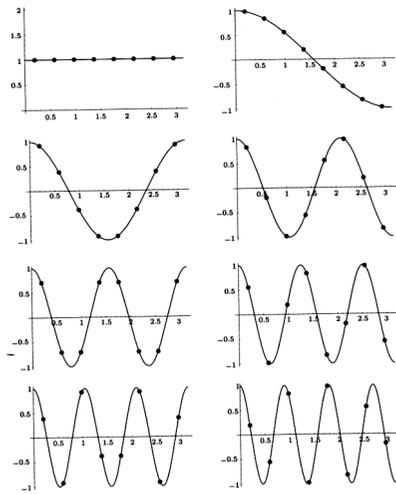


Figure 4.17: Calculating A One-Dimensional DCT.

Kosinusfunktionen für die 1D DCT:
zunehmende Frequenz
entsprechende Abtastpunkte

(Salomon)

DCT: 8x8 2D-Transformation

DCT:

$$G_{ij} = \frac{1}{4} C_i C_j \sum_{x=0}^7 \sum_{y=0}^7 p_{xy} \cos\left(\frac{(2x+1)i\pi}{16}\right) \cos\left(\frac{(2y+1)j\pi}{16}\right),$$

where $C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0, \\ 1, & f > 0, \end{cases}$ and $0 \leq i, j \leq 7$.

IDCT:

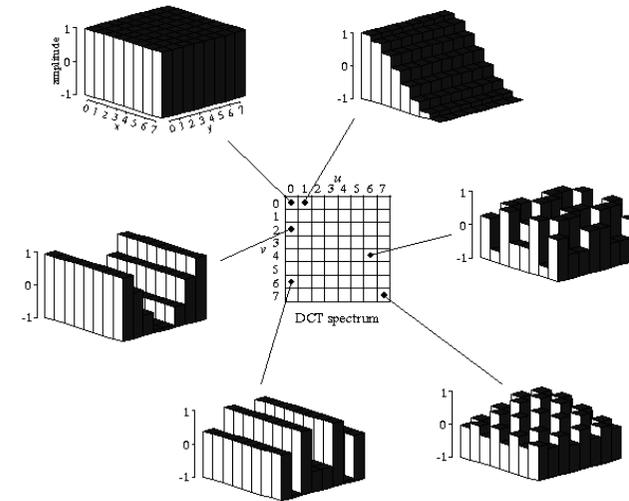
$$p_{xy} = \frac{1}{4} \sum_{i=0}^7 \sum_{j=0}^7 C_i C_j G_{ij} \cos\left(\frac{(2x+1)i\pi}{16}\right) \cos\left(\frac{(2y+1)j\pi}{16}\right),$$

where $C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0; \\ 1, & f > 0. \end{cases}$

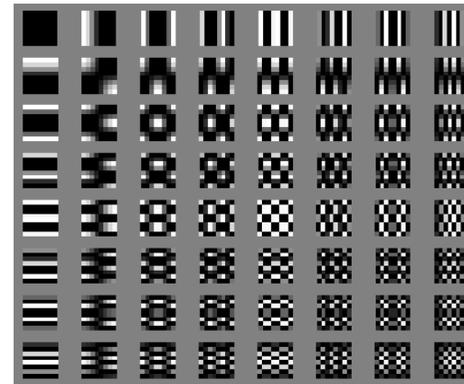
effiziente Implementierung?

- Vorberechnung der cos(x) Terme
- Kombination von 1D-Transformationen (min. Anzahl von Ops)
- zeilenweise Berechnung (Cache-freundlich)
- MMX, SSE, Spezialhardware

DCT: Basisfunktionen

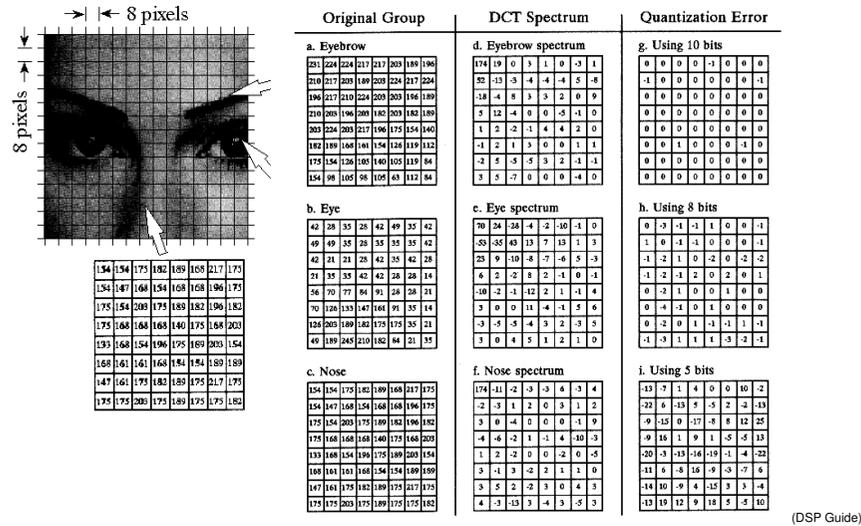


DCT: Basisfunktionen



- die Basisfunktionen der 2D 8x8 DCT
- entsprechende Bilder liefern max. Antwort der DCT
1 (weiß) .. 0 (grau) .. -1 (schwarz)

DCT: Beispiel



JPEG: Übersicht

JPEG := "Joint Picture Experts Group"

- Verfahren zur Kodierung von Grau- und Farbbildern
- 1991 standardisiert
- diverse Varianten, "baseline" oder "lossless"

Kodierung via DCT von Blöcken a 8x8 Pixeln
 YCbCr Farbmodell (Farbsubsampling)
 Kompressionsraten bis ca. 40:1 möglich

- JFIF: Bezeichnung für das Dateiformat (.jpg, .jpeg)
- JBIG: "Joint Bi-level image experts group" (s/w-Bilder)
- JPEG-2000: aktuelle, erweiterte Version

JPEG: Konzepte, Ziele

sehr flexibles Format:

- fast beliebige Bildauflösung
- beliebiges Bild/Pixel-Aspektverhältnis
- unabhängig vom Farbmodell
- für alle Arten von Bildern, aber insb. für Photos
- progressive Kodierung (schnelle Vorschau)
- Implementierung in Software, bei Bedarf in Hardware:
- "Motion-JPEG", Anwendung für Video

JPEG: Beispiel



229 KByte, 3:1, 7 bpp



24 KByte, 32:1, 0.74 bpp



18 KByte, 43:1



12 KByte, 66:1, 0.36 bpp



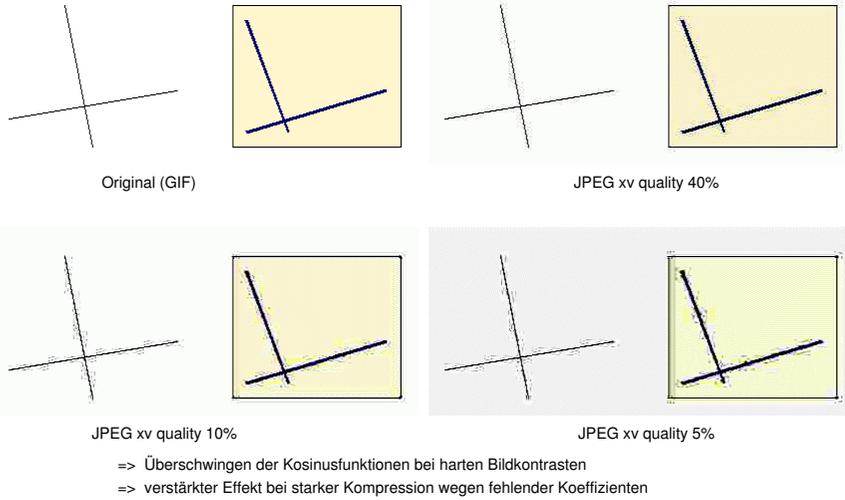
10 KByte, 78:1



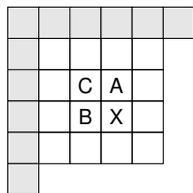
5 KByte, 140:1, 0.17 bpp

=> Vergleich: PPM-raw, 512x512x24: 786 KByte, 24 bpp
 => praktisch optimale Qualität bei 10:1, brauchbar bis ca. 40:1

JPEG: "ringing"



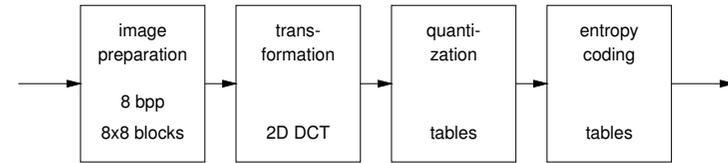
JPEG: verlustfreie Kodierung



Nr.	Vorhersage V:
0	-
1	A
2	B
3	C
4	A+B-C
5	$A+(B-C)/2$
6	$B+(A-C)/2$
7	$(A+B)/2$

- Vorhersage V von Pixelwerten aus Nachbarpixeln
- sieben verschiedene Algorithmen für V
- kodiert Index des Algorithmus und Differenz (X-V)
- anschliessend Huffman-Kodierung
- Kompressionsfaktor ca. 2

JPEG: Verarbeitungsschritte

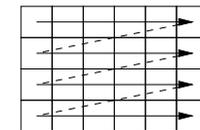


- Eingangsdaten in YUV-Darstellung und auf 8 bits/pixel wandeln
- Aufteilung in Blöcke, feste Größe 8x8
- 2D Kosinustransformation
- Quantisierung der Koeffizienten, feste (oder variable) Tabellen
- kombinierte Huffman-/Runlength-Kodierung

JPEG: Vorbereitung

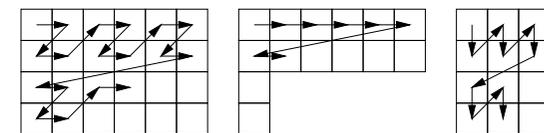
verlustfreie Kodierung mit 2 .. 16 bpp
verlustbehaftete Kodierung intern mit 8 bpp / 12 bpp

"normale" Kodierung:

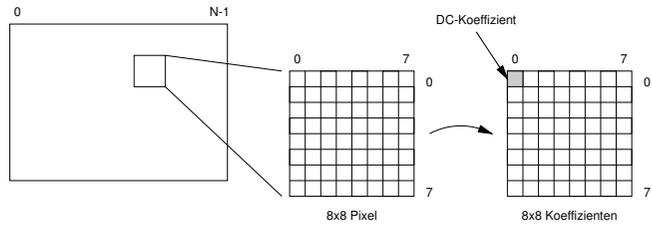


□ = Block mit 8x8 Pixeln

"interleaved"
(progressive)



JPEG: Blöcke



- JPEG transformiert Blöcke, nicht einzelne Pixel
- Annahme: benachbarte Pixel haben ähnliche Werte

=> nach der 2D-DCT liegen die größten Koeffizienten "links oben"

- kein Ausnutzen von Korrelationen über Blockgrenzen hinaus
- Neigung zu "Blockartefakten" bei starker Kompression

JPEG: Quantisierung

Original Group	DCT Spectrum	Quantization Error
a. Eyebrow	d. Eyebrow spectrum	g. Using 10 bits
225 226 224 217 217 203 188 196	174 19 0 3 1 0 -3 1	0 0 0 0 -1 0 0 0
210 217 203 189 203 224 217 224	25 -13 -3 -4 -4 -4 5 -8	-1 0 0 0 0 0 0 -1
196 217 210 224 203 203 196 189	-14 -4 8 3 3 2 0 9	0 0 0 0 0 0 0 0
210 203 196 203 185 203 182 189	5 12 -4 0 0 -5 -1 0	0 0 0 0 0 0 0 0
203 224 203 217 196 175 154 140	1 2 2 -3 4 4 2 0	0 0 0 0 0 0 0 0
182 189 168 141 154 124 119 112	-1 2 1 3 0 0 1 1	0 0 1 0 0 0 -1 0
175 154 126 185 140 105 119 84	-2 5 -2 -2 3 2 -1 -1	0 0 0 0 0 0 0 0
154 98 105 98 105 63 112 84	3 3 -7 0 0 0 -4 0	0 0 0 0 0 0 0 0
b. Eye	e. Eye spectrum	h. Using 8 bits
42 28 35 28 42 49 35 42	70 36 -28 -4 -2 -10 -1 0	0 -3 -1 -1 1 0 0 -1
49 49 35 28 35 35 35 42	-33 -32 43 13 7 13 1 3	1 0 -1 -1 0 0 0 -1
42 21 21 28 42 35 42 28	23 9 -10 -8 -7 -6 5 -3	-1 -2 1 0 -2 0 -2 -2
71 35 35 42 42 28 28 14	6 2 -2 8 2 -1 0 -1	-1 -2 -1 2 0 2 9 1
56 70 77 84 91 28 28 21	-10 -2 -1 -12 2 1 -1 4	0 -2 1 0 0 1 9 0
70 126 133 147 164 91 35 14	3 0 0 11 -4 -1 3 6	0 -4 -1 0 1 0 9 0
126 203 189 182 175 175 35 21	-3 -5 -5 -4 3 2 -3 3	0 -2 0 1 -1 -1 1 -1
49 139 145 210 182 84 21 35	3 0 4 5 1 2 1 0	-1 -3 1 1 1 -3 -2 -1
c. Nose	f. Nose spectrum	i. Using 5 bits
154 154 175 182 189 168 217 175	174 -11 -2 -3 -3 6 -3 4	-13 -7 1 4 0 0 10 -2
154 147 168 154 168 168 196 175	-2 -3 1 2 0 3 1 2	-22 6 -13 5 -5 2 -2 -13
175 154 203 175 189 182 196 182	3 0 -4 0 0 0 -1 9	-9 -15 0 -17 -8 12 25
175 168 168 168 140 175 168 203	-4 -6 -2 1 -1 4 -10 -3	-9 16 1 9 1 -5 -5 13
133 168 154 196 175 189 203 154	1 2 -2 0 0 -2 0 -5	-20 -13 -16 -19 -1 -4 -22
168 101 101 168 154 154 189 189	3 -1 3 -2 2 1 1 0	-11 6 -8 16 -9 -3 -7 6
147 101 175 182 189 175 217 175	3 2 2 -2 3 0 4 3	-14 10 -9 4 -15 3 3 -4
175 175 203 175 189 175 175 182	4 -3 -13 3 -4 3 -5 3	-13 16 12 9 18 5 -5 10

- typisches Beispiel für (gerundete) Wert der Koeffizienten

- weitere Quantisierung: Koeffizienten durch Tabellenwerte teilen

JPEG: Quantisierung: Tabellen

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	193	121	120	101
72	92	95	98	112	100	103	99

Quantisierung für Luminanz (Y)

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Quantisierung für Chrominanz (UV)

- einzelne Koeffizienten nach DCT bereits gerundet
- weitere Quantisierung nach Tabellen,
- einzelner Koeffizient wird durch Tabellenwert geteilt
- vordefinierte Tabellen nutzen physiologische Daten
Beispiel: $Y00' = Y00/16$, $V13' = V13/66$, usw.

JPEG: Quantisierung: Beispiel

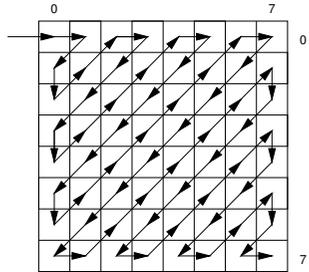
32	4	-3	0	0	0	1	0	70	0	0	0	0	0	0	0
6	-2	0	0	-2	0	0	1	0	0	0	0	0	0	0	0
-7	3	2	0	0	2	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a) (b)

Table 4.53: (a) The Quantized Coefficients of Table 4.49. (b) Those of Table 4.50.

- Beispiel für (typ.) Werte der quantisierten Koeffizienten
- DC-Koeffizient hat großen Wert
- Koeffizienten "links oben" sind klein oder null
- Koeffizienten "rechts unten" sind (fast) alle null
- weitere Kodierung mit Huffman/Runlength-Verfahren

JPEG: ZigZag



DC AC01 AC10 AC20 AC11 AC02 ...
AC03 AC12 AC21 ... AC76 AC77

- Koeffizienten kleiner Frequenz liegen "oben links"
- "ZigZag"-Anordnung der Koeffizienten für die Huffman-Phase
- erzeugt Paare von "00..00"-Folgen und kleinen Integern
- lange "000..0" Folge am Ende

JPEG: Huffman

- Beispiel-Sequenz von Koeffizienten (nach der Quantisierung):

1118, 2, 0, -2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, ...

- (Z, V)-Paare: Anzahl der Nullen vor dem Wert V

DC=1118, (0, 2), (1, -2), (13, -1), (...)

- Kodetabellen für (Z,V)-Paare
- Zahldarstellung mit r bits, r abhängig von Tabellenzeile

(0, 2) -> 01 | 10
(1, -2) -> 1100 | 1110
(13, -1) -> 11111111010 | 1111111111

- am Ende (EOB) Marker, z.B. 64 Koeffizienten -> 50 bits

JPEG: Huffman-Tabellen

Z	R				
	1 6	2 7	3 8	4 9	5 A
0	00	01	100	1011	11010
1	111000	11111000	1111110110	111111110000010	1111111110000011
2	1100	11011	11110001	111110110	11111110110
3	111111110000100	111111110000101	111111110000110	111111110000111	111111110001000
4	11100	1111001	111110111	11111110100	11111110001001
5	111111110001010	111111110001011	11111110001100	11111110001101	11111110001110
6	111010	11110111	11111110101	11111110001111	11111111001000
7	111111110010001	111111110010010	111111110010011	111111110010100	111111110010101
8	111011	111111000	111111110010110	111111110010111	111111110011000
9	111111110011001	111111110011010	111111110011011	111111110011100	111111110011101
A	1111010	1111110111	111111110011110	111111110011111	111111110100000
B	111111110100001	111111110100010	111111110100011	111111110100100	111111110100101
C	1111011	11111110110	111111110100110	111111110100111	111111110101000
D	111111110101001	111111110101010	111111110101011	111111110101100	111111110101101
E	1111010	11111110111	111111110101110	111111110101111	111111110101000
F	111111110110001	111111110110010	111111110110011	111111110110100	111111110110101
G	11111000	11111111000000	111111110110110	111111110110111	111111110111000
H	111111110111001	111111110111010	111111110111011	111111110111100	111111110111101
I	11111001	111111110111110	111111110111111	11111111000000	11111111000001
J	11111111000010	11111111000011	11111111000010	11111111000011	1111111100010
K	11111010	1111111100011	11111111001000	11111111001001	11111111001010
L	11111111001011	11111111001010	11111111001011	1111111100110	1111111100111
M	1111111001	11111111010000	11111111010001	11111111010010	11111111010011
N	11111111010100	11111111010101	11111111010110	11111111010111	11111111011000
O	1111111010	11111111011001	11111111011010	11111111011011	11111111011100
P	11111111011011	11111111011010	11111111011011	1111111101110	1111111101111
Q	11111111011010	11111111011011	1111111101110	1111111101111	1111111101110
R	1111111101110	1111111101111	1111111101110	1111111101111	1111111101110
S	1111111101110	1111111101111	1111111101110	1111111101111	1111111101110
T	1111111101110	1111111101111	1111111101110	1111111101111	1111111101110
U	1111111101110	1111111101111	1111111101110	1111111101111	1111111101110
V	1111111101110	1111111101111	1111111101110	1111111101111	1111111101110
W	1111111101110	1111111101111	1111111101110	1111111101111	1111111101110
X	1111111101110	1111111101111	1111111101110	1111111101111	1111111101110
Y	1111111101110	1111111101111	1111111101110	1111111101111	1111111101110
Z	1111111101110	1111111101111	1111111101110	1111111101111	1111111101110

Table 4.56: Recommended Huffman Codes For Luminance AC Coefficients.

(erster Teil der Tabelle)

- Leerseite

JPEG2000

• JPEG hat sich bewährt
aber diverse Erweiterungen wünschenswert:

- bessere Qualität bei geringen Bitraten (< 0.25 bpp)
- einheitliches Format für S/W-, Grau-, und Farbbilder
- Kombination von verlustfreier und verlustbehafteter Kodierung:
- progressive Kodierung: Vorschau ... verlustfreie Kodierung
- Qualitätsstufen, z.B. 72 dpi Monitor vs. 2400dpi Drucker
- "region of interest" Kodierung
- Robustheit, bessere Korrektur von Bitfehlern
- offene, erweiterbare Architektur
- Unterstützung für Indizierung / Inhaltssuche / MPEG-7
- Unterstützung für Verschlüsselung, Wasserzeichen, ...

(Christopoulos, IEEE Tr. CE 46-4, 1103, 2000)

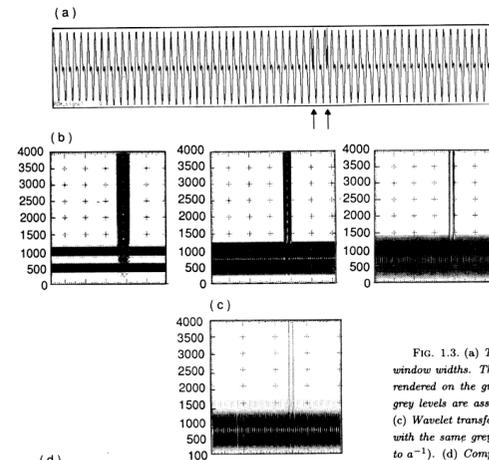
JPEG 2000

ähnliches Prinzip wie JPEG:

- | | | |
|-------------------------------------|---------|--------------|
| | JPEG | JPEG 2000 |
| • Bildvorbereitung, Farbsubsampling | | |
| • lineare Transformation | DCT | Wavelets |
| • Entropiekodierung | Huffman | arithmetisch |
-
- keine feste Blockgröße für die Wavelet-Transformation
 - gesamtes Bild, oder "Tiles" beliebiger Größe
 - verlustfreie / -behaftete Kodierung (Integer/Float, zwei Wavelets)
 - Wavelets kodieren Bild in verschiedener Auflösung
 - daher weniger Neigung zu Blockartefakten
-
- Bildqualität nicht unbedingt besser als JPEG oder PNG
 - nur bei sehr hoher Kompression

(Santa-Cruz et al. SPIE ADIP 23, 4446, 2000)

Wavelet- vs. Fourier-Analyse



Eingangssignal:
zwei Sinussignale 500 Hz/1 KHz,
plus zwei Störimpulse (Pfeile)

Fourieranalyse:
Fenstergröße 12.8 / 6.4 / 3.2 msec.
abnehmende Frequenzauflösung
zunehmende Zeitauflösung

FIG. 1.3. (a) The signal $f(t)$. (b) Windowed Fourier transforms of f with three different window widths. These are so-called spectrograms: only $|T^{w_{10}}(f)|$ is plotted (the phase is not rendered on the graph), using grey levels (high values = black, zero = white, intermediate grey levels are assigned proportional to $\log |T^{w_{10}}(f)|$ in the t (abscissa), ω (ordinate) plane. (c) Wavelet transform of f . To make the comparison with (b) we have also plotted $|T^{w_{10}}(f)|$, with the same grey level method, and a linear frequency axis (i.e., the ordinate corresponds to ω^{-1}). (d) Comparison of the frequency resolution between the three spectrograms and the wavelet transform. I would like to thank Oded Ghitza for generating this figure.

Wavelet-Analyse:
gute Zeitauflösung, mittlere Frequenzauflösung

(Daubechies: Ten lectures on wavelets, 1.2)

JPEG 2000: Beispiel



JPEG 2000, 9819 bytes, 0.3bpp

JPEG, 9643 bytes

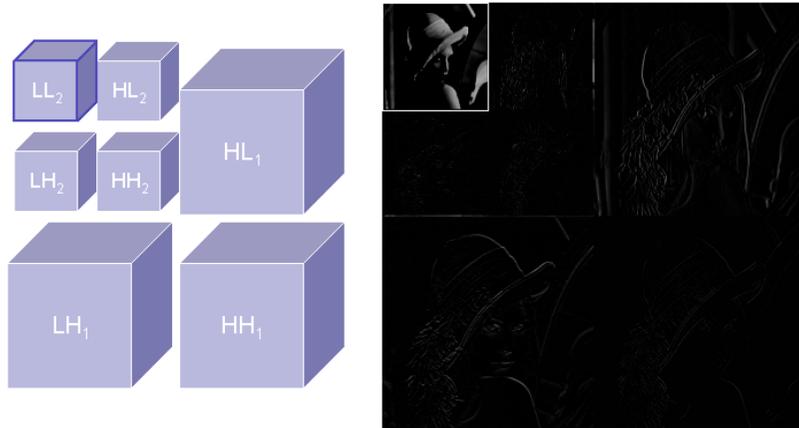
JPEG, 11904 bytes

- Vorführung des JJ2000 Encoders/Decoders

```
java JJ2KEncoder -i lena.ppm -o lena.j2k -rate 0.3
java JJ2KDecoder -i lena.j2k
```

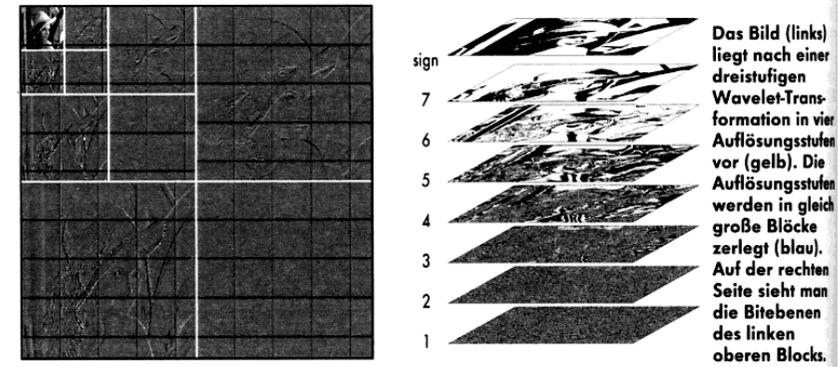
(jj2000.epfl.ch)

JPEG 2000: Skalierung



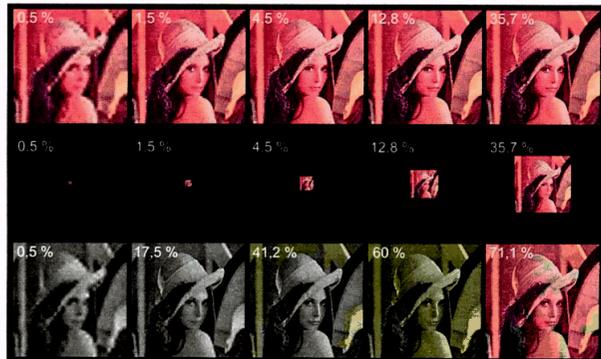
- rekursive Anwendung der Wavelet-Transformation
- Tiefpaßanteile "oben links", Details "unten und rechts"

JPEG 2000: Bitplanes



- weitere Skalierungsmöglichkeit: Übertragung der MSBs zuerst

JPEG 2000: progressive Kodierung



Flexible Progression: Verwendet man nur einen Teil der komprimierten Daten (Prozentangabe), zeigt sich, in welcher Reihenfolge die Pakete in der Datei gespeichert sind. Hier zu sehen (von oben nach unten): qualitative Verbesserung, Steigerung der Auflösung und Komponentenprogression.

- Leerseite

Bildverarbeitung: Filter

- Grundoperatoren: lineare, verschiebungsinvariante Filter
- Beispiel 3x3 Filter zur Glättung:

$$\begin{matrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & 0 & 1 & 1 & \dots \\ \dots & 0 & 0 & 1 & 1 & \dots \\ \dots & 0 & 0 & 1 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{matrix} * \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \begin{matrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & 1/3 & 2/3 & 1 & \dots \\ \dots & 0 & 1/3 & 2/3 & 1 & \dots \\ \dots & 0 & 1/3 & 2/3 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{matrix}$$

- für jeden Bildpunkt: Auslesen der Nachbarpixel
- Multiplikation mit den entsprechenden Filterkoeffizienten
- Summation
- alternative Berechnung via Fouriertransformation

Filter:

- große Auswahl wichtiger Algorithmen
- typ. Matrixgröße von 3x3 bis 15x15 bis ..
- Limit wegen Rechenaufwand: 9 .. 225 .. Operationen / Pixel

Glättung, Weichzeichner (Gauß), Schärfung, Kantendetektion, ...

- Verbesserung durch "separierbare" Filter
- Auftrennung der Matrix in zwei Vektoren
- Anwendung erst zeilen- dann spaltenweise (oder umgekehrt)
- lineare Abhängigkeit von der Filtergröße
- entsprechendes Prinzip auch für nichtlineare Op. (z.B. Median)

Filter:

Beispielmatrizen: Weichzeichner 3x3 und 5x5:
Koeffizienten aus Binomialverteilung

$$\frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad \frac{1}{256} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

- Herleitung und weitere Beispiele: s. Jähne

Bildverarbeitung: Filter-Pipelines

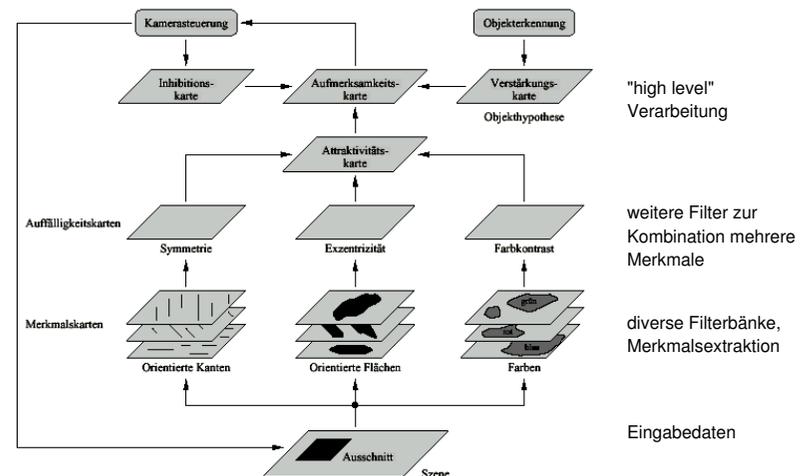


Abbildung 4.1: Schema der Aufmerksamkeitssteuerung in NAVIS

Vektorgraphik

- Aufwand bei Rasterbildern: (etwa) quadratisch mit Bildgröße
- auflösungsunabhängige Formate?

=> "Vektorgraphik"

- Grundobjekte Linien, Rechtecke, usw.
 - Angabe der Koordinaten und Attribute
 - einfache Manipulation
 - Darstellung via Koordinatentransformation und Rasterung
-
- ideal für Zeichnungen, Diagramme, etc.
 - aber nur bedingt für "photorealistische" Darstellungen
 - diverse Editoren: Corel Draw & Co.

Medientechnik | WS 2001 | 18.204

Vektorgraphik: Beispiel FIG-Format

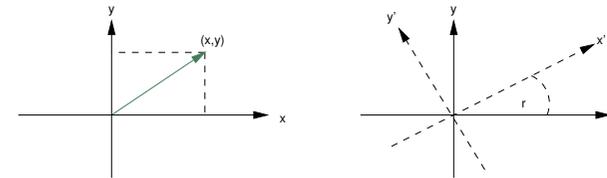
```
#FIG 3.2
Landscape
Center
...
0 34 #4193ff
...
4 0 0 100 ... 2250 8100 Vektorgraphik
...
2 2 0 1 7 0 ... 5
0 0 13500 0 13500 0 9450 0 9450 0 0
...
3 1 0 ... 45
...
```

] Dateiheader, globale Attribute
] 0=benutzerdefinierte Farbe (RGB)
 4=Textobjekt, 0=left justified, 0=black...
 2=Polyline, 2=Box, 0=solid, 1=thickness...
 ... 5=number of points
 ... 5 points (x1,y1) (x2,y2) ...
 3=Spline, 1=closed, 0=solid... 45=npoints.

(www.xfig.org, ftp.x.org/contrib/applications/drawing_tools/xfig)

Medientechnik | WS 2001 | 18.204

Vektorgraphik:



einfache Transformationen: Skalierung, Translation, Rotation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} ax \\ ay \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x0 \\ y0 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos r & \sin r \\ -\sin r & \cos r \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Rasterung via "Bresenham"-Algorithmus

Medientechnik | WS 2001 | 18.204

Fonts: TrueType

- entfällt aus Zeitgründen, siehe Henning 3.2

Medientechnik | WS 2001 | 18.204

Video

- Übersicht, Motivation
- Digitaler Videorekorder, Timeshift, MHP
- Analoge Fernsehtechnik
- Composite-Signal, QAM
- NTSC/PAL/SECAM
- Digitale Videokodierung
- MPEG, H.26x
- Bewegungskompensation, IBP-Frames
- Datenkompression und Datenstrom
- Objektbasierte Kodierung, MPEG-4
- Demos

Medientechnik | WS 2001 | 18.204

Video: Literatur

MPEG-Homepage, Docs, Tutorials, www.mpeg.org

MPEG-Docs: www.csel.it/mpeg/

L. Krich, Fernsehtechnik, Vieweg, 1993

Salomon, Data Compression, Springer, 2000

IBM Journal R&D 43-3, 1997

Communications of the ACM, 34-4, 1991 (JPEG, MPEG-1, H.261)

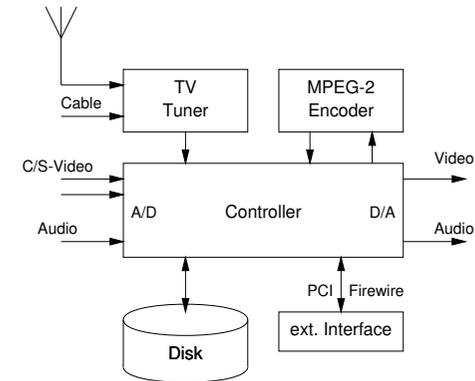
diverse Datenblätter, insb. Brooktree/Conexant Bt8x8

Überblick DVB,MHP,DVR: c't 18/2001, 95ff:

Einführung, MPEG-4, Codec-Test, c't 10/2001, 122ff:

Medientechnik | WS 2001 | 18.204

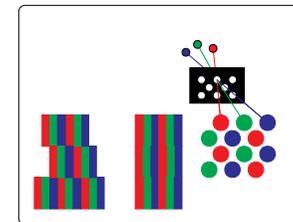
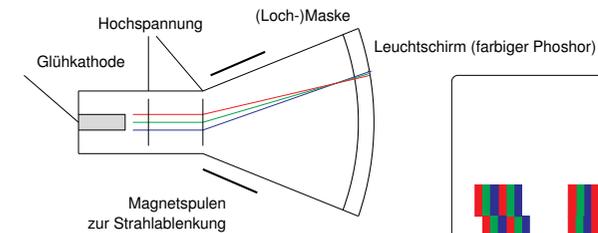
Digitaler Videorecorder, Timeshift



- "digitaler Videorekorder", z.B. MPEG-2 Aufzeichnung
- gleichzeitige Aufzeichnung und (zeitversetzte) Wiedergabe
- erste "single-chip" Lösungen erhältlich

Medientechnik | WS 2001 | 18.204

Fernsehen:

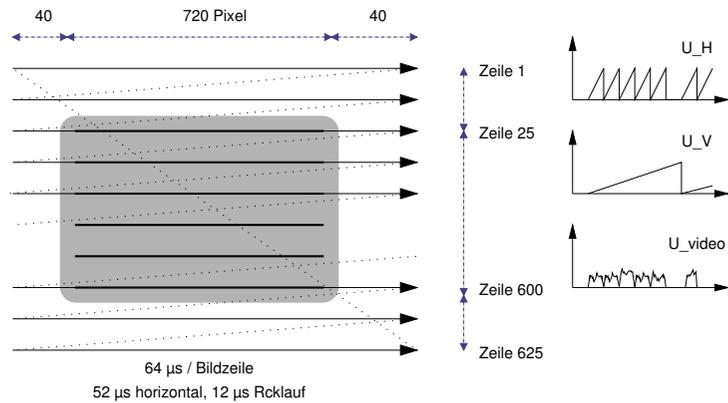


Braun'sche Röhre:

- fokussierter Elektronenstrahl (10+ KV), Leuchtschirm
- Farbfernsehen mit drei unabhängigen Glühkathoden
- magnetische Strahlableitung
- div. Varianten (Loch/Schlitzmaske, inline, ...)

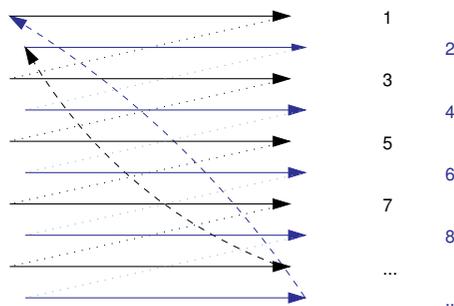
Medientechnik | WS 2001 | 18.204

Fernsehen: Bilderzeugung



- zeilenweise Abtastung
- Strahl dunkel für horizontalen und vertikalen Strahlrücklauf
- PAL: 625 (575) Zeilen / Bild, 800 (720) Pixel / Zeile

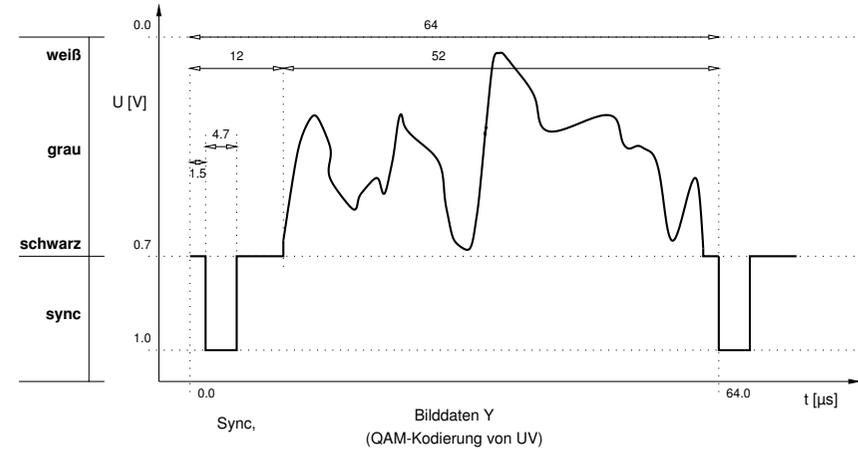
Fernsehen: Interlacing



"Zeilensprungverfahren":

- abwechselnde Übertragung von "Halbbildern"
- wegen zu kurzem Nachleuchten des Bildschirms
- nachträgliche Korrektur ("De-Interlacing") extrem aufwendig

Fernsehen: Signalverlauf



- "composite" Signal: Y-Daten plus Sync-Impulse

PAL, NTSC, SECAM



		f _{horiz} kHz	f _{vert} Hz	Zeilen	Bandbr. MHz	f _{audio} MHz	
NTSC	1953	15.734	60	525	4.2	4.5	
SECAM	1957/1968	15.625	50	625	5.0	5.5	
PAL	1967	15.625	50	625	5.0	5.5	

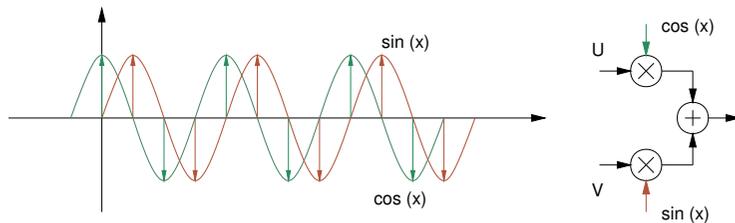
PAL: Prinzip

"phase alternating line"

- Überlagerung von Helligkeit- und Farbsignalen
- abwärtskompatibel zum S/W-Fernsehen
- YUV-Farbmodell
- QAM-Verfahren (Quadratur-Amplituden-Modulation)
- Farbträger mit 4.43361875 MHz
- Phase des V-Signals wechselt mit jeder Bildzeile

=> Phasenabweichungen "mitteln" sich aus

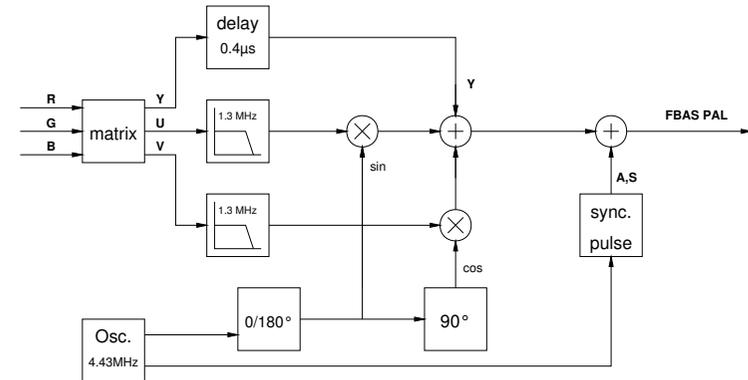
PAL: QAM



"Quadratur-Amplituden-Modulation"

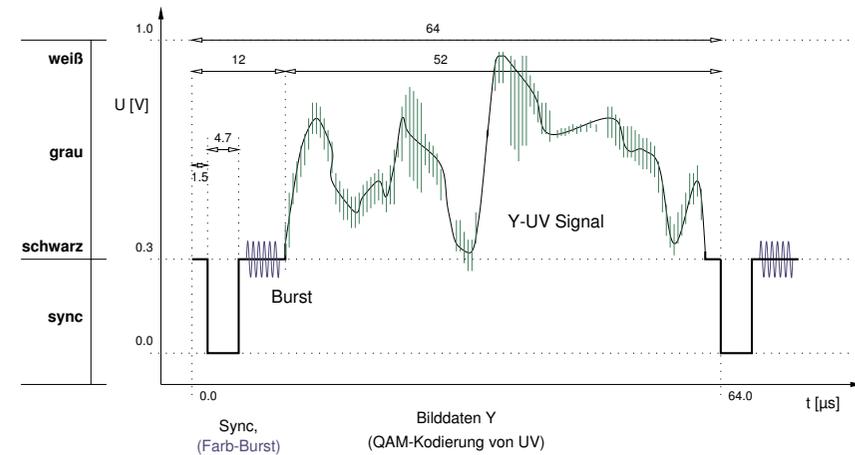
- Übertragen von zwei Signalen (U,V) auf einem Träger
- durch phasenverschobene Multiplikation
- zum Abtastzeitpunkt jeweils einer der sin/cos Terme Null
- Rekonstruktion der Signale erfordert Phase des Trägers
- extrem empfindlich gegen Phasenfehler
- scherzhaft: NTSC = "Never Twice the Same Color"

PAL: Encoder



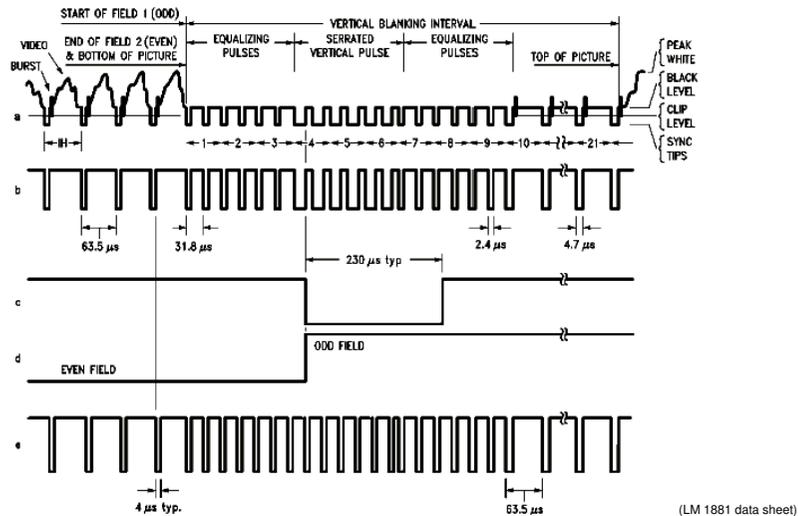
- "FBAS": Farb-Bild-Austast-Synchronisationssignal

PAL: Signalverlauf



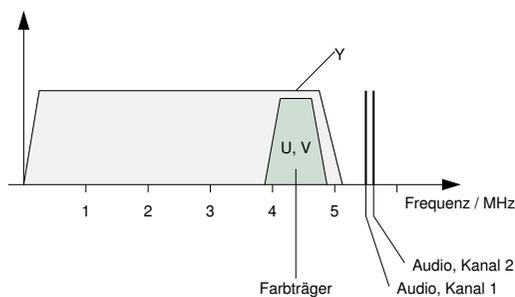
- Burst-Impuls erlaubt Synchronisation auf Phase des Farbträgers

PAL: H/V-Synchronisation



Medientechnik | WS 2001 | 18.204

PAL: Frequenzbereiche



- Helligkeitssignal (Y) mit ca. 5 MHz Bandbreite
- Farbsignale (U,V) in QAM um den Farbträger, 4.43.... MHz
- Audiosignale bei 5.50 und 5.74 MHz

Medientechnik | WS 2001 | 18.204

Video Grabbing

digitale Aufzeichnung von analogem Video?!

- Synchronisation auf das Eingangssignal
- Abtastung, A/D-Wandlung
- Demodulation von Audio, Y/U/V
- erfordert Ausmitteln von Schwankungen
- z.B. autom. Korrektur zu kurzer / zu langer Bildzeilen
- entsprechende Kompensation der Samplezeitpunkte
- De-Interlacing, Korrektur von "Kammeffekten"
- 720 x 576 Pixel RGB888, 25 fps: 30 MB Daten / Sekunde
- diverse Chips verfügbar

Medientechnik | WS 2001 | 18.204

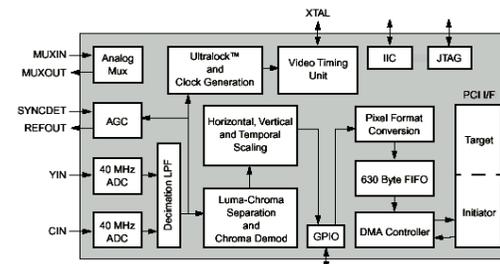
Grabbing: Brooktree 8x8 Chip

Bt848 is a complete, low cost single-chip solution for analog NTSC/PAL/SECAM video capture on the PCI bus. As a bus master, Bt848 does not require any local memory buffers to store video pixel data which significantly minimizes the hardware cost for this architecture. Bt848 takes advantage of the PCI-based system's high bandwidth and inherent multimedia capability. It is designed to be interoperable with any other PCI multimedia device at the component or board level, thus enabling video capture and overlay capability to be added to PCI systems in a modular fashion at low cost. The Bt848 solution is independent of the PCI system bus topology and may be used in a variety of system bus organizations: directly on a motherboard planar bus, on a card for a planar or secondary bus.

Distinguishing Features

- Fully PCI Rev. 2.1 compliant
- Auxiliary GPIO data port and video data port
- Supports image resolutions up to 768x576 (full PAL resolution)
- Supports complex clipping of video source
- Zero wait state PCI burst writes
- Field/frame masking support to throttle bandwidth to target
- Multiple YCbCr and RGB pixel formats supported on output
- Supports NTSC/SECAM/PAL analog input
- Image size scalable down to icon using vertical & horizontal interpolation filtering
- Multiple composite and S-video inputs
- Supports different destinations for even and odd fields
- Supports different color space/scaling factors for even and odd fields
- Support for mapping of video to 225 color palette
- VBI data capture for closed captioning, teletext and intercast data decoding

Functional Block Diagram



Medientechnik | WS 2001 | 18.204

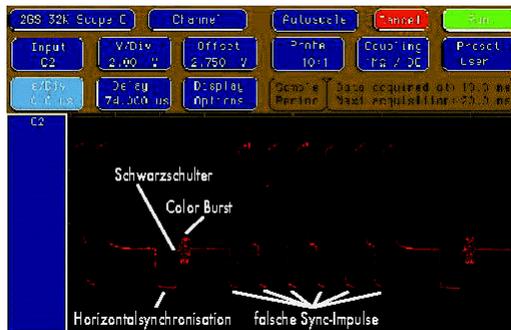
Grabbing: Karten

Vergleichstabelle
analoge WinTV Karten

	BUS Typ	TV-Tuner/ Audio	LKW Stereo Radio	Frame & AVI Clip Capture	Video text	Fernbedien-ung	Dolby Surround	Externe Video-eingänge	Betriebs-system	empf. VK (DM)
WinTV Go Mod. 603	PCI	Ja / Mono	-	Ja	Ja	Nein	Nein	1 x Comp.	Win 95/98/ME NT 4.0/2000	129,-
WinTV Primio-FM Mod. 719	PCI	Ja / Mono	ja	Ja	Ja	Ja	Nein	1 x Comp.	Win 95/98/ME NT 4.0/2000	189,-
WinTV PCI-FM Mod. 718	PCI	Ja / 2-Kanal Stereo	ja	Ja	Ja	Ja	Nein	1 x Comp/ S-Video	Win 95/98/ME NT 4.0/2000	249,-
WinTV Theater Mod. 497	PCI	Ja / 2-Kanal Stereo	Ja	Ja	Ja	Ja	Ja	1 x Comp. + 1 x S-Video	Win 95/98/ME NT 4.0/2000	309,-
WinTV CinemaPro Mod. 288	ISA	Ja / Mono	-	Ja	Ja	Nein	Nein	1 x Comp. (S-Video optional)	Win 3.1x Win 95 Win NT	699,-

(Beispiele von www.hauppage.com, andere Hersteller/Anbieter entsprechend)

Macrovision



Schutz gegen (analoge) Kopien per Videorekorder:

- zusätzliche (falsche) Synchronimpulse
- wechselnde Amplitude zur Verwirrung der Aussteuerung (AGC)
- im unsichtbaren Bereich: Fernseher ignoriert das Signal

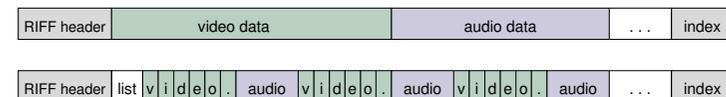
Video: digitale Speicherung ...

- 720x576 Pixel RGB888 bei 25 fps: 30 MB Daten / Sekunde
- unkomprimiertes Video liefert > 100 GByte / Stunde

=> Videodaten müssen extrem komprimiert werden

- Kompression des Einzelbildes (z.B. JPEG)
- Ausnutzen der Korrelationen aufeinanderfolgender Bilder "Bewegungskompensation"
- Synchronisation von Audio, Video, Untertiteln, Suchhilfen, ...
- "interleaved" Dateiformate: AVI, ASF, MPEG, ...
- 32-bit Dateisystem: Dateigröße limitiert auf 4 GB (2 GB)
- erfordert 64-bit Systeme oder Verkettung mehrerer Dateien

AVI: Audio Video Interleaved



normal / interleaved

- RIFF-Dateiformat für Multimedia / Videos
- eingeführt mit Video for Windows (Win 3.11)
- definiert ca. 20 Chunk-Typen
- List-Chunk erlaubt "verwobene" Daten
 - Audiodaten, Videoframes (BMP), Audiodaten, ...
 - Datei muß vor Abspielen nicht voll geladen werden
- Spezifikation in Windows API, Übersicht in c't 94/11 S.327
- mittlerweile von ASF abgelöst (s.u.)

AVI: Beispiel

(0x00000000)	ID:<RIFF> Size:0x000fe964	RIFF: AVI
	Form Type = <AVI >	
(0x0000000c)	ID:<LIST> Size:0x000007d4	Header
	List Type = <hdrl>	
(0x00000018)	ID:<kavih> Size:0x00000038	
(0x00000058)	ID:<LIST> Size:0x00000474	[c't 11/94 327f]
	List Type = <strl>	
(0x00000064)	ID:<strh> Size:0x00000038	
(0x000000a4)	ID:<strf> Size:0x00000428	
(0x000000d4)	ID:<LIST> Size:0x0000005c	
	List Type = <strl>	
(0x000004e0)	ID:<strh> Size:0x00000038	
(0x00000520)	ID:<strf> Size:0x00000010	
(0x00000538)	ID:<vedt> Size:0x00000008	
(0x00000548)	ID:<JUNK> Size:0x00000298	
(0x000007e8)	ID:<LIST> Size:0x000fd184	
	List Type = <movi>	
(0x000007f4)	ID:<LIST> Size:0x000005b0	Daten (MOVI):
	List Type = <rec >	Audio / Video / ...
(0x00000800)	ID:<01wb> Size:0x000005a4	
...	/* einige Male nur Sound */	
(0x00004c98)	ID:<LIST> Size:0x000019da	
	List Type = <rec >	
(0x00004ca4)	ID:<00dc> Size:0x00001422	
(0x000060ce)	ID:<01wb> Size:0x000005a4	
...	/* diverse Male Sound und Bild */	
(0x000db804)	ID:<LIST> Size:0x00002d72	
	List Type = <rec >	
(0x000db810)	ID:<00dc> Size:0x00002d66	
...	/* einige Male nur Bild */	
(0x000fd974)	ID:<idx1> Size:0x00000ff0	Index

Medientechnik | WS 2001 | 18.204

ASF: Advanced Streaming Format

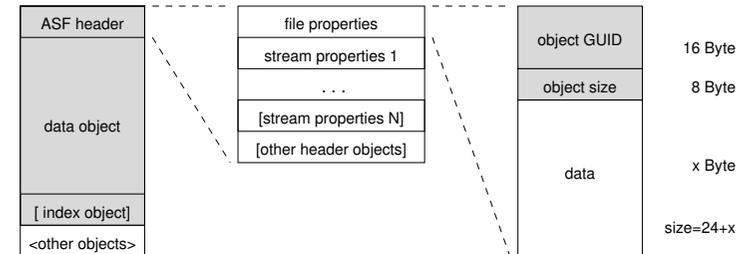
"... an extensible file format designed to store synchronized multimedia data. It supports data delivery over a wide variety of networks and protocols, while still proving suitable for local playback. The explicit goal of ASF is to provide a basis for industry-wide multimedia interoperability, with ASF being adopted by all major streaming solution providers."

[Microsoft, Real Networks 1998]

- RIFF-ähnliche Struktur: header, index, interleaved data
- Chunks (=:"objects") per GUID gekennzeichnet
 - bei Bedarf Registrierung der GUIDs bei Microsoft
 - erlaubt z.B. Codec-Download
 - auch der Player identifiziert sich über seine GUID
 - sehr feine Copyright / Nutzungskontrolle
- siehe ASF-Spezifikation (Version 1.0, 26.02.1998)
- aber aktuelle Version (WMA) von M\$ nicht mehr dokumentiert

Medientechnik | WS 2001 | 18.204

ASF: Object



- Datei besteht aus ASF objects (statt RIFF chunks)
 - oberste Ebene: header / data / [Index] / [more objects]
 - weitere Ebenen durch Einteilen des data-object
 - insbesondere: mehrere Datenströme innerhalb des data-object
 - Header-Object bei Bedarf wiederholt senden
 - Größenfeld erlaubt Objekte bis zu 2**64 Byte

Medientechnik | WS 2001 | 18.204

ASF: GUID

GUID/UUID := "globally/universally unique IDs"

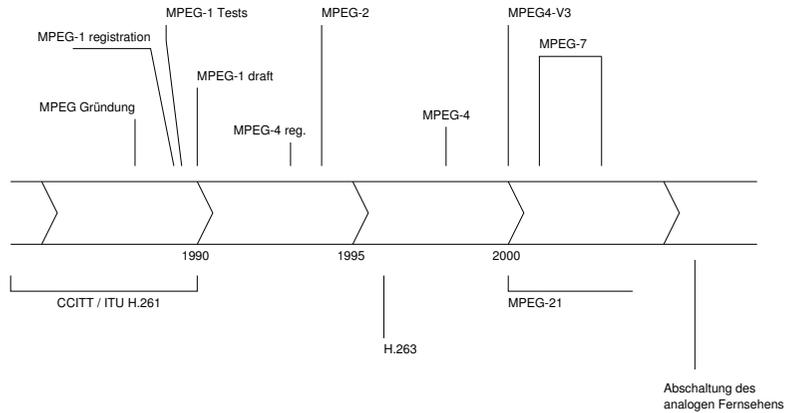
- 128-bit Signatur
- entwickelt für NCS (Apollo), übernommen in OSF/DCE und Windows
 - Ethernet-MAC Adresse plus Zeitmarke (eindeutig)
 - oder 47-bit Zufallsadresse plus Zeitmarke
 - Zeit mit 100ns Auflösung, Sequenznr. zur Korrektur (reboot)
 - eindeutig bis 3400 n.Chr.

ASF-Header:	D6E229D1-35DA-11DA-9034-00A0C90349BE
ASF-Data:	D6E229D2-35DA-11DA-9034-00A0C90349BE
ASF-Index:	D6E229D3-35DA-11DA-9034-00A0C90349BE
usw.	time-low-mid-high seq. ethernet----

- jeder Windows-Rechner eindeutig identifizierbar

Medientechnik | WS 2001 | 18.204

MPEG: Entwicklung



(www.cseit.it/mpeg/, JACM 34/1991)

Medientechnik | WS 2001 | 18.204

MPEG: Übersicht

MPEG-1	Audio/Video-Kodierung, Bitrate bis 1.5 Mbps Video-CD
MPEG-2	Audio/Video-Kodierung, Bitraten bis 15 Mbps, div. Profile digitales Fernsehen (DVB-T/-S/-C), DVD Qualität ausreichend für HDTV ("MPEG-3")
MPEG-4	Audio/Video-Kodierung für mobile Systeme, objektbasierte Kodierung, extrem geringe Bitraten vielfältige Optionen ("face descriptor", "structured audio", ...)
MPEG-7	"Meta"-Kodierung, Medienbeschreibung und -suche, ...
MPEG-21	geplantes universelles "Multimedia Framework" "digital item description", "rights management", ...

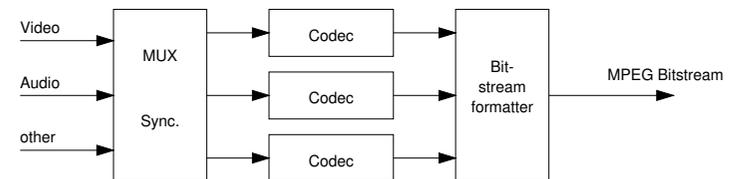
Medientechnik | WS 2001 | 18.204

MPEG: Konzept

- MPEG-Gremien erarbeiten Richtlinien für Medien-Kodierung
- Publikation als ISO/ITC-Standards
- Standards definieren das Datenformat ("bitstream")
- Referenzimplementierung für Decoder verfügbar
- sichert Interoperabilität aller Decoder
- aber Encoder nicht spezifiziert
- erlaubt spätere Verbesserungen
- erlaubt Wettbewerb / Konkurrenz
- Realisierung in Software oder mit Hardware/VLSI
- Tradeoff (Bild-) Qualität / Latenz / Performance

Medientechnik | WS 2001 | 18.204

MPEG: Video



- Standards definieren das Datenformat
- MPEG-1: nur Audio / Video
- MPEG-2: mehrere Audio/Videostreams, Untertitel, ...
- MPEG-4: viele weitere Objekttypen möglich

Medientechnik | WS 2001 | 18.204

MPEG-1:

- erster Standard für digitales Video (CD, DAT, disks)
- digitale Kodierung, Anforderungen:
 - random access, access time < 0.5 sec.
 - fast forward / reverse searches
 - reverse playback
 - audio-visual synchronisation
 - robustness to errors
 - decoding delay < 1 sec. ("interactive")
- Bildformat 352x288 Pixel, 25 fps
- Bitrate gesamt 1.5 Mbps, davon 64..192 kbps für Audio
- Anwendung für die Video-CD, Spieldauer ca. 60 min
- Demo ("goodtime", "pink panther")

Medientechnik | WS 2001 | 18.204

MPEG: vs. H.26x

MPEG-1/-2:

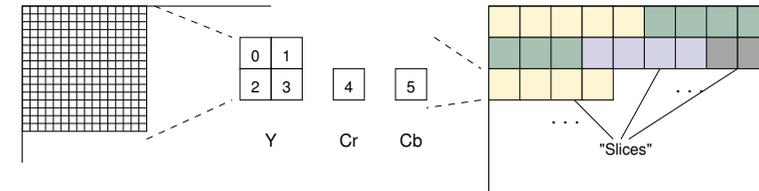
- Kodierung und Speicherung von Videodaten
- keine Echtzeit- / Latenzanforderungen
- aufwendige, komplexe (2-pass) Kodierung möglich

H.261, H.263:

- Video-Telephon, Video-Konferenzen
- für (mehrfache) ISDN-Bandbreite: $p \times 64$ kbps
- erfordert Echtzeit-Kodierung, Übertragung, Dekodierung
- minimale Latenz, möglichst < 0.3s
- sehr ähnliche Algorithmen
- Kodierung basiert auf 8x8 DCT-Blöcken mit Entropiekodierung

Medientechnik | WS 2001 | 18.204

MPEG-1: Bildformat



Bildformat ähnlich wie JPEG:

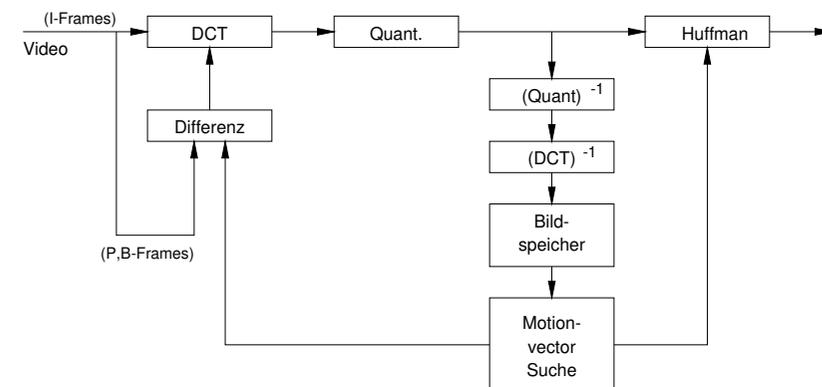
- Blockbildung 8x8 Blöcke für DCT
- 4:2:0 YCrCb-Kodierung
- Makroblöcke aus 4 Blöcken Y, je ein Block Cr Cb

"Slices" aus mehreren Makroblöcken

"Pictures" aufgebaut aus Slices

Medientechnik | WS 2001 | 18.204

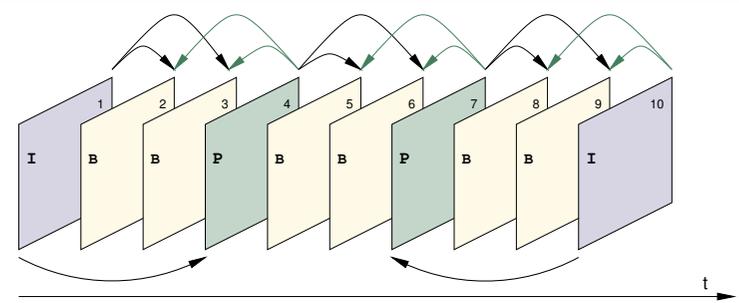
MPEG-1: Blockschaltbild



- Suche nach Bewegungsvektoren besonders aufwendig

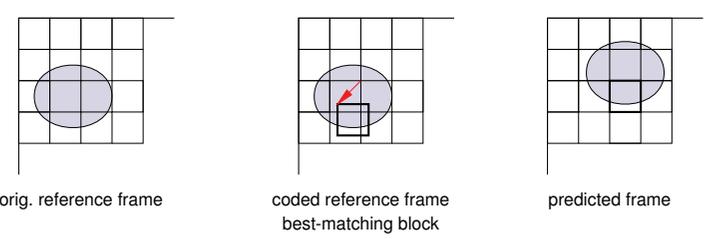
Medientechnik | WS 2001 | 18.204

MPEG-1: IPB-Frames



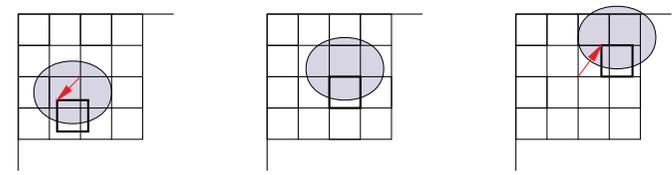
- "Intra-Frames": eigenständig kodiertes Bild
- "Predicted": Vorhersage aus letztem Intraframe
- "Bidirectional": Interpolation aus zwei anderen Bildern (DC-Frames)
nur DC-Koeffizient kodiert, für schnelle Vorschau
- Übertragung: (GOP I1 P4 B2 B3) (GOP I10 P7 B5 B6) (GOP B8 B9) ...

MPEG: Prediction



- "motion compensation" / "motion vector":
- Suche des am besten passenden 8x8 Blocks
 - im dekodierten (d.h. quantisierten!) Referenzbild
 - "Bewegungsvektor": relative Lage des Referenzblocks
 - Kodierung der Differenz zum Referenzblock

MPEG-1: B-Frames



bi-directional prediction:

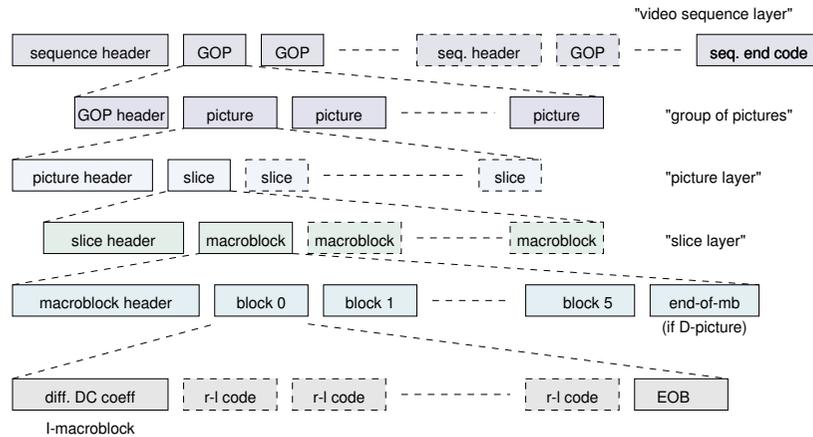
- Suche nach "passenden" Blöcken in beiden Referenz-Frames
- Kodierung von zwei Bewegungsvektoren mv1 und mv2
- Kodierung der Differenz:
$$B(x) = [I1(x+mv1) + I2(x+mv2)] / 2$$
- B-Frames sind besonders klein (und unscharf)

MPEG-1: Huffman-Kodierung

0/1	1s (first)	2	1/1	011s	4
0/1	11s (next)	3	1/2	0001 10s	7
0/2	0100 s	5	1/3	0010 0101 s	9
0/3	0010 1s	6	1/4	0000 0011 00s	11
0/4	0000 110s	8	1/5	0000 0001 1011 s	13
0/5	0010 0110 s	9	1/6	0000 0000 1011 0s	14
0/6	0010 0001 s	9	1/7	0000 0000 1010 1s	14
0/7	0000 0010 10s	11	1/8	0000 0000 0011 111s	16
0/8	0000 0001 1101 s	13	1/9	0000 0000 0011 110s	16
0/9	0000 0001 1000 s	13	1/10	0000 0000 0011 101s	16
0/10	0000 0001 0011 s	13	1/11	0000 0000 0011 100s	16
0/11	0000 0001 0000 s	13	1/12	0000 0000 0011 011s	16
0/12	0000 0000 1101 0s	14	1/13	0000 0000 0011 010s	16
0/13	0000 0000 1100 1s	14	1/14	0000 0000 0011 001s	16
0/14	0000 0000 1100 0s	14	1/15	0000 0000 0001 0011 s	17
0/15	0000 0000 1011 1s	14	1/16	0000 0000 0001 0010 s	17
0/16	0000 0000 0111 11s	15	1/17	0000 0000 0001 0001 s	17
0/17	0000 0000 0111 10s	15	1/18	0000 0000 0001 0000 s	17
0/18	0000 0000 0111 01s	15	2/1	0101 s	5
0/19	0000 0000 0111 00s	15	2/2	0000 100s	8
0/20	0000 0000 0110 11s	15	2/3	0000 0110 11s	11
0/21	0000 0000 0110 10s	15	2/4	0000 0001 0100 s	13
0/22	0000 0000 0110 01s	15	2/5	0000 0000 1010 0s	14
0/23	0000 0000 0110 00s	15	3/1	0011 1s	6
0/24	0000 0000 0101 11s	15	3/2	0010 0100 s	9
0/25	0000 0000 0101 10s	15	3/3	0000 0001 1100 s	13
0/26	0000 0000 0101 01s	15	3/4	0000 0000 1001 1s	14
0/27	0000 0000 0101 00s	15	4/1	0011 0s	6
0/28	0000 0000 0100 11s	15	4/2	0000 0011 11s	11
0/29	0000 0000 0100 10s	15	4/3	0000 0001 0010 s	13
0/30	0000 0000 0100 01s	15	5/1	0001 11s	7
0/31	0000 0000 0100 00s	15	5/2	0000 0010 01s	11
0/32	0000 0000 0011 000s	16	5/3	0000 0000 1001 0s	14
0/33	0000 0000 0010 111s	16	6/1	0001 01s	7
0/34	0000 0000 0010 110s	16	6/2	0000 0001 1110 s	13
0/35	0000 0000 0010 101s	16	6/3	0000 0000 0001 0100 s	17
0/36	0000 0000 0010 100s	16	7/1	0001 00s	7
0/37	0000 0000 0010 011s	16	7/2	0000 0001 0101 s	13
0/38	0000 0000 0010 010s	16	8/1	0000 111s	8
0/39	0000 0000 0010 001s	16	8/2	0000 0001 0001 s	13
0/40	0000 0000 0010 000s	16			

- MPEG "variable-length run-level codes"
- Konzept wie bei JPEG

MPEG-1: Datenstrom



Medientechnik | WS 2001 | 18.204

MPEG-1: Sync-Muster

Start code	Hex	Binary
extension.start	000001B5	00000000 00000000 00000001 10110101
GOP.start	000001B8	00000000 00000000 00000001 10111000
picture.start	00000100	00000000 00000000 00000001 00000000
reserved	000001B0	00000000 00000000 00000001 10110000
reserved	000001B1	00000000 00000000 00000001 10110001
reserved	000001B6	00000000 00000000 00000001 10110110
sequence.end	000001B7	00000000 00000000 00000001 10110111
sequence.error	000001B4	00000000 00000000 00000001 10110100
sequence.header	000001B3	00000000 00000000 00000001 10110011
slice.start.1	00000101	00000000 00000000 00000001 00000001
...
slice.start.175	000001AF	00000000 00000000 00000001 10101111
user.data.start	000001B2	00000000 00000000 00000001 10110010

- vordefinierte 32-bit Werte zur Synchronisation im Datenstrom
- erlauben Wiederaufsetzen nach Fehlern

Medientechnik | WS 2001 | 18.204

MPEG1: Profiling

```
Flat profile of 214.45 secs (3628 total ticks): Thread-1
Interpreted + native  Method
1.9%  0 + 62  java.lang.Object.notifyAll
0.5%  0 + 15  java.io.FileInputStream.readBytes
...
3.6%  14 + 105 Total interpreted

Compiled + native  Method
28.7%  936 + 0  Element.Make_Picture (YUV -> RGB, paint())
10.0%  326 + 0  IDCT.invers_dct
9.6%  314 + 0  motion_data.get_area
6.1%  199 + 0  motion_data.put_area
4.1%  133 + 0  MPEG_video.correct_lum_pixel
3.4%  111 + 0  Huffmann.decode
2.7%  89 + 0  motion_data.copy_area
2.1%  68 + 0  MPEG_video.correct_col_pixel
...
74.0%  2412 + 1  Total compiled (including elided)
```

- Wiedergabe einer MPEG-1 Datei, 352x288
- JDK 1.3 Hotspot Client VM

(mvs.informatik.tu-chemnitz.de/~jan/MPEG/HTML/mpeg_file.html)

Medientechnik | WS 2001 | 18.204

MPEG-2: Konzept

- Audio/Videokodierung, Qualität von MPEG-1 bis HDTV
 - bei entsprechend höheren Datenraten, 4 Mb/s .. 80 Mb/s
 - diverse Profiles / Levels vordefiniert
 - Draft 1993, Standard seit Ende 1994
-
- Mehrkanal-Audio, MPEG-BC/AAC Codecs
 - für DVDs aber meistens Dolby AC3 oder DTS
-
- Untertitel (Sprites)
 - mehrere Videoströme, mehrere Blickwinkel, Branching
 - interaktive, animierte Menüs
-
- "scalable": verschiedene Auflösungen / Anpassung an Bitrate

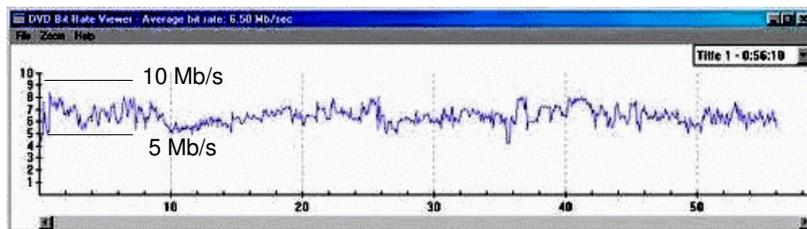
Medientechnik | WS 2001 | 18.204

MPEG: Levels, Profiles

profile	simple	main	SNR scalable	spatial scalable	high profile
B-frames	no	yes			
scalable	no	no	SNR	SNR or spatial	
subsampling	4:2:0			4:2:0/4:2:2	
high-level 1920x1080		< 80 Mbps			< 100 Mbps
high-1440 1440x1152		< 60 Mbps		< 60 Mbps	< 80 Mbps
main-level 720x576	< 15 Mbps	< 15 Mbps	< 15 Mbps		< 20 Mbps
low 360x288		< 4 Mbps	< 4 Mbps		

DVD / DVB benutzt MP@ML, Datenrate bis 15 Mbps

DVD: Datenrate MPEG-2



- Beispiel für Datenrate einer Video-DVD: 5 .. 10 Mb/s
- digitales Fernsehen (DVB-S) mit 6 Mb/s
- MP@ML erlaubt bis 15 Mb/s maximal

MPEG: Audio

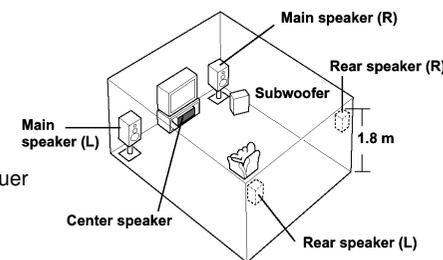
Verfahren	Methode	kHz	Kanäle	kb/s	Qualität
MPEG-1 layer 1	hybrid	32, 44.1, 48	1, 2	32 .. 448	AM / FM / (CD)
MPEG-1 layer 2	hybrid	32, 44.1, 48	1, 2	32 .. 384	AM / FM / (CD)
MPEG-1 layer 3	hybrid	32, 44.1, 48	1, 2	32 .. 320	AM / FM / CD
MPEG-2 AAC	hybrid	16 .. 48	1 .. 5.1	16 ..	AM / FM / CD
Dolby AC-2	trans.	44.1	2	256/ch	CD
Dolby AC-3	trans.	44.1	1 .. 5.1	32 .. 384	CD
DTS	trans.	44.1	1 .. 7.1	768 (5.1)	CD

- derzeit am häufigsten verwendet: MP3, AC-3
- MPEG-4 definiert diverse weitere Codecs
- insbesondere Sprache bei extrem geringer Bitrate

MPEG: Mehrkanalaudio

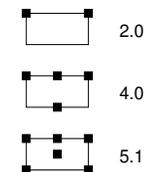
Film als "Erlebnis"?

- erfordert "Eintauchen"
- Übereinstimmung Bild / Ton
- auch für außen sitzende Zuschauer
- "Surround"-Geräusche



Mehrkanal-Audio mit mehreren Lautsprechern:

- Hauptlautsprecher "normal"
- Center direkt über/unter dem Bild
- Surroundlautsprecher seitlich oder an der Rückwand
- Subwoofer seitlich vorne
- X.Y Notation: X vollwertige Kanäle, Y LFE-Kanäle



MPEG: Two Pass Encoding

- möglichst geringe mittlere Datenrate
- möglichst hohe Bildqualität auch in kritischen Szenen
- asymmetrische Anwendungen erlauben aufwendige Kodierung

=> "2 pass encoding"

- erste Kodierung protokolliert Statistiken
- z.B. Bewegungsvektoren, Anzahl notwendiger I-Makroblöcke, Signal-Rausch-Abstand, usw.
- zweite Kodierung mit dynamisch angepasster Qualität:
 - höhere Bitrate in kritischen, stark bewegten Szenen
 - reduzierte Bitrate für "unkritische" Szenen

Medientechnik | WS 2001 | 18.204

H.261: p64

- CCITT/ITU Standard für Videotelephonie
- Kodierung in Echtzeit mit geringer Latenz
- Bandbreite als Vielfaches von ISDN (64kbps)
- $p = 1 \dots 30$ (64 kbps .. 1.92 Mbps)



"Miss America, 176x144"

Auflösungen:

SQCIF	128 x 96
QCIF	176 x 144
CIF	352 x 288
4CIF	(704 x 576)

- Verfahren gleichzeitig mit MPEG-1 entwickelt
- ebenfalls 8x8 DCT und Huffman-Kodierung
- leicht andere Quantisierungsregeln

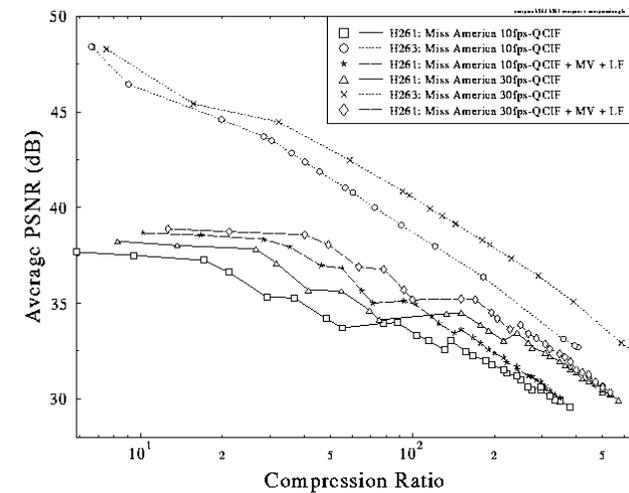
Medientechnik | WS 2001 | 18.204

H.263

- Erweiterung von H.261, ITU-Standard 1996
- höhere Genauigkeit der Bewegungsvorherage (half-pixel)
- B-Frames (bidirektionale Vorhersage)
- Einsparen von Fehlerkorrekturbits
- fünf verschiedene Auflösungen, zusätzlich 4CIF 704x576, 16CIF 1408x1152
- erlaubt höhere Kompression als H.261
- Anwendung für MPEG-4 Videokodierung

Medientechnik | WS 2001 | 18.204

Bildqualität: PSNR H.261 vs H.263



Medientechnik | WS 2001 | 18.204

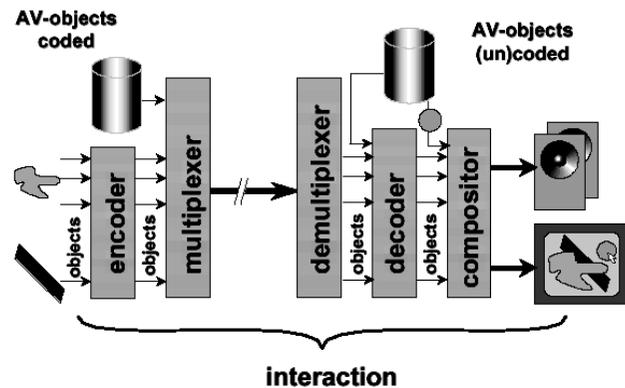
MPEG-4: Konzept

MPEG-4: "Coding of Audio-Visual Objects"

- Satz von standardisierten "Werkzeugen"
- für eine Vielzahl von möglichen Anwendungen
- (broadcast, interactive, conversational)
- insbesondere für mobile Geräte
- daher möglichst geringe Bitraten / hohe Kompression
- aber: Standardformat wichtiger als Kompression
- objektbasiertes Konzept: A&V "objects"
- "compositor" inszeniert die Objekte in (2D oder 3D-) Szenen
- spezielle Kodierung für jedes einzelne Objekt möglich
- unabhängig von der Bitrate

Medientechnik | WS 2001 | 18.204

MPEG-4: Übersicht

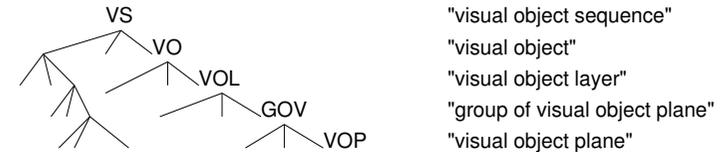


- BIFS: "binary format for scene description"
- SNHC: "synthetic-natural hybrid coding"

Medientechnik | WS 2001 | 18.204

MPEG-4: Videokodierung

- Szene als Baum:



- "texture coding" (die eigentliche Videokodierung):
IBP-Frames, Bewegungskompensation wie MPEG / H.263
- "shape coding": Kodierung nicht-rechteckiger Objekte
- "sprite coding": bel. geformte Sprites zur Überlagerung
- "temporal scaling": automatische Anpassung der Framerate

Medientechnik | WS 2001 | 18.204

MPEG-4: DivX

- Microsoft liefert "MPEG4V3" Codec für Windows
 - Funktionsumfang beschränkt auf H.263 Videocodec
 - trotzdem gut geeignet zur Videokodierung
 - aber Encoder schreibt nur WMV, kein AVI
- => DivX :-): Hack: Freischaltung des Encoders für AVI-Format
- Kombination mit MP3 oder WMA für Audio
 - bis ca. 2h Spielfilm auf einer CD

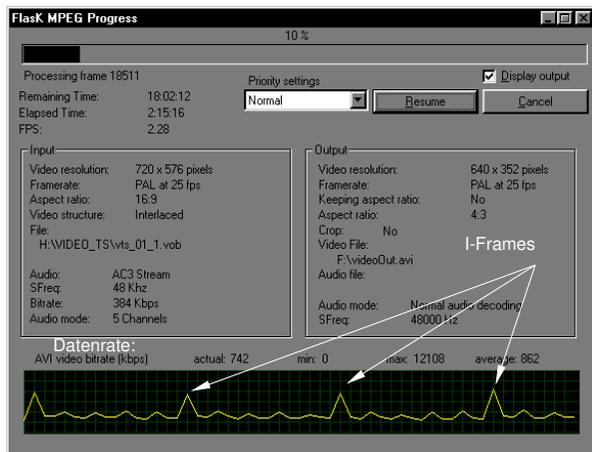
OpenDivX:

- neuentwickelter, Open-Source MPEG-4 Codec
- dekodiert auch DivX 3.xx Datenströme
- Download inklusive Player von divx.com
- Demo

(www.divx.com)

Medientechnik | WS 2001 | 18.204

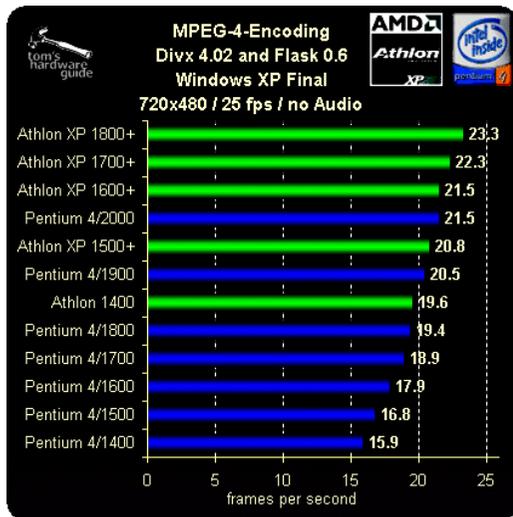
MPEG-4: FlaskMPEG



- Tool zur Konvertierung zwischen Videoformaten
- im Beispiel: DVD (vob) -> DivX (avi)

(www.flaskmpeg.net/)

OpenDivX: Performance



stark abhängig von:

- Speicherbandbreite
- MMX / SSE / 3DNow!
- nicht echtzeitfähig

(www.tomshardware.com)

3ivX: low artifact mode



- Blockartefakte: 3ivX-Codec, 1Mbps, normal / low-artifact mode

(ct 10/2001 122ff)

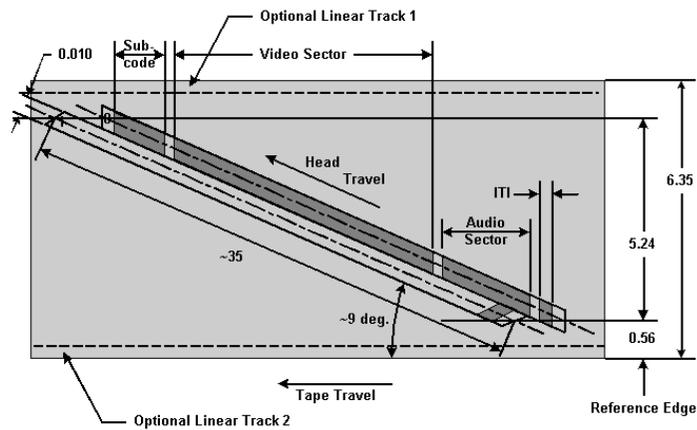
- Leerseite

DV: Digital Video

- aktueller Standard für digitale Camcorder
- im Prinzip volle PAL-Auflösung (abhängig vom CCD-Sensor)
- modifiziertes M-JPEG Verfahren zur Datenkompression
- Datenrate 3.5 MB/s
- digitale Aufzeichnung auf Magnetband
- Datentransfer via FireWire/IEEE 1394 Bus
- www.dvcentral.org/dvwhat.html, www.imagendv.com

Medientechnik | WS 2001 | 18.204

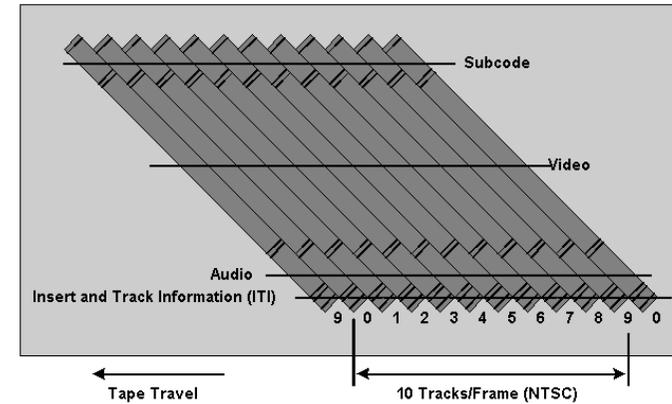
DV: Schrägspuraufzeichnung



- Mechanik ähnlich wie VHS/S-VHS/DAT, aber digitale Aufzeichnung

Medientechnik | WS 2001 | 18.204

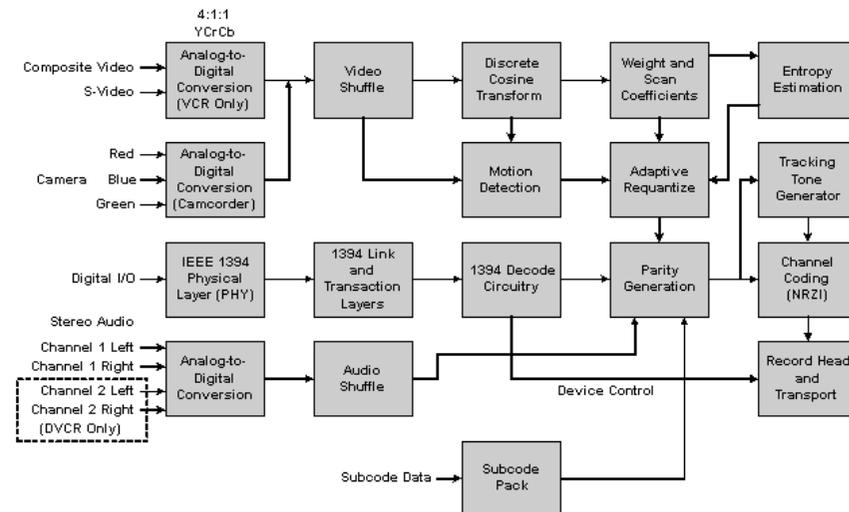
DV: Schrägspuraufzeichnung



- getrennte Bandbereiche für Audio / Video / Subcode

Medientechnik | WS 2001 | 18.204

DV: Datenfluss



Medientechnik | WS 2001 | 18.204

MHP: Aktuell



ARD-Vorsitzender Pleitgen sieht MHP vor endgültigem Durchbruch in Europa

[19.11.2001 16:59]

Der ARD-Vorsitzende Fritz Pleitgen begrüßt die Einführung des digitalen Fernsehstandards Multimedia Home Platform (MHP[1]) auf dem US-amerikanischen Markt. "Dieser Schritt wird der Einführung von MHP als einheitlichem europäischen Standard endgültig zum Durchbruch verhelfen", erklärte Pleitgen am heutigen Montag. Das Konsortium der bedeutendsten US-amerikanischen Kabelnetzbetreiber, Cable Television Laboratories (Cable Labs[2]), hatte vergangene Woche angekündigt, sich für MHP zu entscheiden. MHP ist der offene Standard für digitale Set-Top-Boxen, der interaktive Anwendungen ermöglichen soll.

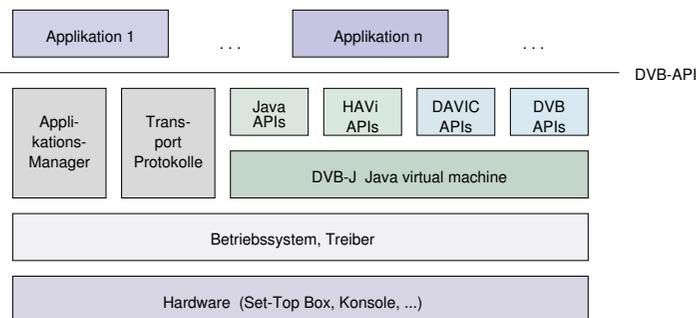
In Deutschland haben sich in den vergangenen Monaten alle wichtigen Programmveranstalter – ARD, ZDF, RTL und die Kirch-Gruppe – sowie die Landesmedienanstalten auf die Einführung von MHP in Deutschland geeinigt[3]. "Versuche von Liberty Media, in Deutschland einen einheitlichen Standard zu verhindern, können nicht akzeptiert werden. Im Interesse des Publikums brauchen wir klare Verhältnisse", forderte Pleitgen. Der US-Investor Liberty, der von der Deutschen Telekom rund 60 Prozent des Kabelnetzes für 5,5 Milliarden Euro gekauft hat[4], will auf MHP aus Kostengründen verzichten[5]. Seinen zehn Millionen Kunden will Liberty Decoder schenken[6]. (dpa) / (anw[7]/c't)

URL dieses Artikels:

<http://www.heise.de/newsticker/data/anw-19.11.01-004/>

Medientechnik | WS 2001 | 18.204

MHP: API



- DVB-J basiert auf JDK 1.1.8 und JMF (mit Erweiterungen)
- zusätzliche APIs für DVB-spezifische Funktionen
- z.B. Graphik-Overlays, Sprites, ...
Beispiele: siehe MHP-Spezifikation, Anhang W

Medientechnik | WS 2001 | 18.204

MHP: Multimedia Home platform

erweiterte Funktionen beim Fernsehen?

- "MPH" Vorschlag und Spezifikation
- vom DVB-Konsortium (digital video broadcasting)
- für Set-Top Boxes, Konsolen, PCs, usw.

drei Stufen / Profiles:

- "enhanced broadcast" Programinfos, Newsticker, ...
 - "interactive broadcast" Spiele, Shopping, ... (Rückkanal)
 - "internet access" Web, Mail, Chat, ...
-
- angepasste Java-Virtual Machine (DVB-J)
 - Standard wird mittlerweile weltweit unterstützt

(www.dvb.org, www.mhp.org)

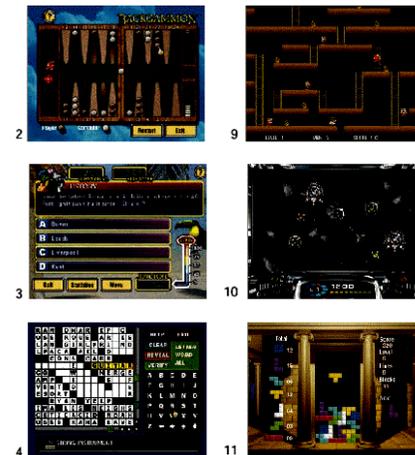
Medientechnik | WS 2001 | 18.204

MHP: Beispiel

Beispiel für MHP Interaktion:

SNAP2 the fun!

- Board Games**
 - 1 Othello
 - 2 Backgammon
 - 3 Trivia
 - 4 Crossword
 - 5 Chinese Solitaire
- Card Games**
 - 6 Solitaire
 - 7 Black Jack
 - 8 Five Card Draw
- Arcade Games**
 - 9 Professional Gold Digger
 - 10 Zero G
 - 11 Tetris™
- Children's Games**
 - 12 Twins Café Match Game
 - 13 Hangman
 - 14 Tic Tac Toe



- Spiele jeweils als Java- oder "native" Versionen

(www.snaptwo.com)

Medientechnik | WS 2001 | 18.204

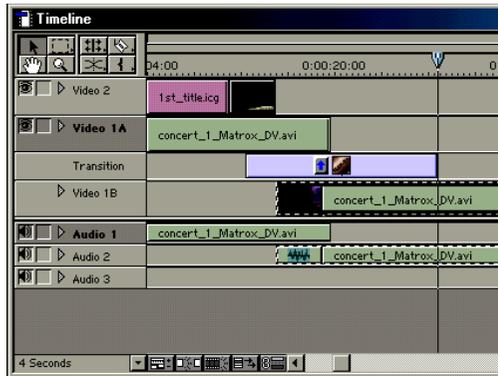
Animationen

"Satz von Multimedia-Daten, die paketweise räumlich korreliert sind und von Paket zu Paket eine zeitliche Korrelation aufweisen. Ein Paket mit intern lediglich räumlicher Korrelation wird als Frame bezeichnet."

(Henning 6.1)

- Definition umfaßt alles: Slideshows .. Videoclips
- Frame = einzelnes Bild, Field = Halbbild (Video)
- Produktion, Movie = gesamte Animation
- Clip = einzelne Szene
- Key Frame = vollständige, eigenständige Einzelbilder
- Animationspfad = Kurve, entlang der sich Objekte bewegen

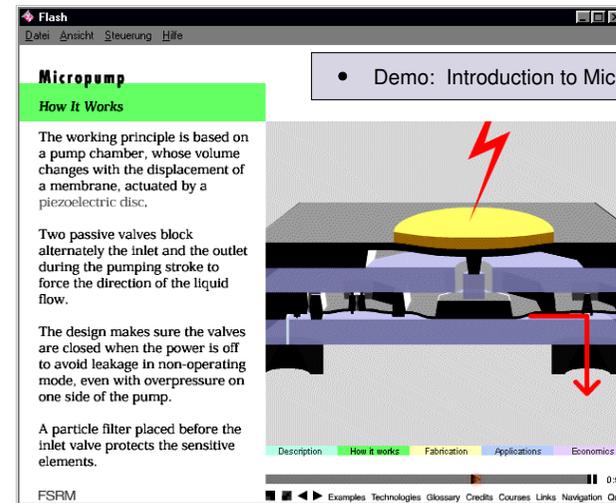
Animation: Timeline



- Anordnung der einzelnen Clips, Blenden, usw.
- Synchronisation von Audio/Video/Untertiteln

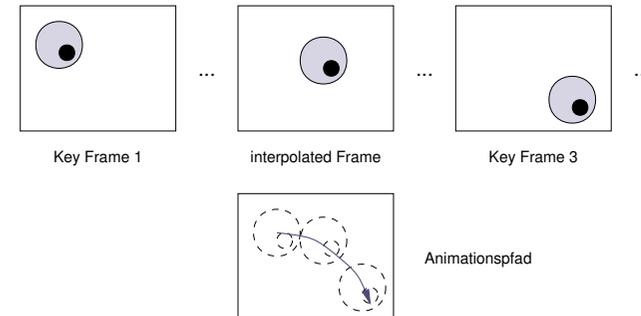
(Screenshot Premiere)

Shockwave, Flash



(www.macromedia.com / www.fsrn.ch)

Animation: Keyframes



- volle Spezifikation aller Objekt-positionen / -eigenschaften
- nur zu bestimmten, ausgezeichneten Zeitpunkten:

=> "Keyframes"

- Software interpoliert Objektbewegungen / -attribute
- an allen notwendigen Zwischenbildern

Haptische Interfaces

- Motivation
- Physiologie
- Sensoren zur Orts- und Kraftmessung
- Verfahren zur Kraftrückmeldung
- Gerätespektrum
- Datenhandschuh
- Head- / Eye-Tracking
- VR-Anwendungen
- DirectInput
- Codebeispiel



Medientechnik | WS 2001 | 18.204

Haptik: Literatur

Thema "haptische / multimodale Interfaces" ist recht neu...
... bisher keine geeigneten Lehrbücher erschienen

Konferenzbände:

Int. Workshop on haptic human-computer interaction, Glasgow, 2000, Springer

Int. Conf. on Cooperative multimodal communication, CMC/95, Eindhoven, Springer

<http://haptic.mech.nwu.edu/intro/gallery/>

www.immersion.com

www.sensable.com

www.logitech.com / www.microsoft.com/hardware

Bargen, Donnelly, Inside DirectX, Microsoft Press, 98

Schäpers, DirectX nicht nur für Spiele, c't 9/99-216, 12/99-238, 15/99-180

Medientechnik | WS 2001 | 18.204

Haptik: Motivation

- verbesserte Mensch-Computer Interaktion
 - durch Ausnutzen des Tastsinns ("full-duplex")
 - Unterstützung von behinderten (blinden) Personen
 - direkte Manipulation von Objekten in CAD-Systemen
 - auch für unsichtbare / verdeckte Objekte
 - Erkennen von Objekten über (simulierte) Oberflächen und Reibung
 - verbessertes "Eintauchen" in VR-Umgebungen
 - insb. Massenmarkt 3D-Actionspiele
- aber:
- Wahrnehmung von Force-Feedback bisher wenig erforscht
 - Sicherheitsmaßnahmen notwendig

Medientechnik | WS 2001 | 18.204

Haptik: Status

- aktuelles Forschungsthema
- Entwurf und Herstellung von Sensoren und Aktoren
- Systemintegration, Programmierschnittstellen
- breites Gerätespektrum, bisher weitgehend mechanisch:

	Freiheitsgrade / Parameter
Lenkräder	1
Joystick	2 .. 3
Stifte, Roboterarme	5 .. 6
Datenhandschuh	6 .. 20
"smart skin"	1000+

- aktuelle Spiele unterstützen zunehmend FF-Geräte
- billigere Sensoren durch Mikrosystemtechnik

Medientechnik | WS 2001 | 18.204

Multimodale Interfaces

"multimodal" := Kombination mehrerer Modalitäten (=Sinne)

- Maus- / Tastatur- / Joystick- / Datenhandschuh
- Spracheingabe
- Eye-Tracking / Gestenerkennung
- vielfältige neue Möglichkeiten zur Rechnersteuerung
- aktuelles Forschungsthema
- bisher keine einheitliche Theorie

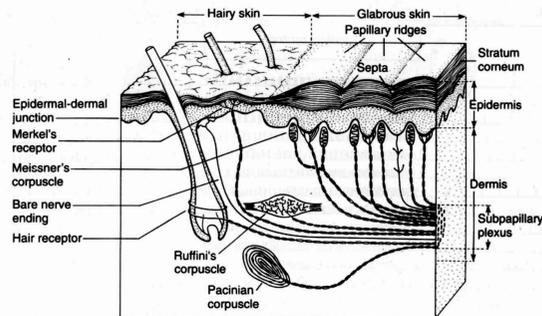
(vgl. Ausschreibung TAMS: "technische Aspekte multimodaler Systeme" am FBI)

Medientechnik | WS 2001 | 18.204

Tastkörperchen

FIGURE 24-2

The location of various receptors in hairy and hairless (glabrous) skin of primates. Receptors are located in the superficial skin, at the junction of the dermis and epidermis, and more deeply in the dermis and in subcutaneous tissue. The receptors of the glabrous skin are: Meissner's corpuscles, located in the dermal papillae, Merkel's receptors, also located in the dermal papillae, and bare nerve endings. The receptors of the hairy skin are: hair receptors, Merkel's receptors (having a slightly different organization than their counterparts in the glabrous skin), and bare nerve endings. Subcutaneous receptors, beneath both glabrous and hairy skin, include Pacinian and Ruffini's corpuscles. (Adapted from Light and Perl, 1984.)



- mehrere Sorten, in der Haut, Haarwurzeln, ...
- unterschiedliche Empfindlichkeiten
- unterschiedliche Wahrnehmung (lokalisiert, großflächig, ...)

(Kandell, Principles of Neural Science)

Medientechnik | WS 2001 | 18.204

Tastsinn: Auflösung

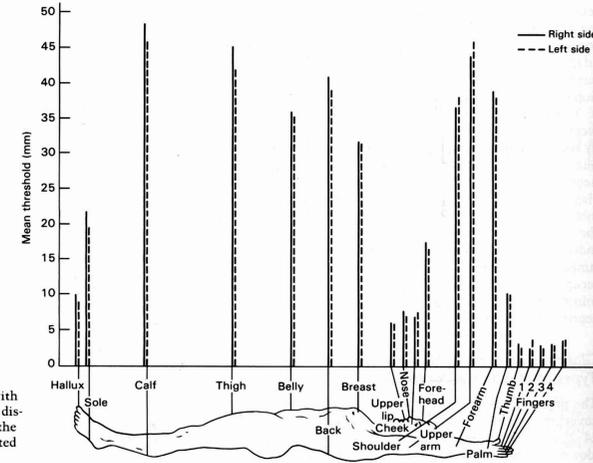


FIGURE 24-7
Two-point discrimination varies with location on body surface. Greatest discriminative capacity is present in the finger tips, lips, and tongue. (Adapted from Weinstein, 1968.)

(Kandell)

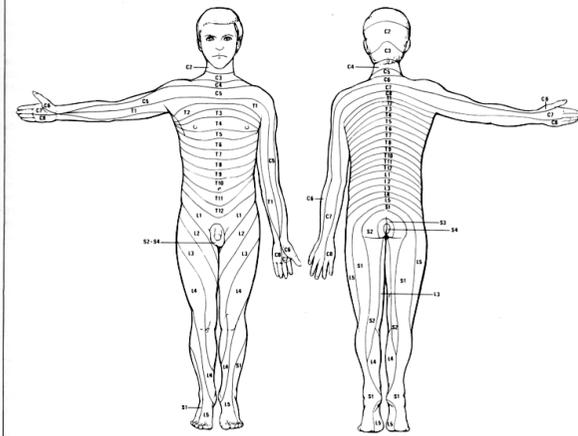
Medientechnik | WS 2001 | 18.204

Tastsinn: Nervenbahnen

FIGURE 25-2

The dermatomes follow a highly regular pattern on the body

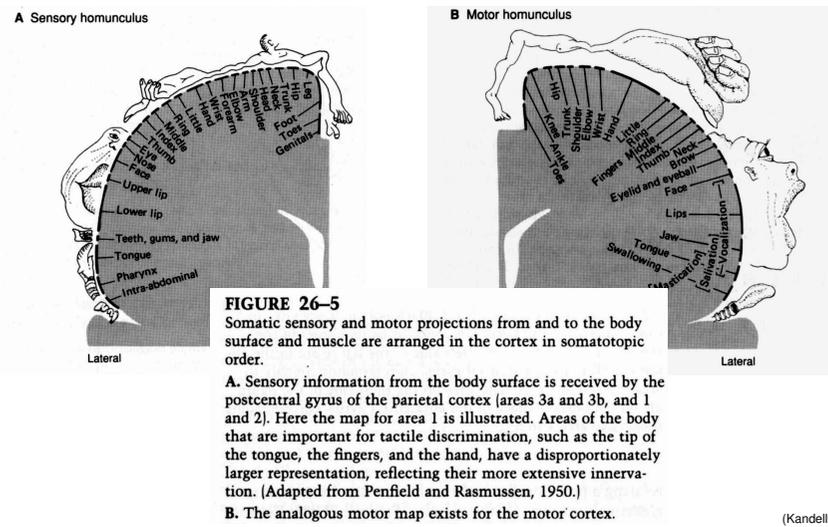
the boundaries of the dermatomes are less distinct than shown here because of overlapping innervation.



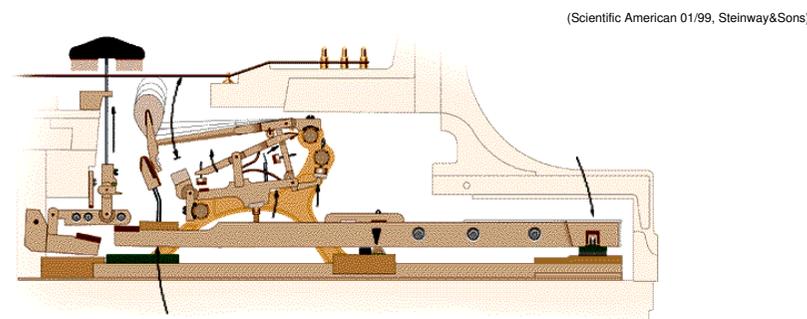
(Kandell)

Medientechnik | WS 2001 | 18.204

Tastsinn: Wahrnehmung

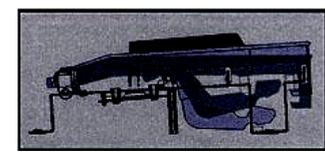


Haptic Interface: Beispiel

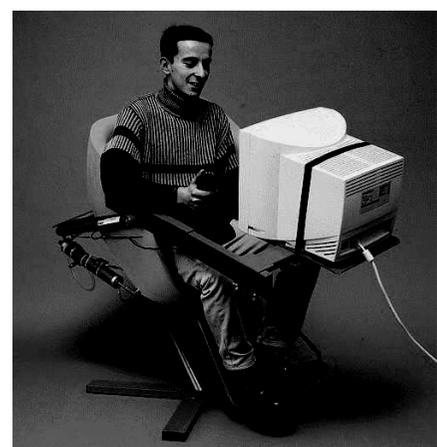


Kunden bezahlen durchaus für gutes "Spielgefühl" :-)

- zum Vergleich:
- Mechanik des Alesis QS8 Synthesizers
 - diverse Kennlinien umschaltbar
 - einstellbare Empfindlichkeit



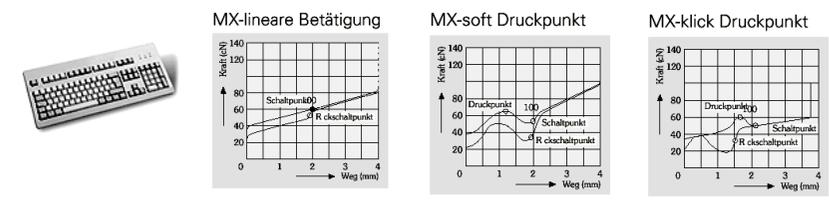
Lage- / Bewegungswahrnehmung:



- Lagewahrnehmung:
- Beschleunigungsmessung
 - "Bogengänge" im Innenohr
 - nicht von außen zugänglich

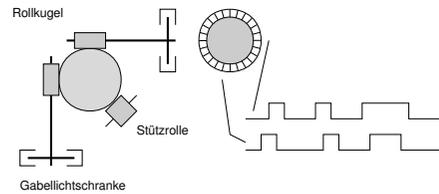
- Simulation / VR:
- Bewegung des Körpers nötig
 - aufwendige Mechanik, z.B.
 - 6-Achsen Hydraulik für prof. Fahr-/Flugsimulatoren
 - Rock'n'Ride: derzeit einzige low-cost Alternative :-)

Tastatur



- Tasten (Mikroschalter/Gummimatte) in QWERTZ-Anordnung
- Organisation als Tastenmatrix
- Ansteuerung und Auslesen im Multiplex-Verfahren
- geringe Datenraten (~ 10 Byte/s), serielles Protokoll
- immer noch das Standard-Eingabegerät

Maus



- Messung von relativen x/y-Bewegungen
- über klassische "Kugelmechanik"
- oder mit optischem Sensor auf gemustertem Untergrund

andere Zeigegeräte:

- Lichtgriffel, druckempfindliche Zeichenstifte
- "Spacemouse" mit x/y/z-Freiheitsgraden

(Zitat)

Medientechnik | WS 2001 | 18.204

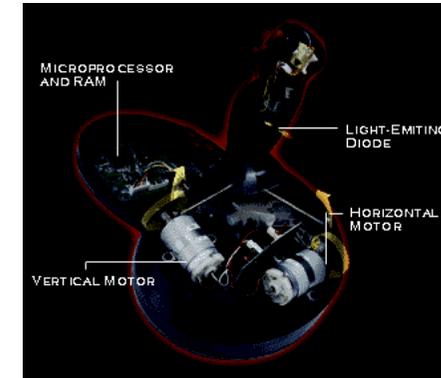
Force-Feedback Joystick

Joystick:

- Arm auf Kugelgelenk
- 2 Freiheitsgrade X, Y
- Rückstellung mit Federn
- diverse Tasten, Coolie-Hat
- evtl. zusätzliche Achsen

mit Force-Feedback:

- X/Y-Achsen mit Motoren statt Federn
- Rückstellkräfte und Vibrationen beliebig einstellbar
- eigener Mikrorechner zur Regelung



Medientechnik | WS 2001 | 18.204

Force Feedback Mouse . . .



- Maus mit Force-Feedback (!)
- absolute statt relativer Position
- eingeschränkter Bewegungsbereich
- höhere Trägheit durch die Mechanik
- kaum nützliche Anwendungen
- (bisher) kein Markterfolg
- Nachfolger iFeel-Mouse:
- normale, vibrierende Maus

(!) *Imagine feeling pictures and links on web pages, the softness corduroy, the rough texture of sand paper. With the Logitech WingMan Force Feedback Mouse, you can do all that and more – all at the ease of USB!*



Medientechnik | WS 2001 | 18.204

Braille-Zeilen



- Darstellung einer Textzeile im Braille-Code
- entsprechende Anzahl einzeln magnetisch betätigter Stifte
- erlaubt Rechnerzugang für Blinde, aber recht teuer

(diverse Hersteller, hier: www.brailletnet.jussieu.fr/accessibilite/livreblanc/handvis.html)

Medientechnik | WS 2001 | 18.204

Hometrainer



TRAININGSROLLEN

■ **Messeneinheit 2002:**
Tacx i-Magic
 Der Hometrainer Tacx T1900 i-Magic wird über eine Steuereinheit am Lenker mit der USB-Schnittstelle des PC verbunden. Durch die reelle Simulation kann auf ausgewählte Routen gewählt werden, oder aber ein Radrennen gegen Gegner auf einer simulierten Radrennbahn fahren.
 1.269,- DM

weitere Modelle:
 Grand Excel 1.125,- DM
 Excel 999,- DM
 Basic 499,- DM
 Swing 299,- DM
 Cycleforce I 249,- DM

Zubehör Hometrainer:
Skyliner sorgt für die natürliche waagerechte Fahrposition 29,- DM
Schweißfänger fängt ihren Schweiß auf und schützt dadurch ihr Fahrrad 29,- DM

- virtuelle Tour-de-France :-)

(Prospekt, Radsport von Hacht, 11/2001)

Medientechnik | WS 2001 | 18.204

Sensible: Phantom

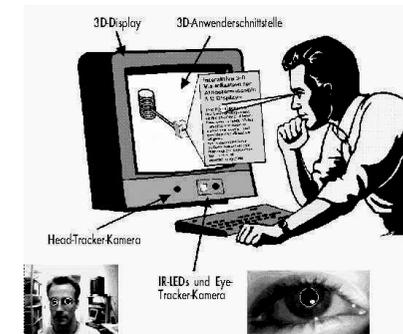


- Positionsmessung des Handgriffs
- 6 Freiheitsgrade: x,y,z-Position, 3 Rotationswinkel
- eingebaute Bremsen für Force-Feedback
- Anwendung z.B. für 3D-Konstruktion

(www.sensible.com)

Medientechnik | WS 2001 | 18.204

Eye- und Head-Tracking



Messung der Augen-/Kopfposition:

- korrekte Benutzerperspektive
- Ansteuerung von 3D-Displays
- LOD: level-of-detail Verfahren
- Darstellung der "Aufmerksamkeitspunkte"
- mit stationären Kameras und (aufwendiger) Bildverarbeitung
- oder Brille mit Sensoren für Pupillenposition
- und Kopfmessung (z.B. Marker am Helm)

Medientechnik | WS 2001 | 18.204

Uni Colorado: Eingabestift

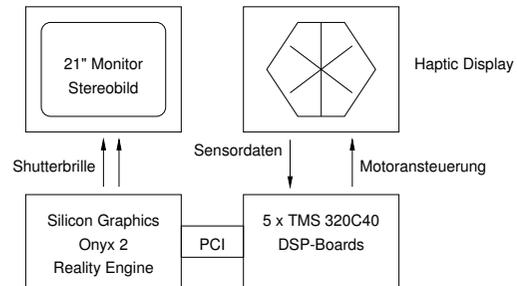


- Prototyp für 6 DOF-Eingabe mit Force-Feedback
- Stift mit 5 Aktuatoren, Sensoren mit 0.08N, Motoren bis 8.0N
- Silicon Graphics Onyx2, Shutterbrille, 5 DSPs zur Motorsteuerung

(<http://osl-www.colorado.edu/Research/haptic/hapticInterface.shtml>)

Medientechnik | WS 2001 | 18.204

Blockschaltbild



- Einlesen der aktuellen Sensordaten (Position, Druck, ...)
- Aktualisierung der virtuellen 3D-Welt
- Berechnen der Bewegungsgleichungen für die Aktuatoren
- Ansteuerung der Motoren

Medientechnik | WS 2001 | 18.204

Beispiel: Moleküldesign



Figure 1: The *Docker* application simulates the forces between a drug and its receptor site in a protein as the user guides it to the minimum-energy configuration.

(Taylor, Scientific Applications of Force Feedback, SIGGRAPH'99)

Medientechnik | WS 2001 | 18.204

CMU: Magnetic Levitation

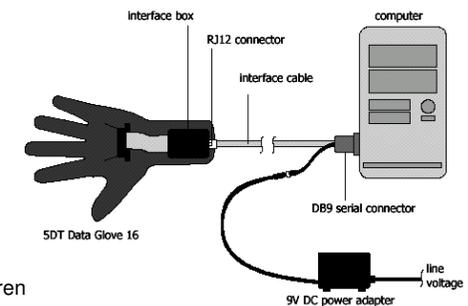


- direkte Ansteuerung des "Griffels"
- über rundum angeordnete Elektromagnete
- Aktuator ist vollkommen frei beweglich
- geringe Masse (und daher Trägheit) des Systems
- erlaubt sehr schnelle Bewegungen
- sehr komplexe Ansteuerung

(www-2.cs.cmu.edu/afs/cs/project/msl/www/haptic/haptic_desc.html)

Medientechnik | WS 2001 | 18.204

Datenhandschuh



"data glove": Handschuh mit Sensoren

- Messung der relativen Handposition (Neigungswinkel)
- Messung der Fingerposition (Dehnungsmeßstreifen / opt. Sensoren)
- A/D-Wandlung der Werte, Übertragung zum Rechner (Kabel / Funk)

=> Standard-Eingabegerät für VR-Anwendungen

=> direkte Manipulation virtueller Objekte

(Abb.: Five Dimension Technologies)

Medientechnik | WS 2001 | 18.204

Datenhandschuh: Freiheitsgrade

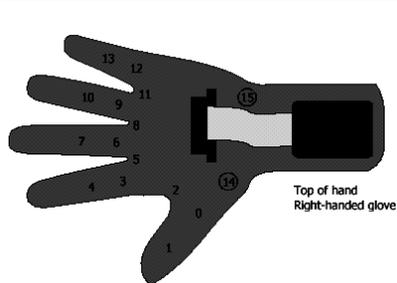


Figure 15 - Sensor mappings for the 5DT Data Glove 16

Sensor	Driver Index	Sensor	Description
0	0		Thumb flexure (lower joint)
1	1		Thumb flexure (second joint)
2	2		Thumb-index finger abduction
3	3		Index finger flexure (at knuckle)
4	4		Index finger flexure (second joint)
5	5		Index-middle finger abduction
6	6		Middle finger flexure (at knuckle)
7	7		Middle finger flexure (second joint)
8	8		Middle-ring finger abduction
9	9		Ring finger flexure (at knuckle)
10	10		Ring finger flexure (second joint)
11	11		Ring-little finger abduction
12	12		Little finger flexure (at knuckle)
13	13		Little finger flexure (second joint)
14	14		Thumb translation [not yet implemented]
15	15		Wrist flexure [not yet implemented]

Beispiel für die Anordnung der Sensoren (5DT Glove16):

- 2 Krümmungssensoren pro Finger
- 4 Sensoren für Fingerspreizung
- Daumen- und Handgelenkneigung geplant
- keine Drucksensoren an den Fingerspitzen (!)

Medientechnik | WS 2001 | 18.204

Datenhandschuh: Datenformat

The measured flexures of each of the sensors are the returned in this packet. The packet consists of 36 bytes and has the following structure:

```
header s1high s1low ... s16high s16low checksum trailer
```

The `header` is always two bytes long:

`0x3C` (an ASCII "<", decimal value 60)

`0x44` (an ASCII "D", decimal value 68)

After the header, two bytes are sent for each sensor. The high order byte is sent first, then the low order byte. The value of the sensor is therefore $(s_high) \times 256 + s_low$.

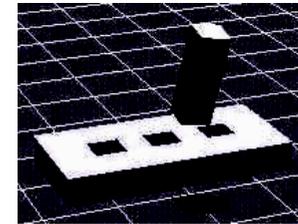
The `checksum` is the least significant byte of the addition of all the sensor values in the packet.

At the end of the packet, a trailing byte with the value of `0x3E` (an ASCII ">", decimal value 62) is sent.

- einfache RS-232 Schnittstelle, 115 kbps
- 100 Samples aller Sensoren pro Sekunde

Medientechnik | WS 2001 | 18.204

Datenhandschuh: VRML

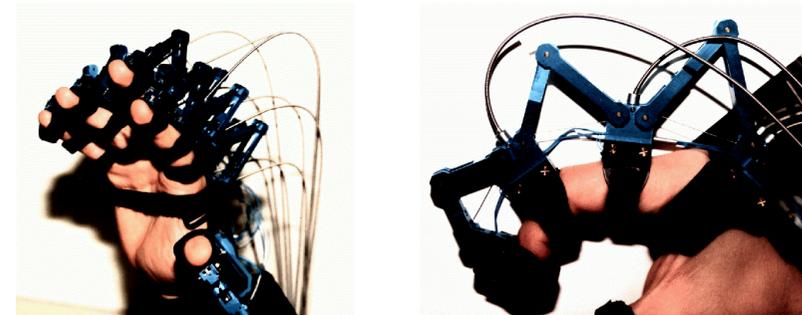


Anwendungsbeispiel:

- direkte Manipulation von Objekten im 3D-Editor
- direkter Zugriff auf verdeckte (unsichtbare) Objekte
- oder die Rückseite von Objekten
- Kombination mit force-feedback wäre ideal ...

Medientechnik | WS 2001 | 18.204

Datenhandschuh: Force-Feedback



- erfordert "Außenskelett" mit vielen Motoren
- komplexe Seilzugmechanik zur Gewichtsreduzierung
- für realistische Effekte beträchtliche Kräfte erforderlich (Beispiel: antippen an eine harte, virtuelle Wand)

(www.caip.rutgers.edu/~bouzit/rp/glove.html)

Medientechnik | WS 2001 | 18.204

Datenhandschuh: Force-Feedback



- Fernsteuerung des Roboterarms mit Kräfteückmeldung
- ideal für fein"fühliges" Arbeiten

hier die Motoren (!)

Datenhandschuh: CyberForce



- CyberGlove: Datenhandschuh, Positionsmessung der Finger
- CyberGrasp: Kräfteückmeldung für die Finger (Seilzüge, s.o.)
- CyberForce: Positionsmessung und Kräfteückmeldung für die Handposition, Arbeitsbereich ca. 50 cm²

=> volle Messung und Manipulation der Handbewegungen

- Demo (Video)

Datenhandschuh: Gestenerkennung



mit Datenhandschuh:

- Tastatureingabe problematisch
- aber neue Anwendungen möglich
- Beispiel: Gestenerkennung

- Leerseite

DirectX: Force-Feedback

DirectX:

- für effizienten Hardwarezugriff unter Windows
- Direct3D, DirectDraw, DirectShow, DirectSound, ...
- Zugriff auf Eingabegeräte via DirectInput
- seit DirectX5 Unterstützung von Force-Feedback Geräten
- also: Joysticks und Lenkräder
- DirectInput definiert Standardfunktionen und Aufrufe
- Applikation listet Geräte und deren Fähigkeiten auf
- Umsetzung der Effekte über gerätespezifische Treiber
- Auswahl und "Download" der Effekte bei Programmbeginn

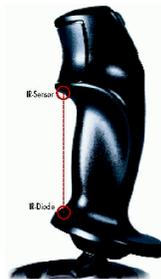
(Inside DirectX)

Medientechnik | WS 2001 | 18.204

DirectInput: Sicherheit . . .

Force-Feedback-Systeme:

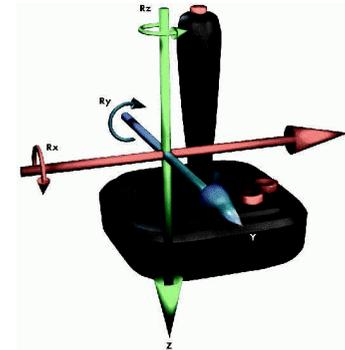
- können den Anwender verletzen
- oder zumindest irritieren
- Gefahr selbst bei "schwachen" Geräten
- Joysticks u.ä. bis ca. 10 N (1Kg)
- Schutzmaßnahmen erforderlich
- Beispiel: Sensor im Microsoft SideWinder Joystick



Medientechnik | WS 2001 | 18.204

DirectInput: Koordinatensystem

- starrer Körper hat 6 Freiheitsgrade
Position (x,y,z)
Lage (Rx,Ry,Rz)
- Zuordnung zu Mausposition
- bzw. Joystick-Achsen
- kartesisches / Polar-Koordinatensystem
- DirectX übernimmt die Umrechnungen
- Kraftwirkung wie "Windrichtung" angeben
- siehe DirectX Dokumentation und DDK-Beispiele



(c't 09/99 216)

Medientechnik | WS 2001 | 18.204

DirectX: Enumeration

```
LPDIRECTINPUTDEVICE8 g_lpDIDevice = NULL;
hr = g_lpDI->EnumDevices(DI8DEVTYPE_GAMECTRL,
                        DIEnumDevicesProc, // Callback function
                        NULL, DIEDFL_FORCEFEEDBACK | DIEDFL_ATTACHEDONLY);
if (FAILED(hr)) ... // No force-feedback joystick available...

BOOL CALLBACK DIEnumDevicesProc( LPCDIDEVICEINSTANCE lpddi, LPVOID pvRef )
{
    HRESULT hr;
    GUID DeviceGuid = lpddi->guidInstance;

    // Create game device, request exclusive access
    hr = g_lpDI->CreateDevice(DeviceGuid, &g_lpDIDevice, NULL);
    hr = g_lpDIDevice->SetCooperativeLevel(g_hwndMain, DISCL_EXCLUSIVE | ... );

    // Set game data format
    hr = g_lpDIDevice->SetDataFormat(&c_dfDIJoystick);
    DIPROPDWORD DIPropAutoCenter;
    DIPropAutoCenter.diph.dwSize = sizeof(DIPropAutoCenter);
    DIPropAutoCenter.diph.dwHeaderSize = sizeof(DIPROPHEADER);
    DIPropAutoCenter.diph.dwObj = 0;
    DIPropAutoCenter.diph.dwHow = DIPH_DEVICE;
    DIPropAutoCenter.dwData = DIPROPAUTOCENTER_OFF;

    hr = g_lpDIDevice->SetProperty(DIPROP_AUTOCENTER, &DIPropAutoCenter.diph);
    return DIENUM_STOP; // One is enough.
} // end DIEnumDevicesProc
```

(DirectX8.1 SDK)

Medientechnik | WS 2001 | 18.204

DirectX: Enumeration der Effekte

```

LPDIRECTINPUTDEVICE8 g_lpDIDevice = NULL;
g_lpDIDevice = ... // Created by CreateDevice

// Try to find a periodic effect on the joystick device
HRESULT hr;
GUID guidEffect;
BOOL EffectFound = FALSE; // global flag

hr = g_lpDIDevice->EnumEffects(
    (LPDIENUMEFFECTSCALLBACK) DIEnumEffectsProc,
    &guidEffect, DIEFT_PERIODIC);
..

if (FAILED(hr)) ... // Internal error

// Callback function
BOOL CALLBACK DIEnumEffectsProc(LPCDIEFFECTINFO pei, LPVOID pv)
{
    *((GUID *)pv) = pei->guid;
    EffectFound = TRUE;
    return DIENUM_STOP; // One effect is enough
}

```

Medientechnik | WS 2001 | 18.204

DirectX: Effekt erzeugen

```

DWORD dwAxes[2] = {DIJOFS_X, DIJOFS_Y};
LONG lDirection[2] = {0, 0};
DIPERIODIC diPeriodic; // type-specific parameters
DIENVELOPE diEnvelope; // envelope
DIEFFECT diEffect; // general parameters

diPeriodic.dwMagnitude = DI_FFNOINALMAX;
diPeriodic.lOffset = 0;
diPeriodic.dwPhase = 0;
diPeriodic.dwPeriod = (DWORD)(0.05 * DI_SECONDS);

diEnvelope.dwSize = sizeof(DIENVELOPE);
diEnvelope.dwAttackLevel = 0;
diEnvelope.dwAttackTime = (DWORD)(0.5 * DI_SECONDS);
diEnvelope.dwFadeLevel = 0;
diEnvelope.dwFadeTime = (DWORD)(1.0 * DI_SECONDS);

diEffect.dwSize = sizeof(DIEFFECT);
diEffect.dwFlags = DIEFF_POLAR | DIEFF_OBJECTOFFSETS;
diEffect.dwDuration = (DWORD)(2 * DI_SECONDS);

diEffect.dwSamplePeriod = 0; // = default
diEffect.dwGain = DI_FFNOINALMAX; // no scaling
diEffect.dwTriggerButton = DIJOFS_BUTTON0;
diEffect.dwTriggerRepeatInterval = 0;
...
LPDIRECTINPUTEFFECT g_lpdiEffect; // global effect object
hr = g_lpDIDevice->CreateEffect(
    guidEffect, // GUID from enumeration
    &diEffect, // where the data is
    &g_lpdiEffect, // where to put interface pointer
    NULL); // no aggregation

```

Medientechnik | WS 2001 | 18.204

DirectX: Effekt abspielen

```

...
diEffect.dwSize = sizeof(DIEFFECT);
diEffect.dwFlags = DIEFF_POLAR | DIEFF_OBJECTOFFSETS;
diEffect.dwDuration = (DWORD)(2 * DI_SECONDS);
...

LPDIRECTINPUTEFFECT g_lpdiEffect; // global effect object

hr = g_lpDIDevice->CreateEffect(
    guidEffect, // GUID from enumeration
    &diEffect, // where the data is
    &g_lpdiEffect, // where to put interface pointer
    NULL); // no aggregation

if (FAILED(hr)) ...

diEffect.dwTriggerButton = DIEB_NOTRIGGER;

// To make a chain saw that starts and keeps going, change the
// dwDuration member as follows:

diEffect.dwDuration = INFINITE;
diPeriodic.dwPeriod = (DWORD)(0.08 * DI_SECONDS);
hr = g_lpdiEffect->SetParameters(&diEffect, DIEP_TYPESPECIFICPARAMS);

// Next, start the effect.
g_lpdiEffect->Start(1, 0);

// The effect keeps running until you stop it.
g_lpdiEffect->Stop();

```

Medientechnik | WS 2001 | 18.204

Immersion: Force-Feedback Motivation

Even under excellent conditions for viewing and hearing, touch substantially improves user performance. Consider this:

How is it that a person can drink a cup of coffee while simultaneously reading a newspaper?

- Release grip on paper, casually adjusting other hand's grip to compensate for paper sag. Perhaps rest bottom of paper in lap (without looking)
- While still reading (perhaps slower), slowly feel around table for coffee mug, adjusting direction based on feel.
- When found, feel for handle and grip.
- Bring mug to lips, keeping mug level, rotating arm and wrist appropriately (by weight and grip pressure).
- Tilt mug back when mug is felt against lips.
- Return mug to table slowly, adjusting course if small collisions detected. Notice that you do not even need to look at the table, you can feel it through the mug!
- And you never lost your position on the page.

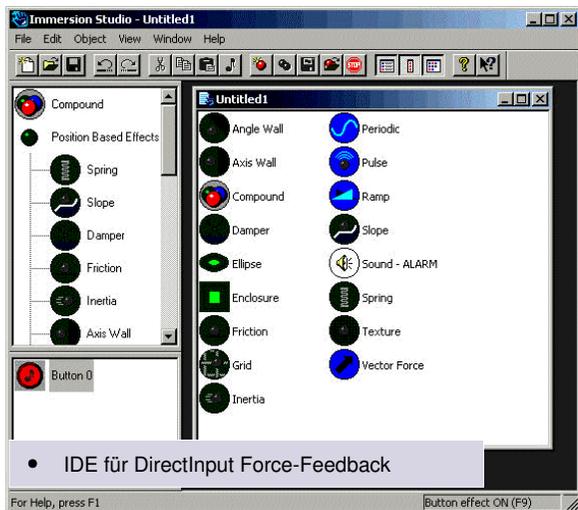
Now imagine without force feedback...

First, try to imagine holding the paper or turning the pages without feeling them!

(www.immersion.com)

Medientechnik | WS 2001 | 18.204

Immersion: iForce Studio



(www.immersion.com)

Medientechnik | WS 2001 | 18.204

Immersion: "Ellipse"

Ellipse Effect (Position Based Effect)

Ellipse can be used to attract the cursor to the inside of the ellipse, keep the cursor outside of the ellipse, or attract the cursor to the border surrounding the ellipse. Common uses of ellipse are for snapping to grid points or nodes, or attracting the cursor to icons. Enclosure effect is only available to TouchSense mouse devices only.



Medientechnik | WS 2001 | 18.204

Immersion: Geräte vs. Effekte

The nature of the device limits the types of touch effects felt. A steering wheel, for instance, turns left and right, so it plays effects on the x-axis only. A joystick, attached to its base at a single point, cannot play effects which require an exploration of 2-dimensional space, like textures and contours. A tactile mouse, free from a base attachment, is unable to exert lateral forces, making it unable play effects like pushes, pulls, gravity, or resistance to movement.

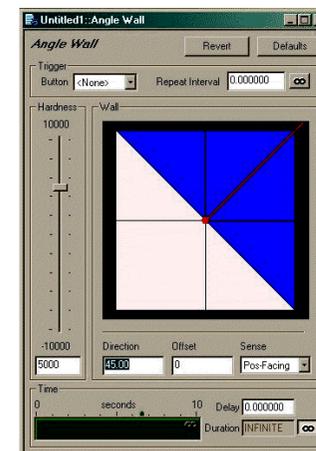
	Tactile Feedback Mouse or Trackball	Joystick	Steering Wheel	Full Force Feedback Gamepad	Full Force Feedback Mouse	Rumble Feedback Gamepad
Periodic	x	x	x	x	x	x
Texture	x				x	
Enclosure	x				x	
Ellipse	x				x	
Spring		x	x	x	x	
Grid					x	
Constant		x	x	x	x	
Ramp		x	x	x	x	
Damper		x	x	x	x	
Friction		x	x	x	x	
Inertia		x	x	x	x	

Medientechnik | WS 2001 | 18.204

Immersion: "Angle Wall"

Angle Wall Effect (Position Based Effect)

Angle Wall is a position-based effect. It should feel like a wall at an arbitrary angle. Based on [Spring](#), it uses the Immersion TouchSense IMM_CONDITION effect with a Spring GUID and the PositiveCoefficient or NegativeCoefficient set to zero.



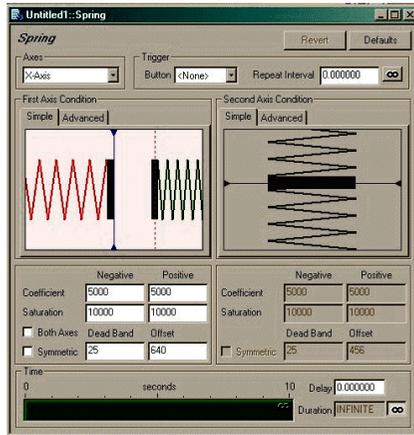
- basiert auf dem "Feder"effekt
- "hardness" = Federkonstante
- keine Wirkung im neg. Bereich
- Richtung frei wählbar

Medientechnik | WS 2001 | 18.204

Immersion: "Spring"

Spring Effect (Position Based Effect)

Spring is a position-based effect. It should feel like compressing a spring. It uses the Immersion TouchSense IMM_CONDITION effect with a Spring GUID.

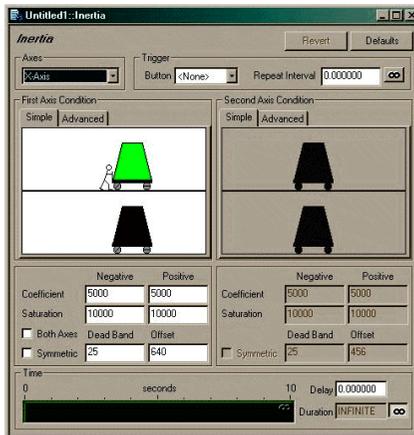


- modelliert eine Feder
- Rückstellkraft proportional zur Auslenkung
- für alle elastischen Medien
- optional mit Sättigungswert
- Richtung frei wählbar

Immersion: "inertia"

Inertia Effect (Position Based Effect)

Inertia is an acceleration-based effect. It should feel like pushing a mass on wheels. It uses the Immersion TouchSense IMM_CONDITION effect with an Inertia GUID.

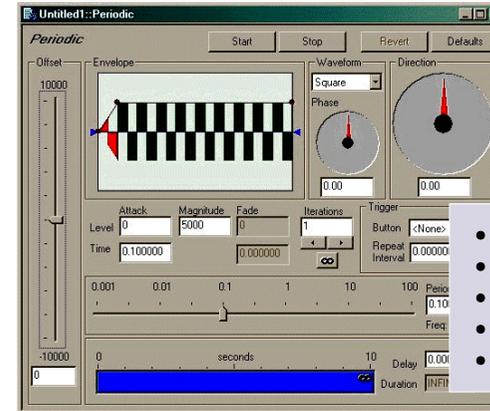


- Trägheitseffekt
- Objekt (Masse) auf Rädern...
- Masse des Objekts
- max. Geschwindigkeit
- Richtung frei wählbar

Immersion: "periodic"

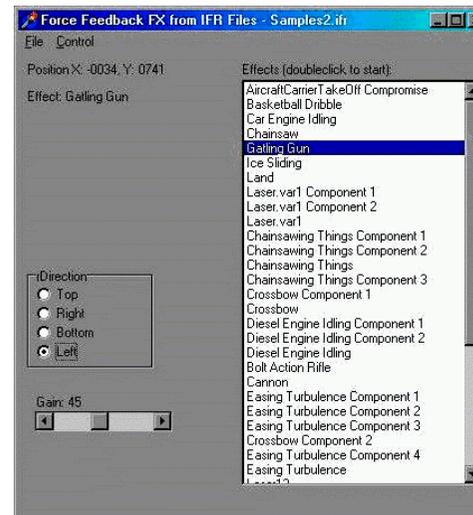
Periodic Effect (Time Based Effect)

The Periodic effect is a force that varies over time, such as a sine wave, square wave, etc. It can feel like a simple back and forth motion or a high frequency vibration. It uses the Immersion TouchSense IMM_PERIODIC effect.

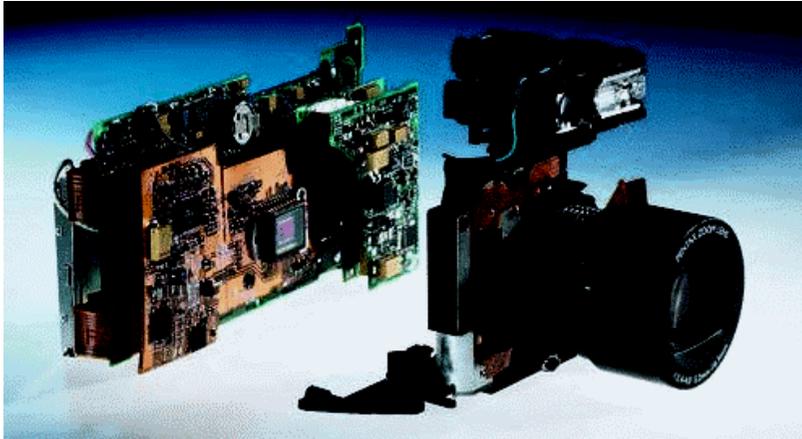


- Vibrationseffekt
- Sinus, Rechteck, usw.
- Frequenz, Amplitude,
- Hüllkurve (Start)
- Richtung frei wählbar

DirectX: Beispiel für Force-Effekte



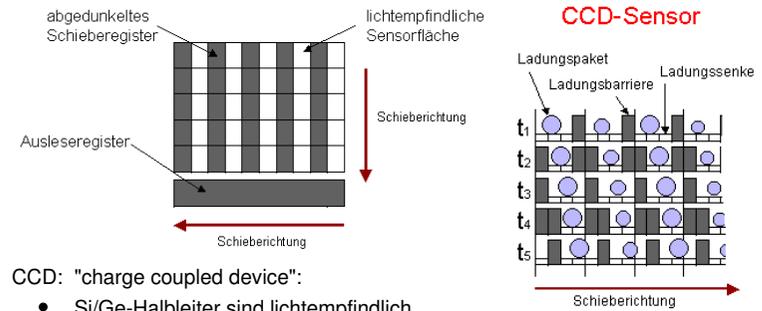
Digitalkamera



- Objektiv, Verschluss, CCD, (32-bit) Rechner

(HP 618 Digitalkamera, ct 04/2001, S.172)

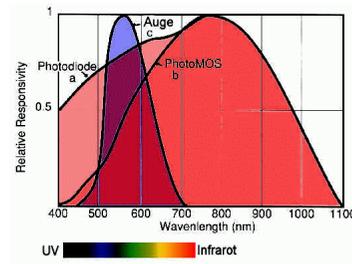
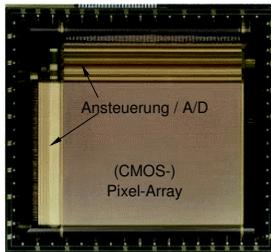
CCD: Prinzip



CCD: "charge coupled device":

- Si/Ge-Halbleiter sind lichtempfindlich
- Photonen erzeugen freie Ladungsträger
- spektrale Empfindlichkeit bis ins Infrarote
- einfache "Streifenelektroden" zur Lokalisierung der Ladungen
- Auslesen via 3/4-Phasen Ansteuerung, "Eimerketten"
- anschließende A/D-Wandlung, digitales Interface

Bildsensoren: CCD, CMOS



CCD: "charge coupled devices"

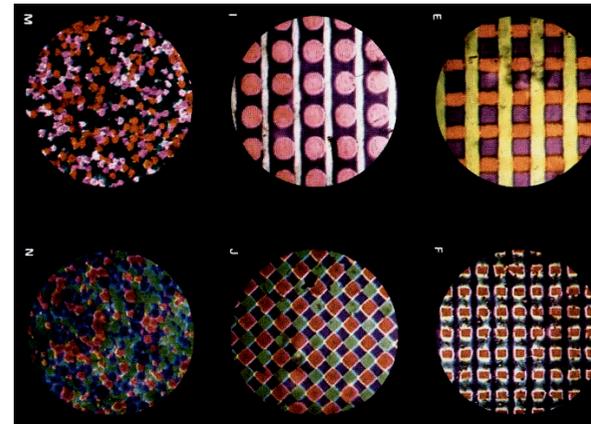
- gute Bildqualität, hohe Auflösung, aber externe Ansteuerung
- spezieller Herstellungsprozess, daher recht teuer

CMOS: auch normale Transistoren sind lichtempfindlich...

- Integration von Sensor-Array und Ansteuerung möglich
- bisher schlechtere Auflösung / Qualität als CCD

(Abb.: Fraunhofer/IMS CIF-Format CMOS-Sensor)

Photographie: Rasterfilme

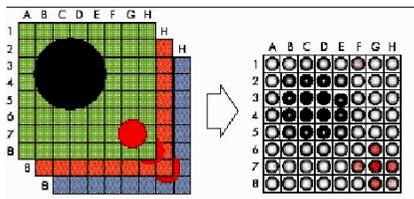


- E: Dufay 1909
- F: Omnicolore 1907
- I: Leto 1913
- J: Baker Duplex 1926
- M: Autochrome 1907
- N: Agfa Farbenplatte 1916

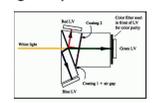
- Beispiele für Raster bei frühen (additiven) Farbfilmen
- aktuelle CCDs ähnlich wie J, aber grün am häufigsten

(Coe: Geschichte der Farbphotographie)

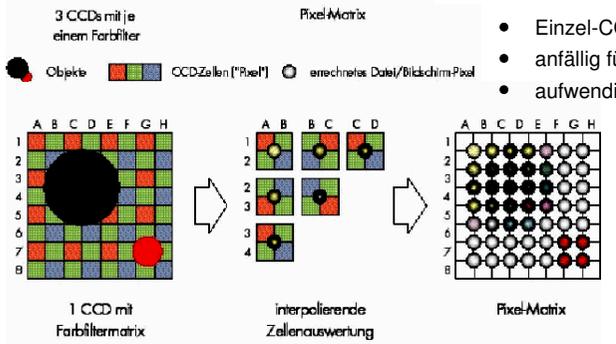
CCD: Bild-Rekonstruktion



- drei separate CCDs:
- Aufnahme direkt als RGB
- teuer, aufwendige Optik

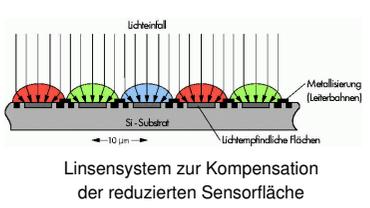
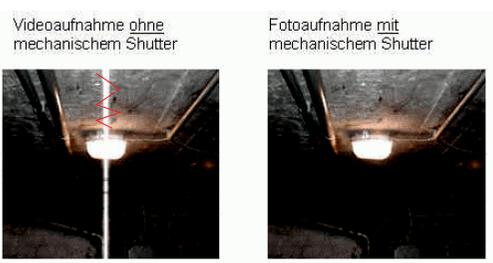


- Einzel-CCD, mit Farbraster
- anfällig für Moire-Muster
- aufwendige Bild-Rekonstruktion



(z.B. c'1 14/1998 S.82ff)

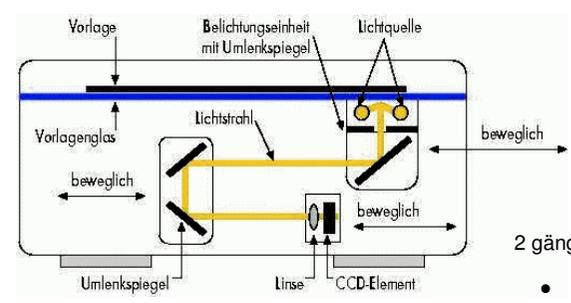
CCD: Blooming



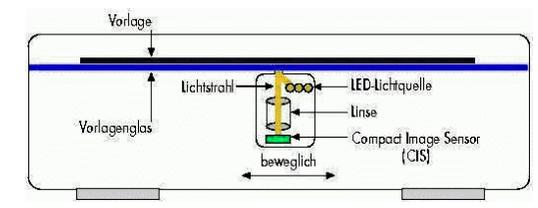
- Sensorelemente sind durchgängig lichtempfindlich
- auch beim Auslesen (Durchschieben) der Daten => mechanischer Verschluss => abgedeckte Flächen (plus Linsen) auf dem Chip
- außerdem: "Ausbluten" überbelichteter Sensoren in Nachbarpixel

(www.ccd-sensor.de / c'1 14/98 084)

Scanner: CCD vs. CIS

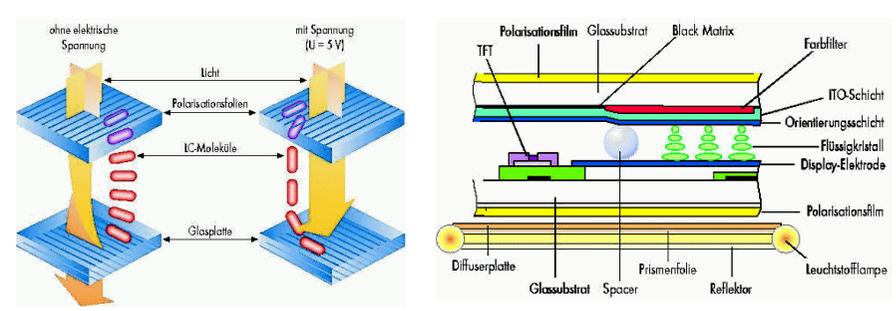


- 2 gängige Bauformen:
- CCD mit Umlenkspiegel
 - Compact Image Sensor



(c'1 24/2000 210ff)

LC-Displays:



- Flüssigkristalle zwischen zwei Polarisationsfiltern
- Matrixansteuerung (passiv oder TFT), Farbfilter
- Hintergrundbeleuchtung
- aufwendige Herstellung, geringer Yield: teuer
- geringe Effizienz

LCD: Display-Technologie vs. Auge

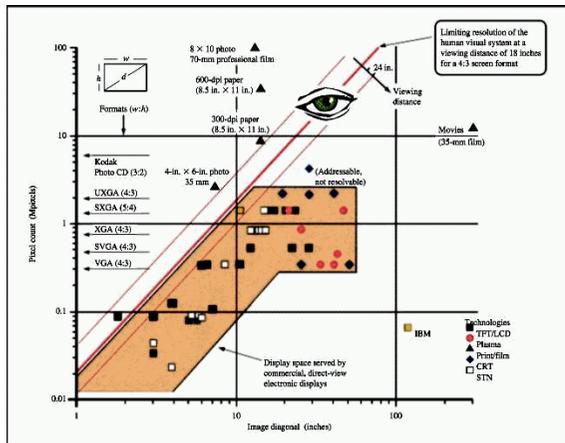


Figure 1
A view of display space showing existing electronic, paper, and film displays.

LCD: Evolution

- 1992: VGA (640x480)
- 1998: SXGA (1280x1024)
- monochrom bis true color
- 8" bis 15" Bilddiagonale
- Auflösung bis 200 dpi (IBM SXGA Prototyp)
- Qualität wie Laserdrucker / Zeitschriften-Farbdruck
- weitere Steigerung nötig?

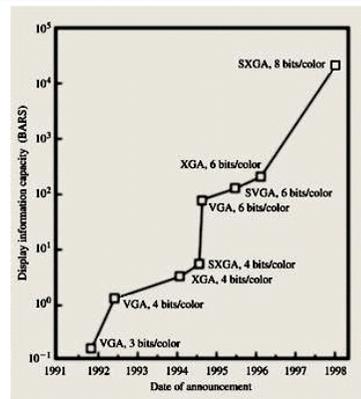
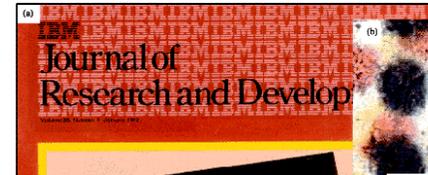
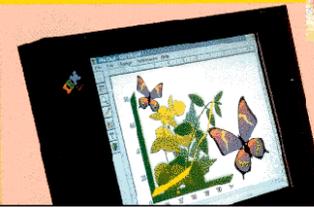


Figure 2
Information capacity increase of IBM TFT/LC displays with time. Information capacity is in units of pixel count times number of colors (billions of addressable retinal stimuli, or BARS).

LCD: 200dpi Prototyp



gedrucktes Cover des JR&D



200dpi LCD Prototyp



- vergleichbare Auflösung
- besserer Kontrast
- LCD subjektiv besser

LCD: 200dpi Prototyp

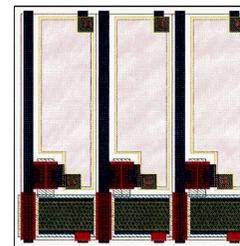


Figure 2
CAD layout of three color subpixels of dimensions $58 \mu\text{m} \times 154 \mu\text{m}$, arranged in a vertically striped mode.

Prototyp am Limit der Technologie:

- Pixelgröße vs. Kapazität vs. Multiplexing
- Kontrastverhältnis (aktive/passive Fläche)
- keine Redundanz möglich (vgl. DRAM)
- 3 x 1.2M Pixel: Ausbeute problematisch

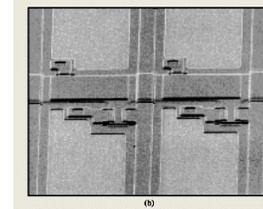
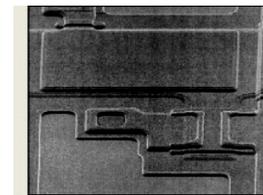


Figure 6
Oblique-angle scanning electron micrographs of the array depicted in Figure 5. The subpixels are $54 \mu\text{m}$ wide.

LCD: Kontakte . . .

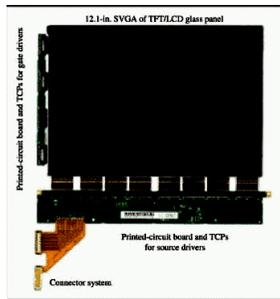


Figure 1
Part of a 12.1-in. SVGA TFT color LCD module for the IBM ThinkPad 560. Eight source-driver tape-carrier packages (TCPs), which provide a bending structure, and four gate-driver TCPs are attached by means of an anisotropic conductive film to the LCD glass panel. Printed-circuit boards are soldered to the input electrodes on the TCPs.

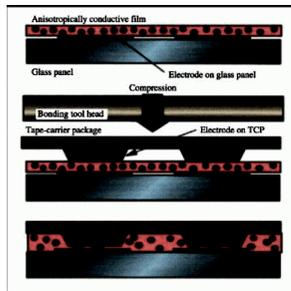


Figure 3
Bonding process for TCPs and LCD glass panels. An ACF is attached to the glass panel and a TCP is pre-attached, with accurate alignment. Final curing is then done. Adhesive fills the space between the electrodes, bonding the TCP and glass panel. Particles between the corresponding electrodes create an electrical contact, and insulation between adjacent electrodes is maintained.

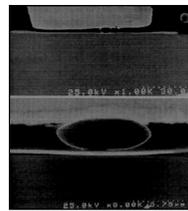
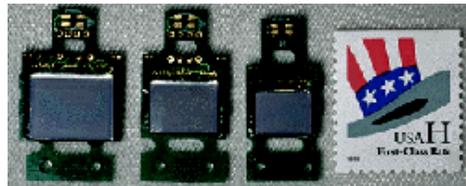
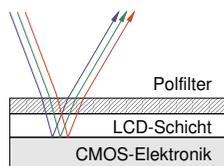


Figure 4
Cross-sectional view of interconnection by a 2-µm-diameter electrically conductive particle for the TCP example. The particle is deformed between a copper lead electrode on the TCP and an aluminum/aluminum-oxide electrode on the glass panel.

[IBM JR&D 1998]

- Ansteuerung per Multiplexing, trotzdem (1280+1024) Anschlüsse
- großflächig, Glas: normales Chip-Bonding unmöglich

LCD: reflektiv, MikroOptical



- bisherige LCDs auf Glassubstrat
- erlaubt sehr große Displays (15", 17" usw.)
- aufwendige Dünnschicht-Technologie

reflektive Displays:

- LCD-Schicht direkt auf entsprechend entworfener CMOS-Logik
- erlaubt sehr kleine Pixel und Miniaturisierung
- Farbdarstellung durch Zeitmultiplex, nacheinander rot-grün-blau

(www.microoptical.com)

LCD: IBM Lightvalve

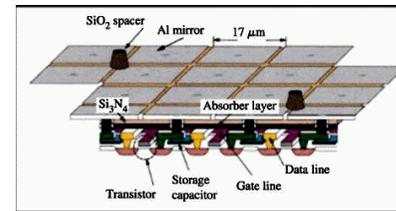


Figure 1

Schematic diagram of a spatial light-modulator array showing the cross section of the light valve used in the prototype monitor. To complete the light valve, a top glass plate with a transparent conductive layer is placed on top of the posts and glued to the silicon substrate along the edges. Finally, the gap between the mirrors and glass is filled with liquid crystal material.

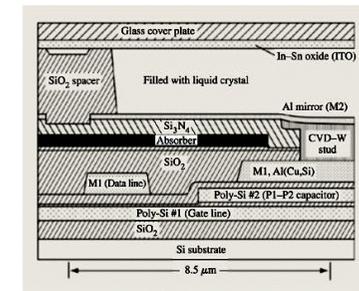


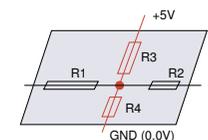
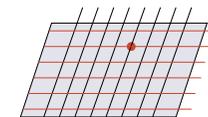
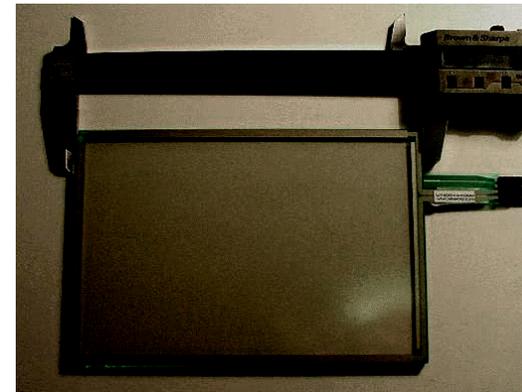
Figure 1

Schematic cross section of a pixel of the four-million-pixel light valve.

- IBM-Prototyp für ein reflektives Display
- oberste Metallschicht bildet den Spiegel, direkte CMOS-Ansteuerung
- normale CMOS-Fertigung nutzbar, 4 MegaPixel

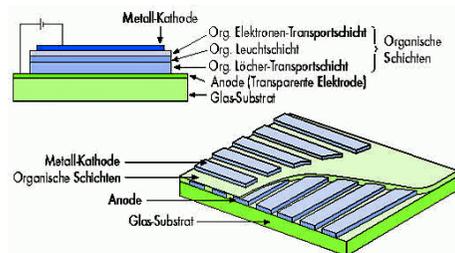
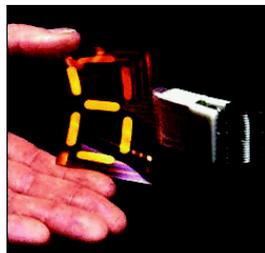
(IBM JRD 42-3)

Touchscreen:



- kapazitiv: orthogonale Streifen-Elektroden, serielles Auslesen
- resistiv: Stiftbedienung, Kontaktposition wird gemessen (billiger, aber geringere Genauigkeit)

organische Displays:



- spezielle Farbstoffe
- Lichtemission bei angelegter Spannung
- im Prinzip beliebige Farbe und Helligkeit
- flexibel (Folie statt Glas als Träger)
- kein Lichtverlust durch Filter
- billige Herstellung: Farbstoffe per "Tintendrucker" aufbringen
- aber noch zu geringe Lebensdauer

organische Farbstoffe:

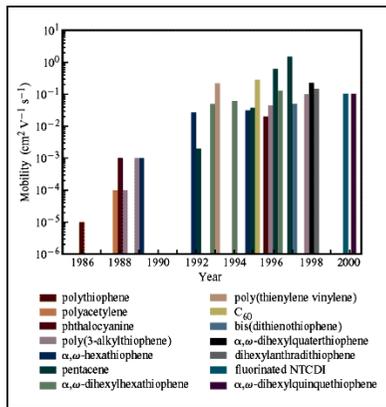
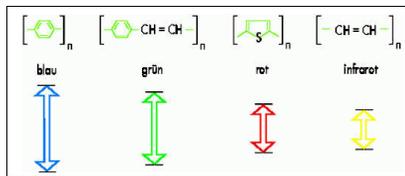
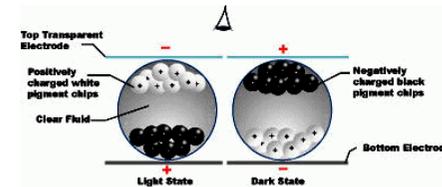
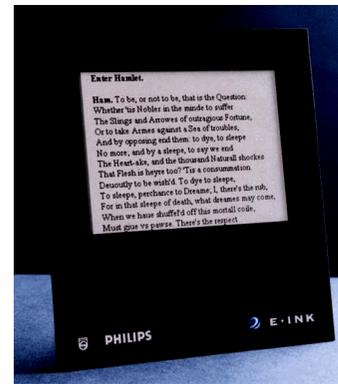


Figure 1
Semilogarithmic plot of the highest field-effect mobilities (μ) reported for OFETs fabricated from the most promising polymeric and oligomeric semiconductors versus year from 1986 to 2000.



- Farbe
- Leuchtdichte
- Entdeckungsjahr
- von Farbstoffen für org. Polymerdisplays

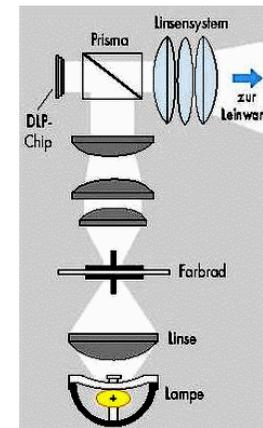
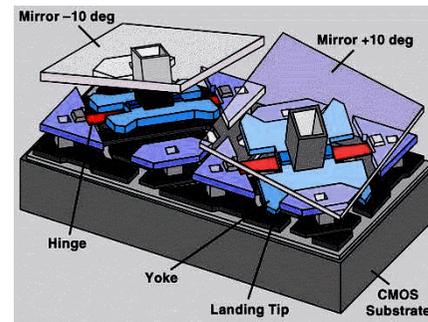
eInk:



(Philips/eInk Prototyp, 80 dpi, Juni 2001)

- schwarz/weiß gefüllte/gefärbte Kugeln auf Trägermaterial
- Ansteuerung wie LCD (passiv oder TFT)
- aber metastabil: daher stromsparend

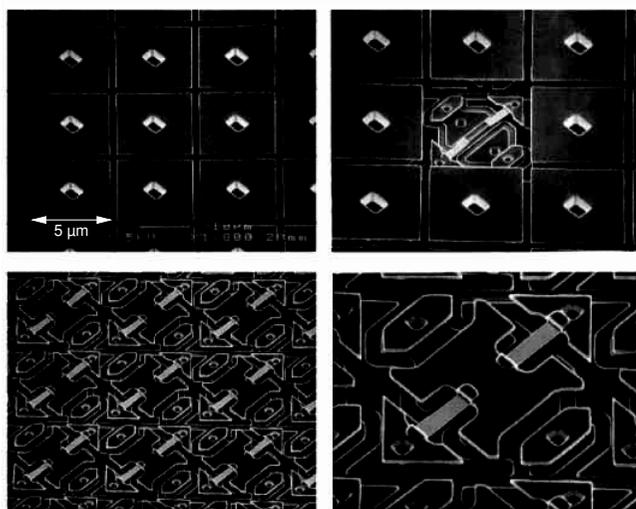
Mikrosysteme: DLP



"Digital Light Processing":

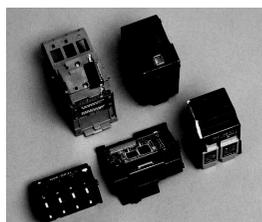
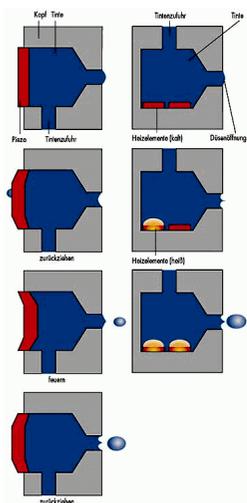
- Mikrospiegel über SRAM, z.B. 1024x768
- Lichtrichtung umschaltbar
- sehr schnell: Zeitmultiplex für Farbe&Helligkeit
- heller und kleiner als LCD-Projektoren

Mikrosysteme: DLP



PC-Technologie | SS 2001 | 18.214

Tintenstrahldrucker

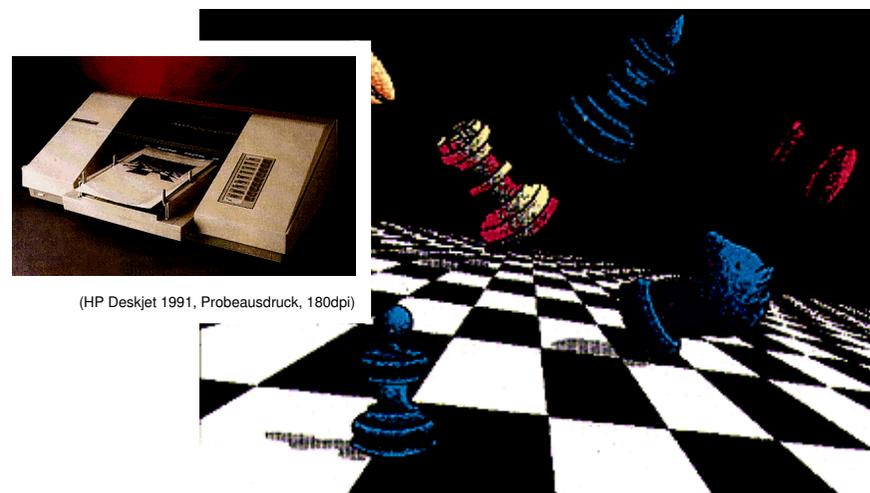


(Druckköpfe Epson, HP, Lexmark, Canon)

- Mechanik mit Schlitten für Druckköpfe
- Aufspritzen von Tintentröpfchen auf das Papier
- Piezokeramik (links, Epson)
- thermisch (rechts, z.B. HP, Canon)
- subtraktive CMYK-Farbmischung
- Rasterung zur Darstellung von Mischfarben

(Abb. aus c't 02/1991 S.94, 07/2001 S.136)

HP Deskjet, 1991 ...

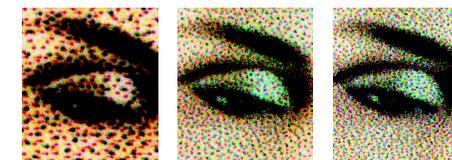
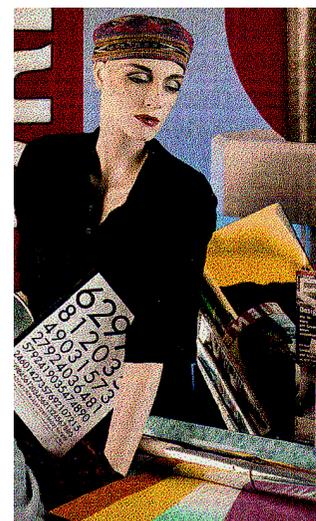


(HP Deskjet 1991, Probeausdruck, 180dpi)

(Abb. aus c't 02/1991 S.94)

Medientechnik | WS 2001 | 18.204

... Rasterung 1997 ...



HP 670c
(4 Farben)

Epson Stylus 800
(4 Farben)

Epson Stylus Photo
(6 Farben)

- im weiter verbesserte Rasterung
- und daher auch Farbwiedergabe
- stark vom Treiber abhängig
- Spezialdrucker für Fotoqualität (z.B. 6 Tinten CMYK, helles CY)

(Abb. aus c't 07/97 S.234 und 09/97 S.84)

Rechnerarchitektur

- welche Rechnerarchitektur für "Medienverarbeitung"?
- Anforderungen für Audio, Video, 2D/3D, VR, Speech, ...
- Anforderungen an Server und Netzwerke
- Bewertung und Leistungsanalyse von Rechnern
- Befehlssätze, RISC vs. CISC
- Einfluss der Speicherhierarchie
- ARM-Architektur
- ARM-Erweiterungen: Thumb, Jazelle, XScale
- Single-Chip Multiprozessoren: MAJC, Piranha
- Netzwerkprozessoren
- SIMD-Befehlssätze: MMX, SSE

Medientechnik | WS 2001 | 18.204

Rechnerarchitektur: Literatur

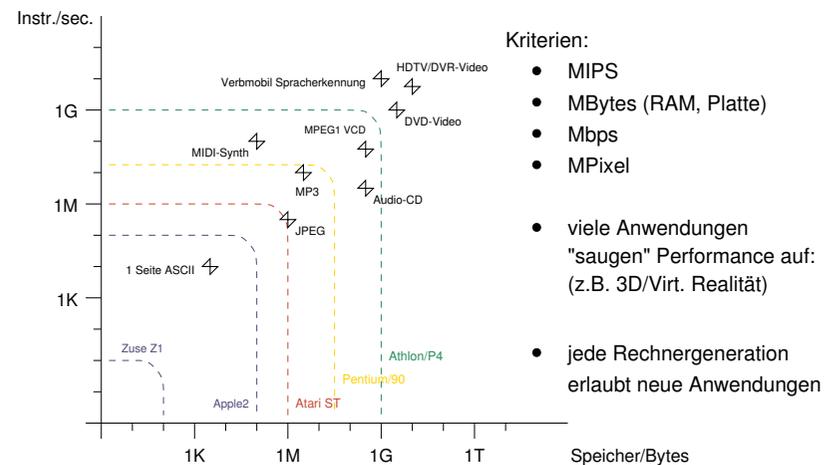
D.L.Hennessy & D.A.Patterson, Computer architecture, a quantitative approach, MKP
 A.S.Tanenbaum, Computer architecture, 4th ed.,
 S.Furber, ARM system-on-chip architecture, Addison-Wesley
 M.D.Hill (Ed), Readings in Computer Architecture, MKP 2000

Intel x86, MMX, SSE, SSE2 documentation, www.developer.intel.com
 Compaq Piranha, Proc. ISCA 2000
 Sun MAJC processor documentation and whitepapers, www.sun.com
 Netzwerkprozessoren: Proc. HotChips 12, 2000, www.hotchips.org

Benchmarks: www.spec.org

Medientechnik | WS 2001 | 18.204

Medien: Anforderungen



Medientechnik | WS 2001 | 18.204

Medienverarbeitung: Speicherzugriffe

- Mediendaten liegen als 1D-/2D-Arrays vor (Audio, Video, Images, ..)
 - große Datenmengen / hohe Datenraten (HDTV: 100 MB/s)
 - geringe Genauigkeit der einzelnen Datenworte
 - geringe "Lokalität" (Samples/Pixel/Frames unabhängig)
 - klassischer Cache wirkungslos
 - "Streaming" mit kleinen Puffern
 - aber auch "klassische" Algorithmen und Datenstrukturen
 - z.B für LZW/BWT/Huffman-Datenkompression
 - z.B Hashtabellen/Wörterbücher für Spracherkennung (Verbomobil: 1 GB)
 - und natürlich Floating-Point für Simulationen (z.B. Objektphysik in 3D-Spielen, VR, haptische Interfaces)
- => "Medien"-Architekturen müssen alles beherrschen

Medientechnik | WS 2001 | 18.204

Medienverarbeitung: I/O

stark unterschiedliche Anforderungen, je nach Medium:

Audio, Wiedergabe (48 KHz, 16bit, 6 Kanäle)	0.6 MB / s
Audio, Studio-Mischpult (100 Kanäle)	9.6 MB / s
Video, 640x480x24, 25fps	23 MB / s
Video, 1280x960x24, 25fps	92 MB / s
3D, Transformation, 100K Dreiecke, 50 fps, je 32-bit	60 MB / s
3D, 1024x768x24, 50fps, trilinear (8/pixel), 3x overdraw:	8490 MB / s
Datenhandschuh, 20 Mestreifen, 8 bit, 25fps	0.0005 MB / s
Force-Feedback, 30 Motoren, 25 fps	

- zum Vergleich: PCI 132 MB/ps, AGP4X 1GB/s
- mehrfache Anforderungen fr Medien-Server

Medientechnik | WS 2001 | 18.204

Medienverarbeitung: Server

- Auslieferung statischer Dateien:
- erfordert wenig CPU-Performance
- aber hohen I/O-Durchsatz
- einzelne Client-Anfragen voneinander unabhängig
- im Prinzip vollständig parallele Bearbeitung möglich
- Multiprozessor / SMP-Systeme
- aber: Synchronisation, z.B. wegen Logfiles
- Beispiel "video on demand":
 - 1 Stream MPEG-2 mit 1 MB/s
 - 100 Streams mit je 1 MB/s: 100 MB/s
 - usw...
- Beispiel: Compaq Piranha 8xAlpha SOC

Medientechnik | WS 2001 | 18.204

Medienverarbeitung: Netzwerk

- Telekommunikation: steigender Anteil Daten vs. Voice
- "packet" vs. "switched" Übertragungen

- exponentiell steigende Datenmengen
- entsprechend komplexeres Routing (erst Recht für ATM)
- neue Anforderungen, z.B. QoS
- potentiell großer Markt

=> neue Geräteklasse "Netzwerkprozessoren"

- effiziente Bearbeitung von Routing-Anfragen
- zugehörige Pufferung / Bearbeitung der Datenpakete
- erweiterte Zugriffs- und Rechtekontrolle (in Echtzeit)
- interessante, parallele Rechnerarchitekturen

(siehe HotChips12 Session)

Medientechnik | WS 2001 | 18.204

Medienverarbeitung: Mobilgeräte

- zunehmende Verbreitung von "mobile appliances"
- z.B. Mobiltelefone, PDAs, MP3-Player, ...
- Multimedia-Anwendungen erwünscht (MP3, Video, UMTS, ...)
- erforderliche Rechenleistung wie aktuelle Desktop-Systeme
- 200+ MIPS, 64+ MByte RAM, Hintergrundspeicher
- aber möglichst geringer Stromverbrauch
- spezielle Prozessoren mit zugehörigen Hilfschips
- meistens: RISC mit DSP-Erweiterungen
- Powermanagement, low-voltage, gated-clocks
- spezielle CMOS-Technologie (derzeit 0.13 .. 0.18µm)
- Kriterium: mW / MIPS

Medientechnik | WS 2001 | 18.204

Amdahl's Gesetz

"Speedup" durch Parallelisierung? [Gene Amdahl, 1967]

System 1: berechnet Funktion X, zeitlicher Anteil $0 < F < 1$
 System 2: Funktion X' ist schneller als X mit "speedup" SX:
 $SX = \text{Zeitbedarf}(X) / \text{Zeitbedarf}(X')$

Amdahl's Gesetz: $S_{\text{gesamt}} = \frac{1}{(1-F) + F/SX}$

=> Optimierung lohnt nur für häufige Operationen !!

=> Beispiele:

$SX = 10, F = 0.1, S_{\text{gesamt}} = 1 / (0.9 + 0.01) = 1.09$
 $SX = 2, F = 0.5, S_{\text{gesamt}} = 1 / (0.5 + 0.25) = 1.33$
 $SX = 2, F = 0.9, S_{\text{gesamt}} = 1 / (0.1 + 0.45) = 1.82$
 $SX = 1.1, F = 0.98, S_{\text{gesamt}} = 1 / (0.02 + 0.89) = 1.10$

RISC: Designphilosophie

- minimaler, regulärer Befehlssatz
- optimale VLSI-Implementierung
- Compiler erledigt den Rest
- Berücksichtigung von Amdahl's Gesetz
- umfangreiche Performance-Simulationen (Benchmarks)

ursprüngliche RISC Entwurfsentscheidungen:

- + 32-bit Prozessor, 4 GByte Adressraum
- + 32 Universalregister (ausser RISC/SPARC)
- + 32-bit Befehlswoorte, wenig Formate
- Pipeline-Abhängigkeiten (delayed branches)
- Spezialregister (MIPS mult/div)

RISC: RISC-I und Mips

ca 1980: 801-Nachfolgeprojekte:

- Berkeley RISC-I "reduced instruction set computer"
- Stanford MIPS "microprocessor w/o interlocked pipeline stages"
- Compiler-gerechte Architektur
- single-Chip VLSI-Implementierung

bessere Performance als 8086/68000:

- sauberer Befehlssatz RISC
- "hardwired" Controller statt Microcode
- Pipeline
- viele Register, weniger Speicherzugriffe auch für CISC möglich!
- gut optimierende Compiler
- Caches, insbesondere I-Cache

SPECint: DLX

Instruction	compress	eqntott	espresso	gcc (cc1)	li	Integer average
load	19.8%	30.6%	20.9%	22.8%	31.3%	26%
store	5.6%	0.6%	5.1%	14.3%	16.7%	9%
add	14.4%	8.5%	23.8%	14.6%	11.1%	14%
sub	1.8%	0.3%		0.5%		0%
mul				0.1%		0%
div						0%
compare	15.4%	26.5%	8.3%	12.4%	5.4%	13%
load imm	8.1%	1.5%	1.3%	6.8%	2.4%	3%
cond. branch	17.4%	24.0%	15.0%	11.5%	14.6%	16%
uncond branch	1.5%	0.9%	0.5%	1.3%	1.8%	1%
call	0.1%	0.5%	0.4%	1.1%	3.1%	1%
return, jmp ind	0.1%	0.5%	0.5%	1.5%	3.5%	1%
shift	6.5%	0.3%	7.0%	6.2%	0.7%	4%
and	2.1%	0.1%	9.4%	1.6%	2.1%	3%
or	6.0%	5.5%	4.8%	4.2%	6.2%	5%
other (xor, not)	1.0%		2.0%	0.5%	0.1%	1%
load FP						0%
store FP						0%
add FP						0%
sub FP						0%
mul FP						0%
div FP						0%
compare FP						0%
mov reg-reg FP						0%
other FP						0%

"load/store architecture"
keine mul/div Befehle (!)

FIGURE 2.26 DLX instruction mix for five SPECint92 programs. Note that integer register-register move instructions are included in the add instruction. Blank entries have the value 0.0%.

SPECint: x86

Instruction	compress	eqntott	espresso	gcc (cc1)	li	Int. average
load	20.8%	18.5%	21.9%	24.9%	23.3%	22%
store	13.8%	3.2%	8.3%	16.6%	18.7%	12%
add	10.3%	8.8%	8.15%	7.6%	6.1%	8%
sub	7.0%	10.6%	3.5%	2.9%	3.6%	5%
mul				0.1%		0%
div						0%
compare	8.2%	27.7%	15.3%	13.5%	7.7%	16%
mov reg-reg	7.9%	0.6%	5.0%	4.2%	7.8%	4%
load imm	0.5%	0.2%	0.6%	0.4%		0%
cond. branch	15.5%	28.6%	18.9%	17.4%	15.4%	20%
uncond. branch	1.2%	0.2%	0.9%	2.2%	2.2%	1%
call	0.5%	0.4%	0.7%	1.5%	3.2%	1%
return, jmp indirect	0.5%	0.4%	0.7%	1.5%	3.2%	1%
shift	3.8%		2.5%	1.7%		1%
and	8.4%	1.0%	8.7%	4.5%	8.4%	6%
or	0.6%		2.7%	0.4%	0.4%	1%
other (xor, not,...)	0.9%		2.2%	0.1%		1%
load FP						0%
store FP						0%
add FP						0%
sub FP						0%
mul FP						0%
div FP						0%
compare FP						0%
mov reg-reg FP						0%
other (abs, sqrt,...)						0%

vgl. DLX: 26% / 9%

FIGURE D.15 80x86 Instruction mix for five SPECint92 programs.

(H&P)

SPECfp: DLX

Instruction	doduc	ear	hydro2d	mdljdp2	su2cor	FP average
load	1.4%	0.2%	0.1%	1.1%	3.6%	1%
store	1.3%	0.1%		0.1%	1.3%	1%
add	13.6%	13.6%	10.9%	4.7%	9.7%	11%
sub	0.3%		0.2%		0.7%	0%
mul						0%
div						0%
compare	3.2%	3.1%	1.2%	0.3%	1.3%	2%
load imm	2.2%		0.2%	2.2%	0.9%	1%
cond. branch	8.0%	10.1%	11.7%	9.3%	2.6%	8%
uncond branch	0.9%	0.4%		0.4%	0.1%	0%
call	0.5%	1.9%			0.3%	1%
return, jmp ind	0.6%	1.9%			0.3%	1%
shift	2.0%	0.2%	2.4%	1.3%	2.3%	2%
and	0.4%	0.1%			0.3%	0%
or		0.2%	0.1%	0.1%	0.1%	0%
other (xor, not)						0%
load FP	23.3%	19.8%	24.1%	25.9%	21.6%	23%
store FP	5.7%	11.4%	9.9%	10.0%	9.8%	9%
add FP	8.8%	7.3%	3.6%	8.5%	12.4%	8%
sub FP	3.8%	3.2%	7.9%	10.4%	5.9%	6%
mul FP	12.0%	9.6%	9.4%	13.9%	21.6%	13%
div FP	2.3%		1.6%	0.9%	0.7%	1%
compare FP	4.2%	6.4%	10.4%	9.3%	0.8%	6%
mov reg-reg FP	2.1%	1.8%	5.2%	0.9%	1.9%	2%
other FP	2.4%	8.4%	0.2%	1.2%	2%	2%

Adressarithmetik

load / store FP

FP-Arithmetik

FIGURE 2.27 DLX Instruction mix for five programs from SPECfp92. Note that integer register-register move instructions are included in the add instruction. Blank entries have the value 0.0%.

(H&P)

SPEC: x86 vs. DLX

	compress	eqntott	espresso	gcc (cc1)	li	Int. avg.
Instructions executed on 80x86 (millions)	2226	1203	2216	3770	5020	
Instructions executed ratio to DLX	0.61	1.74	0.85	0.96	0.98	1.03
Data reads on 80x86 (millions)	589	229	622	1079	1459	
Data writes on 80x86 (millions)	311	39	191	661	981	
Data read/modify/writes on 80x86 (millions)	26	1	129	48	48	
Total data reads on 80x86 (millions)	615	230	751	1127	1507	
Data read ratio to DLX	0.85	1.09	1.38	1.25	0.94	1.10
Total data writes on 80x86 (millions)	338	40	319	709	1029	
Data write ratio to DLX	1.67	9.26	2.39	1.25	1.20	3.15
Total data accesses on 80x86 (millions)	953	269	1070	1836	2536	
Data access ratio to DLX	1.03	1.25	1.58	1.25	1.03	1.23

x86 (CISC): 3% mehr Befehle als RISC...

x86 braucht mehr Lese-Befehle

und 3X mehr Store-Befehle

FIGURE D.17 Instructions executed and data accesses on 80x86 and ratios compared to DLX for five SPECint92 programs.

	doduc	ear	hydro2d	mdljdp2	su2cor	FP average
Instructions executed on 80x86 (millions)	1223	15220	13342	6197	6197	
Instructions executed ratio to DLX	1.19	1.19	2.53	2.09	1.62	1.73
Data reads on 80x86 (millions)	515	6007	5501	3696	3643	
Data writes on 80x86 (millions)	260	2205	2085	892	892	
Data read/modify/writes on 80x86 (millions)	1	0	189	124	124	
Total data reads on 80x86 (millions)	517	6007	5690	3820	3767	
Data read ratio to DLX	2.04	2.36	4.48	4.77	3.91	3.51
Total data writes on 80x86 (millions)	261	2205	2274	1015	1015	
Data write ratio to DLX	3.68	33.25	38.74	16.74	9.35	20.35
Total data accesses on 80x86 (millions)	778	8212	7965	4835	4782	
Data access ratio to DLX	2.40	2.05	4.44	4.40	3.44	3.35

20X mehr FP-Stores wegen mangelnder Register (!)

FIGURE D.18 Instructions executed and data accesses for five SPECfp92 programs on 80x86 and ratio to DLX.

(H&P)

SIMD: Flynn-Klassifikation

SISD "single instruction, single data"

=> jeder klassische PC

SIMD "single instruction, multiple data"

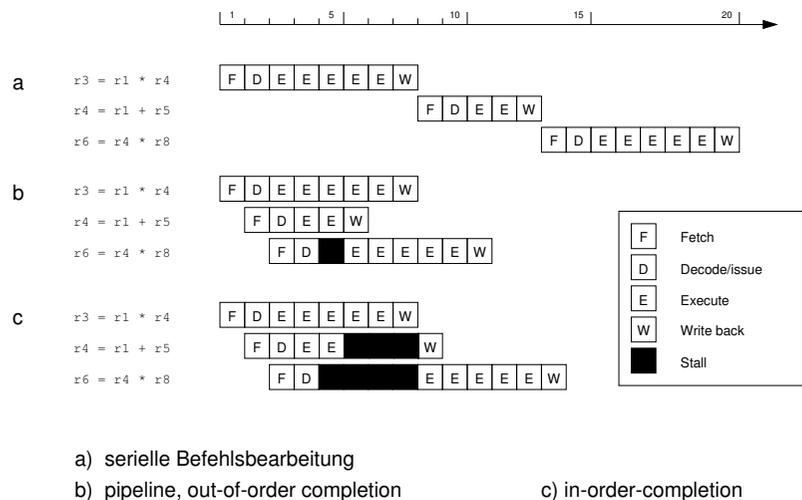
=> Feldrechner/Parallelrechner
=> z.B. Connection-Machine 2: 64K Prozessoren
=> eingeschränkt: MMX&Co: 2-8 fach parallel

MIMD "multiple instruction, multiple data"

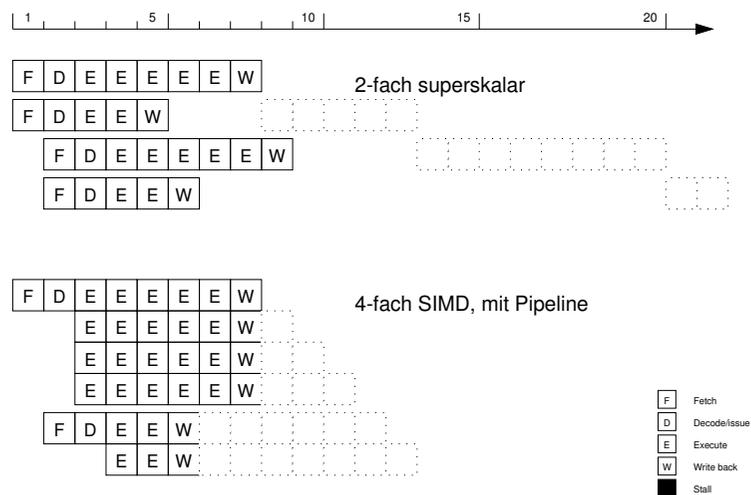
=> Multiprozessormaschinen
=> z.B. vierfach PentiumPro-Server

MISD :-)

Befehlspipeline: in order / out of order



Superskalar, SIMD



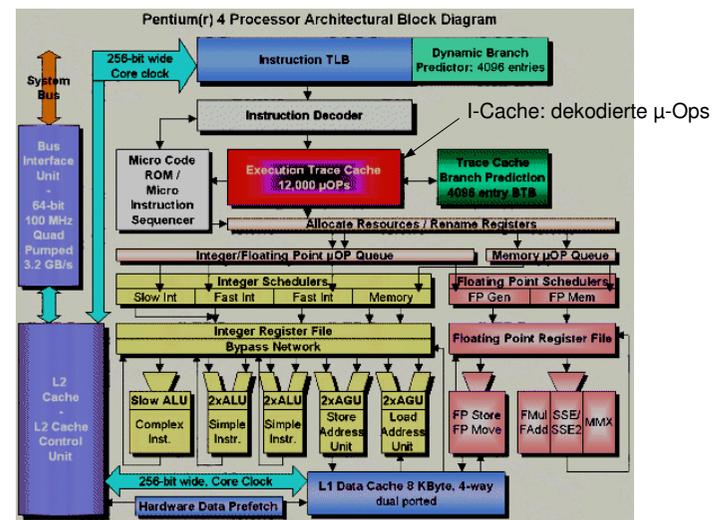
superskalare Prozessoren

RISC vs. CISC für superskalare Prozessoren:	RISC	CISC
komplexe Befehlsdekodierung		•
mehrfache Funktionseinheiten	•	•
komplexes Steuerwerk (Scoreboard etc.)	•	•
out-of-order execution	•	•
große on-chip Caches	•	•
Speicherzugriffe sind das Nadelöhr	•	•

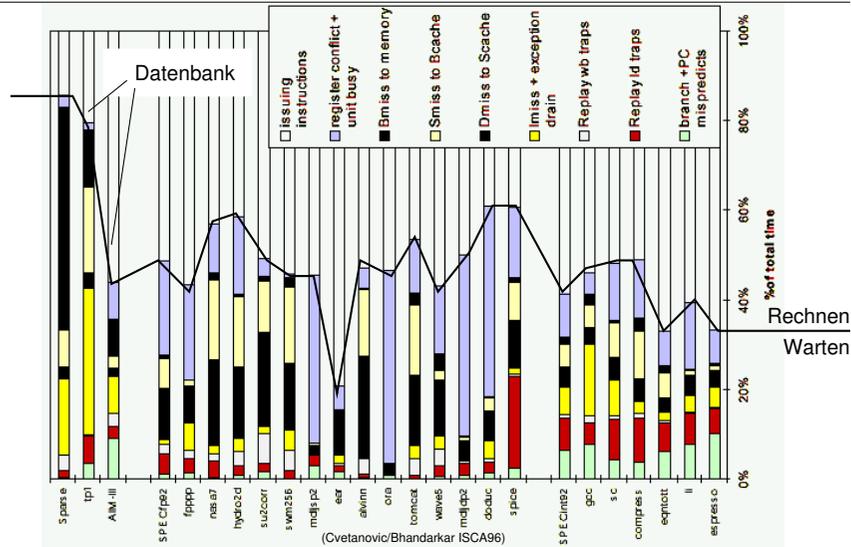
=> extreme Komplexität für RISC und CISC

- Marktbedeutung der IA-32 erlaubt große Investitionen
- bessere Chiptechnologie zuerst für x86 (Intel, AMD)
- alle x86-Prozessoren seit Pentium sind superskalar
- vgl. AMD K7 Präsentation (extern)
- K7 verwaltet bis zu 72 "instructions in flight"

Pentium IV

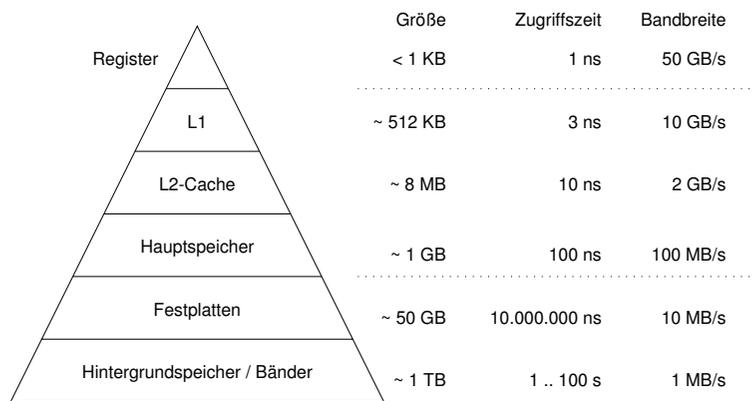


Benchmarks: Beispiel Alpha 21164



Medientechnik | WS 2001 | 18.204

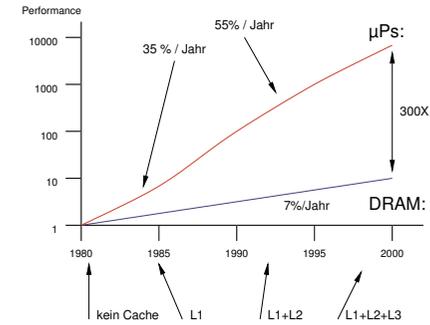
Speicherhierarchie



- "Diskontinuität" zwischen Halbleiter- und Magnetspeichern

Medientechnik | WS 2001 | 18.204

DRAM: Performance Gap



- DRAM-Kapazität: 60% / Jahr, Latenz: 7% / Jahr
 - Prozessor-Performance: 55% / Jahr
 - Kluft vergrößert sich ständig
- => Speicherhierarchie mit Caches notwendig

Medientechnik | WS 2001 | 18.204

DRAM: Performance Gap: Was tun?

schnellerer Speicher notwendig ...

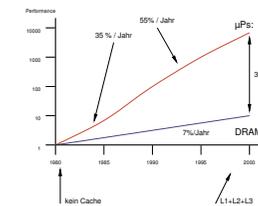
- aber DRAM inhärent langsam
 - SRAM sehr teuer
- => DRAM besser ausnutzen
- SDRAM, SDRAM-DDR
 - RAMBUS, SLDRAM

=> Speicherhierarchie verbessern

- größere, schnellere Caches
- bessere Cache-Organisation
- Prefetch-Optimierungen

=> neue Konzepte?

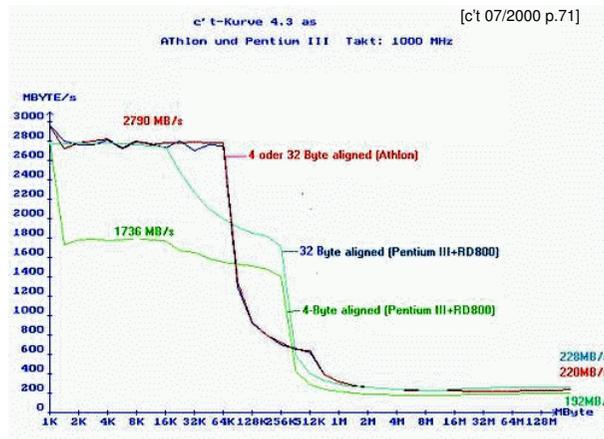
- IRAM



"Cache: a safe place for hiding or storing things"
Websters dictionary

Medientechnik | WS 2001 | 18.204

x86: ctkurve



- Messung der Cache-Transfertrate vs. Blockgröße (random)
- Caches deutlich sichtbar: Pentium 16K/256K, Athlon 64K/512K

Cache: Compulsory / Capacity / Conflict

3 Arten Cache-Misses:

- compulsory (cold start / first reference)
erster Zugriff auf einen Block
- capacity Cache zu klein für alle benötigten Blöcke;
Blöcke müssen ausgetauscht werden
=> Cache vergrößern
- conflict (collision misses / interference misses)
bei direct mapped / set associative Caches:
mehrere Blöcke im gleichen Set benötigt
=> Organisation verbessern, etwa 4fach assoz.
=> victim buffers

Cache: Speicherzugriffe

- Speicherzugriffe limitieren Performance (für von-Neumann Rechner)
- effiziente Ausnutzung von Caches erforderlich

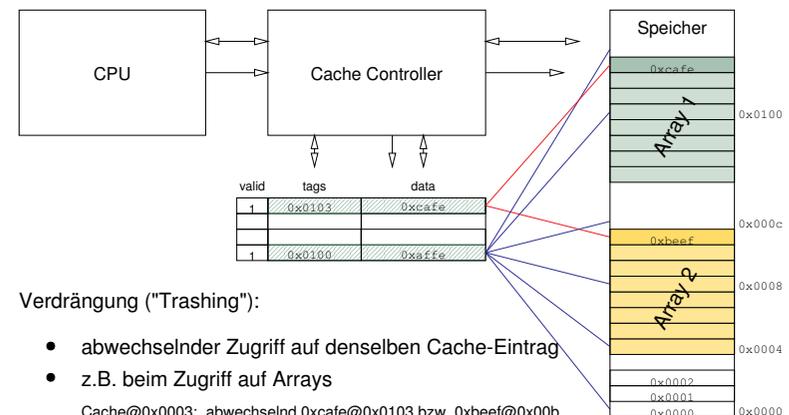
aber:

- Cachegröße oft nicht ausreichend, Beispiel Bildverarbeitung:
- 1024x1024x8: 1 MByte / Bild mind. zwei Bilder erforderlich

also:

- sorgfältige Optimierung der Algorithmen / Filter
- möglichst lokale Zugriffe / "Blocking"-Techniken
- Ausnutzen von Prefetch-Befehlen
- völlig andere Zugriffsmuster als SPEC/OPC-Benchmarks
- evtl. Probleme mit Direct-Mapped Caches (Verdrängung)

Cache: Trashing (direct mapped)



Verdrängung ("Trashing"):

- abwechselnder Zugriff auf denselben Cache-Eintrag
- z.B. beim Zugriff auf Arrays
Cache@0x0003: abwechselnd 0xc0afe@0x0103 bzw. 0xc000b@0x000b
- ständiges Neuladen, massiver Performanceverlust
- bessere Cache-Organisation / OS-Unterstützung

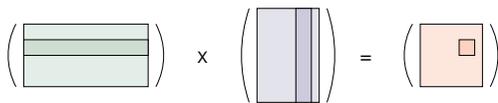
Cache: direct-mapped conflict misses

```
static void filterF(char* in1, char* out1)
{
    register int i0,i1,i2;
    register int x, int y;
    register char *in,*out;
    in = in1;
    out = out1;
    for( y=0; y < YRES; y++ ) {
        i0 = (int)in[0];
        i1 = (int)in[1];
        /* ignore boundary pixels, over/underflow for this benchmark */
        for( x=1; x < XRES-1; x++ ) {
            i2 = (int)in[x+1];
            out[x] = (char)( (i0 + (2*i1) + i2) / 4 );
            i0 = i1; i1 = i2;
        }
        in += XRES;
        out += XRES;
    }
} /*filterF*/
```

- read a byte from one array, compute, store result in second array, a byte at a time.
- If the arrays line up on top of each other in a direct-mapped cache, there is massive cache-thrashing.

execution time	via array	size:	[comp.arch	posting]					
SYS	511	512	513	1023	1024	1025	2047	2048	2049
CRIM	0.2	0.3	0.2	0.8	7.3*	0.9	3.7	33.4*	3.4 D
INDIGO4K	0.2	0.3	0.2	0.8	9.4*	0.8	3.2	37.9*	3.2 D
IN4K-fix	0.2	0.2	0.2	0.8	0.8	0.8	3.3	3.2	3.2 D
HP 720	0.3	0.7	0.3	1.1	2.7*	1.0	4.2	10.8*	4.2 D
HP 735	0.1	0.6*	0.1	0.6	2.7*	0.6	2.4	11.1*	2.6 D
HP 735	0.1	0.7*	0.1	0.6	2.7*	0.6	2.2	10.8*	2.2 D
Gwy486-66	0.3	0.3	0.3	1.3	1.4	1.3	5.5	5.5	5.5 SA?

Cache: Blocking

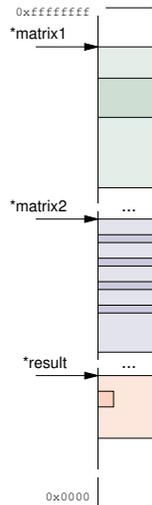


Beispiel Matrixmultiplikation:

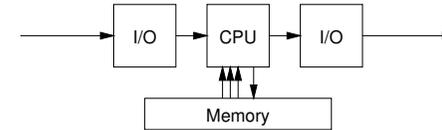
- Zugriff auf drei Speicherbereiche
- unterschiedliche Zugriffsmuster
- gesamte Daten passen weder in L1 noch in L2
- Zugriffszeiten Reg/L1/L2/Mem: z.B. 1:2:10:100

=> "Blocking":

- Aufteilung des Algorithmus in Cache-gerechte Häppchen
- aber abhängig von Cache und Applikation
- oft Performancegewinn um Faktor 2..10 (..100) möglich



Rechnerarchitektur? Bsp. Bildverarbeitung

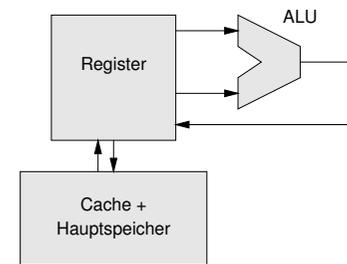


- oft viele Filter oder gar Filterbänke erforderlich
- lokale Operatoren erfordern (m*m) Operationen pro Bildpunkt
- Berechnung muß für jedes Bild wiederholt werden

Beispiel: 1024x1024x8 Grauwertbild, 50 Filter à 7x7 Pixel:
1024x1024 x 50 x 7x7 Operationen/Bild = 2.56 GOP/Bild

- Speicherbedarf für einzelne Bilder unproblematisch
- aber enorme Anforderungen an Rechenleistung

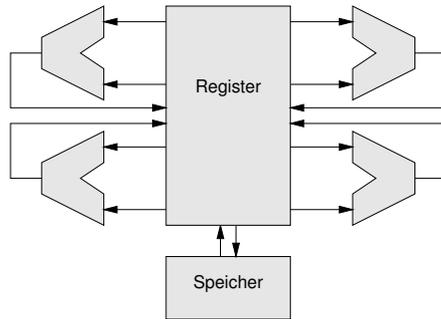
Rechnerarchitektur: von-Neumann



von-Neumann Prinzip:

- CPU mit Registern und ALU
- aber nur wenige (8 .. 64) Register
- selbst einzelne Filteroperation erfordert Speicherzugriffe
- extreme Anforderungen an Caching und Speicherverwaltung

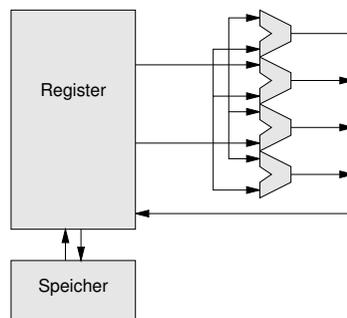
Rechnerarchitektur: mehrere ALUs ?



- viele Operationen im Prinzip massiv parallel berechenbar
- kaum Datenabhängigkeiten, wenn separate Quell- / Zielbilder
- scheitert am Speicherzugriff (von-Neumann bottleneck)

Medientechnik | WS 2001 | 18.204

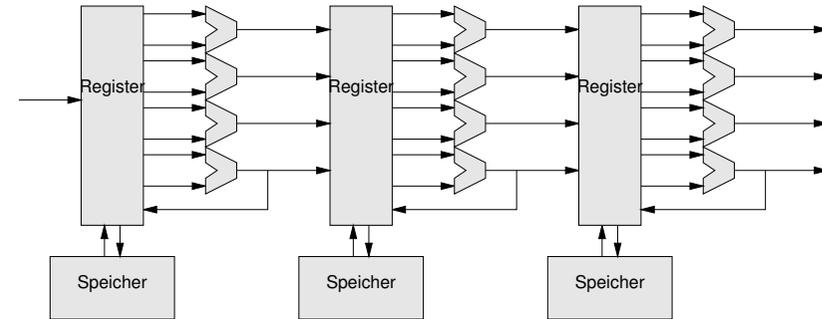
Rechnerarchitektur: VIS, MMX, ...



- Datenwortbreite 8 bit pro Pixel, aber 32/64 bit Datenpfade
- SIMD-Parallelverarbeitung, z.B. 8 Pixel parallel in einem Register
- keine Mehrbelastung für das Speicherinterface
- Details später

Medientechnik | WS 2001 | 18.204

Rechnerarchitektur: Streaming ...



- höhere Performance nur mit höherer Parallelität
- lokale Speicher/Register erforderlich
- "Streaming"-Architekturen, Realisierung z.B. mit FPGAs
- Details später

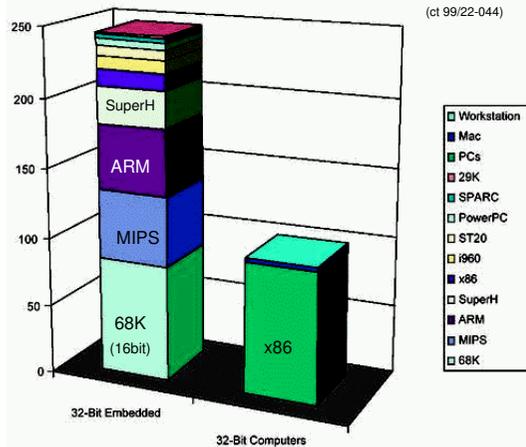
Medientechnik | WS 2001 | 18.204

ARM

- Marktführer für 32-bit Mikroprozessoren / Mikrocontroller
- derzeit ca. 60% Marktanteil: vor MIPS (30%) und anderen
- StrongArm ausgewählt für WindowsCE/PocketPC
- einzige relevante europäische Prozessorfirma:
www.arm.com
- "fabless" keine eigenen Fabriken
- "IP" Lizenzierung der Prozessoren als "intellectual property"
- "cores" fertige Designs für Integration in Chips
- ARM entwickelt von "Acorn Computers, Ltd", 1983-1985
- für den Nachfolger des BBC-Homecomputers (mit 6502 CPU)
- weil damalige 16-bit CPUs zu langsam (insb. für Interrupts)
- vollwertiger, aber sehr billiger RISC-Prozessor

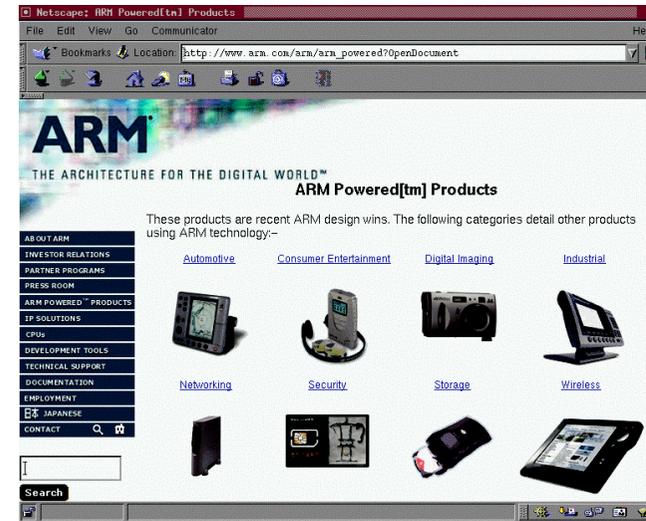
Medientechnik | WS 2001 | 18.204

Mikroprozessoren: 32-bit Markt '99

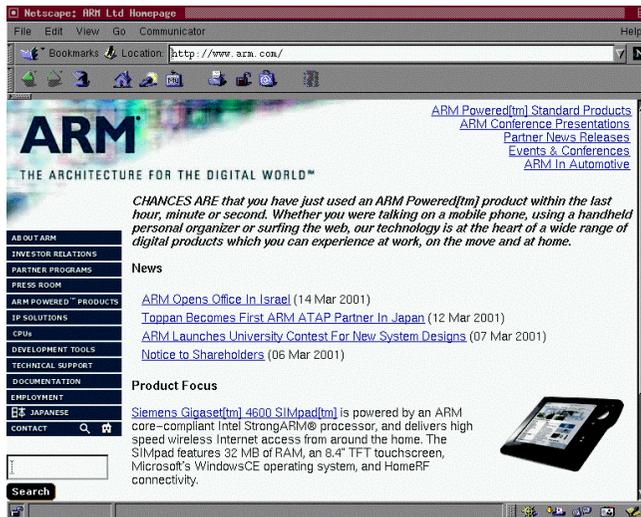


- pro Jahr (1999): 250 Mio. 32-bit μ Ps, plus ca. 100M in PCs
- zusätzlich ca. 5 Mrd. 4/8/16-bit Microcontroller

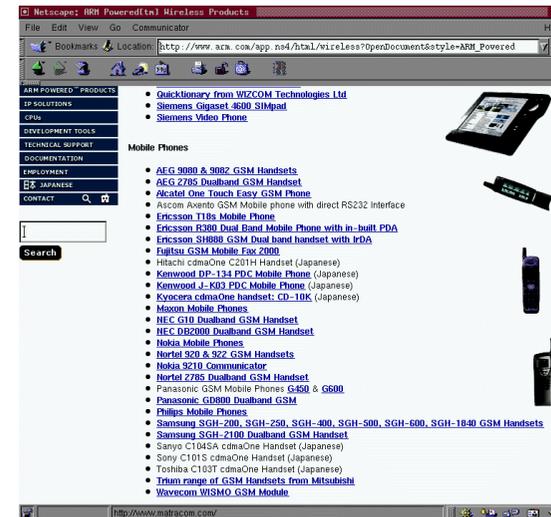
ARM: ARM powered products (3/2001)



ARM: "Advanced RISC Machines"



ARM: im Handy-Markt

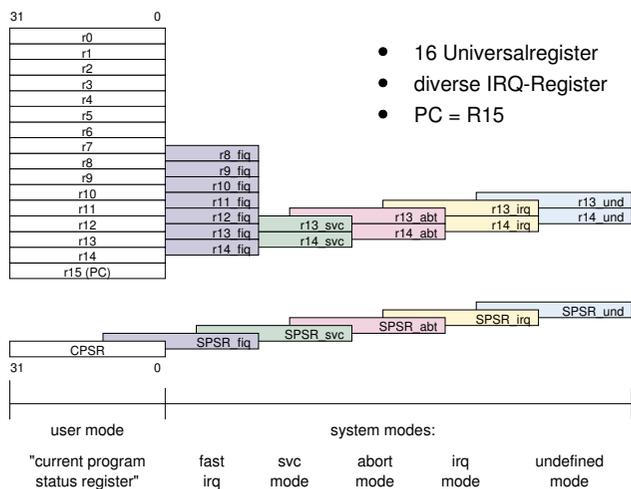


ARM: Architektur

- 32-bit RISC - nach Vorbild von RISC-I bzw. MIPS
 - load/store Architektur
 - 16 Universalregister
 - 3-Adress-Befehle, ein festes 32-bit Format
 - Ausführung aller Befehle in einem Takt (soweit möglich)
- aber kein Cache (aus Kostengründen)
- 3-stufige Pipeline
- erlaubt optimale Ausnutzung der Speicherbandbreite
- interessanter Befehlssatz (z.B. "predication", PC=R15, ...)
- Spezialregister für schnelle Interrupt-Behandlung
- Koprozessor-Interface für spätere Erweiterungen

Medientechnik | WS 2001 | 18.204

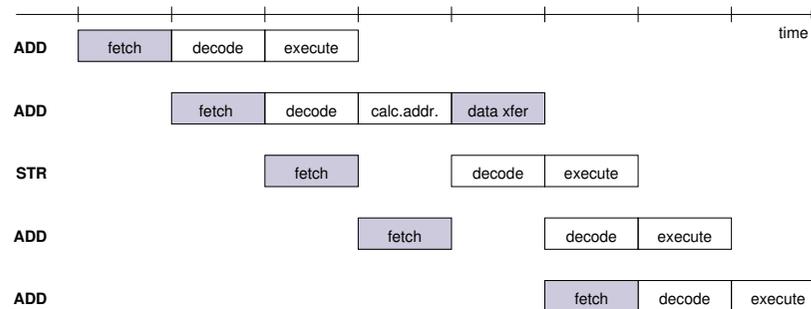
ARM: Register



(Furber, S39)

Medientechnik | WS 2001 | 18.204

ARM: 3-stufige Pipeline



- Konzept mit 3-stufiger Pipeline, ohne Cache
- Lade-/Speicherbefehle benötigen 4 Takte:
fetch / decode / address calc. / data transfer
- Beispiel für Nutzung der Pipeline

Medientechnik | WS 2001 | 18.204

ARM: Befehlssatz (1)

- übliche 3-Adress-Befehle:


```

ADD r0, r1, r2 ; r0 := r1 + r2
ADC r0, r1, r2 ; r0 := r1 + r2 + C (add with carry)
SUB r0, r1, r2 ; r0 := r1 - r2
RSC r0, r1, r2 ; r0 := r2 - r1 + C - 1 (reverse sub)
...
AND r0, r1, r2 ; r0 := r1 AND r2
EOR r0, r1, r2 ; r0 := r1 XOR r2
...
MVN r0, r2 ; r0 := NOT r2 (move negated)
...
CMP r1, r2 ; set CC on (r1 - r2) (compare)
TST r1, r2 ; set CC on (r1 AND r2) (bit test)
...
            
```

Medientechnik | WS 2001 | 18.204

ARM: Befehlssatz (2)

- optional: Verschiebung des zweiten Operanden:

```
ADD r3, r2, r1, LSL #3; r3 := r2 + r1<<3
ADD r5, r5, r3, LSL r2; r5 := r5 + r3<<(2^r2) (!)
```

mögliche Shifts: LSL, LSR, ASL, ASR, ROR, RRX

- Immediate-Operanden: $(0..255) * 2^{(2*\#rot)}$
- deckt viele wichtige Immediate-Werte ab

```
ADD r3, r3, #1 ; r3 := r3 + 1
AND r8, r7, #&FF ; r8 := r7[7:0]
```

cond	00 1	opcode	S	Rn	Rd	#rot	8-bit imm.
31	2827262524	19	15	11	8 7	0	

Medientechnik | WS 2001 | 18.204

ARM: Befehlssatz (3)

- 64-bit Addition (r1,r0) + (r3,r2)

```
ADDS r2, r2, r0 ; r2:=r2+r0, S=set CC (carry)
ADC r3, r3, r1 ; r3:=r3+r1+c
```

- Unterprogramm für $(10*r0)$

```
MOV r0, #3
BL TIMES10 ; Funktionsaufruf
...
TIMES10: MOV r0, r0, LSL #1 ; X2 r0:=r0<<1
ADD r0, r0, r0, LSL #2 ; X5 r0:=r0+r0<<2
MOV pc, r14 ; Rücksprung
```

Medientechnik | WS 2001 | 18.204

ARM: Beispielcode

```
// for (i=0; i < 10; i++) { a[i] = 0; }
MOV r1, #0 ; value to store in a[i]
ADR r2, a[0] ; r2 points to a[0]
MOV r0, #0 ; i=0
LOOP: CMP r0, #10 ; i<10?
BGE EXIT ; if i>=10 goto exit
STR r1,[r2,r0,LSL #2]; a[i]=0
ADD r0, r0, #1 ; i++
B LOOP
EXIT ...

... // store and load multiple registers:
STMFD r13!, {r0-r2,r14}; save work regs & link reg.
BL SUBROUTINE ; branch and link
...
LDMFD r13!, {r0-r2,pc} ; restore work regs & return
```

Medientechnik | WS 2001 | 18.204

ARM: Predicated Instructions

31	2827	8 7 6 5 4	0
N	Z	C	V
unused			
I	F	T	mode

Current Program Status Register

- bedingte Ausführung aller (!) Befehle
- abhängig von den Statusflags (negative, zero, carry, overflow)
- 16 Varianten: EQ NE .. GE LT .. Always NeVer

```
...
CMP r0, #5 ; if (r0 < 5)
BLLT SUB1 ; then call SUB1
BLGE SUB2 ; else call SUB2
...
```

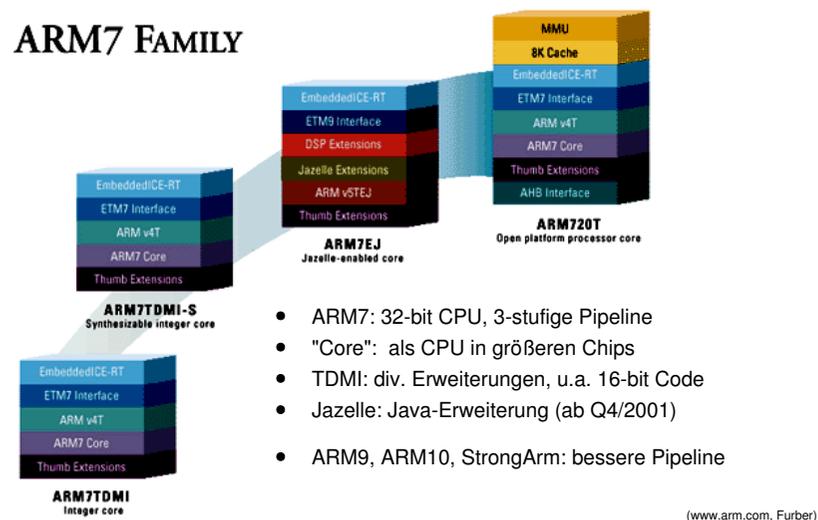
Medientechnik | WS 2001 | 18.204

ARM: Exceptions

Exception	Mode	Vector address
Reset	SVC	0x00000000
Undefined instruction	UND	0x00000004
Software interrupt (SWI)	SVC	0x00000008
Prefetch abort (instr. memory fault)	Abort	0x0000000C
Data abort (data access mem.fault)	Abort	0x00000010
IRQ (normal interrupt)	IRQ	0x00000018
FIQ (fast interrupt)	FIQ	0x0000001C

- Exceptions speichern Werte von PC und CPSR
- verzweigen zu den Vektoren (dort dann Sprung zum Handler)
- spez. Befehle zum Rücksprung
- zusätzliche Arbeitsregister für FIQ, Handler ab 0x0000001C

ARM: Roadmap



ARM: TDMI...

diverse Architekturvarianten:

Thumb 16-bit Befehlssatz, für minimale Programmgröße
 Debug Debug-Interface
 Multiplier Multiplizierer, 32x32 -> 64 bit
 ICE In-Circuit Emulator: Echtzeit-Breakpoints usw.
 Piccolo DSP-Erweiterung mit Puffern und schnellen MAC
 Jazelle Hardware zur direkten Ausführung von Java-Bytecodes

AMBA "advanced microcontroller bus architecture"

ARM 7 einfache 3-stufige Pipeline, kein Cache
 ARM 9 klassische 5-stufige Pipeline, I/D-Caches
 StrongARM mehrstufige Pipeline, hohe Taktraten (Compaq/intel)

...

(Furber, www.arm.com)

ARM: Thumb-Befehlssatz

- starker Kostendruck für eingebettete / mobile Geräte
- möglichst kleiner Programmspeicher wünschenswert
- auch zu Lasten von Performance

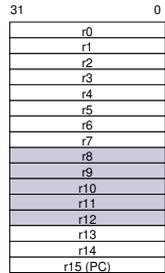
=> "Thumb"-Modus mit kompakterer Befehlskodierung

- 16-bit Befehlskodierung mit 2-Adress-Befehlen
- eingeschränkter Befehlssatz, Zugriff nur auf r0..r7, r13, r14, pc
- Umschalten zum ARM-Modus jederzeit möglich:
- für Exceptions / Betriebssystem / vollen Befehlssatz

Thumb vs. ARM-Befehlssatz:

- ca. 40% mehr Befehle pro Programm
- ca. 70% der Programmgröße
- ca. 30% weniger Stromverbrauch für Speicherzugriffe
- ca. 45% schneller (16-bit Speicher) / 40% langsamer (32-bit Speicher)

Thumb: Register



- Lo/Hi Register: r0 .. r7 / r8 .. r15
- Hi-Register nur eingeschränkt benutzbar
- r13 Stackpointer
- r14 Linkregister
- r15 Programmzähler

- 16-bit 2-Adress Befehlsformat
- alle Befehle setzen Condition Codes

- Implementierung durch Umkodierung in äquivalente ARM-Befehle

- wenig reguläre Befehlskodierung
- um 16-bit Raum voll auszunutzen

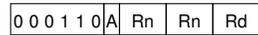
Thumb: Beispiele für Befehle



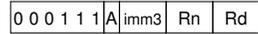
B<cond> <label>



LDR|STR{B} Rd, [Rn, #offset5]



ADD|SUB rd, rn, rm



ADD|SUB rd, rn, #imm3



Thumb software interrupt

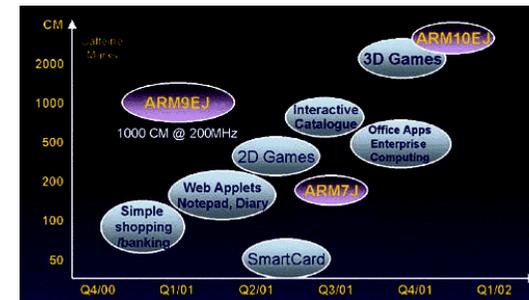
...

ARM: Thumb-Beispiel

```

...
CODE32                ; program starts in ARM mode
ADR    r0, START+1; thumb entry address
BX     r0           ; jump to thumb area
CODE16                ;
START ADR    r1, TEXT ; r1 -> "Hello, world"
LOOP  LDRB   r0, [r1] ; get the next byte
      ADD    r1, r1, #1 ; increment pointer **T
      CMP    r0, #0    ; check for end of text
      BEQ    DONE     ; finished?
      SWI    SWI_putchar; if not, print char (ARM mode!)
      B     LOOP;      ; ... next iteration
DONE  SWI    SWI_exit  ;
...
TEXT DATA = "Hello, world", &0a,&0d,&00
    
```

ARM: Jazelle



- Java ideal zur Erweiterung von mobilen/eingebetteten Geräten
- aber Java-Interpreter zu langsam / JIT zu speicheraufwendig

"Jazelle":

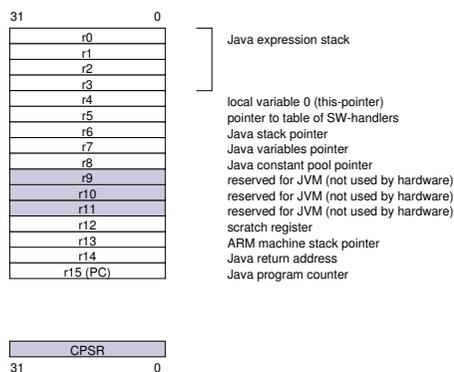
- Befehlssatzerweiterung, direkte Ausführung von Java-Bytecode

ARM: Jazelle Konzept

direkte Ausführung der wichtigsten Java-Bytecodes:

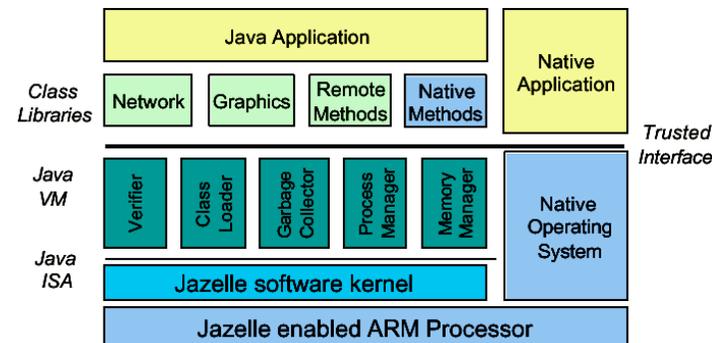
- zusätzlicher Betriebsmodus (ARM | Thumb | Jazelle)
- Befehlsdecoder für Bytecodes (ca. 12.000 Gatter)
- ALU angepaßt an Java-Datentypen und Stack-Adressierung
- Unterstützung für "komplexe" Bytecodes
- z.B. Funktionsaufrufe ("invokevirtual", "invokestatic")
- Exceptions und OS-Aufrufe wechseln in ARM-Modus
- bestehende ARM Betriebssysteme können verwendet werden
- Mischung aus Java / C / Assemblercode problemlos möglich
- hohe Performance, kompakter Code, kein JIT-Compiler notwendig

ARM: Jazelle Register



- alle ARM Register übernehmen Spezialaufgaben
- vier Register (r0..r3) als Rechenregister ("stack cache")
- Auslagern / Einlagern der Register auf den Stack automatisch

ARM: Jazelle, Schichten



- Software-Kernel implementiert die "komplexen" Java-Bytecodes
- obere Softwareschichten (Class Libraries, Apps) voll portabel
- Jazelle-Befehlssatz auch im "Armulator" und on-chip Debugging

ARM: Jazelle und Umfeld

	Execution Performance CM/MHz	Real-time System Performance	Memory Cost	Hardware Implementation Cost	Legacy Code / RTOS support
Software Emulation (SUN JDK, ARM9)	0.67		~16kbyte	-	Yes
Software Emulation (ARM JDK, ARM9)	1.7		~16kbyte	-	Yes
JIT	6.2*	Poor	>100k byte	-	Yes
Co-processor (eg Jedi Tech. JSTAR)	2.9		-	~25k gates	Yes
Dedicated Processor	3		-	20-30k gates	No
ARM with architecture extensions	5.5	Excellent	~8kByte	~12k gates	Yes

The only solution to meet all of the performance & application requirements.

*Note: JIT performance excludes compilation overhead.

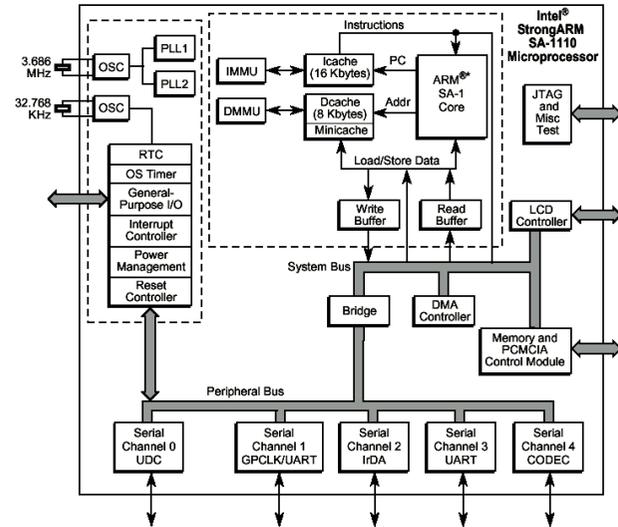
- guter Kompromiß aus HW/SW-Aufwand und Performance
- JIT-Compiler besser wegen zusätzlicher Optimierungen

StrongARM

- gemeinsam entwickelt von DEC und ARM
 - Harvard-Architektur mit separaten I/D-Caches
 - klassische 5-stufige Pipeline
 - mit Forwarding / Bypass
 - on-Chip MMU für Multitasking-Betriebssysteme
-
- Design ausgelegt für sehr hohen Takt
 - bei geringem Stromverbrauch
 - SA-110: 2.5 Millionen Transistoren, 200 MHz Takt, < 1 Watt
 - zugehöriger Supportchip liefert weitere Schnittstellen
-
- eingesetzt in den Compaq "Itsy" PDA-Prototypen
 - derzeit u.a. in den meisten Windows-CE Organzern
 - von Microsoft als einzige CPU für "Pocket-PC 2002" ausgewählt

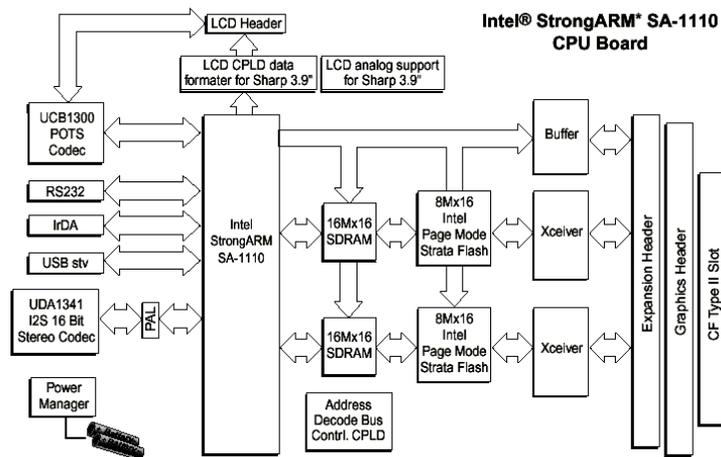


StrongARM: Blockschaltbild SA-1110



(Intel SA1110 Datenblatt)

StrongARM: Intel SA1110



CPU mit vielen Schnittstellen on-chip, Gesamtsystem: nur CPU+Speicher+IO

StrongARM: fuer PocketPC

Der Pocket PC wird erwachsen (Update)
 [06.09.2001 12:14]

Microsoft wird heute auf der **DemoMobile-Konferenz**[1] im kalifornischen La Jolla seine nächste Windows-CE-Version für stiftbediente Organizer vorstellen, die vor allem Unternehmen ansprechen soll. Das bisher unter dem Codenamen Merlin bekannte Betriebssystem heißt nun Pocket PC 2002, beruht nach wie vor auf **Windows CE 3.0**[2] und läuft - wie bereits in c't 18/2001 berichtet - nur noch auf **ARM-Prozessoren**[3].

Die selbstaufgelegte Prozessor-Monogamie bietet eine Menge Zündstoff, da einige OEM-Partner wie **Casio**[4] bisher MIPS- oder SH3-Prozessoren eingesetzt hatten. Zudem führte Microsoft die Vielfalt unterstützter CPUs stets als Vorteil gegenüber Hauptkonkurrent **Palm**[5] an, der seinem bis dato einzigen Prozessorlieferanten Motorola auf Gedeih und Verderb ausgeliefert ist. Doch Microsofts Produkt-Manager Matt Taylor versucht zu beschwichtigen: "Wir unterstützen zwar nur eine Prozessorarchitektur, die aber von verschiedenen Herstellern lizenziert wird."

Pocket PC 2002 ist lediglich ein Zwischen-Update und bietet keine dramatisch veränderte Oberfläche, wie es beim Wechsel von Windows CE 2.1 auf 3.0 der Fall war. Die Icons sind jetzt etwas bunter und erinnern stark an Windows XP. Die Oberfläche lässt sich mit Skins weitgehend anpassen, der "Today"-Bildschirm ist übersichtlicher und bietet mehr Platz für die eigentlichen Informationen. Über zusätzliche Symbole in der Taskleiste kann man jetzt schneller auf die Audio-Eigenschaften, eingegangene Nachrichten oder anstehende Termine zugreifen.

Aus der Kritik am mangelhaften Taskmanagement der **Vorversion**[6] hat Microsoft zwar Konsequenzen gezogen, die aber zu kurz greifen. Mit dem jetzt standardmäßig vorhandenen Smart-Minimize-Button werden Anwendungen lediglich in den Hintergrund befördert, wo sie aber weiterhin Speicher belegen, was wiederum zu Lasten der Performance gehen kann.

Die bei der ersten Pocket-PC-Generation lediglich auf CD mitgelieferte Wörterkennungssoftware Transcriber befindet sich jetzt vorinstalliert im ROM. Eine weitere neue Texteingabevariante ist der Block-Recognizer, ein offensichtliches

Xscale: Übersicht

- Intels neueste Version der ARM-Architektur
- 7-stufige, komplexe Pipeline für höhere Taktrate
- zusätzlicher ALU (MAC) für Signalverarbeitung
- zugehörige neue (SIMD) Befehle
- dreifacher Cache: 32 KB I, 32 KB D, 2 KB "mini data"
- getrennte Befehls- und Daten-MMU
- diverse Pufferspeicher, separate 32-bit Load/Store-Busse
- flexibles Power-Management
- "speed step" Technik regelt Spannung vs. Taktrate
- Fertigung in 0.18µ

(intel XScale whitepaper, und SA 1110 Datenblatt)

Xscale: Signalverarbeitung

SMLAxy	32 <= 16x16 + 32
SMLWxy	32 <= 32*16 + 32
SMLALxy	64 <= 16*16 + 64
SMULsy	32 <= 16*16
SMULwy	32 <= 32*16
QADD	rz <= saturate(rx + ry)
QDADD	rz <= saturate(rx + 2*ry)
QSUB	...
...	...

- Auswahl der DSP-Befehle
- Intel liefert "integrated performance primitives" Bibliotheken, "filtering, ..., transforms, MP3, H.263, MPEG-4, GSM, JPEG"
- Konzept und Details zu SIMD-Befehlssätzen s.u.

(Intel IPP datasheet)

Xscale: Performance

Cache blocking techniques, such as strip-mining, are used to improve temporal locality of the data. Given a large data set that can be reused across multiple passes of a loop, data blocking divides the data into smaller chunks which can be loaded into the cache during the first loop and then be available for processing on subsequent loops thus minimizing cache misses and reducing bus traffic.

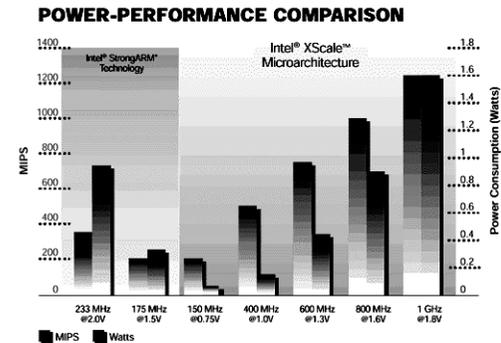
As an example of cache blocking consider the following code:

```
for(i=0; i<10000; i++)
  for(j=0; j<10000; j++)
    for(k=0; k<10000; k++)
      C[j][k] += A[i][k] * B[j][i];
```

The variable A[i][k] is completely reused. However, accessing C[j][k] in the j and k loops can displace A[i][j] from the cache. Using blocking the code becomes:

```
for(i=0; i<10000; i++)
  for(j1=0; j<100; j++)
    for(k1=0; k<100; k++)
      for(j2=0; j<100; j++)
        for(k2=0; k<100; k++)
          {
            j = j1 * 100 + j2;
            k = k1 * 100 + k2;
            C[j][k] += A[i][k] * B[j][i];
          }
```

Xscale: Performance



- XScale mit 0.13µm CMOS-Prozeß, 1 GHz Takt, < 2 Watt
- gleichzeitig bessere Performance durch bessere Architektur
- klarer Kandidat für zukünftige Mobilgeräte

(www.developer.intel.com)

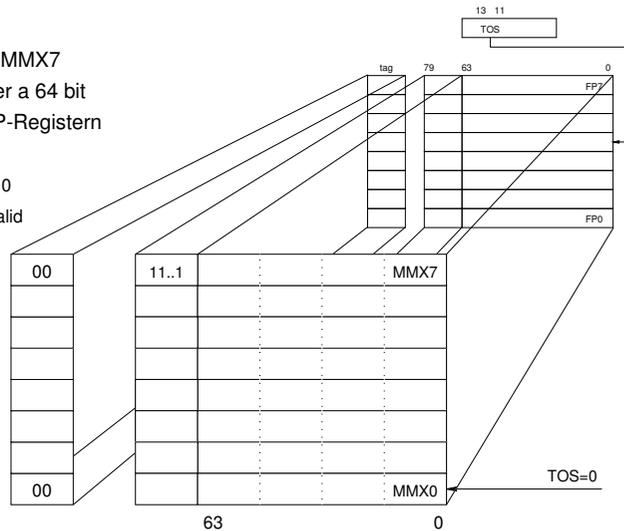
MMX: Entwurfsentscheidungen

Kompatibilität zu alten Betriebssystemen / Apps:

- keine neuen Register möglich => FP-Register nutzen
- keine neuen Exceptions => Überlauf ignorieren
- bestehende Datenpfade nutzen => saturation arithmetic
- möglichst wenig neue Opcodes => 64 bit
- alte Prozessoren und neue Software => Code doppelt
- => MMX DLLs
- Test-Applikationen: => 16 bit dominiert
- (audio/image/MPEG-1/3D-Graphik/...) => MMX DLLs
- keine Tools => Assembler
- optimierte Libraries verfügbar

MMX: Register

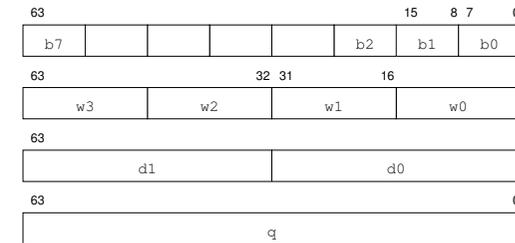
- MMX0 .. MMX7
- 8 Register a 64 bit
- in den FP-Registern
- FP NaN
- FP TOS = 0
- tag 00 = valid



MMX: Datenformate

64-bit Register, 4 Datentypen:

- packed byte *8 / packed word *4 / packed doubleword *2 / quadword
- Zugriff abhängig vom Befehl



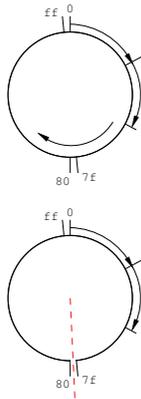
MMX: Befehlssatz

EMMS (FSAV / FRESTOR)		clear MMX state (handle FP regs)
MOVQ	mm1, mm2/mem64	move 32 bit data
MOVQ	mm1, mm2/mem64	move 64 bit data
PACKSSWB	mm1, mm2/mem64	pack 8*16 into 8*8 signed saturate
PUNPCKH	mm1, mm2/mem64	fancy unpacking (see below)
PACKSSDW	mm1, mm2/mem64	pack 4*32 into 4*16 signed saturate
PAND	mm1, mm2/mem64	mm1 AND mm2/mem64 / auch OR/XOR/NAND
PCMPEQB	mm1, mm2/mem64	8*a==b, create bit mask / auch GT
PADDB	mm1, mm2/mem64	8*add 8 bit data
PSUBD	mm1, mm2/mem64	2*sub 32 bit data / signed wrap
PSUBUSD	mm1, mm2/mem64	2*sub 32 bit data / unsigned saturate
PSLL	mm1, mm2/mem64/imm8	shift left mm1 / auch PSRA/PSRL
PMULL/HW	mm1, mm2/mem64	4*mul 16*16 store low/high 16 bits
PMADDWD	mm1, mm2/mem64	MAC 4*16 -> 2*32
insgesamt 57 Befehle		(Varianten B/W/D S/US)

MMX: "Saturation Arithmetic"

was soll bei einem Überlauf passieren?

- wrap-around
..., 125, 126, 127, -128, -127, ...
- saturation
..., 125, 126, 127, 127, 127, ...
- Zahlenkreis "aufgeschnitten"
- gut für DSP-Anwendungen



paddw (wrap around):

a3	a2	a1	7FFFh
+			
b3	b2	b1	0004h

a3+b3	a2+b2	a1+b1	8003h

paddsw (saturating):

a3	a2	a1	7FFFh
+			
b3	b2	b1	0003h

a3+b3	a2+b2	a1+b1	7FFFh

MMX: "packed multiply add word"

für Skalarprodukte:

```

vector_x_matrix_4x4( MMX64* v, MMX64 *m ) {
    MMX64 v0101, v2323, t0, t1, t2, t3;

    v0101 = punpckldq( v, v ); // unpack v0/v1
    v2323 = punpckhdq( v, v ); // unpack v2/v3

    t0 = pmaddwd( v0101, m[0] ); // v0|v1 * first 2 rows
    t1 = pmaddwd( v2323, m[1] ); // v2|v3 * first 2 rows
    t2 = pmaddwd( v0101, m[2] ); // v0|v1 * last 2 rows
    t3 = pmaddwd( v2323, m[3] ); // v2|v3 * last 2 rows

    t0 = padd( t0, t1 ); // add
    t2 = padd( t2, t3 ); // add
    v = packssdw( t0, t2 ); // pack 32->16, saturate
}
    
```

pmaddwd

a3	a2	a1	a0
*			
b3	b2	b1	b0

a3*b3+a2*b2		a1*b1+a0*b0	

MMX: "packed compare"

Vergleiche / Sprungbefehle:

- schlecht parallelisierbar
- Pipeline-Abhängigkeiten

pcmpgtw:

23	45	16	34
>			
31	7	16	67

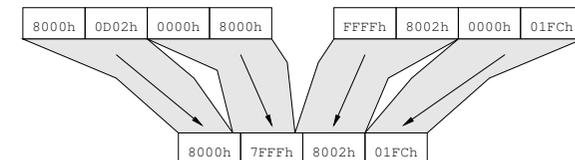
0000h	FFFFh	0000h	0000h

=> keine Sprungbefehle in MMX
=> compare-Operationen setzen Bitmasken

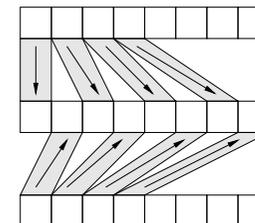
- Bitmasken für logische Ops verwendbar
- Beispiel: chroma-keying

MMX: packssdw / punpckhbw

packssdw: pack with saturation 32 -> 16 signed data:



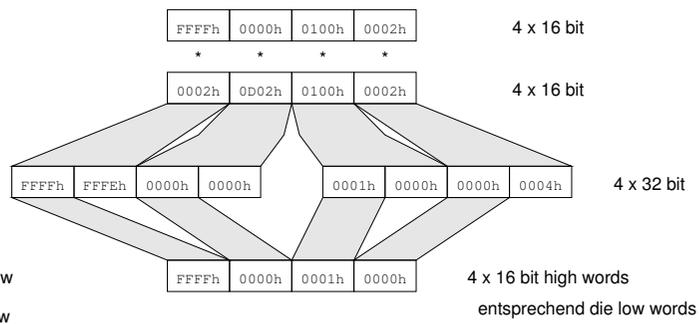
punpckhbw:



punpcklbw: lower 32 bits

MMX: pmullw / pmullhw

pmull[lh]w: multiply 4 words, write low/high byte of results:

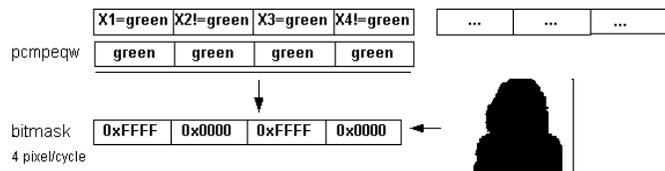


mit Packbefehlen kombinieren, wenn 32-bit Resultate gewünscht

MMX: Chroma Keying (1)

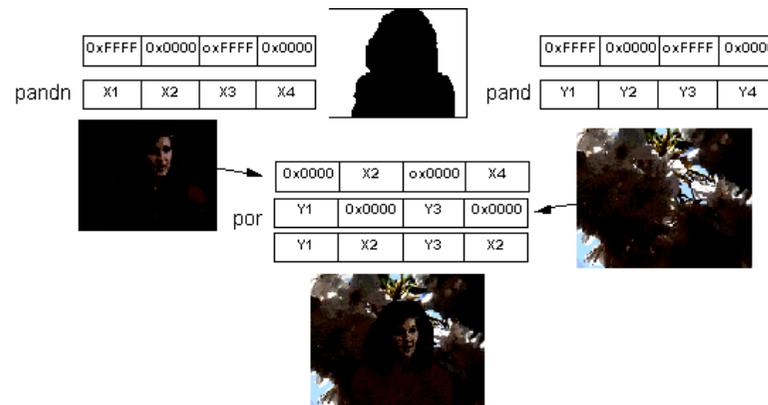
"Wetterbericht":

- MMX berechnet 4 Pixel / Takt
- keine Branch-Befehle
- Schritt 1: Maske erstellen (high-color: 16 bit/pixel)



[Intel MMX appnote]

MMX: Chroma Keying (2)



MMX: Zufallszahlen

$$x(t) = (x(t-1) * 47989) \& 0xFFFF;$$

```
QuadWord DithMultVal = 0x4f314f314f314f31;
QuadWord DithRegInit = 0x4f31994d2379bb75;
```

```
Init:
MOVQ mm0, DithRegInit;
```

```
Loop:
PMULLW mm0, DithMulVal // x(t) -> x(t+1)
MOVQ [result64], mm0 // 3 clocks
```

- PMULLW latency 3, throughput 1 (on Pentium)
- bis zu vier Zufallszahlen pro Takt (U/V pipelines genutzt)

3Dnow! Prefetch

Speicherzugriffe in Multimedia-Applikationen:

- reguläre Speicherzugriffsmuster
- ungewöhnliche Lokalität
- viele Daten werden (pro Frame) nur einmal benötigt
- aber regelmässig (in jedem Frame)
- Performance stark von optimaler Cache-Ausnutzung abhängig

=> prefetch-Befehl

- quasi normaler Ladebefehl, aber ohne Zielregister
- gewünschte Daten werden in L1/L2-Cache geladen
- löst keine Exceptions / Page Faults aus

=> "memory streaming"

=> auch für andere Anwendung gut nutzbar (etwa Numerik)

3Dnow! Division / Quadratwurzel

- Rechenwerk für Division / Sqrt ist sehr aufwendig
- möglichst wenig Chipfläche für 3Dnow!
- teilweise nur geringe Genauigkeit benötigt
- etwa Shading/Beleuchtungsberechnung für 3D-Graphik

=> Division und Quadratwurzel per Approximation

- erster Befehl liefert 14/15 bit Approximation
- aus Lookup-Table und Interpolation
- mit vollem Takt
- zusätzliche Befehle für Newton-Iteration
- quadratische Konvergenz: zwei Iterationsschritte für volle Genauigkeit
- wenig Hardwareaufwand
- voll in Pipeline integriert, maximaler Durchsatz

3Dnow! Division / Quadratwurzel

- Hardware-Dividierer ist sehr aufwendig
- oft wird nicht die volle Genauigkeit benötigt, z.B. Beleuchtungsberechnung bei 3D-Graphik

- Iteration zur Berechnung von 1/b:

$$x' = x * (2 - b*x)$$

(14-Bit Precision)

MOVD	MM0, [mem]	;	0		w	
PFRCPP	MM0, MM0	;	1/w		1/w	(approx.)
MOVQ	MM2, [mem]	;	y		x	
PFMUL	MM2, MM0	;	y/w		x/w	

(24-Bit Precision)

MOVD	MM0, [mem]	;	0		w	
PFRCPP	MM1, MM0	;	1/w		1/w	(approx.)
PUNPCKLDQ	MM0, MM0	;	w		w	(MMX instruction)
PFRCPPIT1	MM0, MM1	;	1/w		1/w	(intermed.)
MOVQ	MM2, [mem]	;	y		x	
PFRCPPIT2	MM0, MM1	;	1/w		1/w	(full prec.)
PFMUL	MM2, MM0	;	y/w		x/w	

3Dnow! Quadratwurzel

- Iteration zur Berechnung von 1/sqrt(b):

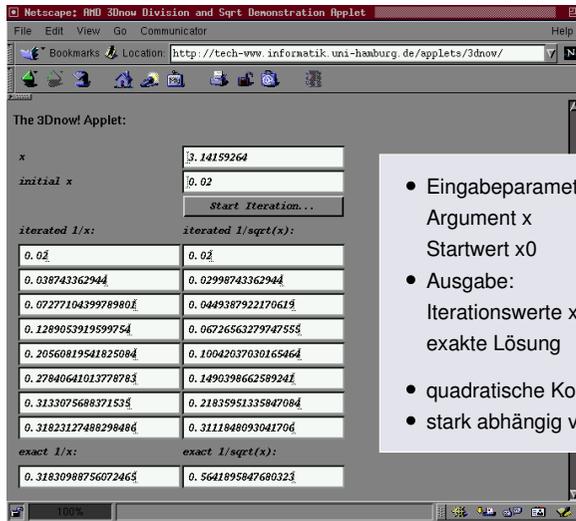
$$x' = 0.5 * x * (3 - b*x^2)$$

- separate Befehle zur ersten Schätzung (14 bit)
- zwei weitere Befehle zur ersten und zweiten Iteration
- Resultat mit 24-bit Genauigkeit
- abschließende Multiplikation für sqrt(x) statt 1/sqrt(x)

(24-Bit Precision)

MOVD	MM0, [mem]	;	0		a	
PFRSQRT	MM1, MM0	;	1/(sqrt a)		1/(sqrt a)	(approx.)
MOVQ	MM2, MM1	;	X_0 = 1/(sqrt a)			(approx.)
PFMUL	MM1, MM1	;	X_0 * X_0		X_0 * X_0	step 1
PUNPCKLDQ	MM0, MM0	;	a		a	(MMX instr.)
PFRSQRT1	MM1, MM0	;	(intermediate)			step 2
PFRCPPIT2	MM1, MM2	;	1/(sqrt a)			(full prec.) step 3
PFMUL	MM0, MM1	;	(sqrt a)		(sqrt a)	

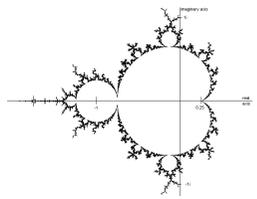
3Dnow! Div/Sqrt-Applet



- Eingabeparameter:
Argument x
Startwert x0
- Ausgabe:
Iterationswerte x0,x1,...
exakte Lösung
- quadratische Konvergenz
- stark abhängig vom Startwert

3D Now! Apfelmännchen

```
Function IterPasD
(I,R :Double; Grenze, Tiefe :Paratyp):Paratyp;
var A,B,C:double;
Begin
Count:= 0;
A:=0; B:=0;
Repeat
C:= SQR(A) - SQR(B) + R;
B:= 2*A*B + I;
A:= C;
INC (Count);
Until (abs (A) >Grenze) or (Abs (B) > Grenze)
or (Count=Tiefe);
IterpasD:=Count;
End;
```

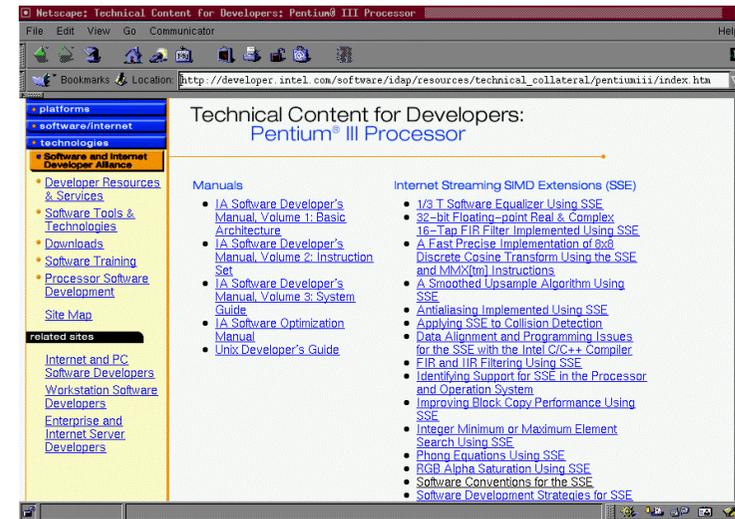


3D Now! Apfelmännchen

```
; Quadriere (A + jB)**2 = A**2 - B**2 + j 2*A*B
; Entry MM0 ;A | B
; MM1 ;1 | -1
; MM2 ;R | I
;
loop:
MOVQ MM3,MM0 ;MM3=A | B
MOVQ MM4,MM0 ; oh weh
PSLLQ MM3,32 ; das Vertauschen ist
PSRLQ MM4,32 ; sehr mühsam ...
POR MM3,MM4 ;MM3=B | A

PFMUL MM3,MM0 ;MM3= A*B | A*B
PFMUL MM0,MM0 ;MM0= A**2 | B**2
PFMUL MM0,MM1 ;MM0= A**2 | -B**2
PFACC MM0,MM3 ;MM0= A**2 - B**2 | A*B+A*B
PFADD MM0,MM2 ;MM0= A**2 - B**2 + R | 2*A*B+I
; = A(n+1) | = B(n+1)
; iA = INT(A) | iB = Int(B)
PF2ID MM4,MM0
MOVQ iA,MM4
; Sieh nach, ob A oder B > GRENZE ist
...
dec CX ; iteration counter
jnz loop
```

ISSE: Homepage / Literatur



ISSE: Entwurfsentscheidungen

- Markt fordert 3D
- mindestens doppelte FP-Performance notwendig

2-fach oder 4-fach SIMD?

- 128-bit machbar (FP bereits 80-bit)
 - bereits 2 64-bit ALUs auf dem Prozessor
- => 4-fach SIMD

"already register-starved IA32 architecture"

- => neue Register, 128-bit
- => erfordert OS-Unterstützung

- 70 neue Befehle
- sowohl "packed" als auch "scalar ISSE instructions"

ISSE: "Streaming"

typisch für Medienverarbeitung:

- hohe Datenmenge / Datenrate
- geringe Lokalität: viele Daten (Pixel) werden nur 1x benötigt

=> Cache-"Pollution"
 => herkömmliche Cache-Strategien nutzlos
 => ALUs müssen auf die Daten warten

Performance leidet (extrem)

- 1GHz, 8x SIMD, 100 nsec Speicher: 800 OPs / 1 Zugriff

Streaming:

- Cache-Nutzung anpassen
- Prefetch: Daten rechtzeitig anfordern
- Speicherlatenz fast perfekt versteckt

(für Media-Apps.)

ISSE: Prefetch-Befehl

PREFETCHh—Prefetch Data Into Caches

Opcode	Instruction	Description
0F 18 /1	PREFETCHT0 m8	Move data from m8 closer to the processor using T0 hint.
0F 18 /2	PREFETCHT1 m8	Move data from m8 closer to the processor using T1 hint.
0F 18 /3	PREFETCHT2 m8	Move data from m8 closer to the processor using T2 hint.
0F 18 /0	PREFETCHNTA m8	Move data from m8 closer to the processor using NTA hint.

Description

Fetches the line of data from memory that contains the byte specified with the source operand to a location in the cache hierarchy specified by a locality hint:

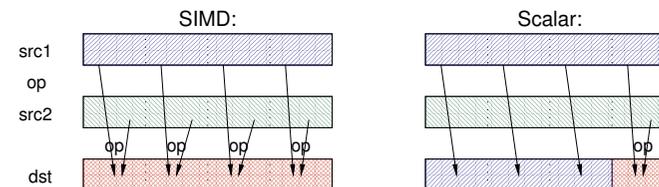
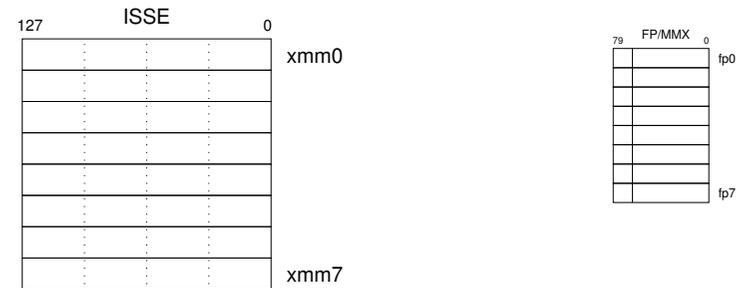
- T0 (temporal data)—prefetch data into all cache levels.
- T1 (temporal data with respect to first level cache)—prefetch data in all cache levels except 0th cache level
- T2 (temporal data with respect to second level cache)—prefetch data in all cache levels, except 0th and 1st cache levels.
- NTA (non-temporal data with respect to all cache levels)—prefetch data into non-temporal cache structure. (This hint can be used to minimize pollution of caches.)

The source operand is a byte memory location. (The locality hints are encoded into the machine level instruction using bits 3 through 5 of the ModR/M byte. Use of any ModR/M value other than the specified ones will lead to unpredictable behavior.)

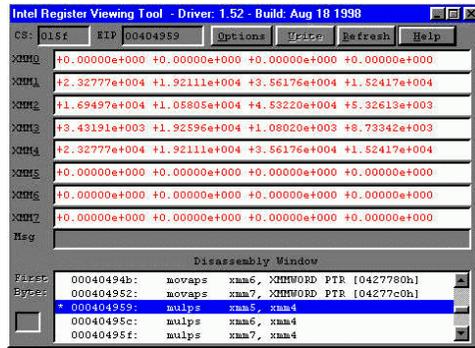
If the line selected is already present in the cache hierarchy at a level closer to the processor, no data movement occurs. Prefetches from uncacheable or WC memory are ignored.

The PREFETCHh instruction is merely a hint and does not affect program behavior. If executed, this instruction moves data closer to the processor in anticipation of future use.

ISSE: Register



ISSE: Register Viewing Tool



Softwareentwicklung für MMX / ISSE / 3Dnow:

- nur rudimentäre Compiler- und Tool-Unterstützung
- oft handoptimierter Assembler wg. bester Performance

ISSE: Programmierung

Intel VTune Performance Enhancement Environment:

- optimierender Compiler mit ISSE-Unterstützung:
 - Intrinsics
 - Vector Class Library
 - Vectorization
 - Intel Performance Library Suite
- C-Funktionen, Compiler inlining
- Klassen, inlining durch Compiler
- optimierender Compiler
- erfordert 16-Byte Alignment aller Datentypen
- umfangreiche Profiling-Tools
- sehr teuer

ISSE: Programmierung mit "Intrinsics"

```
float xa[SIZE], xb[SIZE], xc[SIZE];
float q;

void do_c_triad() {
    for( int j=0; j < SIZE; j++ ) {
        xa[j] = xb[j] + q*xc[j];
    }
}
```

ISSE-Programmierung mit "Intrinsics" und VTUNE:

```
#define VECTOR_SIZE 4
__declspec(align(16)) float xa[SIZE], xb[SIZE], xc[SIZE];
float q;

void do_intrin_triad() {
    __m128 tmp0, tmp1;

    tmp1 = _mm_set_ps1(q);
    for( int j=0; j < SIZE; j+= VECTOR_SIZE ) {
        tmp0 = _mm_mul_ps( *((__m128 *) &xc[j]), tmp1 );
        *((__m128 *) &xa[j]) =
            _mm_add_ps(tmp0, *((__m128 *) &xb[j]));
    }
}
```

ISSE: AoS / SoA

Array of Structures:

- Daten lokal
- Anordnung schlecht für SIMD

```
struct
{
    float A, B, C;
} AoS_data[1000];
```

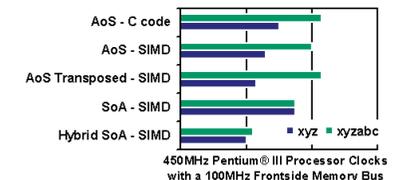
Structure of Arrays:

- Anordnung optimal für SIMD
- aber im Speicher "verstreut"

```
struct
{
    float A[1000], B[1000], C[1000];
} SoA_data;
```

=> Hybrid SoA - SIMD

```
struct
{
    float A[8], B[8], C[8];
} Hybrid_data[125];
```



ISSE: FIR-Filter

Nutzen von MMX/ISSE für Filter?

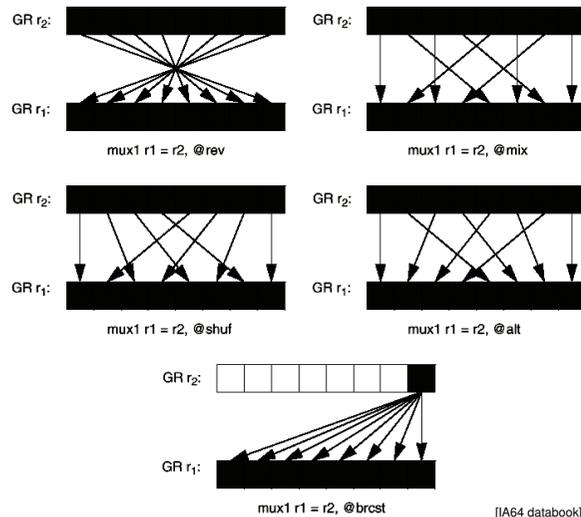
- MMX und ISSE für 16-bit Integer
- ISSE für 32-bit Gleitkommawerte
- maximal vierfache Leistung gegenüber skalarem Code

aber:

- erfordert Operanden-Alignment (16-Byte Grenzen)
- z.B. durch Duplizierung der Daten/Koeffizienten-Arrays
- Arraygrößen Vielfache von 4
- Multiplikation parallel, aber Akkumulation schwierig

=> siehe Intel Appnote
 "32-bit FP FIR Filter implemented using SSE"

ISSE2: mux1-Befehl (IA64)



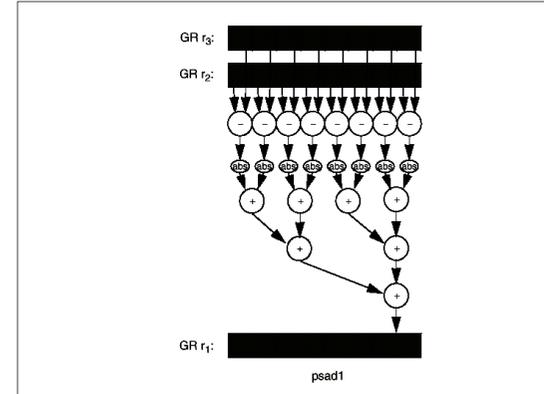
ISSE2: psad1-Befehl (IA64)

Parallel Sum of Absolute Difference

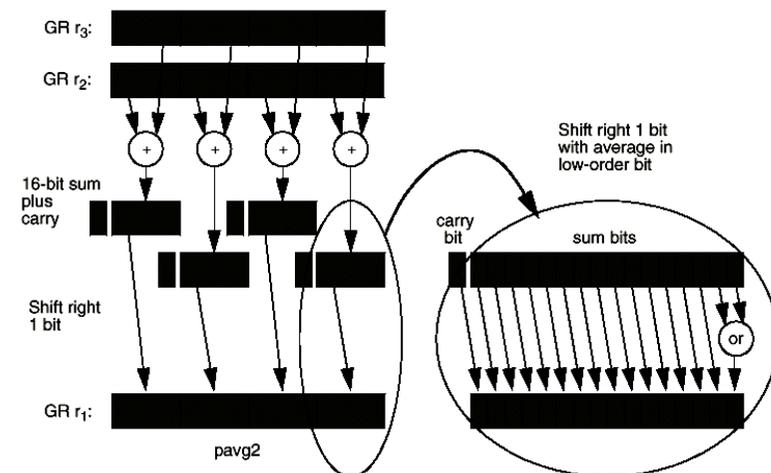
Format: (qp) psad1 r1 = r2, r3

Description: The unsigned 8-bit elements of GR r2 are subtracted from the unsigned 8-bit elements of GR r3. The absolute value of each difference is accumulated across the elements and placed in GR r1.

Figure 7-36. Parallel Sum of Absolute Difference Example



ISSE2: pavg2-Befehl (IA64)



Ausblick: Aktuelle Architekturen

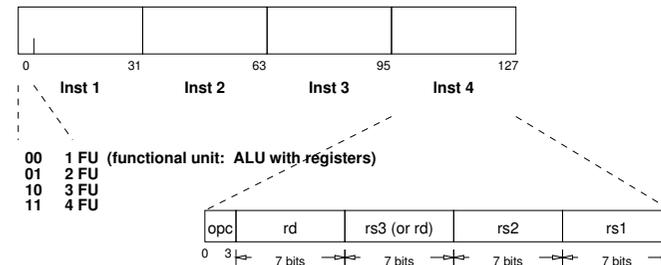
- Miniaturisierung ermöglicht immer höhere Integration
- => wohin mit all den Transistoren?
- noch größere Caches?
 - noch komplexere superskalare Prozessoren?
 - höhere Parallelität: VLIW, Multiprozessoren, Multithreading?
- drei aktuelle Beispiele:
- MAJC: VLIW-Konzept für schnelle Java-Verarbeitung
 - Piranha: 8 CPUs pro Chip für Hochleistungsserver
 - iFLOW Address Prozessor: Routing in Hardware

MAJC

- neue Prozessorarchitektur von Sun
- u.a. zur Unterstützung von Java und Multimedia
- dreifache Parallelität:
 - VLIW-Konzept, je 4 ALUs pro Prozessor
 - on-chip Multithreading des einzelnen Prozessors
 - on-chip Multiprozessoren mit schneller Kommunikation
- diverse I/O-Komponenten on-Chip
- Compilerunterstützung zur Code-Parallelisierung
- "open source", Verilog Quelltext unter SCSL verfügbar
- bisher kein Markterfolg

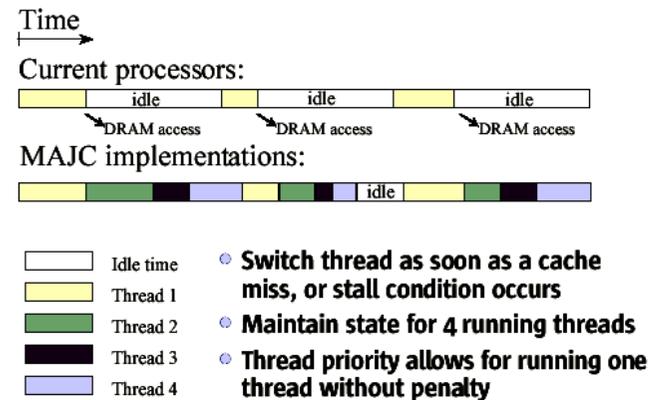
MAJC: VLIW

- dynamisches Befehlsformat, 32 .. 128 bit
- entweder einzelner Befehl oder Gruppen aus bis zu 4 Befehlen
- dadurch wesentlich kompakterer Code als andere VLIW



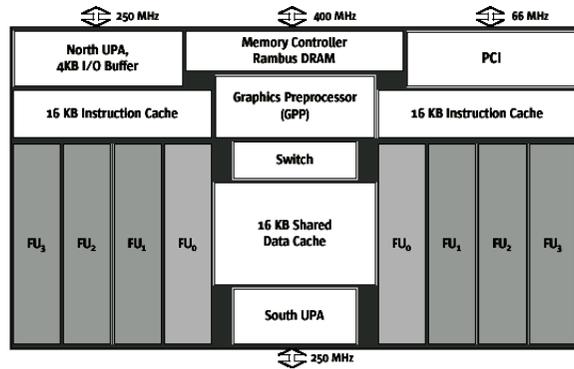
- vgl. Intel/HP IA64

MAJC: Multithreading



- Thread-Wechsel zum Verdecken langsamer Speicherzugriffe
- vgl. Intel "Jackson-Technology", Dissertation X. Fan (1994)

MAJC: Dual-Processor



MAJC-5200 Prozessor:

- 2 CPUs, je 4 FU (ALUs) auf einem Chip
- separate I-Caches, kleiner shared D-Cache
- 2 "UPA"-Interfaces zur Kaskadierung mehrerer Chips

MAJC: Performance

CATEGORY	APPLICATION	DESCRIPTION	PERFORMANCE
Image Processing	JPEG Encode	Baseline JPEG Encode of 8-bit sample images, JPEG 2000 test set	78 MB/s
	Image Segmentation	Segments an 8-bit sample image, text and pictures	365 MB/s
	2D FFT	1024*1024 8-bit per sample Image	30 frames/sec.
	Color Conversion	1024*1024 RGB -> YCbCr	7.34 ms (1 CPU)
Video	MPEG-2 decode	Main level, main profile, 5 Mbps interlaced sequence	> 2 streams
	H.263	CIF decode/encode, 15 fps, 128 kbs	6 decode 1 encode
Audio/Speech	AC-3 Decode	5.1 channels @ 384 kps	7.5% of 1 CPU
	G.723.1 Codec	Floating-point version of the voice codec @ 5.3kbps compressed data rate	100 channels
	G.729.A Codec	Floating-point version of the voice codec	100 channels

Piranha

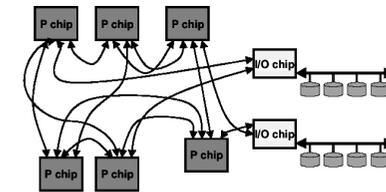


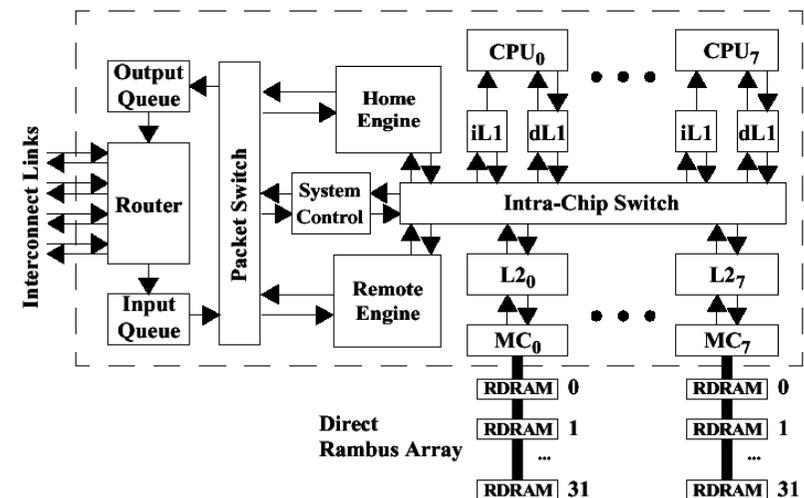
Figure 3. Example configuration for a Piranha system with six processing (8 CPUs each) and two I/O chips.

Forschungsprojekt von DEC/Compaq

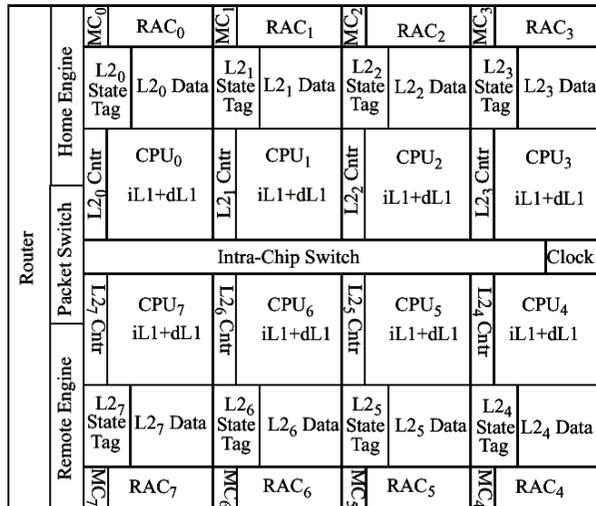
- 8 "einfache" Alpha-CPU's auf einem Chip
- 1 MB gemeinsamer L2-Cache
- 64 KB I-Cache, 64 KB D-Cache pro CPU
- 1 Rambus-Speicherinterface pro CPU, 1.6 GB / s
- Kaskadierung mehrerer Chips über direkte Links
- insgesamt 32 GB /s I/O-Bandbreite pro Chip
- separater Chip für I/O, u.a. mit PCI-Controller

(Barroso et al., Piranha: a scalable architecture based on single-chip multiprocessing, ISCA 2000)

Piranha: 8 CPUs on Chip



Piranha: Floorplan, Caches



Medientechnik | WS 2001 | 18.204

Piranha: Performance

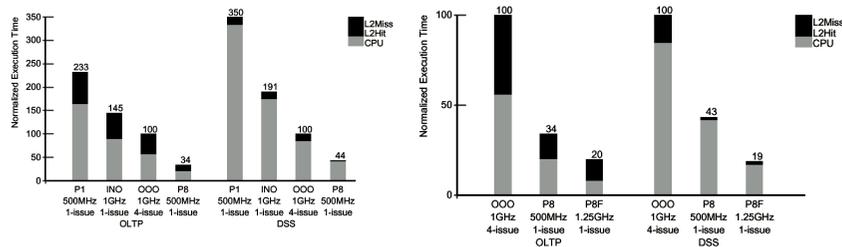


Figure 5. Estimated performance of a single-chip Piranha (8 CPUs/chip) versus a 1GHz out-of-order processor.

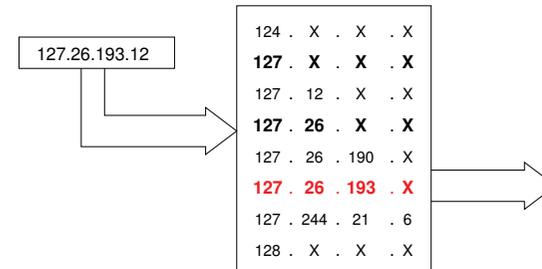
Figure 8. Performance potential of a full-custom Piranha chip for OLTP and DSS.

Piranha (1 CPU, 8 CPUs) im Vergleich mit Einzelprozessoren

- "INO": superskalärer Prozessor (21164, in order completion)
- "OOO": superskalärer Prozessor (21364, out of order completion)
- zwei Datenbank-Benchmarks simuliert: "OLTP" entspricht TPC-C, "DSS" entspricht TPC-D

Medientechnik | WS 2001 | 18.204

iFlow: Routing Processor



IP-Routing:

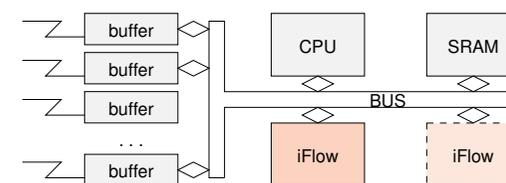
- Router liest Ziel-IP, sucht "longest matching prefix"
- Problem: Routing-Tabellen können sehr groß werden
- bis zu Millionen Anfragen pro Sekunde (Gigabit-Netzwerk)
- normaler Rechner zu langsam

=> spezielle "Netzwerkprozessoren" und -chips

(Beispiel iFLOW: www.siliconaccess.com, www.hotchips.org)

Medientechnik | WS 2001 | 18.204

iFlow: Konzept



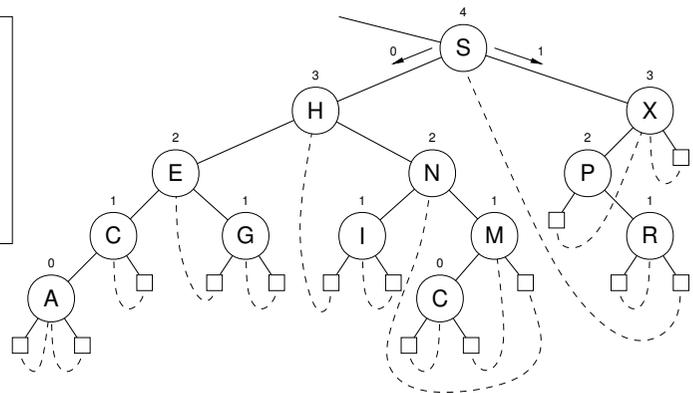
- übliche Router verwenden MAC-Chips, CPU, SRAM
- aber: zu langsam für Gigabit-Netzwerke
- Spezialchip zur Verwaltung der Routing-Tabellen
- pipelined-Lesezugriffe für das eigentliche Routing
- parallel dazu Schreibzugriffe zum Update der Tabellen
- massiv parallele VLSI-Realisierung
- parallele Tabellenzugriffe über 8000-bit Busse

Medientechnik | WS 2001 | 18.204

Patricia-Algorithmus

"practical algorithm to retrieve information coded in alphanumeric" (J.D.Morrison)

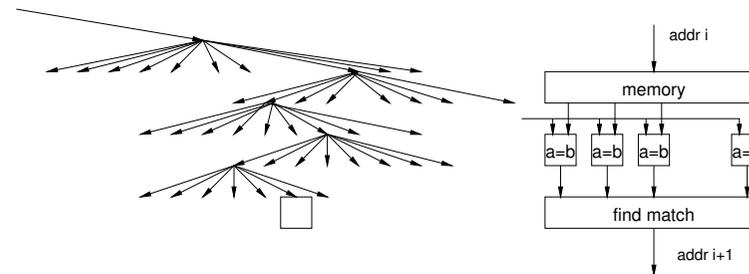
A	00001
S	10011
E	00101
R	10010
C	00011
H	01000
I	01001
N	01110
G	00111
X	11000
M	01101
P	10000
L	01100



- $O(\ln N)$ Operationen/Lookup

(Sedgewick, algorithms in C++, chapter 17)

iFlow: Parallel Lookup



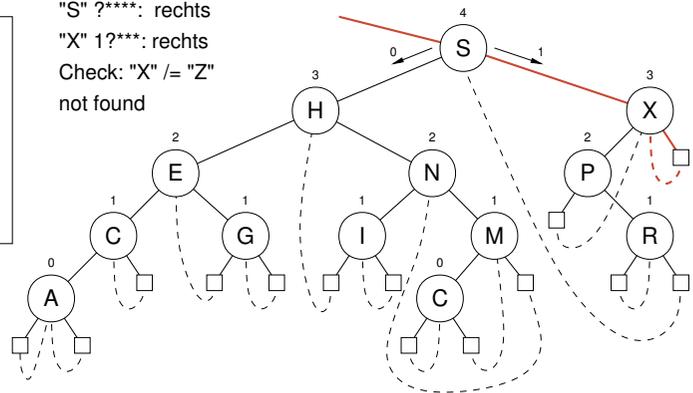
- Hardware-Realisierung des Patricia-Algorithmus
- Suchbaum: mehrfache Verzweigung, dafür nur 5 Stufen
- volles Pipelining, separate Speicherbereiche für jede Stufe
- Entscheidung über parallelen Vergleich von bis zu 512 Werten
- unterste Ebene enthält die Routing- und Verwaltungs-Daten

Patricia: Beispiel

Suche nach "z", 11010:

"S" 1?***: rechts
 "X" 1?***: rechts
 Check: "X" != "Z"
 not found

A	00001
S	10011
E	00101
R	10010
C	00011
H	01000
I	01001
N	01110
G	00111
X	11000
M	01101
P	10000
L	01100



(Sedgewick, algorithms in C++, chapter 17)

iFlow: Chiplayout

- externe Folie(n): iFlow Präsentation von Hotchips 12

CD, DVD: Agenda

- Grundlagen der CD-Technik
- CD-ROM
- CD-R, CD-RW
- DVD und Video
- DVD-R, DVD-RAM
- ISO-9660 Dateisystem
- UDF / Packet-Writing



Medientechnik | WS 2001 | 18.204

CD/DVD: Literatur

www.disctrionics.co.uk/

www.fadden.com/cdrfaq

www.dvddemystified.com/dvdfaq.html

www.ping.be/~pin11466/formtxt.html

www.unik.no/~robert/hifi/dvd/

www.phoenix.com/techs/specs.html

(übersichtliche Kurzbeschreibungen zu CD/DVD)

(alles rund ums Thema CD-R und CD-RW)

(DVD-FAQ, viele mirrors weltweit)

(schöne Übersicht)

(umfangreiche Link-Sammlung zu DVD)

(El Torito Format für bootfähige CDs)

diverse c't Artikel:

02/93 178f CD "color books" und Dateiformate

DVD: 23/99 S.100f

diverse Standards, insbesondere ECMA-xxx (frei verfügbar), www.ecma.ch

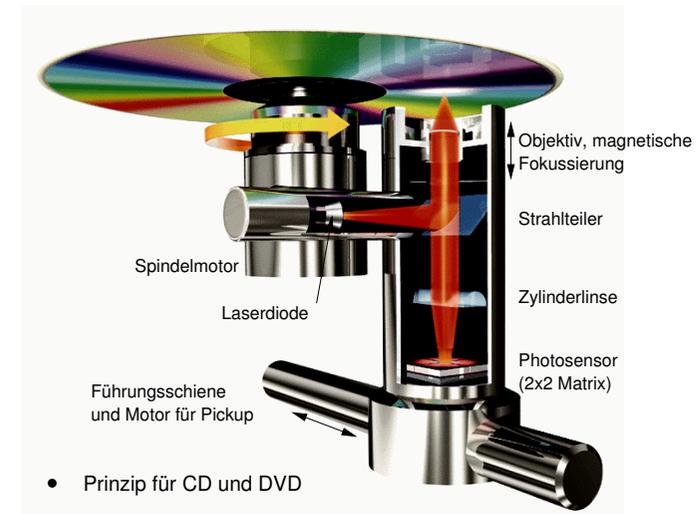
ECMA-119 ISO-9660 Dateisystem

ECMA-267 DVD-ROM Spezifikation

SCSI-3 MMC SCSI MultiMedia Command Set

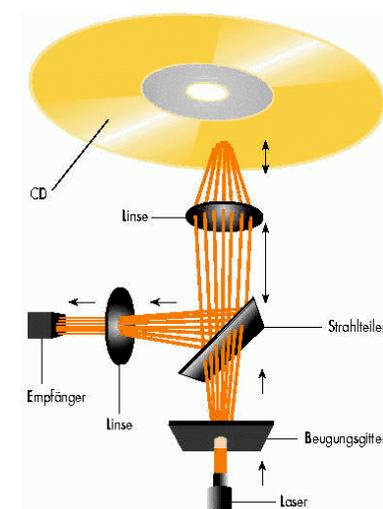
Medientechnik | WS 2001 | 18.204

CD: Aufbau eines Players



Medientechnik | WS 2001 | 18.204

CD: Multibeam-Technik

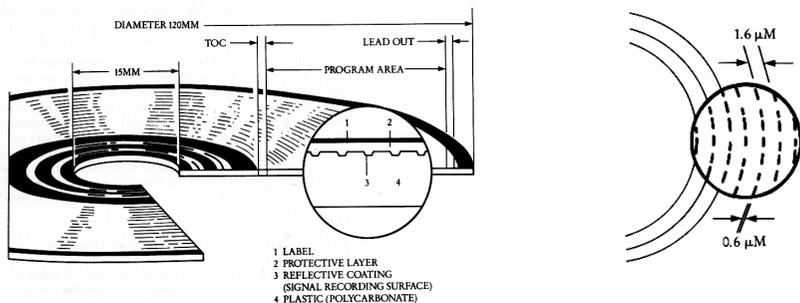


- Strahlteiler
- komplexer Empfänger mit mehreren Photodioden
- liest mehrere Spuren gleichzeitig
- statt höherer Drehzahl

(Testbericht in ct 08/99 74)

Medientechnik | WS 2001 | 18.204

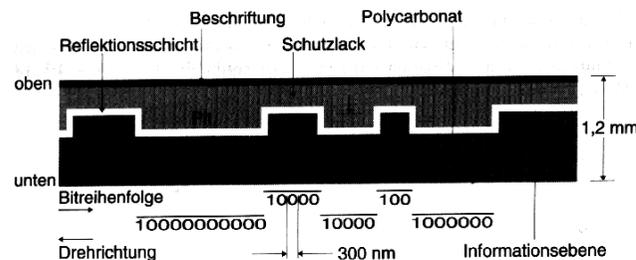
CD: Prinzip



- Polycarbonaträger, 12cm Durchmesser
- eingeprägte Vertiefungen ("pits") bilden die Daten
- spiralförmige Datenspur, 1,6µm Abstand, ca. 16000 Windungen
- Fertigungsmängel fest eingeplant => leistungsfähige Fehlerkorrektur

(Abbn. aus: CD-ROM - The new Papyrus)

CD: Schichtaufbau



- Polycarbonatschicht ~ 1.2 mm
- Größe der Pits / Lands ~ 1.0 µm
- Interpretation: Land = 0, Pit = 1, Wechsel Land/Pit = 1
- Achtung: Kratzer oben zerstören die Daten

CD: Reflexion

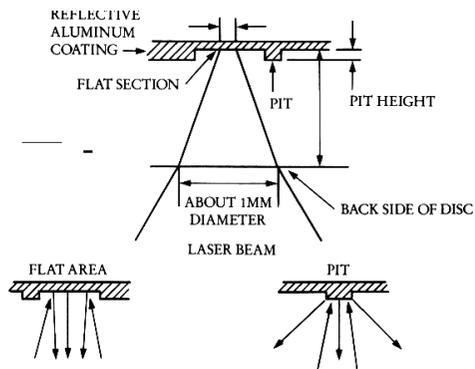
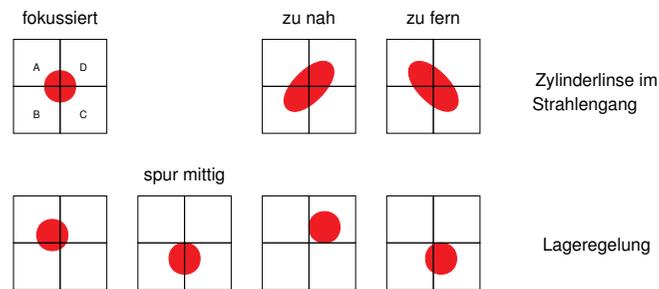


Figure 4. Relation between pits and photodetector output.

- Lands reflektieren das Laserlicht
- Pits streuen das Laserlicht

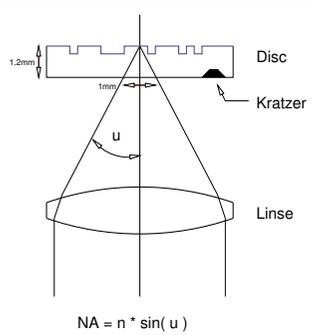
CD: Fokussierung, Spurregelung, ...



Sensorfeld mit 2x2 Photodioden zur Regelung:

- Fokussierung: aus Differenzsignal $(A+C) - (B+D)$
- Spurregelung: aus Differenzsignal $(A+B) - (C+D)$
- Nutzsignal: Land/Pit-Übergänge aus Summensignal
- Drehzahl: aus Taktfrequenz des Nutzsignals

CD: numerische Apertur



Brechungsindex 'n' eines Materials:

- Vakuum = 1
- Luft ~ 1
- Diamant = 2.4

NA := Maß für Auflösungsvermögen des Objektivs

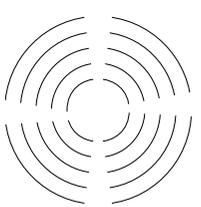
	NA	Öffnungswinkel
CD	0.45	24 .. Grad
DVD	0.5 .. 0.6	.. 37 Grad

- Auflösungsvermögen wie bei Mikroskopen (!)
- asphärische Linsen notwendig
- Kratzer/Staub auf der Oberfläche stören kaum

CD: CAV vs. CLV

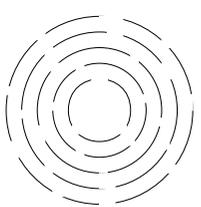
Constant Angular Velocity

(Floppy, aktuelle CD-ROM Laufwerke)



Constant Linear Velocity

(Audio/Video CD, DVD)



- Audiodaten: konstante Datenrate sinnvoll: CLV
- Drehzahl: innen hoch, außen langsam
- Angabe "48x"-Laufwerk: 48x Datenrate (CLV) der Audio-CD
- CAV erlaubt Spurwechsel ohne Drehzahländerung
- aktuelle CD-ROMs: CAV soweit per Daten/Fehlerrate möglich

Programme "CD-Bremse", "CD-Bänschmaak": home.t-online.de/home/Joern.Fiebelkorn/

CD: EFM

Eight-to-Fourteen Modulation:

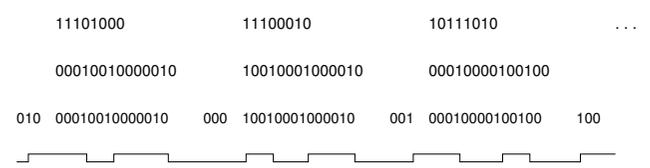
- selbsttaktende Aufzeichnung, NRZI
- minimal 2 Nullen, maximal 11 Nullen zwischen Einsen

data bits	channel bits
0000 0000	01001000100000
0000 0001	10000100000000
0000 0010	10010000100000
0000 0011	10001000100000
0000 0100	01000100000000
0000 0101	00000100010000
...	... via lookup table

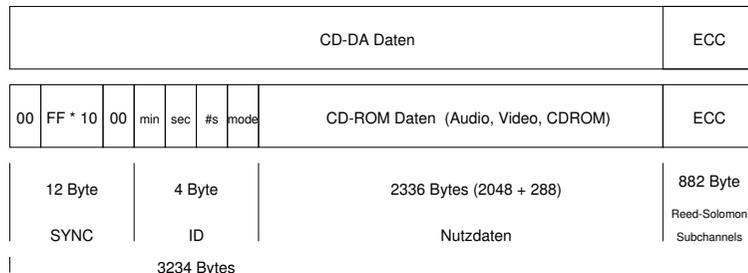
- zusätzlich 3 "Merge-Bits" zwischen zwei Codeworten einfügen
- eigentlich 8-17 Modulation
- DVD verwendet verbessertes 8-16 Verfahren

CD: Kodierung der Daten:

- Nutzdaten
- Nutzdaten in Frames einteilen
- Reed-Solomon Checksumme an Frames anfügen
- 14-bit EFM-Daten aus 8-Bit Nutzdaten
- 17-bit EFM mit Merge-Bits
- 17-bit EFM, Sync-Pattern anfügen
- Pits and Lands

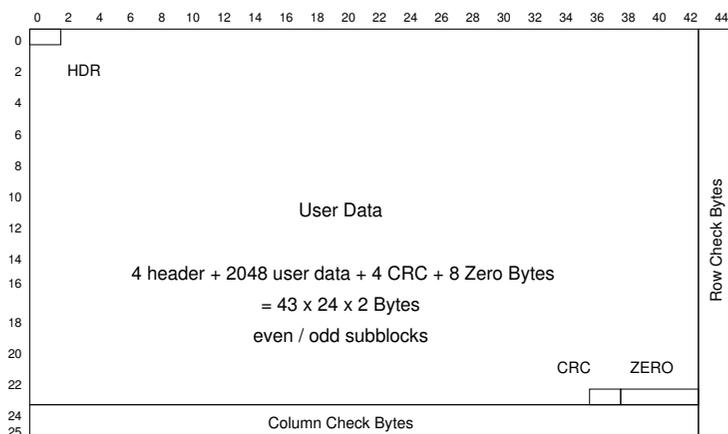


CD: Sektoren (Frames)



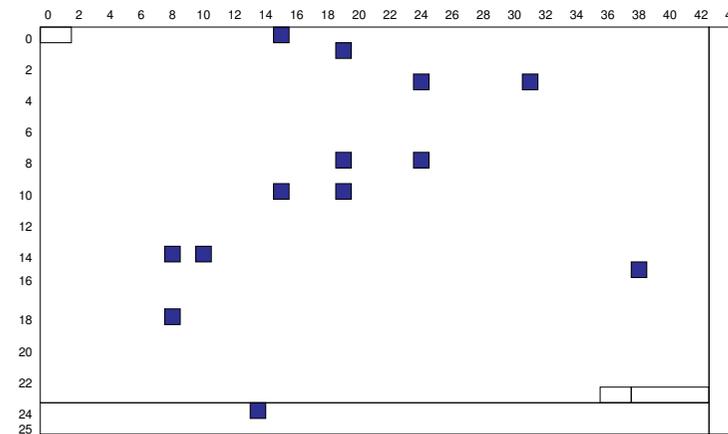
- 75 Sektoren pro Sekunde, 0 .. 74
- Numerierung per (minute, second, sector)
- 60 Minuten: 270.000 Frames (553 MB)
- 74 Minuten: 333.000 Frames (682 MB)

CD-ROM LEC Reed-Solomon Code



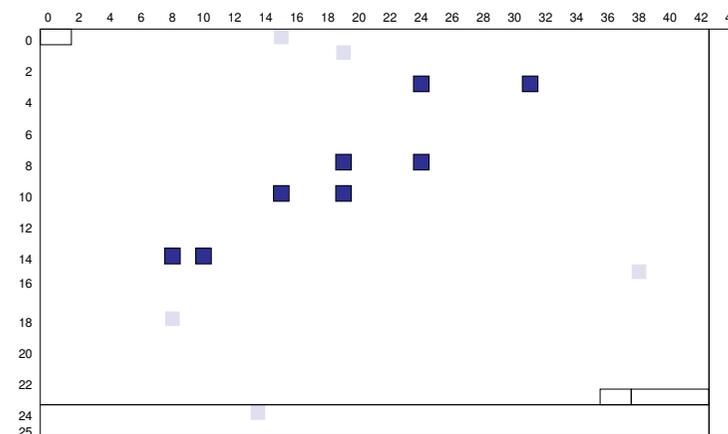
- Layered Error Correction (CD-ROM Mode 1)

CD: Reed-Solomon Code



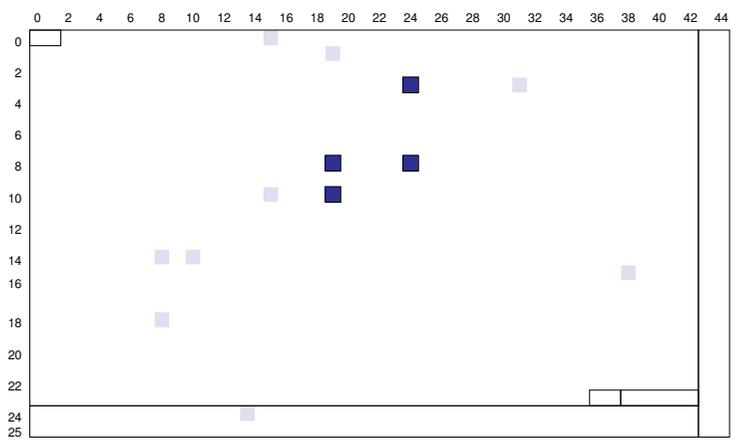
- Beispiel: Fehler vor der Korrektur

CD: Reed-Solomon Code



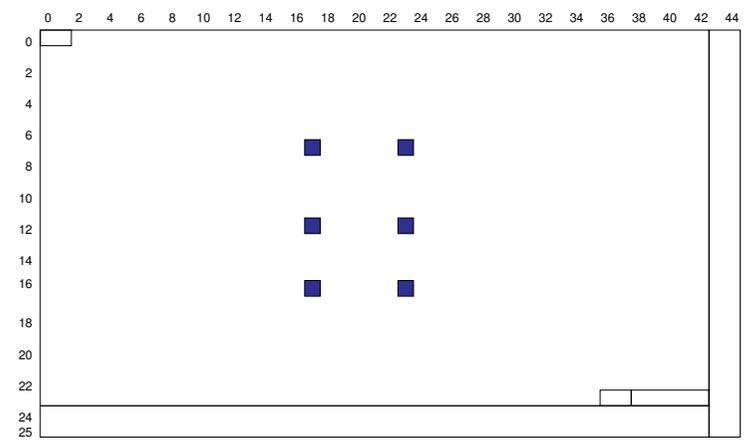
- erster Schritt: alle Einzelfehler in Zeilen korrigiert

CD: Reed-Solomon Code



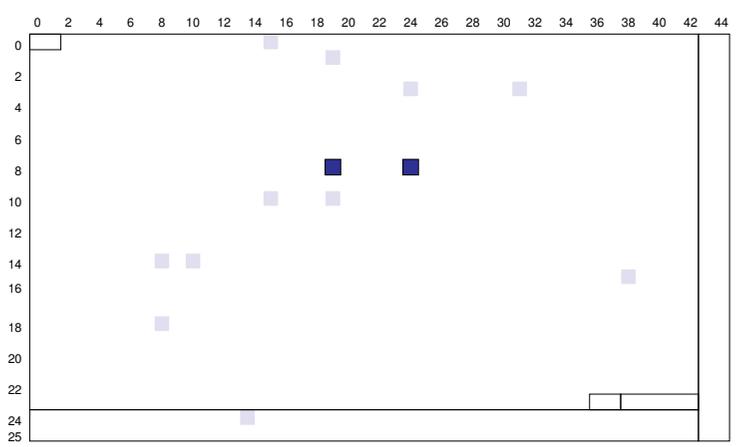
- zweiter Schritt: alle Einzelfehler in Spalten korrigiert

CD: Reed-Solomon Code



- keine Einzelfehler in Zeilen/Spalten, trotzdem korrigierbar
- Bitfehlerrate CD-ROM besser als 1E-13

CD: Reed-Solomon Code



- dritter Schritt: wieder die Zeilen korrigiert, dann wieder die Spalten

CD: "colors"

Red-Book CD-Audio	Mixed Mode	(1980)
Yellow Book CD-ROM		(1984 ?)
Mode 1 (zus. Fehlerkorrektur)		
ISO 9660 & Co Dateisysteme		
Mode 2 (ohne zus. ECC)		
CD-ROM XA		
Form 1	Form 2	(1989)
Green Book CD-I		
Orange Book CD-R/RW	Photo CD	(1990)
Blue Book CD-Extra		
White Book Video-CD		plus diverse Mischformate . . .

CD: Datenformate Daten / Audio

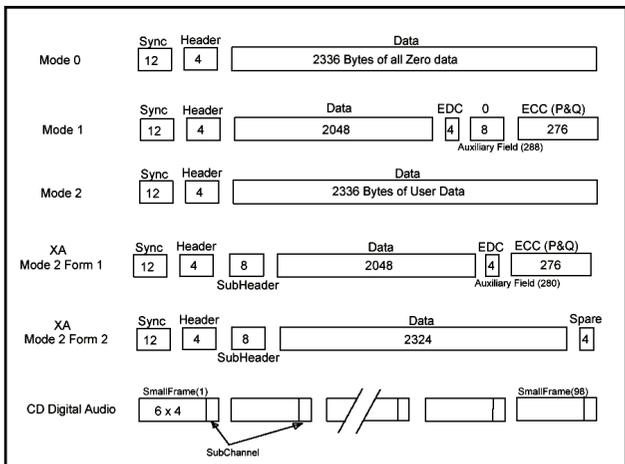
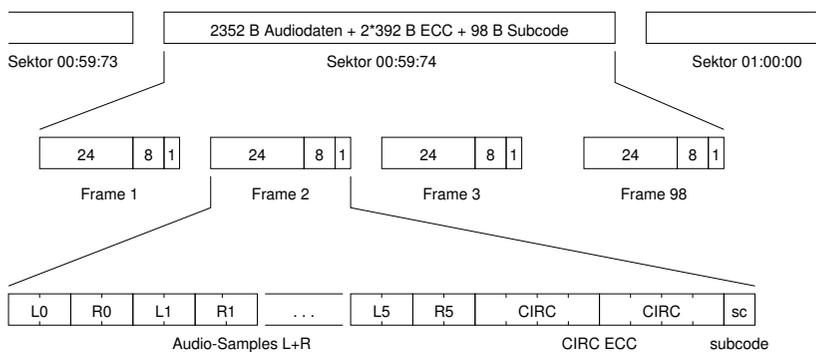


Figure 2 - CD-ROM Sector Formats

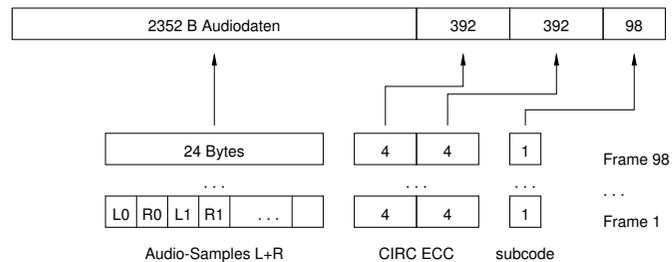
- Tradeoff: Kapazität vs. Fehlerkorrektur

CD: Audioformat, Sektoren, Frames



- 75 Sektoren pro Sekunde
- 98 Frames a 24 Bytes (+ECC) pro Sektor
- je 1 Byte Subcode pro Sektor

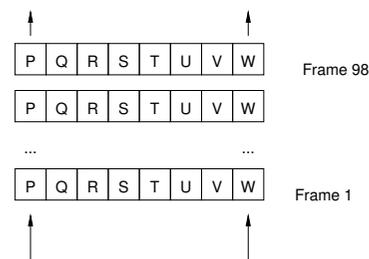
CD: Audioformat, konzeptionell



- 98 Frames a 24 Bytes pro Sektor
- 24 Bytes: je 6 Samples linker/rechter Kanal
- ein Byte Subcode pro Frame
- $75/s * 2352B = 44100/s * 16b * 2 / 8 = 176 KB/s$
- ECC korrigiert Bursts bis zu 7000 fehlenden Bits

CD: Subchannels

Subchannels / Subcode:

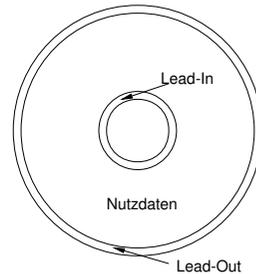


- 1 Byte Subcode pro Frame
- P markiert Start eines Tracks
- Q für Inhaltsverzeichnis der CD (TOC)
- R..W abhängig vom Format
z.B. konstant Null bei CD-ROM
Nutzung für CD-Text
Datenrate: $75 * 98 \text{ bit/s} = 918 \text{ B/s}$

CD: Lead In / Lead Out

Lead-In:

- spezieller Bereich am Anfang (innen) der CD
- Nutzdaten konstant Null
- Q-Subcode enthält das Inhaltsverzeichnis
- bis zu 99 Tracks erlaubt: ca. 9 MByte
- Lead-Out kennzeichnet Ende der CD
- Nutzdaten und Subcodes konstant Null
- Multisession-CDs:
je 1 Lead-In/Out Bereich pro Session
plus Master-Lead In / Lead-Out



CD: Photo-CD

- Kodak / Philips 1993
- basiert auf CD-ROM/XA
- Filme (Kleinbild) werden mit 2000 dpi gescannt
- Auflösung 3072x2048 Pixel (optional 6144x4096)
- bis ca. 100 Photos
- multisession-Format (erlaubt mehrere Filme)
- proprietäres Datenformat
- mehrere Auflösungen: 192x128 bis 3072x2048 Pixel
- vergleichsweise hohe Kosten
- Markterfolg nur im Profi-Bereich
- neuer Versuch als "Picture-CD" (mit Intel/Adobe Software)
1024x1536 Pixel, JPEG-Format



CD: Mixed Mode CD

- kombiniert Audio-Tracks mit Daten/Video-Tracks
- Audio CD-Player erwarten nur Audio-Tracks:
 - ältere Player lesen Daten als Audio
 - Knacksen (Vorsicht: extreme Lautstärke)
 - neuere Player schalten den Track stumm
 - alternative Anordnung: Datentrack(s) ganz hinten
- "kranke" Block- bzw. Min/Sek/Frame-Adressierung
- (siehe Beispiel auf nächster Folie)
- wird von allen aktuellen PC-Laufwerken unterstützt

CD: Mixed Mode CD (Beispiel)

Block Description	Logical Address (Decimal)	Track Relative logical address	Absolute M/S/F Address ¹	Track / Index	Track Relative M/S/F Address	Sector Contains Info or Pause	Mode Audio or Data	CD Data Mode ²
Lead-in Area	---	---	---	0/-	---	---	Audio	---
Pre-gap	---	---	00/00/00	1/0	00/02/00 ⁷	Pause	Data	Null
1st Track data	0000 ⁴	0	00/02/00 ⁵	1/1	00/00/00	Info	Data	L-EC
2nd track data	6000 ⁴	0	01/22/00 ⁵	2/1	00/00/00	Info	Data	L-EC
	7500	1500	01/42/00	2/2	00/20/00	Info	Data	L-EC
Post gap	9000	3000	02/02/00	2/3	00/40/00	Pause	Data	Null
Pause-silence	9150	-150 ⁶	02/04/00	3/0	00/02/00 ⁷	Pause	Audio	---
3rd track audio	9300 ⁸	0	02/04/00 ⁹	3/1	00/00/00	Info	Audio	---
	1400	2250	02/34/00	3/2	00/03/00	Info	Audio	---
4th track audio	21975 ⁸	0	04/53/00 ⁹	4/1	00/00/00	Info	Audio	---
Pre-gap part 1	30000	-225 ⁶	06/40/00	5/0	00/03/00 ⁷	Pause	Audio	---
Pre-gap part 2	300075	-150	06/41/00	5/0	00/02/00 ⁷	Pause	Data	Null
5th track data	30225	0	06/43/00	5/1	00/00/00	Info	Data	L-EC
Last Information	263999 ¹⁰	233 774	58/39/74	5/1	51/56/74	Info	Data	L-EC
Post-gap	---	233 775	58/40/00	5/2	51/57/00	Pause	Data	Null
Lead-out area	264000 ¹¹	0	58/42/00	AA/ ¹³	00/00/00	Pause	Audio	---

CD: LBA/MSF Umrechnung

Table 207- LBA to MSF translation

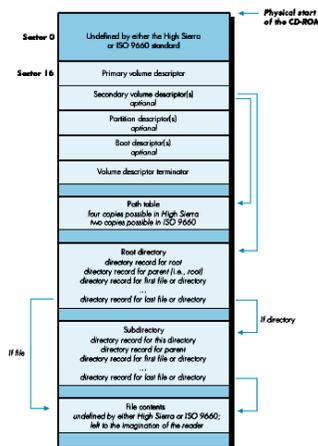
Condition	Formulac
$-150 \leq LBA \leq 404849$	$M = IP\left(\frac{LBA + 150}{60 \cdot 75}\right)$ $S = IP\left(\frac{LBA + 150 - M \cdot 60 \cdot 75}{75}\right)$ $F = IP(LBA + 150 - M \cdot 60 \cdot 75 - S \cdot 75)$
$-45150 \leq LBA \leq -151$	$M = IP\left(\frac{LBA + 450150}{60 \cdot 75}\right)$ $S = IP\left(\frac{LBA + 450150 - M \cdot 60 \cdot 75}{75}\right)$ $F = IP(LBA + 450150 - M \cdot 60 \cdot 75 - S \cdot 75)$
$00/00/00 \leq MSF \leq 89/59/74$	$LBA = (M \cdot 60 + S) \cdot 75 + F - 150$
$90/00/00 \leq MSF \leq 99/59/74$	$LBA = (M \cdot 60 + S) \cdot 75 + F - 450150$

- logische Blockadresse vs. Minute/Sekunde/Frame

CD-ROM: ISO 9660

Standard-Dateiformat für CD-ROMs

- Daten starten in Sektor 16 (00:02:16)
- DOS-kompatibel (FAT)
- Dateinamen mit 8+3 Zeichen
- bis zu 8 Verzeichnisebenen
- Level-2 erlaubt Namen bis 32 Zeichen
- plattformunabhängig
- Dateien müssen linear vorliegen
- keine späteren Änderungen möglich
- Level-3 erlaubt fragmentierte Dateien



CD-ROM: Joliet

- Microsoft-Erweiterung von ISO-9660
- erlaubt Windows95-Dateinamen
- Namen bis 64 Zeichen, inklusive Sonderzeichen

- integriert in Windows 9x/2K
- ebenfalls in neueren Linux-Versionen

www-plateau.cs.berkeley.edu/people/chaffee/joliet.html

CD-ROM: Rock-Ridge und andere

Rock-Ridge:

- Erweiterung von ISO-9660 für Unix-Systeme
- erlaubt lange Dateinamen
- Unix-style Datei-Attribute (owner, permissions)
- symbolische Links
- abwärtskompatibel (ISO-9660 Systeme sehen die 8+3 Daten)
- ftp.yimi.com/pub/rockridge/

Macintosh HFS:

- CD-ROM Format mit Apple's hierarchical file system
- völlig inkompatibel mit ISO-9660
- benötigt entsprechende Treiber

CD-ROM: El Torito

bootfähige CD-ROMs?

- "El-Torito" Spezifikation von Phoenix und IBM (1994)
- Grundidee: BIOS ersetzt Laufwerk A: durch die CD-ROM
- basiert auf ISO-9660 Dateisystem
- Sektor 16 wie gehabt für Primary Volume Descriptor
- Sektor 17 als Boot Record Volume Descriptor
- erlaubt mehrere Boot-Sektoren pro CD
- Zugriff über BIOS/DOS INT-13 Schnittstelle
- CD-ROM kann als Live-Filesystem genutzt werden

Medientechnik | WS 2001 | 18.204

CD-ROM: El Torito

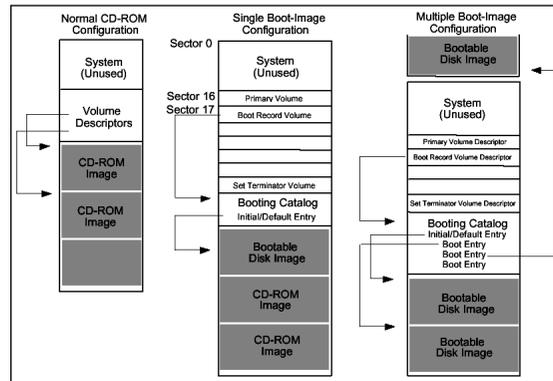
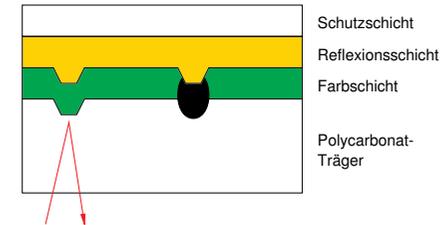


Figure 1. Three types of CD-ROM configuration:

1. The Normal CD-ROM configuration is not bootable, uses Root Directory and CD-ROM drivers to access CD-ROM images.
2. A BIOS with Single Boot-Image capability accesses the Initial/Default Entry to access a single bootable disk image. After loading the operating system, the system can revert to standard CD-ROM drivers and the Root Directory to access CD-ROM images.
3. A BIOS with Multiple Boot-Image capability can access any one of a number of Bootable Disk Images listed in the Booting Catalog. After loading the operating system, the system can access other items in the disk image with standard INT 13 calls or return to normal access of CD-ROM images using CD-ROM drivers and the Root Directory.

Medientechnik | WS 2001 | 18.204

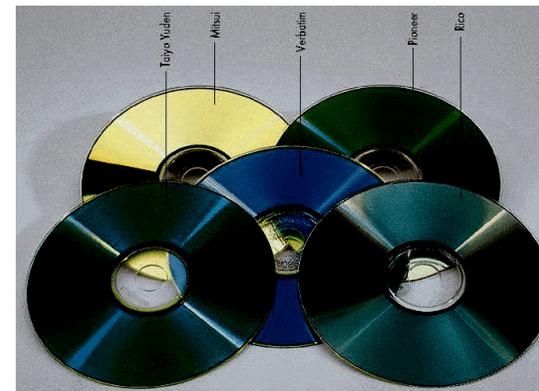
CD-R: Prinzip



- mechanische Prägung (Pits/Lands) nicht praktikabel
- statt dessen: Farbstoff durch Laserimpuls zerstören
- etwas andere Reflexionsdaten als CD
- Spurführung des Pickups erfordert Daten:
=> Rohlinge enthalten vorbereitete Spiralspur (siehe DVD pre-groove)

Medientechnik | WS 2001 | 18.204

CD-R: Rohlinge, Farbstoffe

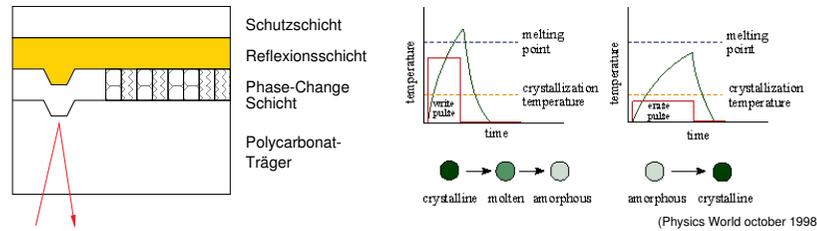


Cyanin Phthalocyanin Metallkomplex-Azo-Farbstoff

- diverse Farbstoffe, aber Haltbarkeit, Schreibeigenschaften ähnlich

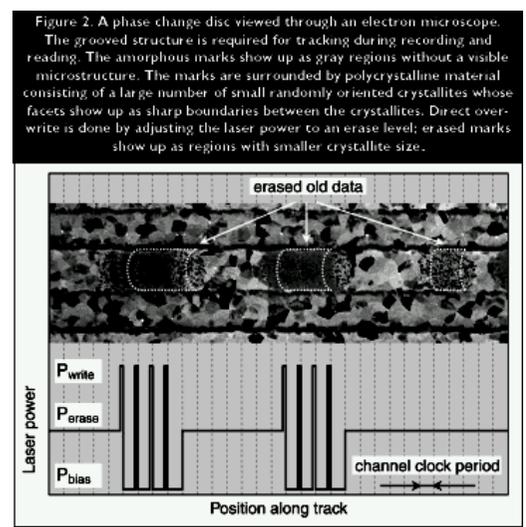
Medientechnik | WS 2001 | 18.204

CD-RW: Prinzip

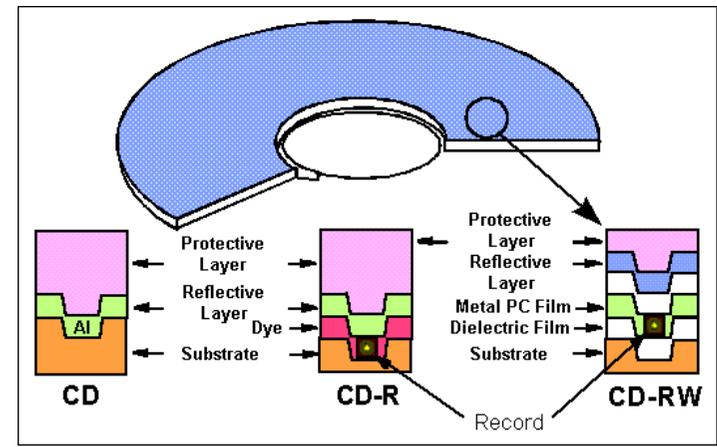


- Phase-Change Verfahren für wiederbeschreibbare CDs
- Material mit kristalliner / amorpher Struktur
- deutlich kleinere Reflexionsänderung als bei CD/CDR
- Umschalten durch schwache/starke Laserimpulse
- schnelle Abkühlung: amorph, langsame Abkühlung: kristallin
- bis zu 100.000 Mal wiederbeschreibbar (theoretisch)

CD-RW: amorph / polykristallin

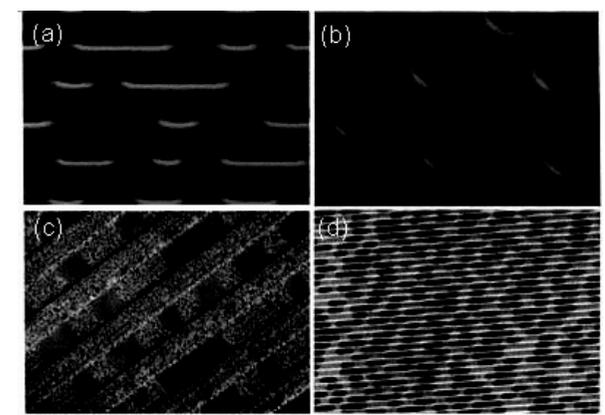


CD-RW: Aufbau CD / CDR / CDRW



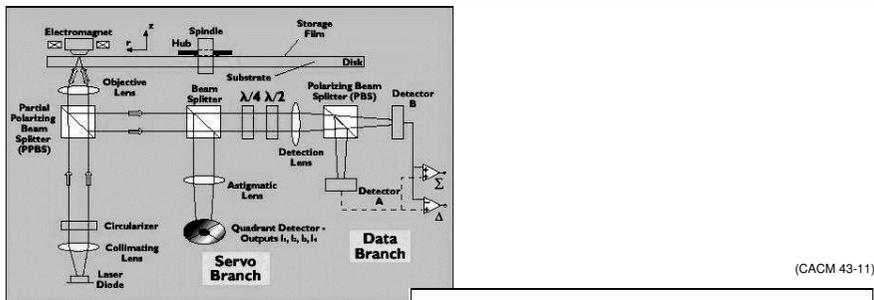
- CD-Pressung "parallel"
- CD-R / CD-RW Schreiben sequentiell, entlang der Rohspur

Pits: CD, CD-R, CD-RW, MO

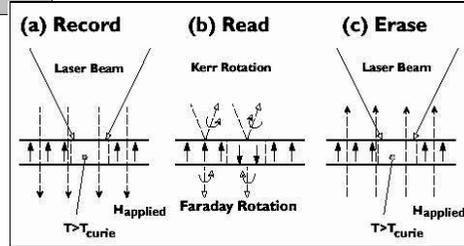


- a) CD (Pits gepreßt)
- b) CD-R (Pits gebrannt)
- c) CD-RW (amorph/kristallin)
- d) MO (Kerr-Effekt)

magneto-optische Verfahren (MO)



- Mechanik wie bei CD
- zusätzlich Magnet (gegenüber Pickup)
- Intensitätsdifferenz durch Polarisation



Medientechnik | WS 2001 | 18.204

CD-R: erweitertes Lead-In

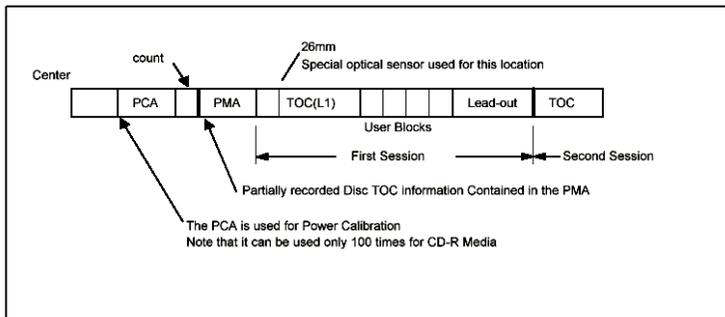


Figure 3 - CD-R/RW Disc Layout

- erweiterte Lead-In Zone (weiter innen als normale CD)
- u.a. Kalibrierung der Laserintensität beim Schreiben
- Audio/Datenformat unverändert

Medientechnik | WS 2001 | 18.204

CD-R: "Überbrennen"

- Länge der Rohspur definiert die Kapazität der CD-R/RW
- spezielle Rohlinge (80 Min) mit engerer Rohspur

"Überbrennen":

- angegebene Kapazität enthält >90 Sekunden Lead-Out plus einige Sekunden Reserve
- verkürztes Lead-Out erlaubt mehr Daten
- muß von Brenner und Software unterstützt werden (zB. www.feurio.com)
- evtl. Probleme mit älteren / abgenutzten Playern
- alternativ für Audio: Daten minimal stauchen

Medientechnik | WS 2001 | 18.204

CD-R: Buffer-Underrun

- sequentielles Schreiben der CD-R:
- => Brenner benötigt kontinuierlichen Datenstrom
typische Puffergröße 2..4 MB

Problem Buffer-Underrun:

- CD-R entspricht nicht mehr den Normen
- Rohling defekt (CD-R) / neu formatieren (CD-RW)

www.burn-proof.com:

- Brenner rechtzeitig (kontrolliert) stoppen
- Position auf der CD-R merken (Spur, Position 100 µm)
- neu aufsetzen, sobald Daten verfügbar
- Fehlerkorrektur beseitigt die Lücke ("burst error")
- wird von einigen neuen Brennern unterstützt

Medientechnik | WS 2001 | 18.204

CD-R: Audio-Grabbing

digitales Auslesen von CD-DA:

- optimal mit Audio-Playern (Digitalausgang, 1X Speed)
- Digitalausgänge an CD-ROMs selten / oft fehlerhaft

"Packet"-Interface problematisch:

- in alten Laufwerken schlecht implementiert
- Audio-Format hat keine fortlaufenden Sektor-IDs
- mm:ss:ff-Marken: ff-Werte fehlen manchmal
- Packet vs. Streaming: Probleme beim Wiederaufsetzen
- nur einfache Fehlerkorrektur, kein LEC

=> gutes Laufwerk notwendig

=> mehrfaches Lesen / Korrelation der Daten (cdparanoia)

CD: Audio Grabbing via SCSI3 MMC

Table 95 - CD-DA (Digital Audio) Data Block Format

Bit	7	6	5	4	3	2	1	0
0	Left Channel (Lower Byte)							(LSB)
1	(MSB)	Left Channel (Upper Byte)						
2	Right Channel (Lower Byte)							(LSB)
3	(MSB)	Right Channel (Upper Byte)						
2348	Left Channel (Lower Byte)							(LSB)
2349	(MSB)	Left Channel (Upper Byte)						
2350	Right Channel (Lower Byte)							(LSB)
2351	(MSB)	Right Channel (Upper Byte)						

If the CD Drive does not support the CD-DA Stream-Is-Accurate capability, See Table 230 - CD Capabilities and Mechanical Status Page, then the digital audio data must be read as a continuous stream. If while streaming the drive must stop, there will be a non recoverable error generated READ ERROR - LOSS OF STREAMING. This is due to the 1 second uncertainty of the address. (i.e. there is no header in CD-DA data). Reissuing the command may not return exactly the same data as the previous try. When the drive supports the stream accurate capability, there will be no error, only some time delay for rotational latency.

UDF: Dateisystem

- CDR Medien sind nur einmal beschreibbar
 - ISO-9660 erwartet TOC und Directories an fester Position
- => spätere Änderungen unmöglich

UDF-Dateisystem: "universal disk filesystem"

- basiert auf ISO 9660
- aber erweitertes, flexibleres Dateisystem
- "virtual allocation tables"
- gültiges Directory jeweils im letzten geschriebenen Block
- dort Verweise auf Dateien und ältere Directory-Blöcke
- keine Beschränkung der Verzeichnis-Schachtelungstiefe
- Finalisieren der CD erzeugt volles ISO 9660 Dateisystem
- www.osta.org

UDF: Packet Writing

- CDR Medien sind nur einmal beschreibbar

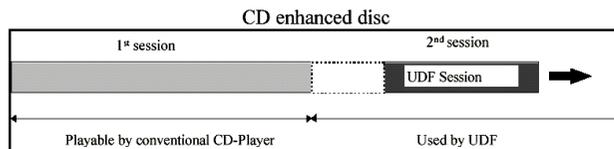
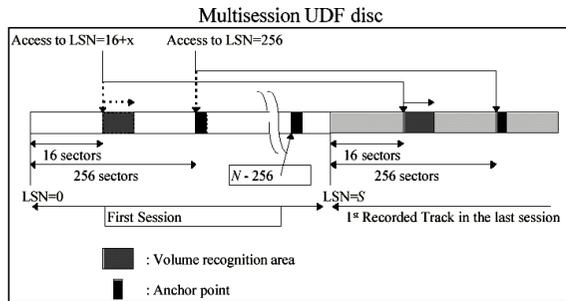
UDF-Packet Writing:

- Dateien in einzelnen kleinen Paketen schreiben
- zunächst ohne TOC im Lead-In

"virtual allocation tables":

- gültiges Directory jeweils im letzten geschriebenen Block
- dort Verweise auf Dateien und ältere Directory-Blöcke
- Dateien können immer noch nicht gelöscht werden
- neues Directory ohne Verweis auf gelöschte Datei schreiben
- Datei modifiziert:
- neue Datei schreiben, neues Directory schreiben

UDF: Multisession / enhanced disks



DVD: Konzept und Anforderungen

Anforderungen für DVD-Video:

- 135 Minuten Spieldauer pro Seite
- bessere Auflösung als die Laserdisc
- Surround-Audio in CD-Qualität
- Audiospuren für 3-5 Sprachen
- Untertitel in mehreren Sprachen
- diverse Bildformate (Letterbox, Pan, Widescreen)
- Interaktion wie bei Video-CDs
- Jugendschutz
- Kopierschutz
- CD-kompatibel
- Herstellungskosten ähnlich wie CDs

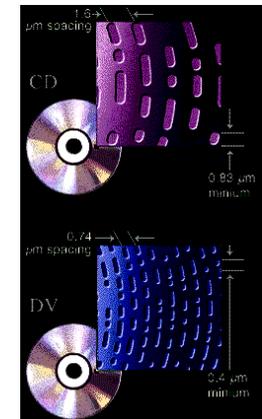
MPEG-2, AC3:
=> ca. 6 Mb/s
=> 4-5 GB / Seite

DVD: Verbesserungen

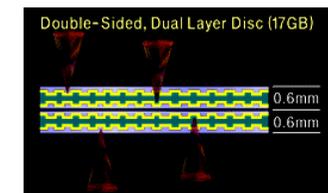
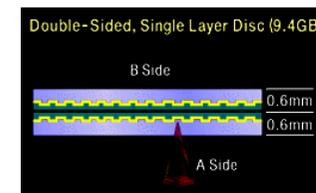
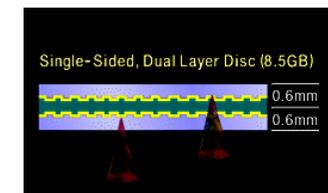
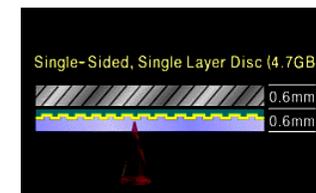
höhere Kapazität der DVD gegenüber der CD:

- kleinere Pits, kleinerer Spurbestand
- veränderte Header-Strukturen
- weniger Parity-Bits
- Weglassen der Subcodes
- 2048-Byte Sektoren
- kleinerer nicht-genutzten Innenteil (Radius CD 25 mm, DVD 24 mm)

=>	DVD-5	single layer	4.4 GB	6.7x
	DVD-9	double layer	8.0 GB	12.3x
	DVD-10	double side	8.8 GB	13.5x
	DVD-18	DS / DL	15.9 GB	24.4x CD-ROM



DVD: 4 Formate



- single/double side
- single/dual layer (äußere Schicht dabei halbdurchlässig)

DVD: Datenformat (Sektorformat)

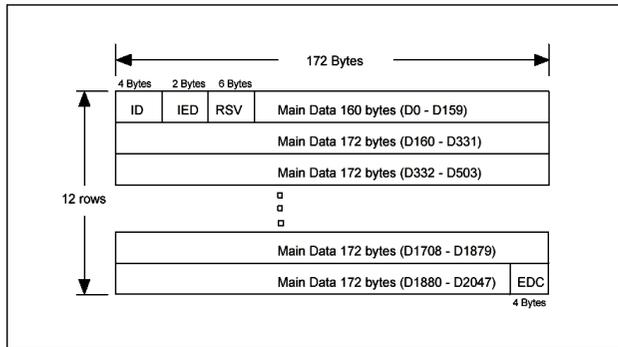


Figure 13 - Sector Layout

- eindeutige Block-ID, 4-Byte layered ECC
- Sektor enthält 2 KByte Nutzdaten
- vergleiche CDROM

DVD: Datenformat (Header)

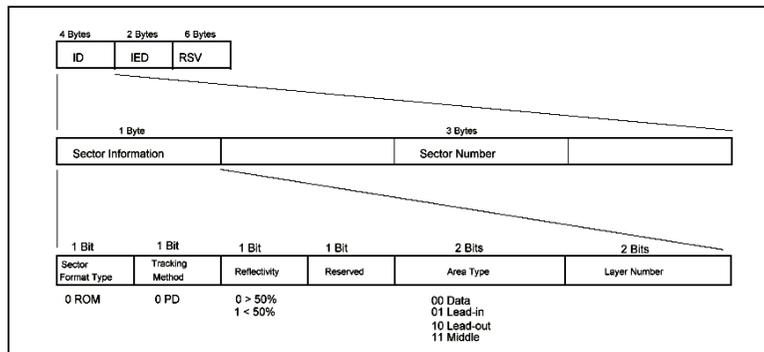


Figure 11 - Header Layout

- vollständige Information für jeden einzelnen Block
- 2-bit layer number: Seite 1/2, außen/innen

DVD: Sektoranordnung dual-layer

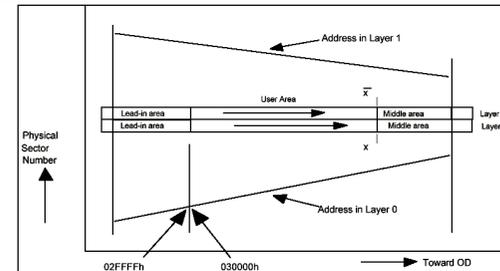


Figure 10 - Opposite Track Path Description

Layer-Umschaltung durch Fokussierung (schnell)

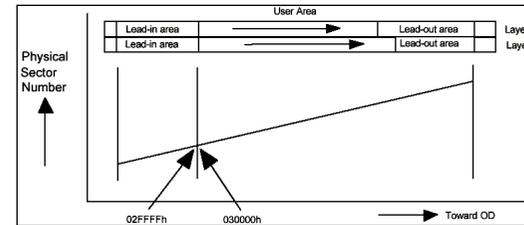
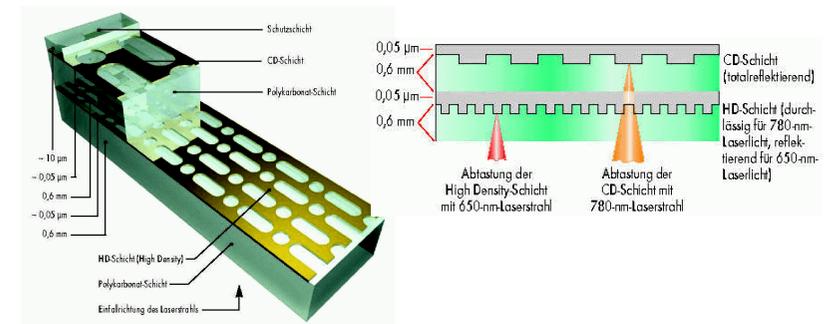


Figure 9 - Parallel Track Path Description

Layer-Umschaltung erfordert Kopfneupositionierung

DVD: Dual-Layer Audio CD



- Kompatibilität mit Audio-CDs
- zweite (DVD)-Schicht mit Stereo "bitstream", 2.8 Mb/s
- vgl. DVD-Audio

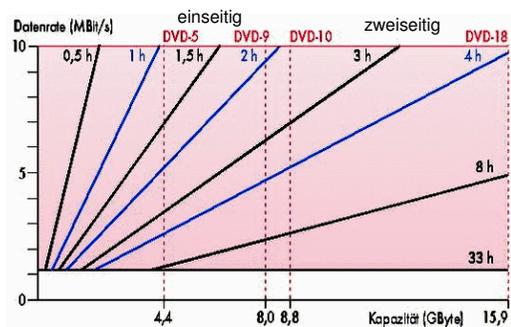
DVD-Audio

- Spezifikation für DVD-Audio seit Q1/1999
- nutzt die DVD-5 (4.7 GB)

diverse Audioformate werden unterstützt:

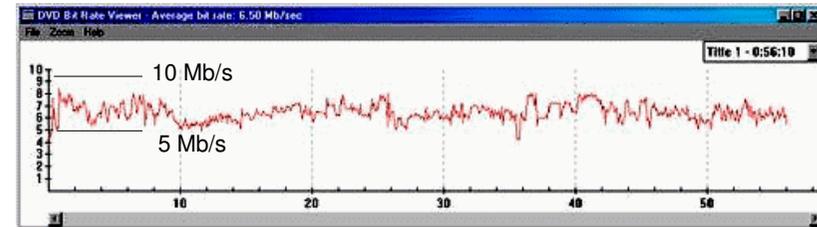
- Abtastraten 44.1 / 48 / 88.2 / 96 KHz
- Quantisierung mit 12 / 20 / 24 bit
- mindestens 74 min. Spieldauer für alle Modi
- 16 bit, 44.1 Stereo, 7 Stunden Spieldauer
- 24 bit, 96 KHz, 2-6 Kanäle Surround
- 24 bit, 192 KHz Stereo
- Dolby Digital, DTS, MPEG-AAC, ...
- plus Standbilder und Textinformationen
- bisher kaum erhältlich

DVD: Video, Datenrate vs. Spieldauer



- Formate: 720x576x25 PAL / 720x480x29.97 NTSC
 - 2 Stunden Spieldauer gewünscht, bei 5 GB
 - typische Datenrate für MPEG-2 mit AC3-Audio
 - Digitales Fernsehen: DF1 sendet MPEG-2 mit
- 5.5 Mb/s
1.5 .. 9.8 Mb/s
6.8 Mb/s

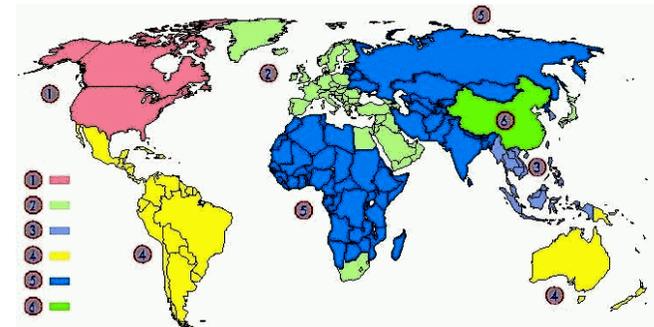
DVD: Datenrate MPEG-2



- Beispiel für Datenrate einer Video-DVD: 5 .. 10 Mb/s
- digitales Fernsehen (DVB-S) mit 6 Mb/s
- MP@ML erlaubt bis 15 Mb/s maximal

(ct 20/99, Sesamestreet-DVD)

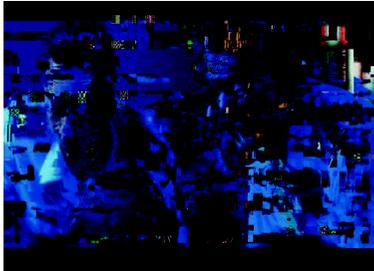
DVD: Region Codes



DVD-Video spezifiziert Region-Codes

- zeitversetzte Veröffentlichungen zwischen USA / EU / Japan
- Sicherung des Kino-Marktes
- Region 0 ist universell nutzbar
- Code in Laufwerks-Firmware, typisch höchstens 5x wechselbar

DVD: CSS



- direkte Kopie einer DVD-Video
- Daten großteils unlesbar
- nicht alle DVDs sind verschlüsselt

"Content Scrambling System":

- Schutz vor digitalen (=perfekten) Raubkopien
- verschlüsselte Übertragung zwischen Laufwerk und Decoder (HW/SW)
- komplexes Challenge-Response-Protokoll zur Authentifizierung
- Codes im Lead-In der DVD gespeichert, dort nicht zugreifbar
- Verfahren nicht publiziert, nur für Hersteller zugänglich

DVD: DeCSS

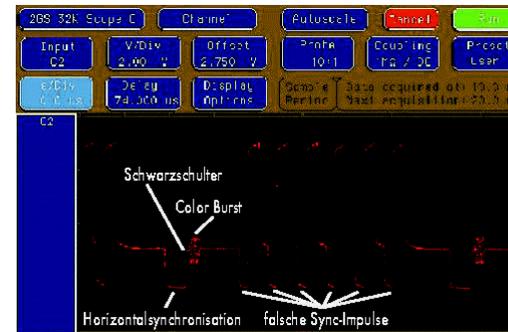
mittlerweile ist CSS ge crackt:

- diverse Angriffspunkte in den Windows Treiberschichten
- Screenshots -> AVI
- Software-Player cracken
- algorithmische Angriffe möglich wegen 40-bit Schlüssel

DeCSS:

- Windows-GUI
 - dekodiert DVD-Daten auf die Festplatte
 - verwendet Player-Key aus Xing Software-Player
 - Verbreitung via Internet / Abmahnungen durch DVD-Anwälte
- => ermöglicht DVD-Player für Linux
=> rechtliche Situation unklar

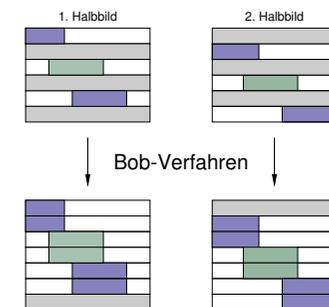
Macrovision



Schutz gegen (analoge) Kopien per Videorekorder:

- zusätzliche (falsche) Synchronimpulse
- wechselnde Amplitude zur Verwirrung der Aussteuerung (AGC)
- im unsichtbaren Bereich: Fernseher ignoriert das Signal

DVD: Interlace-Probleme



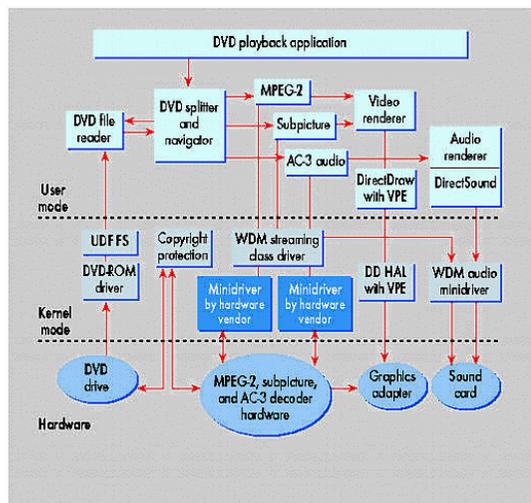
De-Interlacing:

- Kino: Vollbilder (24 Bilder/s)
- Fernsehen: Halbbilder:

PAL, Secam:	25 / 50
NTSC:	30 / 60
- Monitore: Vollbilder bei hoher Wiederholrate

- => komplexe Umrechnung notwendig
=> sonst schlechte Bildqualität (Fransen, Kammefekte)
=> Kinofilme laufen um Faktor 25/24 zu schnell . . .

DVD: Windows-Treiber ...



DVD: Laufwerke Q4/99

Benchmark-Ergebnisse DVD-ROM-Laufwerke						
	maximale Datenrate DVD-R	abspielbar von Fehler-DVD	Audio-Graben	mittlere Zugriffszeit	mittlere Datenrate	mittlere Datenrate
	MB/s/s besser»	Prozent besser»	Datenrate	ms «besser	CD-R	CD-RW
DVD-ROM_ATAPI						
Altecy DVD0-320E	6,5	44	3,8	127	2,9	0,6
ACOpen DVD9632	5,4	99	10	97	3,5	2,1
Creative L-Encore 6X	8,1	79	9	134	2,7	1,6
Guillemot Theater 6X	8,0	59	6	88	3,7	2,1
Hitachi GD-2500 BX	8,1	76	13	131	2,7	1,6
NEC DV-5500	6,2	83	6	85	4,1	0,9
Panasonic SR-8583	6,8	48	1	132	2,5	0,9
Philips FCD632	8,1	44	6	93	3,7	2,1
Pioneer DVD-AU35	5,4	100	12	92	3,4	3,4
Sony DDU22016	6,6	20	11	146	2,0	1,1
DVD-ROM_SATA						
Pioneer DVD-U035	5,4	34	10	94	3,5	3,2
Toshiba SD-M1201	6,5	60	6	124	3,7	2,2
DVD-RAM						
ACOpen DVD-5205	2,7	94	9	117	1,7	1,7
Hitachi GF-1000	2,7	56	10	265	1,5	1,7

In **abspielbar von Fehler-DVD** stellt die Menge dar, die von der Fehler-DVD abspielbaren Material in Prozent. Als DVD-Player diente Pioneer DVD 1.60 unter Windows 98. Von jedem Clip wurde die rückläufige Spielzeit bis zum Hängenbleiben gezählt. Stöckende Passagen weichen wir um die Hälfte ihrer Dauer. Wenn ein ganzer Clip stattdessen ruckelte, erzielte er nur 1 bis 5%. Bei ein oder zwei kleinen Aussetzern in einem Clip wurde dieser mit 98 bis 99% gewertet. Das Ergebnis ist der Mittelwert der Clips von 2 bis 6.

Die in der Spalte **Audio-Graben** angegebenen maximalen Datenraten gelten grundsätzlich nur für fehlerlos geladene Tracks. Im Unterschied zum Lesen von CD-ROM arbeiten die Laufwerke beim Graben weitgehend mit konstanter Datenrate (CLV = Constant Linear Velocity). Zum Audio-Graben dienten die beiden Program-

me WinDAC32 in der Version 1.48 (www.winac.de) sowie als Gegencheck CD Speed 99 Version 0.41 Beta (<http://www.cd-speed.com>).

Die **mittlere Datenrate** vereint die Messungen von fünf CD-Recordables mit unterschiedlichen Farb- und Reflexionsstärken (Cyan/Gold, Cyan/Silber usw.) zu einem Durchschnittswert.

Zum Benchmark und zum Abspielen der DVD-Videos mit Softwaredecodern stand ein Asus P2B mit Pentium-II-333 bei 60 MHz busfertig unter Windows 98 zur Verfügung. Die DVD-Laufwerke waren an verschiedenen IDE-Kanälen als Master beziehungsweise an einem Wide Ultra SCSI-Adapter SYMBIOS151U von Symbios logic angeschlossen. Zur Anzeige diente eine Diamond Viper V550 mit 16 MB RAM. Die Geschwindigkeitsmessungen wurden mit einer speziellen, nicht öffentlichen Version von H2Bench unter Windows 98 durchgeführt.

- DVD 6x, Tendenz steigend

DVD: DVD-R

DVD-Recordable:

- voll kompatibel mit DVD-Video, DVD-Audio, DVD-ROM
- kann auf jedem DVD-Player abgespielt werden
- Recorder sind noch extrem teuer

	Ver 1.0	Ver 1.9 / 2.0
Seiten	1 / 2	1 / 2
Kapazität	3.95 GB	4.7 GB
verfügbar	1997	1999
Pit µm	0.44 x 0.80	0.40 x 0.74
Verfahren	Farbstoffe wie CDR, 635 nm Laser	

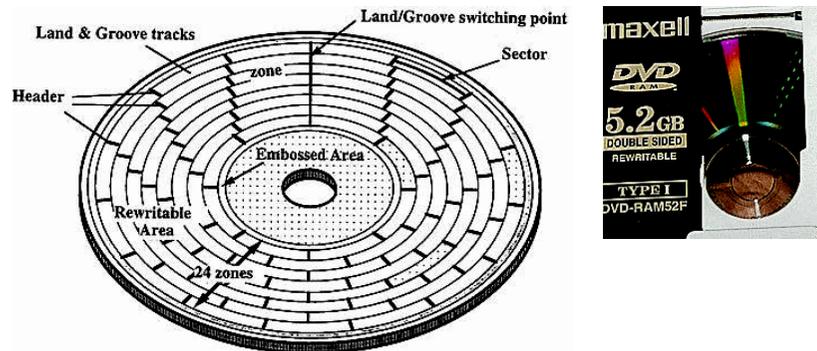
DVD: DVD-RAM

wiederbeschreibbare DVDs:

- mehrere, untereinander inkompatible Verfahren
- nicht mit DVD-R kompatibel
- Phase-Change-Technik wie CD-RW
- zoned CLV, wobbed pre-groove, usw.

	Ver 1.0	Ver 1.9 / 2.0
Seiten	1 / 2	1 / 2
Kapazität	2.6 GB	4.7 GB
verfügbar	1997	1999
Pit µm	0.41 x 0.74	0.28 x 0.615
Verfahren	phase change wie CD-RW	

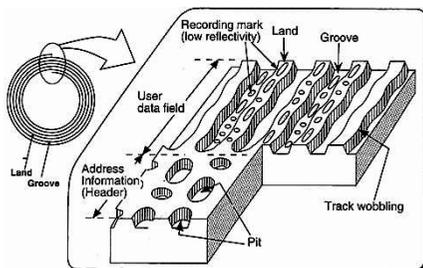
DVD: DVD-RAM Sektoren



- Disk ist in 24 Zonen eingeteilt
- innerhalb einer Zone konstante Umdrehungsgeschwindigkeit

Medientechnik | WS 2001 | 18.204

DVD: DVD-RAM Pregroove

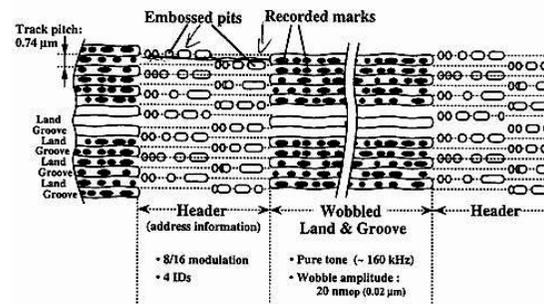


CD/DVD-Rohlinge enthalten eine Roh-Spur:

- Roh-Spur für Spurführung des Schreib/Lese-Kopfes
- Track-Wobbling für Drehzahlregelung
- regelmässig Header-Sektoren
- DVD-RAM: Daten abwechselnd in Lands und Grooves

Medientechnik | WS 2001 | 18.204

DVD: DVD-RAM Datenaufzeichnung



- Rohlinge enthalten fertige Header-Zonen
- DVD-RAM Typ, Kapazität, Schreibgeschwindigkeit, usw.
- Aufzeichnung nur in die Datenbereiche
- UDF: max 1 Partitit a 2.3 GB / Seite
- FAT 16: mehrere Partitionen a 2 GB / Seite

Medientechnik | WS 2001 | 18.204

DVD: Vergleich mit CD und DVR

Figure 3. Three generations of optical disc systems. Progress in "areal density," or bit count per unit area, takes big steps; a CD holds 650MB, a DVD 4.7GB, and a DVR 22GB) by reducing the spot size through a shorter wavelength and stronger objective lens (with higher numerical aperture). The electron micrographs show read-only discs with replicated pit patterns.

CD	DVD	DVR

(CACM 43/2000)

Medientechnik | WS 2001 | 18.204

Frameworks

- Multimedia-Frameworks
- Motivation, Ziele
- Komponenten: Player, Encoder, Server
- Streaming, RTP
- Windows-Media
- Windows Media Player
- Digital Rights Management
- Java Media Framework
- Konzept, Klassenhierarchie, Status
- Registry, Codec-Auswahl
- Codebeispiele

Medientechnik | WS 2001 | 18.204

Literatur

Microsoft DirectX 8.1 documentation, msdn.microsoft.com/library/
Bargen, Donnelly, Inside Direct X, Microsoft Press, 1998

Java Media Framework API Guide, Sun Microsystems, 1999
JMF API specification, JMF examples, java.sun.com/products/java-media/jmf
Gordon, Talley, Essential JMF, Prentice Hall, 1999

Internet RFCs, hier: RFC 1889 (RTP/RTCP)

www.microsoft.com/windowsmedia
www.real.com
www.apple.com/quicktime/
java.sun.com/products/java-media/

Medientechnik | WS 2001 | 18.204

Frameworks: Ziele

Infrastruktur ("Rahmenwerk") für

- Wiedergabe von Multimedia,
- möglichst viele Medientypen
- von lokalen Datenträgern
- oder Streaming über Netzwerke / Internet
- Synchronisation der beteiligten Medien
- Verwaltung der benötigten Codecs und Renderer
- Unterstützung der benötigten Netzwerkprotokolle

zusätzlich

- Unterstützung für Medienproduktion / -capturing
- Rechte-Management (Copyright, Kopierschutz, Pay-per-use, ...)
- Medien-Server

Medientechnik | WS 2001 | 18.204

Frameworks: IBM Hotmedia Marketing . . .



A single, easy-to-use authoring environment provides the basis for creating a wide range of multimedia effects, resulting in a *single file* that contains a variety of multiple interactive elements and is easily added to a Web page. Just like that.

With HotMedia, today's Web professionals can take advantage of a number of opportunities in e-business, such as:

- e-tour: *Customers can take a virtual tour of any environment*
- e-care: *... or view and hear operational and assembly instructions online*
- e-tail: *... and interactively experience products and services.*



Medientechnik | WS 2001 | 18.204

Frameworks: Beispiele

- Microsoft WindowsMedia
 - Real Networks
 - Apple Quicktime
 - Sun Java Media Framework
(Linux Simple Direct Layer, aber nur Ansätze zu Tools: xmps, xine, ...)
(MPEG-4)
- enorme Marktbedeutung, daher:
- Player meistens gratis verfügbar
 - Bindung an proprietäre (undokumentierte) Protokolle
 - Authoring-Tools für "low quality" frei verfügbar
 - aber professionelle Versionen kommerziell
 - Medien-Server "schweineteuer"
- zunehmende Bedeutung von "Digital Rights" Tools

Medientechnik | WS 2001 | 18.204

Frameworks: Grundfunktionen

- Lesezugriff auf Medien-Dateien oder -Streams
 - Erkennung des jeweiligen Datenformats (z.B. AVI oder Qt)
 - Bereitsstellen / Zugriff auf Codecs (z.B. MPEG-2)
 - zusätzliche Filter (z.B. YUV nach RGB)
 - Gerätezugriff zur Ein- und Ausgabe (z.B. Framebuffer, DV-Kamera)
- Player-Applikation w/o GUI
 - Player-Utilities: playlists, cddb-Zugriff, ...
- Streaming, Quality of Service (QoS)
 - Sicherheits- und Copyright-Funktionen

Medientechnik | WS 2001 | 18.204

Frameworks: Synchronisation

- Video 25fps: jedes Frame jeweils 40 ms
- Audio 48KHz: jedes Sample 20.8 μ s
usw.

aber: wie genau muss die Synchronisation passen?

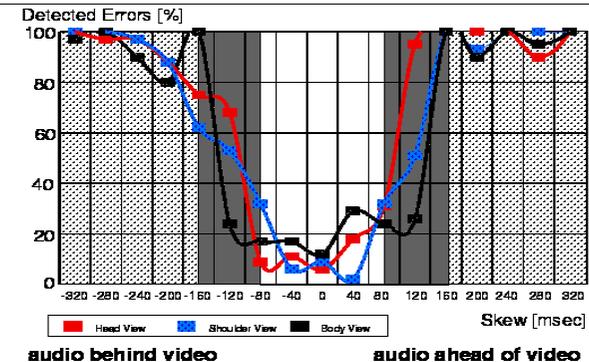
- Puffergrösse / Latenz / QoS (re-transmission) optimieren
- => Experimente mit Testpersonen
- Versatz Video / Musik / Sprache variieren
 - individuelle Bewertung der wahrgenommenen Fehler

Empfehlungen (Beispiele, siehe Steinmetz 5.6 für Details):

Audio (Versatz von Stereosamples)	+/- 11 μ s
Audio (Dialog mehrerer Personen)	+/- 120 ms
Audio (Hintergrundmusik)	+/- 500 ms
Video (Lippensynchronisation)	+/- 80 ms

Medientechnik | WS 2001 | 18.204

Beispiel: Lippen-Synchronisation



Abhängigkeiten:

- Video-Inhalt, z.B. Sprache vs. Ereignisse (z.B. Hammerschlag)
- Video-Hintergrund, wg. Ablenkung
- Kamera-Einstellung (Totale vs. Figur vs. nur Kopf)
- nur Sprache oder Hintergrundgeräusche oder Musik
- Eigenheiten der jeweiligen Sprache

(siehe Steinmetz, Kapitel "media synchronization")

Medientechnik | WS 2001 | 18.204

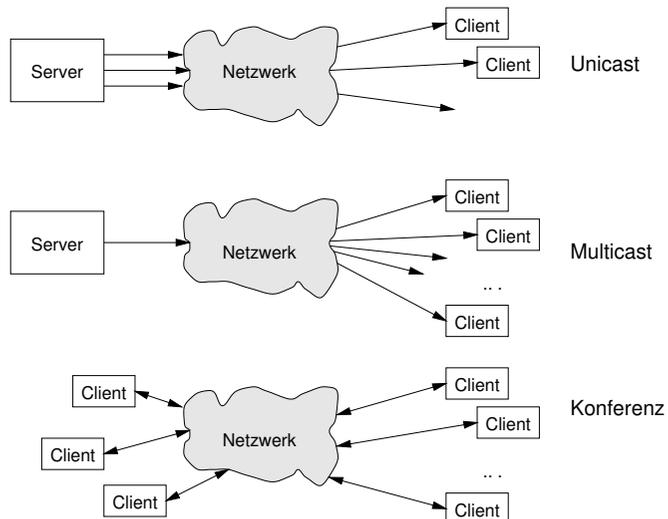
Streaming: Definition

"Streaming Media / Audio" :=

- Echtzeit-Übertragung von Mediendaten
- von einem Server zu einem / vielen Clients
- ohne vorheriges Laden der gesamten Daten
- Beschränkung auf Bitrate des Kanals, mit Schwankungen
- erfordert Fehlertoleranzmechanismen
- Tradeoff Qualität / Robustheit / Bitrate
- Audio auch standalone ("Internet Radio")
- aber meistens Video / Animationen mit synchronem Audio
- großes Marktpotential erwartet ("Video on demand", ...)
- Microsoft WMA, RealAudio, Apple Quicktime, Liquid Audio, ...
- Mediensuche / Tauschbörsen (Napster) / Copyright-Fragen

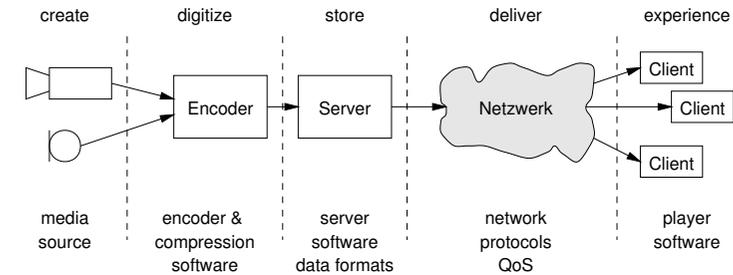
Medientechnik | WS 2001 | 18.204

Streaming: Szenarien



Medientechnik | WS 2001 | 18.204

Streaming: System



funktionierendes System erfordert alle Komponenten:

- alle Anbieter liefern "integrierte Lösung"
- Encoder + Server + Management + Player (+ Verschlüsselung)
- Microsoft, RealNetworks, Apple, ...
- Live-Übertragung nur mit Echtzeit-Encoder

Medientechnik | WS 2001 | 18.204

Streaming: Bandbreiten

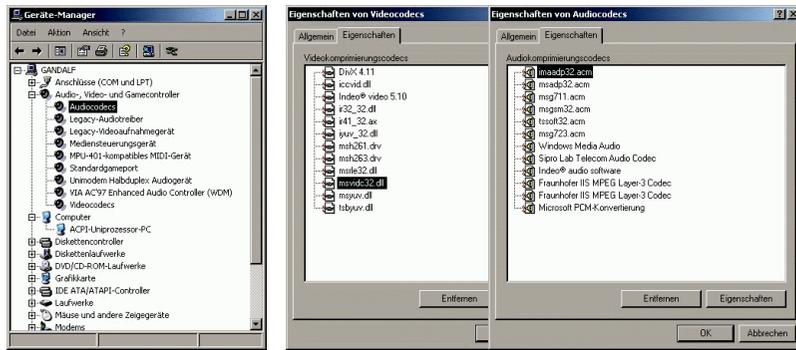
"streaming" ist stark von verfügbarer Bandbreite abhängig:

Modem	28 .. 56	Kb/s	(bis ca. 4 Mb/s)
ISDN	64	Kb/s	
DSL	128 .. 768	Kb/s	
Intranet	10 .. 100	Mb/s	
MPEG-2 (DVD)	4 .. 9	Mb/s	
MPEG-1, CDDA	1.5	Mb/s	
MPEG-4 (DivX)	1.5	Mb/s	
MP3 Audio	64 .. 320	Kb/s	
RealAudio 8	10 .. 160	Kb/s	

- => selbst MP3 nicht per Modem streaming-fähig
- => Bildtelefonie (H.263 QCIF 15fps) erfordert mindestens ISDN

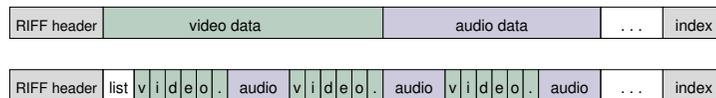
Medientechnik | WS 2001 | 18.204

Windows Media: Medientypen



- vorinstallierte Codecs in Windows XP Home (plus OpenDivX)
- u.a. MPEG-1, -2, -4, H.26x, Indeo 5, diverse Audioformate
- automatischer Download weiterer Codecs

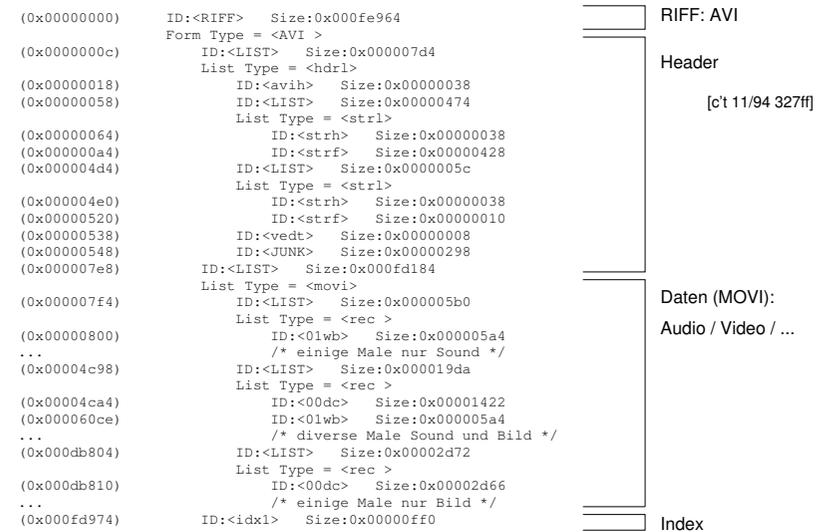
AVI: Audio Video Interleaved



normal / interleaved

- RIFF-Dateiformat für Multimedia / Videos
- eingeführt mit Video for Windows (Win 3.11)
- definiert ca. 20 Chunk-Typen
- List-Chunk erlaubt "verwobene" Daten
 - Audiodaten, Videoframes (BMP), Audiodaten, ...
 - Datei muß vor Abspielen nicht voll geladen werden
- Spezifikation in Windows API, Übersicht in c't 94/11 S.327
- mittlerweile von ASF bzw. WMA/WMV abgelöst (s.u.)

AVI: Beispiel



ASF: Advanced Streaming Format

"... an extensible file format designed to store synchronized multimedia data. It supports data delivery over a wide variety of networks and protocols, while still proving suitable for local playback. The explicit goal of ASF is to provide a basis for industry-wide multimedia interoperability, with ASF being adopted by all major streaming solution providers."

(Microsoft, Real Networks 1998)

- RIFF-ähnliche Struktur: header, index, interleaved data
- Chunks (= "objects") per GUID gekennzeichnet
 - bei Bedarf Registrierung der GUIDs bei Microsoft
 - erlaubt z.B. Codec-Download
 - auch der Player identifiziert sich über seine GUID
 - sehr feine Copyright / Nutzungskontrolle
- siehe ASF-Spezifikation (Version 1.0, 26.02.1998)
- aber aktuelle Version (WMA/WMV) nicht mehr dokumentiert

ASF: GUID

GUID/UUID := "globally/universally unique IDs"

- 128-bit Signatur
- entwickelt für NCS (Apollo), übernommen in OSF/DCE und Windows
 - Ethernet-MAC Adresse plus Zeitmarke (eindeutig)
 - oder 47-bit Zufallsadresse plus Zeitmarke
 - Zeit mit 100ns Auflösung, Sequenznr. zur Korrektur (reboot)
 - eindeutig bis 3400 n.Chr.

```
ASF-Header:      D6E229D1-35DA-11DA-9034-00A0C90349BE
ASF-Data:       D6E229D2-35DA-11DA-9034-00A0C90349BE
ASF-Index:     D6E229D3-35DA-11DA-9034-00A0C90349BE
USW.           time-low-mid-high|seq.|ethernet----
```

- u.a. auch jeder Windows-Rechner eindeutig identifizierbar

Windows Media: Codec-Auswahl

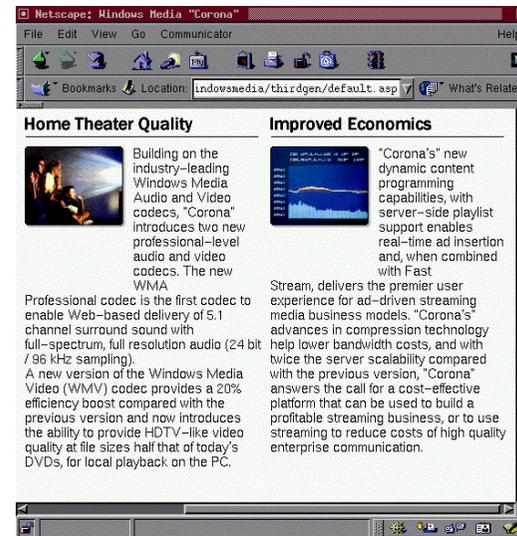
AVI-Format (seit Windows 3.11):

- RIFF-Datei, einzelne Chunks mit "fourcc" Kennung
- Auswahl zum "fourcc" passender Codecs via Registry
- Microsoft verwaltet zentrale Liste aller registrierten Codecs
www.microsoft.com/hwdev/archive/devdes/fourcc.asp
www.webartz.com/fourcc/fcccodec.htm

ASF-/WMx-Format:

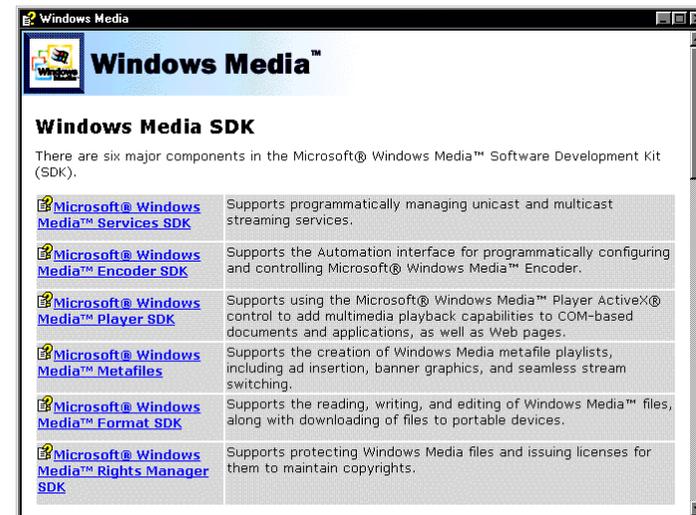
- Codecs-Verwaltung via GUID
- optionale Chunks für Codec-Downloadseiten
- automatischer Download via www.microsoft.com
- Player-Update entfernt "nicht unterstützte" Codecs ...

Windows Media: 3rd generation

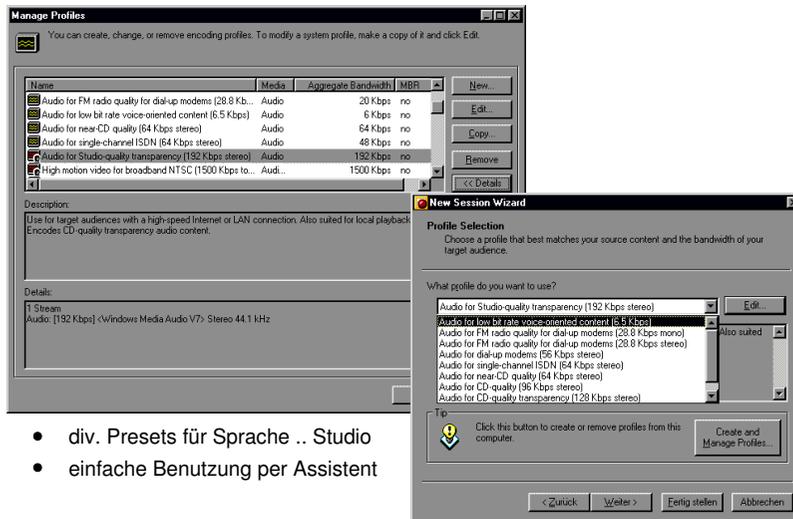


(www.microsoft.com, 07.01.2002)

WindowsMedia: Authoring



WindowsMedia: Audio Profiles, Wizard

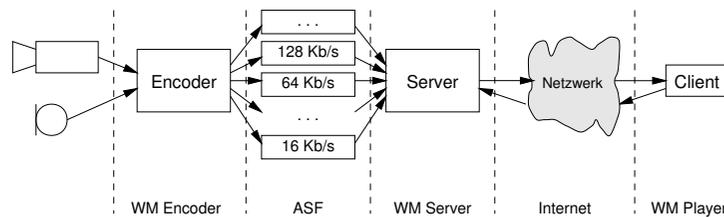


- div. Presets für Sprache .. Studio
- einfache Benutzung per Assistent

WindowsMedia: Security

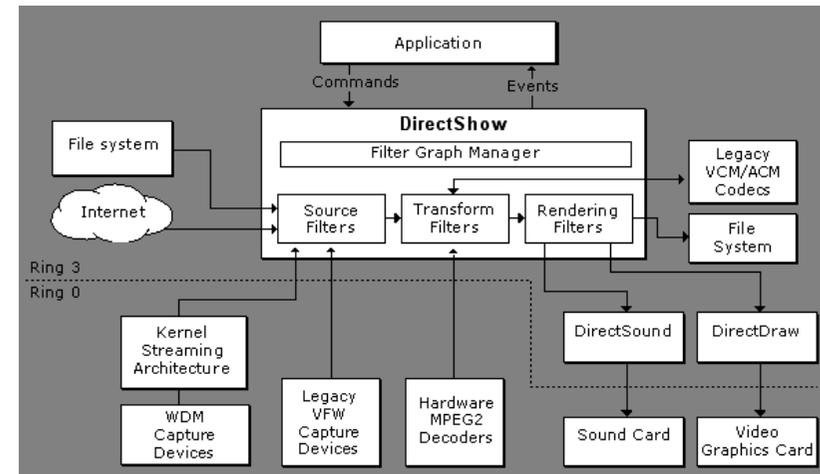
- mögliche Restriktionen für WMA/WMV-Dateien:
 - play on PC
 - counted play
 - start date
 - expire date
 - burn to Audio CD
 - counted CD burn
 - transfer to portable device
 - ... and many more
- Player überprüft vorhandene Lizenzen
- oder versucht, neue Lizenz zu erhalten:
 - "might issue a licence silently, so the consumer is unaware of the process"
 - "... supports pre-delivered licences"

WindowsMedia: Intelligent Streaming

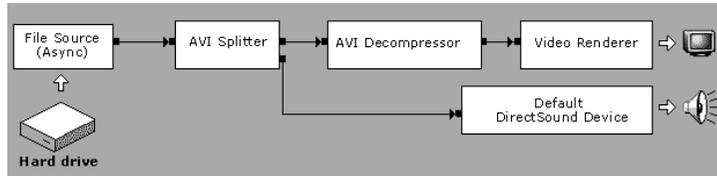


- Encoder erzeugt (optional) spezielle ASF-Datei:
 - mehrere parallele Datenströme
 - mit verschiedenen Bitraten
 - entsprechende Qualitätsstufen
- Player sendet Feedback über Bandbreite und Paketverluste
- Server sendet nur den Datenstrom der max. möglichen Qualität

DirectShow: Übersicht



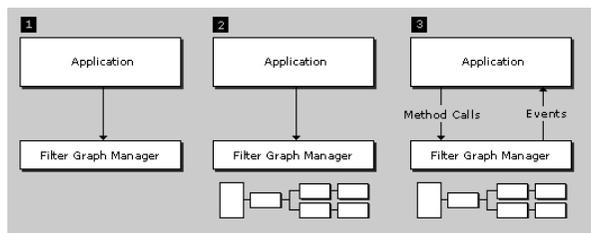
DirectShow: Filter



"Media Streaming":

- DirectShow besteht aus einzelnen (COM-) Komponenten:
- Codecs, Multiplexer/Splitter, Renderer, usw.
- Verkettung nur über definierte Schnittstellen ("pins")
- zentraler FilterGraphManager zur Synchronisation
- Applikation kombiniert einfach die einzelnen Filter

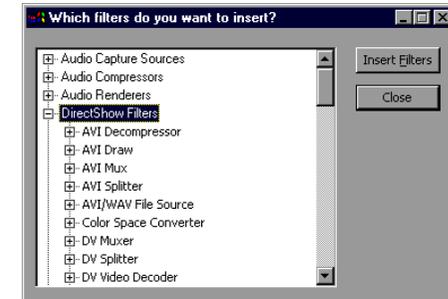
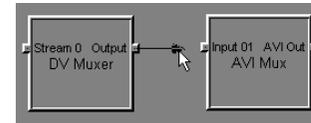
DirectShow: FilterGraphManager



Konstruktion des Filter-Graphen:

- "per Hand", Applikation gibt die Filter fest vor
- automatisch via FilterGraphManager:
 - Suche aller passenden Filter für Eingabe- / Ausgabeformate
 - Einbau evtl. benötigter Formatkonverter
 - Algorithmus terminiert, da gerichtete/zyklenfreie Graphen

DirectShow: GraphEdit



GraphEdit:

- interaktiver Editor zum Aufbau der Filtergraphen
- "drag and drop"
- Zugriff auf alle installierten Codecs / Filter
- direktes Austesten des Graphen
- schreibt Quelltext für FilterGraphManager

(msdn.microsoft.com/library/default.asp?url=/library/en-us/dx8_c/directx_cpp/htm/usinggraphedit.asp)

DirectShow: Simple Player

```
#include <dshow.h> // simple player demo, DirectX 8.1
void main(void)
{
    IGraphBuilder *pGraph;
    IMediaControl *pMediaControl;
    IMediaEvent *pEvent;
    CoInitialize(NULL); // initialize COM DLLs

    // Create the filter graph manager and query for interfaces.
    CoCreateInstance(CLSID_FilterGraph, NULL, CLSCTX_INPROC_SERVER,
        IID_IGraphBuilder, (void **)&pGraph);
    pGraph->QueryInterface(IID_IMediaControl, (void **)&pMediaControl);
    pGraph->QueryInterface(IID_IMediaEvent, (void **)&pEvent);

    // Build the graph. IMPORTANT: Change string to a file on your system.
    pGraph->RenderFile("C:\\Example.avi", NULL);

    // Run the graph.
    pMediaControl->Run();

    // Wait for completion.
    long evCode;
    pEvent->WaitForCompletion(INFINITE, &evCode);

    // Clean up.
    pMediaControl->Release();
    pEvent->Release();
    pGraph->Release();
    CoUninitialize();
}
```

JMF: Java Media Framework

- Java-API zur Verarbeitung und Präsentation zeitbasierter Medien
- JMF 1.0 (nur Wiedergabe) von Sun, Intel, SGI
- JMF 2.0 (inkl. Capturing) von Sun und IBM
- diverse Audio-/Video-Codecs
- pure-Java oder "performance pack" Versionen
- Streaming via RTP/RTCP

Ziele:

- Integration von "Medien" in Java-Applikationen und -Applets
- einfache Programmierschnittstelle
- JSR 135: Multimedia API for J2ME

(www.javasoft.com/products/java-media/jmf)

Medientechnik | WS 2001 | 18.204

JMF: Medientypen

		"cross platform"		"performance packs"	
		pure-Java	Solaris	Windows	
PCM,ADPCM	.wav	D,E	D,E	D,E	
G.711 (u-law)	.wav	D,E	D,E	D,E	
GSM	.gsm	D,E	D,E	D,E	
MP2,MP3	.mp3	D	D/E	D,E	
MIDI	.midi	--	D	D	
Windows ACM		--	--	D,E	
JPEG	.jpg	D	D/E	D/E	
Qt Cinepak	.mov	D	D/E	D	
MPEG-1	.mpeg	--	D	D	
H261/H263		- / D	D / D,E	D / D,E	
Flash	.swf	D	D	D	
Hotmedia	.mvr	D	D	D	
Windows VCM	.avi	--	--	D/E	

(in JMF 2.1.1, java.sun.com/products/java-media/jmf)

Medientechnik | WS 2001 | 18.204

JMF: Browser-Unterstützung

```

...
<applet code=SimplePlayerApplet width=320 height=300>
  <param name=file value="sun.avi">
</applet>
...

```

JMF läuft auch als Applet:

- in Java-fähigen Browsern (JDK 1.1+)
- erfordert lokale Installation des JMF
- kein ständiger Codec-/Code-Download
- allerdings: Class-Download << Medien-Download ...
- SecurityManager regelt Zugriff auf (Capture-) Devices

Medientechnik | WS 2001 | 18.204

JMF: Player / PlayerBean



- media location
- name
- size, position
- aspect ratio
- audio volume
- show control panel
- show caching controls
- start, stop, search, loop

- einfacher Medienplayer, minimale GUI:
- Transportkontrolle, Lautstärkeregelung, Medieninfo
- auch als Komponente zur direkten Integration in Anwendungen:
javax.media.bean.playerbean.MediaPlayer

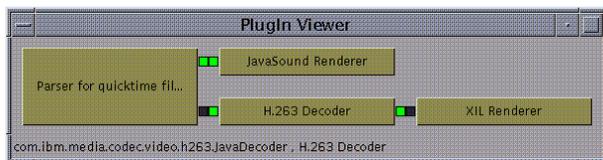
Medientechnik | WS 2001 | 18.204

JMF Player: Beispielcode

```
public class SimplePlayerApplet extends Applet implements ControllerListener {
    Player player = null; // the media player
    Component visualComponent = null; // the (video) display AWT component
    Component controlComponent = null; // AWT component for the controls

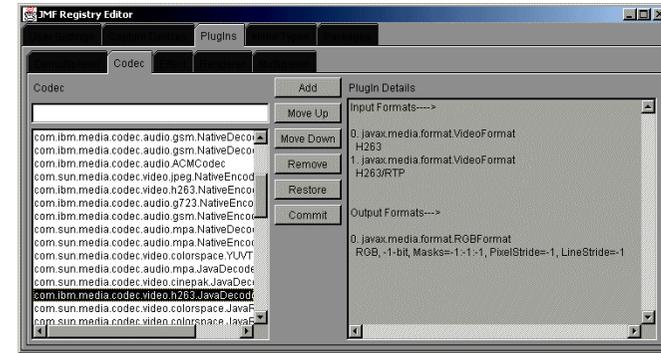
    public void init() {
        String mediaName = getParameter( "FILE" );
        URL mediURL = new URL( getDocumentBase(), mediaName );
        MediaLocator mrl = new MediaLocator( url.toExternalForm() );
        player = Manager.createPlayer(mrl); // create a player
        player.addControllerListener(this); // we handle the player events
    }
    public void start() { // starts the player (including prefetch)
        if (player != null) player.start();
    }
    public void stop() {
        if (player != null) { player.stop(); player.deallocate(); }
    }
    public void destroy() {
        player.close();
    }
    public synchronized void controllerUpdate(ControllerEvent event) {
        if (event instanceof RealizeCompleteEvent) { // player ok, create GUI now
            controlComponent = player.getControlPanelComponent();
            visualComponent = player.getVisualComponent();
            panel.add(controlComponent); panel.add(visualComponent);
        } else if (event instanceof CachingControlEvent) { // update progress bar
        } else if (event instanceof ...) { // handle other events
        }
    }
}
```

JMF: Codec-Graph



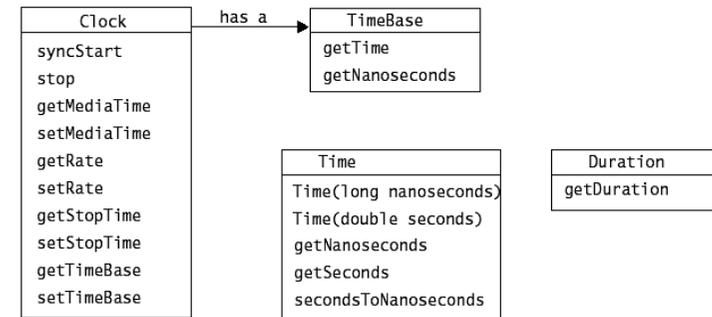
- Konzept wie bei DirectShow:
- Demultiplexer, Codec, Effect, Multiplexer, Renderer
- jeweils als abgeleitete Klasse von "Player" bzw. "Processor"

JMF: Registry



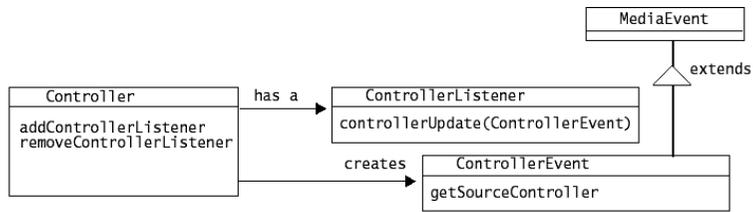
- Liste der Default-Codecs in JMF 2.1.1 ("Windows performance pack")
- JMF "registry" enthält Liste der vorhandenen Codecs
- zusätzlich Download via Classloader

JMF: Clock, TimeBase, Time



- interne Zeitberechnung mit 1ns Auflösung
- Time, Duration: Abstraktion von Zeit bzw. Zeitintervallen
- Clock: Klasse zur Synchronisation

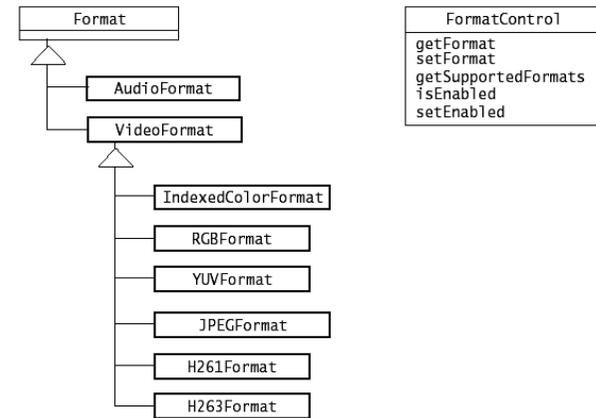
JMF: Events



ControllerListener / ControllerEvent

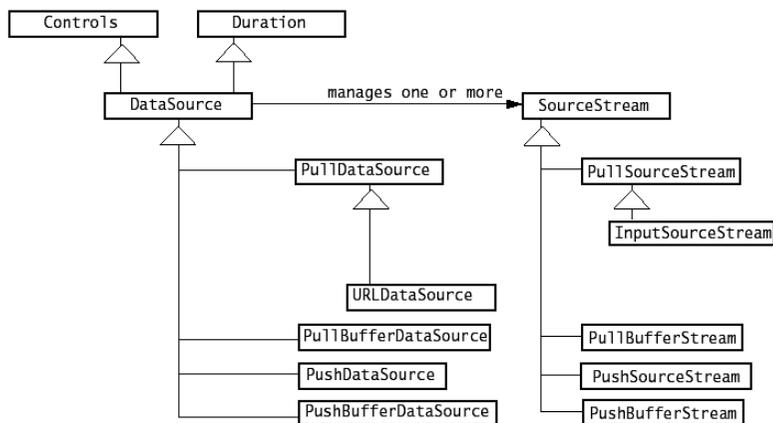
- Abstraktion für alle (!) Medien-relevanten Ereignisse, z.B:
- Player fertig initialisiert
- Filmende
- uvm.

JMF: Media format



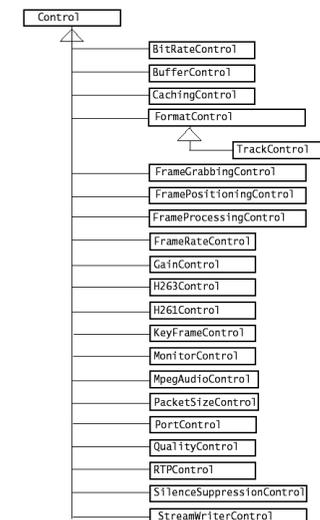
- kleiner Ausschnitt aus der "Format" Klassenhierarchie
- entsprechende Hierarchie für die Audio-/ weitere Datentypen

JMF: Data model

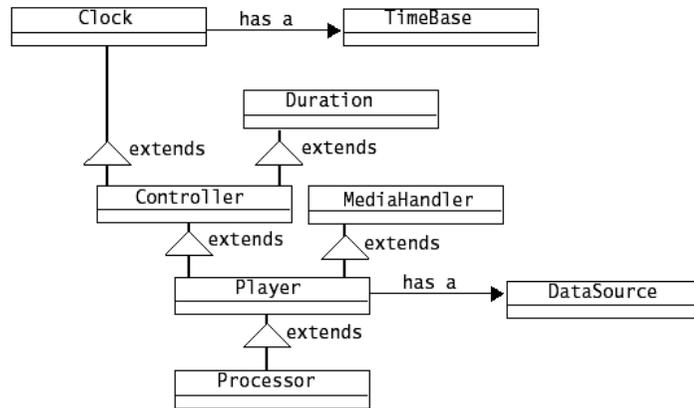


- Datenquellen sowohl als push- oder pull-Typen

JMF: Control

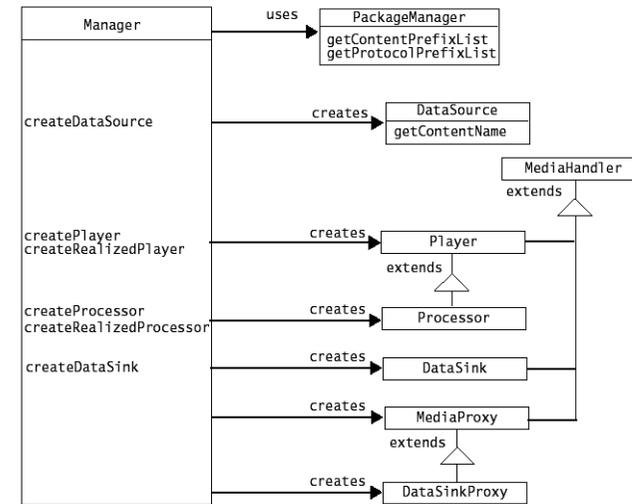


JMF: Controller



- Player: nur Medien-Wiedergabe (aus einer DataSource)
- Processor: Daten-Manipulation (z.B. Filterung)

JMF: Manager

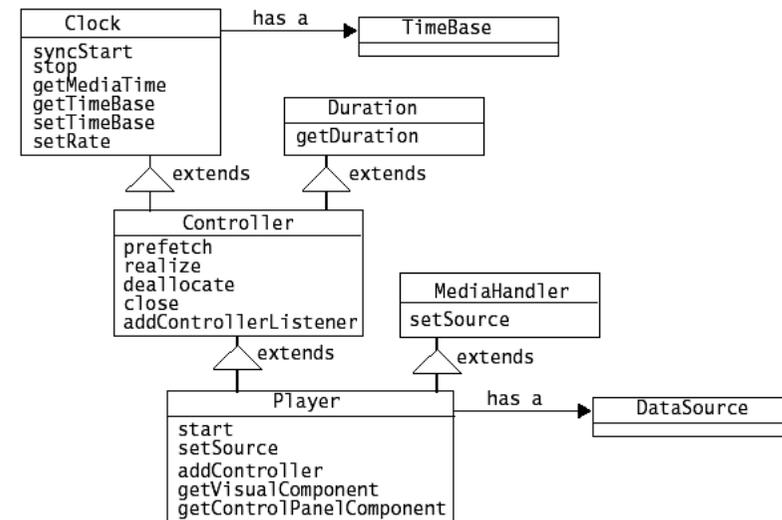


JMF: Managers

JMF uses four managers:

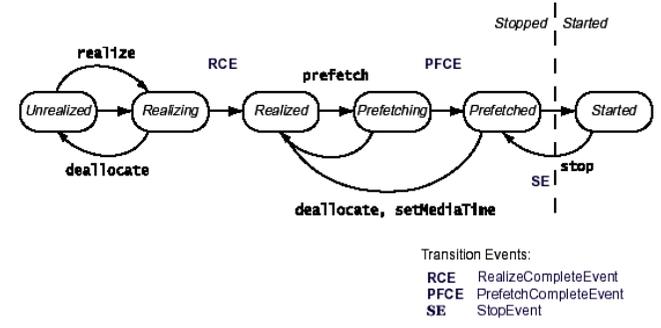
- **Manager**—handles the construction of Players, Processors, DataSources, and DataSinks. This level of indirection allows new implementations to be integrated seamlessly with JMF. From the client perspective, these objects are always created the same way whether the requested object is constructed from a default implementation or a custom one.
- **PackageManager**—maintains a registry of packages that contain JMF classes, such as custom Players, Processors, DataSources, and DataSinks.
- **CaptureDeviceManager**—maintains a registry of available capture devices.
- **PlugInManager**—maintains a registry of available JMF plug-in processing components, such as Multiplexers, Demultiplexers, Codecs, Effects, and Renderers.

JMF: Player class diagram



JMF: Player states

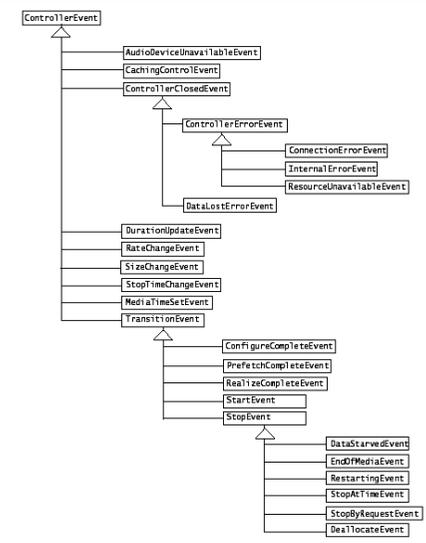
A Player can be in one of six states. The Clock interface defines the two primary states: *Stopped* and *Started*. To facilitate resource management, Controller breaks the *Stopped* state down into five standby states: *Unrealized*, *Realizing*, *Realized*, *Prefetching*, and *Prefetched*.



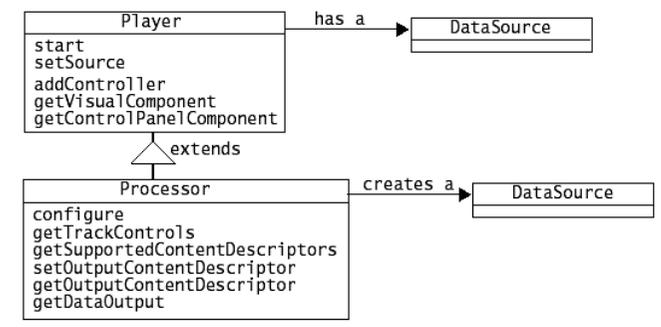
JMF: Player state transitions

Method	Unrealized Player	Realized Player	Prefetched Player	Started Player
addController	NotRealizedError	legal	legal	ClockStartedError
deallocate	legal	legal	legal	ClockStartedError
getControlPanelComponent	NotRealizedError	legal	legal	legal
getGainControl	NotRealizedError	legal	legal	legal
getStartLatency	NotRealizedError	legal	legal	legal
getTimeBase	NotRealizedError	legal	legal	legal
getVisualComponent	NotRealizedError	legal	legal	legal
napToTimeBase	ClockStoppedException	ClockStoppedException	ClockStoppedException	legal
removeController	NotRealizedError	legal	legal	ClockStartedError
setMediaTime	NotRealizedError	legal	legal	legal
setRate	NotRealizedError	legal	legal	legal
setStopTime	NotRealizedError	legal	legal	StopTimeSetError if previously set
setTimeBase	NotRealizedError	legal	legal	ClockStartedError
syncStart	NotPrefetchedError	NotPrefetchedError	legal	ClockStartedError

JMF: Controller event hierarchy

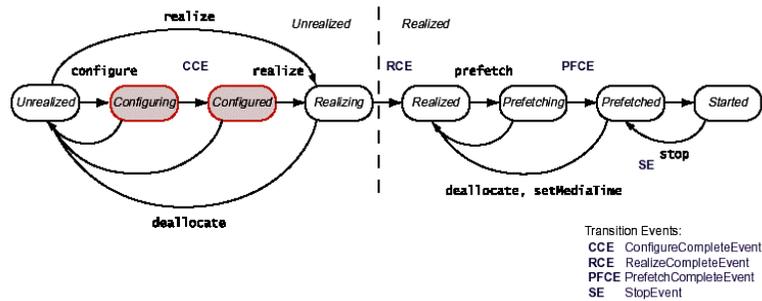


JMF: Processor



- Player liest Mediendaten aus seiner DataSource
- Processor liest und schreibt (manipulierte) Daten
- Basisklasse für alle Filter und Codecs

JMF: Processor states



- "unrealized" / "realized": "noch nicht fertig" / "benutzbar"
- "prefetching" Füllen der internen Puffer
- "start" / "stop" Abspielen / Stopp
- "setMediaTime" Suche / Navigation im MediaStream

Medientechnik | WS 2001 | 18.204

JMF: FrameSeek

```

import javax.media.control.FramePositioningControl;

class Seek extends Frame implements ControllerListener, ActionListener {
    ... // GUI stuff
    Player p; FramePositioningControl fpc;

    // given a DataSource, create a player and use that player for playback
    public boolean open(DataSource ds) throws Exception {
        p = Manager.createPlayer(ds);
        p.addControllerListener(this);
        p.realize();

        fpc = (FramePositioningControl)p.getControl(
            "javax.media.control.FramePositioningControl");
        totalFrames = fpc.mapTimeToFrame(duration);
        System.err.println("Total # of video frames: " + totalFrames);
        ...
        int currentFrame = fpc.mapTimeToFrame(p.getMediaTime());
        int randomFrame = (int)(totalFrames*Math.random());
        randomFrame = fpc.seek( randomFrame );
    }

    public static void main(String [] args) throws Exception {
        MediaLocator ml = new MediaLocator( argv[0] );
        DataSource ds = Manager.createDataSource( ml );
        Seek seek = new Seek();
        if (!seek.open(ds)) System.exit(0);
    }
}
  
```

Medientechnik | WS 2001 | 18.204

JMF: Lightweight Player in Swing

```

import javax.media.*;
...
public class MDIApp extends Frame {
    JMFFrame jmframe = null;
    JDesktopPane desktop;
    Player player = null;

    public MDIApp() {
        super("Java Media Player");
        setLayout( new BorderLayout() );
        desktop = new JDesktopPane();
        desktop.setDoubleBuffered(true);
        add("Center", desktop);
        setMenuBar( createMenuBar() );
        setSize(640, 480);
        setVisible(true);

        UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
        Manager.setHint(Manager.LIGHTWEIGHT_RENDERER, new Boolean(true));
        ...
        player = Manager.createPlayer(url);
    }

    public void controllerUpdate(ControllerEvent ce) {
        ...
        if (ce instanceof PrefetchCompleteEvent) {
            desktop.add( player.getVisualComponent() );
        }
        ...
    }
}
  
```

Medientechnik | WS 2001 | 18.204

Quicktime: Java API

```

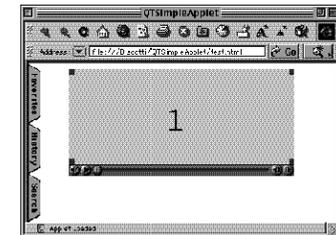
public class QTSimpleApplet extends Applet {
    private Drawable myQTContent;
    private QTCanvas myQTCanvas;

    public void init () {
        try {
            QTSession.open();
            setLayout( new BorderLayout() );
            myQTCanvas = new QTCanvas (QTCanvas.kInitialSize, 0.5F, 0.5F);
            add( myQTCanvas, "Center" );

            QTFile file = new QTFile( getCodeBase().getFile() +
                getParameter("file") );
            myQTContent = QTFactory.makeDrawable( file );
        } catch (Exception qtE) { ... }
    }

    public void start () {
        try {
            if (myQTCanvas != null)
                myQTCanvas.setClient( myQTContent, true );
        } catch (QTException e) { ... }
    }
    ...
}
  
```

- "arguably simpler than C code"
- "C code differs for Mac and Windows"



Medientechnik | WS 2001 | 18.204

MPEG-7

- Motivation
 - Anwendungsgebiete
 - Content Description
-
- Document Description Language (DDL)
 - Description Schemes (DS)
 - Descriptions (D)
- Beispiele:
- Audio: Beschreibung von Melodien
 - Audio: Beschreibung und Vergleich von Klangfarben
 - Video: Erkennung von Szenenwechseln (Schnitte)
 - Video: automatische low-level Szenenbeschreibung

(www.csell.it/mpeg/)

Medientechnik | WS 2001 | 18.204

Literatur

MPEG Homepage, www.csell.it/mpeg/
 MPEG-7 Homepage: www.mpeg-7.com/

diverse Konferenz/Workshop-Beiträge und Tutorials auf obigem Server, u.a.:
 E.J.Delp, Image and video databases: who cars?, MPEG7 IMA tutorial, 2001
 P. Salembier, Status of MPEG-7, IBC 2000, Amsterdam

zum Vergleich: inhaltsbasierte Bildsuche (Gnu Image finding tool):
viper.unige.ch/ www.gnu.org/software/gif/
 "Suchen ohne Worte", c't 15/2001, 162ff

Medientechnik | WS 2001 | 18.204

MPEG-7: Motivation

- A/V-Material zunehmend digital verfügbar
- weltweiter Zugriff via Internet / Datenbanken / Tauschbörsen

wie sucht und findet man Multimedia-Daten?

- effiziente Algorithmen für Volltextsuche bekannt (inverted tables)
 - z.B. Internet-Suchmaschinen (Google, Altavista)
 - oder (manuelle) Klassifizierung (Yahoo)
 - aber: bisher keine entsprechenden Algorithmen für A/V
- => Annotation von A/V-Daten mit Textbeschreibungen
- Metabeschreibung (data about data)
 - nicht unbedingt in den A/V-Daten selbst enthalten
 - unabhängig vom Format der A/V-Daten

(Herre)

Medientechnik | WS 2001 | 18.204

MPEG-7: Ziele

- MPEG-Standard zur
- Beschreibung des Inhalts audio-visueller Information
 - zur schnellen Suche und Identifikation von Inhalten
 - Beschreibung diverser Aspekte der Medien:
 - "low-level, structure, semantic, models, collections, creation, ..."
 - für eine Vielfalt von Anwendungen
 - unabhängig von Datenformat der Medien selbst
 - auch zur Beschreibung von analogem Material
 - für Audio, Sprache, Bilder, Video, 3D-Graphik, ...
 - Szenenbeschreibung der Kombination mehrerer Medien

(Salembier)

Medientechnik | WS 2001 | 18.204

MPEG-7: Anwendungen . . .

vielfältige Anwendungen denkbar:

- Organisation und Suche in AV-Datenbanken (Bilder, Video, Radio, ...)
- Programmauswahl bei Rundfunk / Fernsehen
- Überwachung (z.B. Stauwarnungen, Maschinensteuerung, ...)
- Luftbildauswertung (z.B. Kartographie, Ökologie, Exploration)
- Journalismus (z.B. Suche nach Personen und Ereignissen)
- E-commerce, Teleshopping (z.B. Suche nach bestimmten Stoffen)
- Personalisierte News-Services (z.B. im push-services im Internet)
- Unterhaltung (z.B. Suche nach einem Karaoke-Stück)
- Kultur (z.B. Museen)
- Ausbildung, Multimedia, ...
- uva.

MPEG-7: Anfragen . . .

Beispiel für mögliche Anfragen an MPEG-7 Beschreibungen:

- Text-basierte Suche, z.B. nach Schlüsselwörtern:
- z.B. alle Filme, deren Beschreibung das Wort "MPEG" enthält
- Semantische Beschreibungen
- Suche nach ähnlichen Bildern
- z.B. ausgehend von einer Vorlage des Anwenders
- Suche nach Musikstücken
- z.B. ausgehend von der Melodie oder einem Rhythmus
- Suche nach "low-level" Eigenschaften
- z.B. alle Filme mit charakterischen Objektbewegungen (Trajektorien)

MPEG-7: Leistungsumfang



MPEG-7:

- Definition der Beschreibungen
- inklusive des zugehörigen Datenformats

nicht standardisiert (vorgesehen für späteren Wettbewerb):

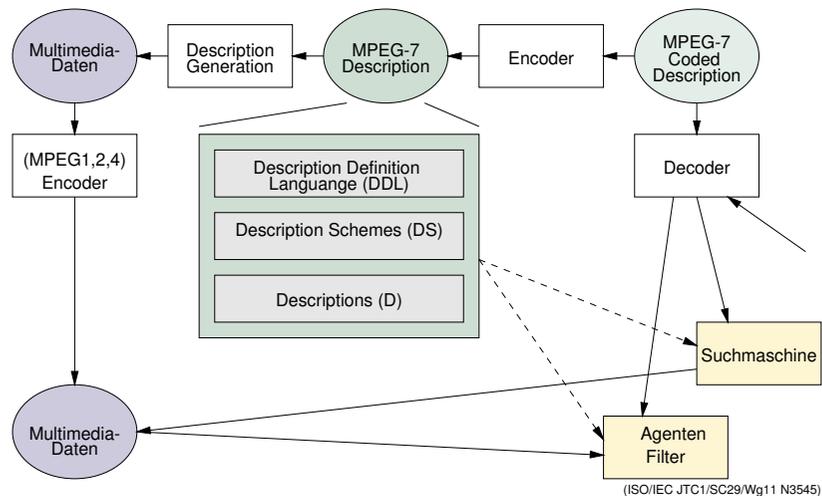
- die Erzeugung der Beschreibungen
Merkmals-Extraktion, Indizierung, Annotation, Authoring, ...
- das Auswerten der Beschreibungen
Suche, Browser, Filter, ...

MPEG-7: Teile

MPEG-7 = ISO/IEC 15938

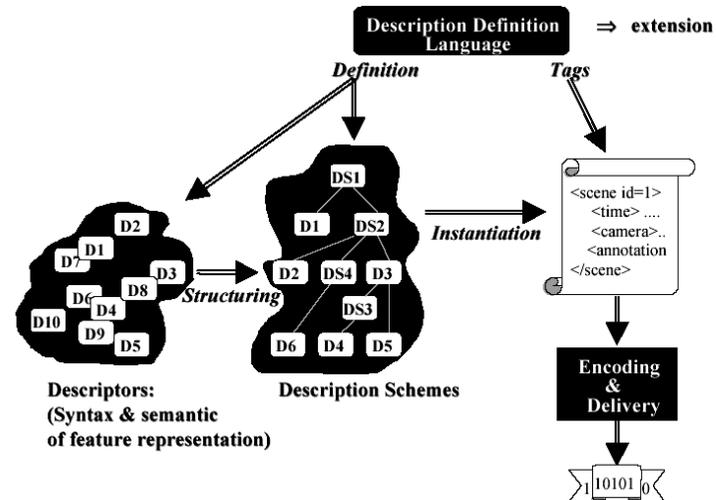
- 1) Systems
- 2) Description Definition Language (DDL)
- 3) Visual
- 4) Audio
- 5) Multimedia Description Schemes (DS)
- 6) Reference Software

MPEG-7: Blockdiagramm



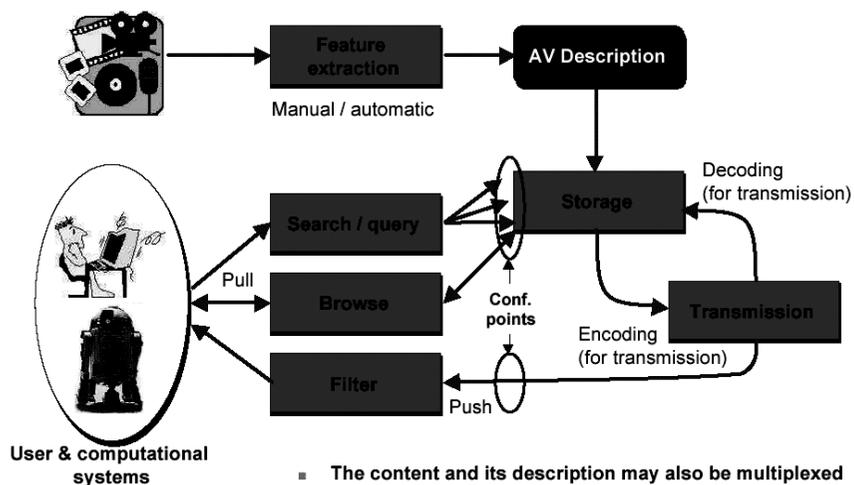
Medientechnik | WS 2001 | 18.204

MPEG-7: D, DS, DDL, Kodierung



Medientechnik | WS 2001 | 18.204

MPEG-7: Informationsflüsse



Medientechnik | WS 2001 | 18.204

MPEG-7: DDL

"Description Definition Language":

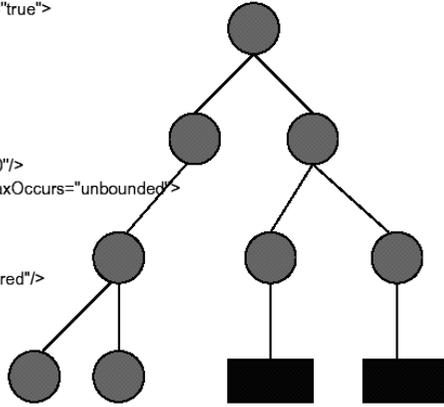
- Data z.B. MPEG-4 Video, CDDA, Word-Dokument
- Descriptor Beschreibung eines Merkmals
- Description Schema Struktur/Semantik von Descriptors
- basiert auf XML
 - einfache Datentypen, Elemente
 - Vererbung, abstrakte Datentypen
- Erweiterungen durch MPEG-7:
 - Array- und Matrix-Datentypen
 - Datentypen für mimeType, countryCode, regionCode, usw.
 - typisierte Referenzen

Medientechnik | WS 2001 | 18.204

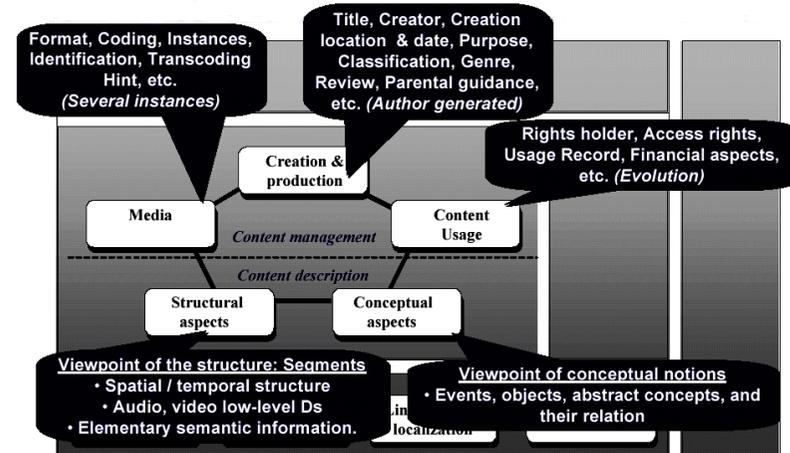
MPEG-7: DDL Beispiel

```

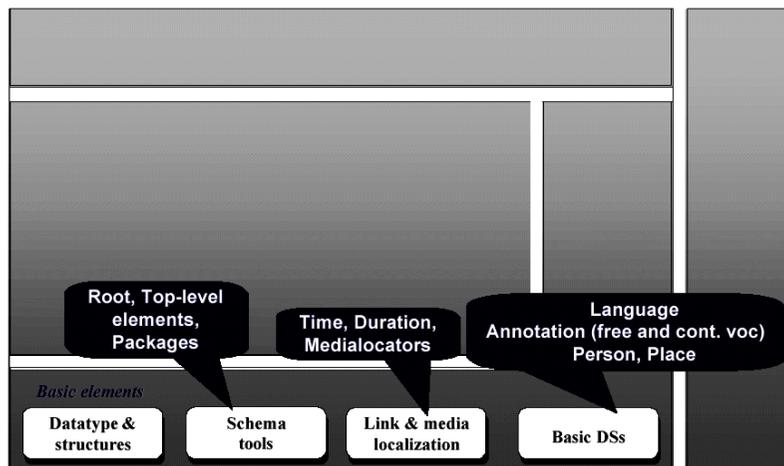
<complexType name="AudioLLDScalarType" abstract="true">
  <complexContent>
    <extension base="mpeg7:AudioDType">
      <choice>
        <element name="SegmentSummary">
          <complexType>
            <sequence>
              <element name="Mean" type="float" minOccurs="0"/>
              <element name="OtherMethod" minOccurs="0" maxOccurs="unbounded">
                <complexType>
                  <simpleContent>
                    <extension base="float">
                      <attribute name="label" type="string" use="required"/>
                    </extension>
                  </simpleContent>
                </complexType>
              </element>
            </sequence>
          </complexType>
        </element>
      </choice>
    </extension>
  </complexContent>
</complexType>
  
```



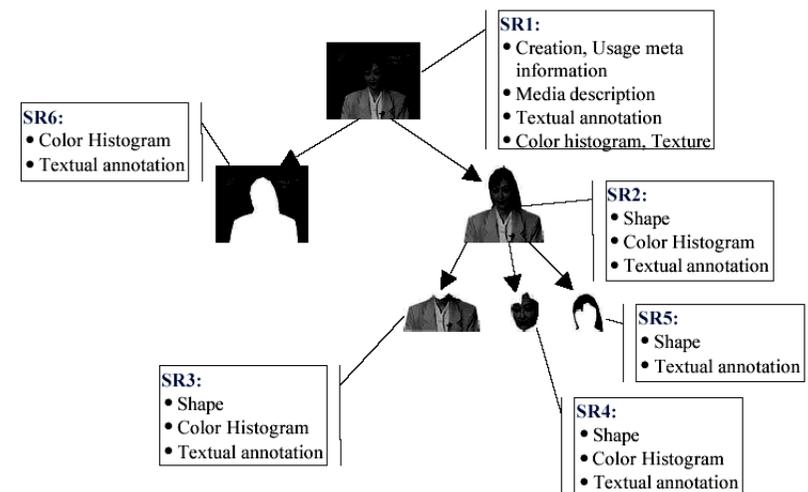
MPEG-7: Content Management



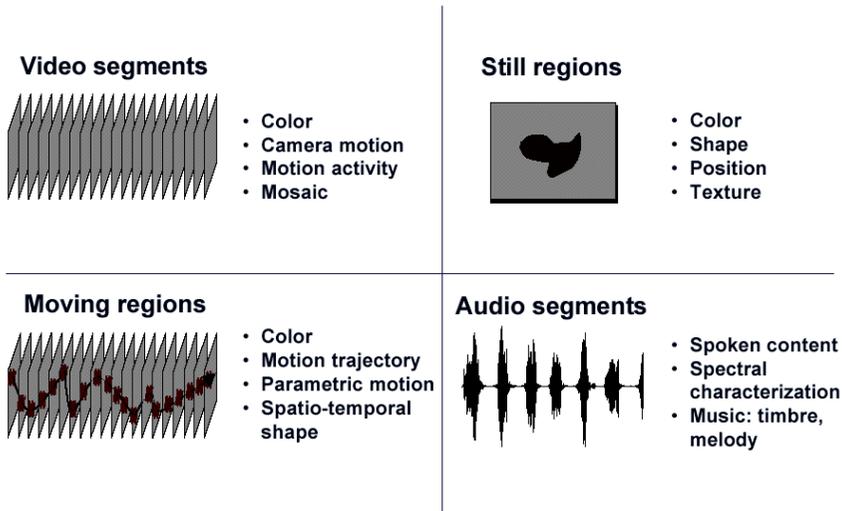
MPEG-7: DDL Basic Elements



MPEG-7: Segment Tree

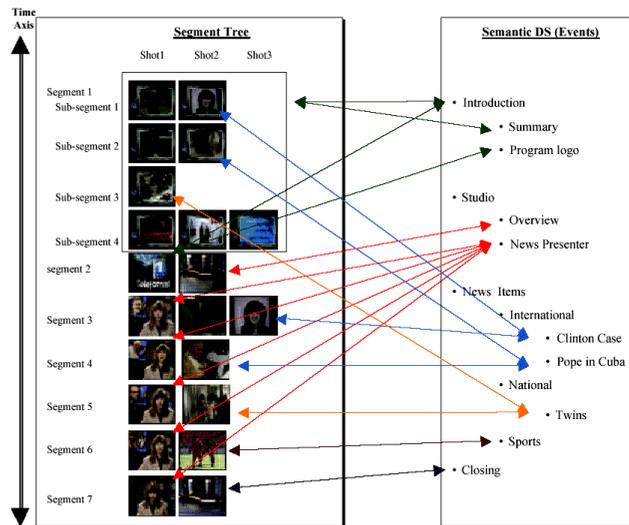


MPEG-7: low-level AV Descriptors



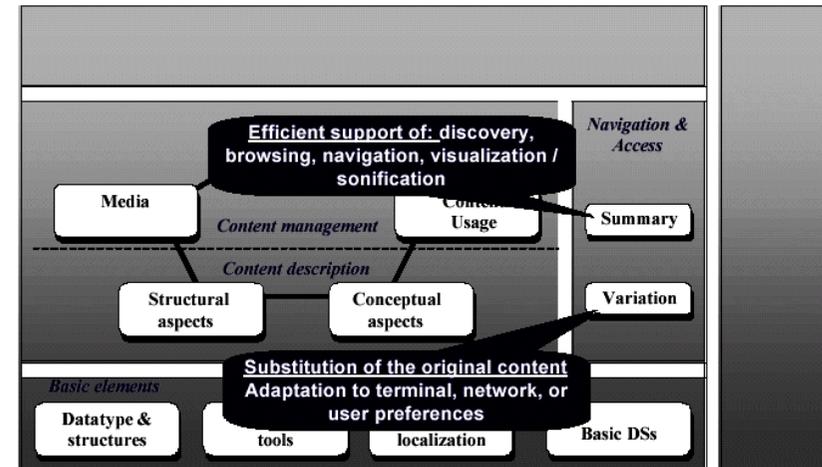
Medientechnik | WS 2001 | 18.204

MPEG-7: Events



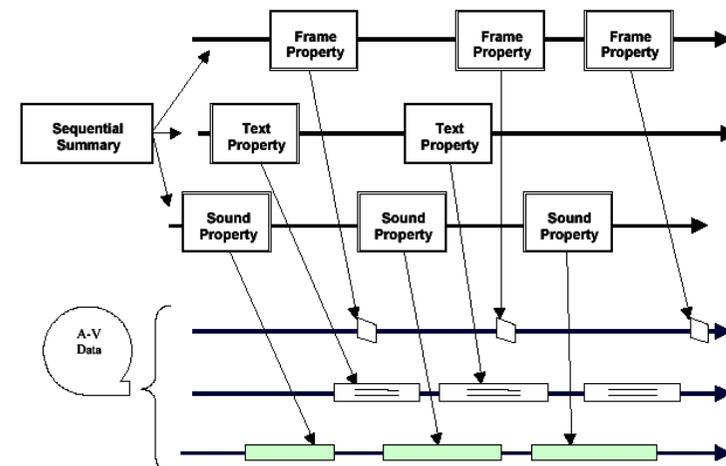
Medientechnik | WS 2001 | 18.204

MPEG-7: Navigation



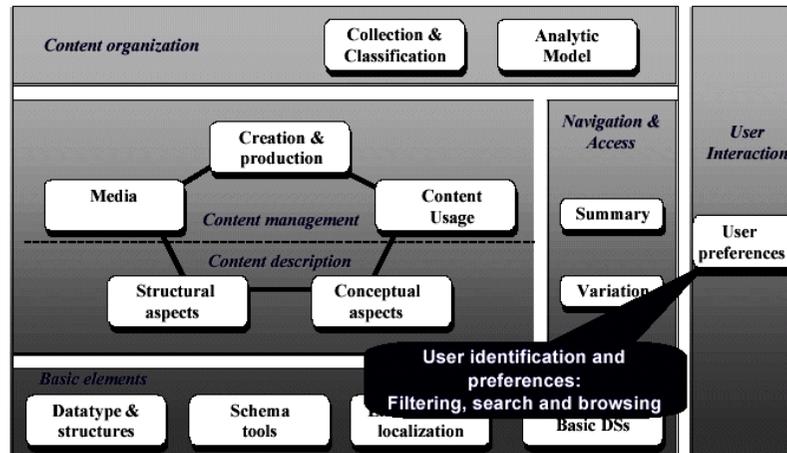
Medientechnik | WS 2001 | 18.204

MPEG-7: Sequential Summary



Medientechnik | WS 2001 | 18.204

MPEG-7: User Interaction



Medientechnik | WS 2001 | 18.204

MPEG-7: Audio

einige aktuell untersuchte Anwendungen:

- Beschreibung von Sprache, Spracherkennung
- Framework zur autom. Erkennung von Audiodaten:
 - Klangfarben, Instrumentfamilien
 - Melodiebeschreibung und -erkennung
 - robuste Identifizierung von Musikstücken
- low-level Merkmale;
 - Wellenformen, Spektrum, Cepstrum
 - Signalparameter, Grundfrequenz, Obertöne
 - Klangfarben, Hüllkurven
 - usw.

Medientechnik | WS 2001 | 18.204

MPEG-7: Melody Description

- Menschen erinnern Musik oft über Melodien
 - Melodien als Vorlage zur Suche nach Musikstücken
 - aber Vorsingen über Mikrophon sehr fehleranfällig:
 - falsche Tonart, anderes Tempo, veränderter Rhythmus, ...
 - Erinnerung nur unvollständig, usw.
- => kompakte und robuste Repräsentation?
- Folge von relativen Tonhöhen ("pitches"):
 - invariant gegen Transponieren und Klangfarbe
 - 5-stufige Werteskala für Tonhöhendifferenz: (-2, -1, 0, +1, +2)
 - robust gegen ungenaues Vorsingen und die meisten Fehler
 - zusätzlich Abspeichern eines (quantisierten) Rhythmus
 - Erzeugen der Beschreibung z.B. aus MIDI-Dateien

Medientechnik | WS 2001 | 18.204

MPEG-7: Beispiel "Moon River"

Contour: 2 -1 -1 -1 -1 1 1
Beat Number: 1 4 5 7 8 9 9 10

```

<!-- MelodyContourDS description of "Moon River" -->
<!-- (7 intervals = 8 notes total) -->
<Contour>
  <ContourData>2 -1 -1 -1 -1 -1 1</ContourData>
</Contour>
<!-- Meter of melody -->
<Meter>
  <Numerator>3</Numerator>
  <Denominator>4</Denominator>
</Meter>
<!-- Beat positions of notes -->
<!-- (8 notes = 1 more than number of intervals) -->
<Beat>
  <BeatData>1 4 5 7 8 9 9 10</BeatData>
</Beat>

```

Medientechnik | WS 2001 | 18.204

MPEG-7: Audio Matching

- inhaltsbasierte Erkennung von Audiodaten
- durch robusten Vergleich mit Referenzdaten
- Robustheit notwendig: Erkennung trotz Anwendung von:
 - lineare Filter (Lautstärke, Filter, Equalizer, ...)
 - nicht-lineare Filter (Kompression, MP3-Kodierung, ...)
 - geschnittenen Daten

Anwendungen:

- gezielte Suche nach bestimmten Musikstücken
- Suche nach ähnlichen Stücken (z.B. E-Commerce)
- "Audio Fingerprinting"
- z.B. zur Überwachung von Verwertungsrechten
- automatische Zuordnung von Metadaten (wie CDDB, ID3v2)

MPEG-7: Audio Matching

aber wie?

- AudioSpectrumFlatness() Descriptor
- beschreibt Spektrum des Audiosignals
- in mehreren Frequenzbändern (z.B. tonal - noise)
- robust gegenüber fast allen Filteroperationen
- sehr kompakt kodierbar, z.B. 4 Werte/s mit 8 bit/Wert
- (aber Binärformat noch nicht in MPEG-7 spezifiziert)

MPEG-7: Audio Matching

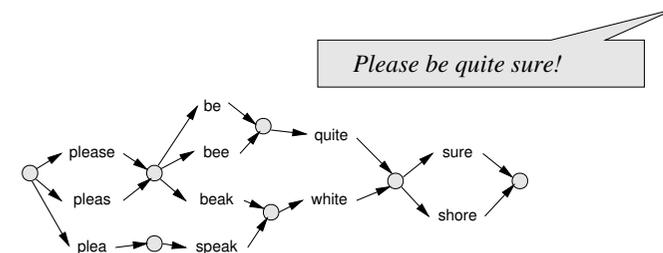
Erkennungsrate des Algorithmus? Demo:

- Testdatenbank mit 15.000 Stücken (Pop/Rock, je 30 sec.)

Originalsignal:	100.0%
Ausschnitt (20 sec.):	99.9%
MP3 (96 kb/s stereo):	100.0%
MP3 & Ausschnitt:	99.7%
Lautsprecher/Mikrophon:	99.3%

- Signaturen insgesamt 15 MByte (1KB/Stück/30 sec.)
- sehr schnelle Erkennung (0.25 sec auf P3-500 / 80x Echtzeit)
- aber: Skalierbarkeit? Verhalten bei sehr ähnlichem Material?

MPEG-7: Speech Description



- Spracherkennung meistens mit HMM (hidden markov models)
- Eingabedaten sehr oft mehrdeutig

MPEG-7 speech description:

- Speicherung der "lattices" anstelle des erkannten Textes
- erlaubt spätere Auswahl der richtigen Deutung

Bild-/Videodatenbanken: das Problem

extreme Datenmengen von Videodatenbanken:

- unkomprimiertes Video kaum handhabbar
- aber welches Kompressionsverfahren ist geeignet?
- Bsp (MPEG-2, 6Mb/s): 90.000 Bilder/h, 3 GB/h
- Archiv eines Senders: 68 GB/day, 24.800 GB/year, 788M frames/year
- ideale Datenbank sammelt viele Sender...
- zum Vergleich: Google derzeit 3G Webseiten (à 100 KB ?!)
- Verwaltung der Daten? Suche in komprimierten Daten möglich?
- Klassifikation der Daten? automatisch oder manuell ...
- Formulierung von Anfragen? Anfragesprachen?
- Browsing der Datenbank? Index, Zusammenfassungen, ...
- Auslieferung der Daten: I/O-Bandbreite, Streaming, QoS?

Medientechnik | WS 2001 | 18.204

TV/Video: Marktbedeutung

Prozentsatz der US-Haushalte mit

- mindestens einem Fernseher: 98 %
- zwei Fernsehern: 34%
- drei oder mehr Fernsehern: 40 %
- mindestens einem Videorekorder 84 %
- "the average American watches 3hrs 35mins of TV each day"

Zahlen für Europa / Deutschland ?!

- kein Wunder, dass die GEZ mir nicht glaubt :-)

(Delp, IMA, www.oc-profam-net.org/media/tv_statistics.htm)

Medientechnik | WS 2001 | 18.204

Interesse an interaktivem TV . . .

- how appealing is interactive TV?

very appealing	14%
somewhat appealing	34%
not very appealing	21%
not at all appealing	29%
don't know / not sure	2%

Ursache / Probleme ?

- reine Konsumhaltung: "Fernseher leergucken"
- Potential wird nicht erkannt, vgl. single- vs. multiplayer Games
- ???

(Angus Reid Group, Red Herring, August 2000, ot of 1000 Americans)

Medientechnik | WS 2001 | 18.204

What do users want?

time-shifting programs	47%
video conferencing	36%
video on demand	35%
getting many more channels	33%
being able to control camera angles	30%
using TV to surf the web	24%
using TV to write and receive email	24%
play games with groups of people who have iTV	14%
shopping over TV	12%

- und was wollen die Anbieter / Sender ?!

(ebenda)

Medientechnik | WS 2001 | 18.204

Bild-/Videodatenbanken: Wozu?

drei Anwendungs-Szenarien:

- Heimanwender-Datenbank
- Video-on-Demand
- Digitale Bibliotheken
- weitere?

Medientechnik | WS 2001 | 18.204

Heim-Datenbanken . . .

Szenario:

- billige Digitalkameras und Videokameras
- jeder hat seinen PC, seine Webseite, seine Kameras
- Erwartung: in 10 Jahren über 90% aller Bilder und Videos digital

=> Markt für "Heim"-Bild- und Videodatenbanken !?

- Suche nach den Hochzeitsfotos / der Einschulung / usw.
- Aufbau von Bildserien / Geschichten ("wie die Kinder wachsen")

Problem:

- mehr als 60 Mrd. Fotos pro Jahr aufgenommen ...
- ... aber jedes Foto weniger als 1 Mal angeschaut

=> Sammlung im Schuhkarton reicht auch in Zukunft aus
=> vermutlich keine Marktbedeutung

Medientechnik | WS 2001 | 18.204

Video-on-demand . . .

Szenario:

- Anwender wollen gezielt nach (Unterhaltungs-) filmen suchen
- Videodatenbank erlaubt die effizienten Suche
- personalisierte Informationen / Präferenzen
- Datenbank zugänglich via WWW oder das DVB- / Kabelnetz

Problem:

- Durchschnittsanwender wählen nach einfachen Kriterien:
Film-Kategorien / Schauspieler / Filmkritiken / Mundpropaganda / ...
- das sind alles Text-Informationen
- keine komplexen Suchfunktionen notwendig

=> sondern nur eine gute Programmübersicht / -zeitschrift

Medientechnik | WS 2001 | 18.204

digitale Bibliotheken . . .

Szenario:

- vernetzte Datenbanken für Schule / Ausbildung
- natürlich auch für (kommerzielle) Recherchen
- erst sekundär auch zur Unterhaltung
- Datenbank wird von Profis (nicht Heimanwendern) verwaltet / gepflegt

Beispiel: Datenbank mit allen Bundesliga-Spielen:

- alle Anwender: Wiederholung interessanter Szenen
- Reporter: Recherche / Vorbereitung von Reports
- Talentscout: Suche nach Talenten
- Fan: "zeig mir das letzte Tor von St. Pauli"
- usw.

=> dieses Szenario könnte (sollte) funktionieren

Medientechnik | WS 2001 | 18.204

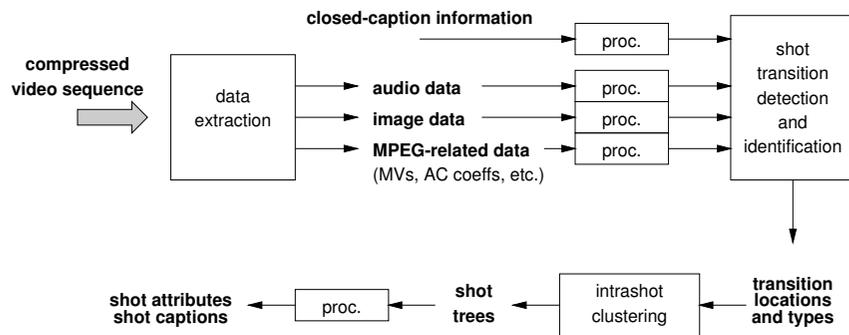
Was ist der "Inhalt" eines Films?



- "Play it again Sam!"
- Man facing woman
- Casablanca
- Ingrid Bergman
- Humphrey Bogart
- Famous movies
- Close-up shot
- "Not Sports"

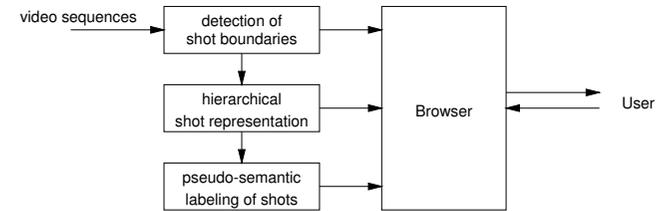
Content is dependent on the particular user group querying the system

Video-Analyse: Beispiel



- automatisches Erzeugen von Szenenbeschreibungen
- direkt aus den (komprimierten) Eingabedaten

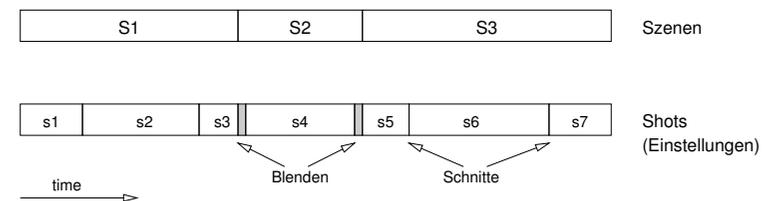
ViBE: Videodatenbank



vier Grundfunktionen:

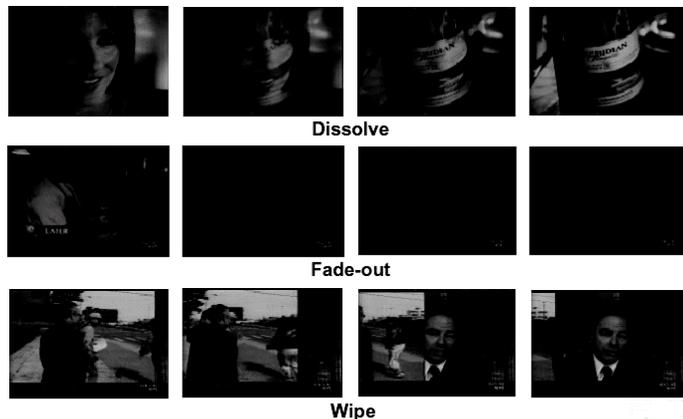
- Erkennung von Szenenwechseln, Erkennung von Szenen
- hierarchische Repräsentation von einzelnen Shots
- pseudo-semantische Benennung von Shots
- interaktives Browsen der Datenbank mit "relevance feedback"
- Framework mit Option zur Integration weiterer Komponenten

ViBE: temporale Segmentierung



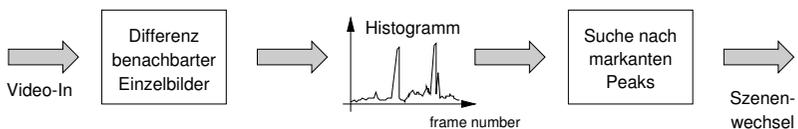
- automatische Auftrennung eines Films in zusammengehörige Szenen
- Zuordnung aufgrund inhaltlicher oder visueller Merkmale
- erfordert die Erkennung von Szenenwechseln (shot boundaries)
- und möglichst auch die Art der Szenenwechsel

ViBE: Szenenwechsel



- harte Schnitte, Überblenden, Ausblenden, Wischblenden, usw.
- Übergänge oft typisch für bestimmte Inhalte / Genres / usw.

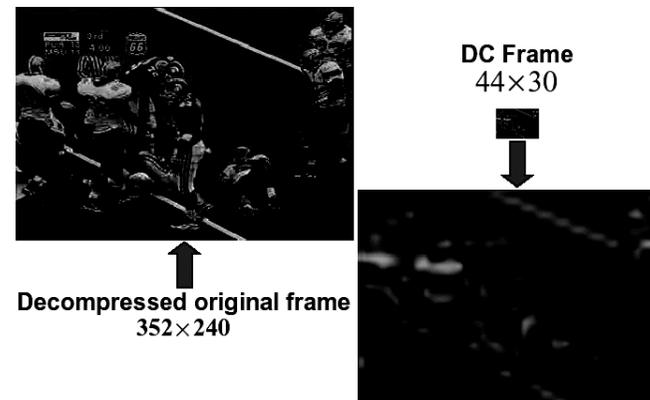
ViBE: Erkennung von Schnitten



diverse Algorithmen vorgeschlagen:

- pixelbasierte Differenz aufeinanderfolgender Einzelbilder
- Grauwert- oder Farbhistogramme
- Kantenerkennung
- Auswertung der "Motionvectors" aus der Bewegungskompensation
- Modellbasierte Verfahren
- Klassifikation der Bildinhalte
- zeitbasierte Verfahren, Adaption an "typische" Szenenlänge

ViBE: DC-Frames



- Berechnung verwendet nur die (MPEG-) DC-Koeffizienten
- dramatisch reduzierte Datenmenge für die Erkennung
- aber Auflösung evtl. zu gering (Details stecken in den AC-Koeffizienten)

ViBE: Testdatensatz

	frames	cuts	dissolves	fades	others
soap opera	67582	337	2	0	0
talk show	107150	331	108	1	6
sports	78051	173	45	0	29
news	58219	297	7	0	6
movies	54160	262	15	6	1
cspan	90269	95	19	0	0
TOTAL	455431	1495	196	7	42

Testdaten mit Videosequenzen, insg. 10 Stunden Video:

- 6 unterschiedliche Genres
- jeweils MPEG-1, 1.5 Mb/s, CIF 352x240
- aus Fernsehaufnahmen (Werbung herausgeschnitten)

ViBE: Performance mit den Testdaten

	Tree Classifier			Sliding Window			Simp. Thresholding		
	Detect	FA	MC	Detect	FA	MC	Detect	FA	MC
<i>soap</i>	0.941	13.3	0	0.916	99	0	0.852	24	0
<i>talk</i>	0.942	32.3	7.5	0.950	45	1	0.968	171	15
<i>sports</i>	0.939	82.5	34.8	0.785	59	1	0.925	251	73
<i>news</i>	0.958	38.0	0.75	0.886	61	0	0.926	212	1
<i>movies</i>	0.821	43.3	2	0.856	25	0	0.816	25	3
<i>cspan</i>	0.915	54.3	8.5	0.994	40	0	0.943	3	20

Fairly constant performance across video genres

- drei verschiedene Algorithmen untersucht
- Tree-Classifer erreicht fast gleichmässige Erkennungsrate
- kein Verfahren ist für alle Fälle optimal

ViBE: "pseudo semantic labeling"

- automatische Klassifikation / Annotation von Szenen
- aufgrund von "mid-" und "low-level" Merkmalen
- insbesondere ohne Bild-"verstehen"
- möglichst gute Korrelation mit "high-level" Beschreibung (Semantik)
- möglichst einfache Berechnung - z.B. ohne Dekompression der Videos

ausgewählte Beispiel-Merkmale:

- "Head-Shoulders" (Sprecher in der Szene - oder nicht?)
- Innen- / Außenszene
- Actionszene (viel Bewegung)
- künstliche / natürliche Umgebung

ViBE: "head shoulders label"



gibt es eine sprechende Person in der Szene ?

- Suche nach "Haut" in den einzelnen Videoframes . . .
- Auswertung von Helligkeit und Chrominanz
- liefert Kandidaten für skin / no-skin Bereiche
- anschließend Segmentierung und Zusammenfassung von Bereichen
- zusätzliche Auswertung von Textur und Bewegungsinformation

ViBE: "skin detection"



- als "skin" erkannte Bereiche nach der Segmentierung

ViBE: "face recognition"

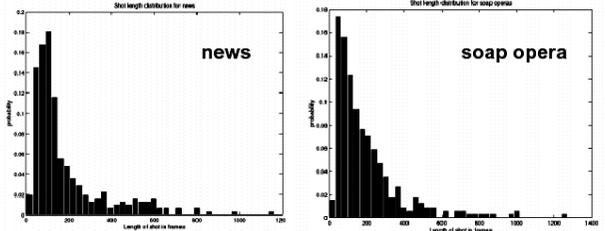
Face Recognition Results

	Shots	Faces	Detect (%)	FA(%)	Correct(%)
news1	231	76	73.7	16.1	80.5
news2	72	29	93.1	23.3	83.3
news3	78	33	87.9	15.6	85.9
news4	103	42	90.5	13.1	88.3
news5	188	51	76.5	13.9	83.5
movie	142	92	84.8	28.0	80.3
drama	100	90	94.4	20.0	93.0
total	914	413	85.2	17.0	84.0

- noch verbesserungsfähig . . .

ViBE: "shot length distribution"

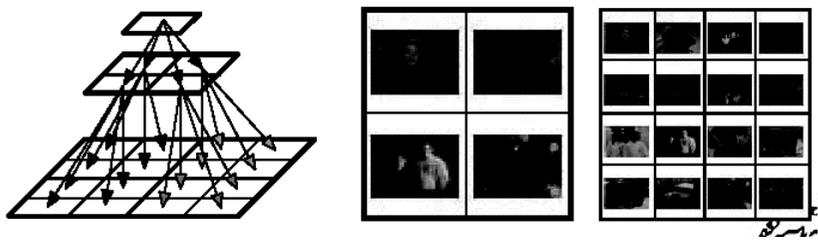
- Shot boundaries are "man-made" according to editing rules
- Shot length is an indication of editing pattern
- Shot length distributions for different genres are different



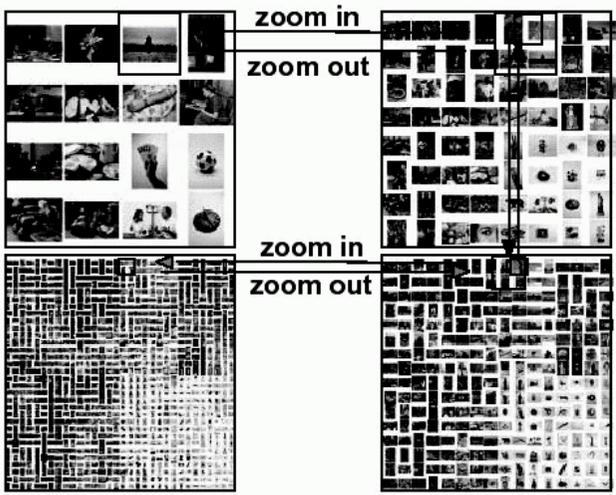
- Erkennung des Genres aus dem Histogramm der Szenenwechsel

ViBE: hierarchische Organisation

- Organize database in a pyramid structure
 - Top level of pyramid represents global variations
 - Bottom level of pyramid represents individual images
- Spatial arrangement makes similar images neighbors
- Embedded hierarchical tree structure



ViBE: Browser und Navigation



MPEG-4 Audio

- Motivation
 - "objekt-basierte" Kodierung
 - Übersicht über MPEG-4 Audio
-
- Structured Audio
 - SAOL - orchestra language
 - SASL - score language
 - SASBF - sample bank format
 - Tools
-
- Einordnung SA vs. MIDI vs. VST
 - Demo

Medientechnik | WS 2001 | 18.204

Literatur

MPEG Homepage, www.cseit.it/mpeg/

MPEG-4 Audio Homepage: sound.media.mit.edu/mpeg4

J. Lazarro, Structured Audio Book, www.cs.berkeley.edu/~lazarro/sa/book/

diverse Konferenz/Workshop-Beiträge und Tutorials auf dem MPEG-Server, u.a.:

Workshop-Vorträge auf dem 106th / 108th AES convention workshop

B. Grill, MPEG-4 scalable audio coding

J. Herre, MPEG-4 general audio coding

E. Schreier, MPEG-4 structured audio

Hendrich, Digitale Audioverarbeitung, tech-www.informatik.uni-hamburg.de/lehre/

Medientechnik | WS 2001 | 18.204

MPEG-4 Audio:

ursprüngliches Ziel von MPEG-4:

- Audio-/Video-Kodierung bei sehr geringen Bitraten
- für mobile Systeme

Paradigmenwechsel während der Entwicklung:

- eigenständige Medien-"Objekte"
- mit entsprechenden Attributen und Methoden
- Framework mit einer Vielfalt von Codecs / Verfahren
- BIFS-Szenengraph zur Integration der Objekte (Raum, Zeit)

für Audio:

- unterschiedliche Codecs für Sprache / Musik / Synthese
- zusätzliche Effekte (Hall, Chorus, ...) und 3D-Audio
- Dekodierung und Mix erst während der Wiedergabe

Medientechnik | WS 2001 | 18.204

MPEG-4 Audio: Übersicht

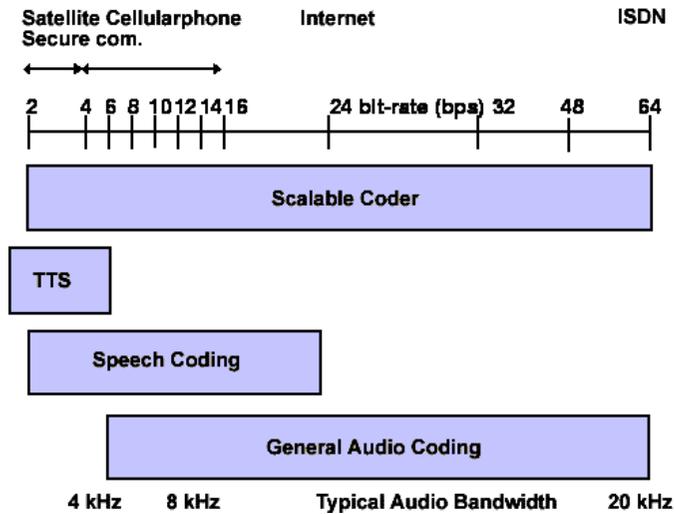
MPEG-4 definiert fünf Audio-Codecs:

- GA: general audio coder, verbesserte Version von MP3/AAC
 - Speech: speech coder, sehr geringe Bitraten
 - SC: scalable audio
 - TTS: text-to-speech
 - SA: structured audio, algorithmische Audiobeschreibung
 - AudioBIFS: Binary Format for Scenes: Integration der Objekte, 3D-Positionierung, Mixer, Effekte
-
- Anwendungen können jeweils geeignetsten Codec auswählen
 - GA-Codec basiert auf den MPEG-1/2 Codecs
 - aber bessere Audioqualität oder geringere Bitraten

(siehe AES 106 und AES 108 workshops, www.cseit.it/mpeg/)

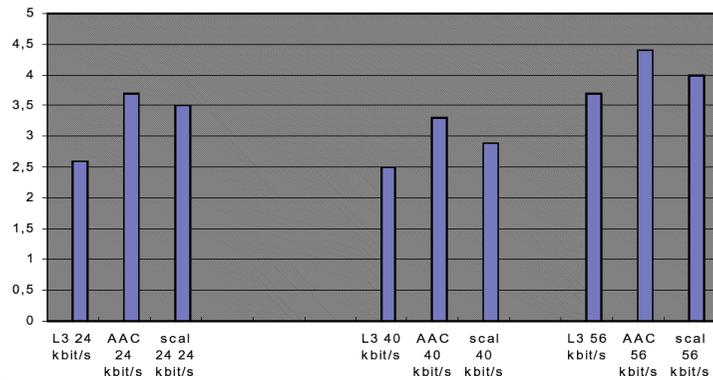
Medientechnik | WS 2001 | 18.204

MPEG-4 Audio: Codec vs. Bitrate



Medientechnik | WS 2001 | 18.204

MPEG-4 Audio: Bitraten vs. Qualität



- Hörtests mit trainierten Testpersonen
- subjektive Bewertung, Skala 5.0 .. 1.0
- CD-nahe Qualität bei 64 kb/s (für stereo)
- im Notfall "akzeptabel" bei 16 Kb/s



Medientechnik | WS 2001 | 18.204

MPEG-4 Audio: Perceptual Noise Substitution

"perceptual coding" Prinzip in MPEG-1/2:

- Ausnutzen der Maskierungseffekte und der Hörschwelle
- Bitzuteilung für Frequenzbänder abhängig von MNR/SNR-Werten
- Unterscheidung tonale / atonale Schallanteile

PNS in MPEG-4 general audio:

- Encoder erkennt Rauschanteile im Signal
- aber keine direkte Kodierung dieses Rauschens
- sondern Parameterextraktion (Amplitude, Spektrum)
- "noise substitution flag" im Bitstream
- Decoder erzeugt passendes pseudozufälliges Rauschen
- extrem kompakte Repräsentation von Geräuschen

Medientechnik | WS 2001 | 18.204

Audio-Synthese:

- Wiedergabe kodierter / komprimierter Daten
- Reproduktion von Musikinstrumenten
 - leichter spielbar
 - billiger, portabel, flexibler, leiser/lauter, ...
 - Rekonstruktion defekter, verlorener Instrumente
- neuartige Klänge
 - Vielfalt der möglichen Algorithmen
 - Erinnerung: Wahrnehmung: Grundton plus Spektrum
- Sprachsynthese

Medientechnik | WS 2001 | 18.204

Musiksynthese: Klassifikation

musikalische Gesichtspunkte:

- Eingabegeräte
- Spieltechniken und -hilfen
- Klangmöglichkeiten
- Echtzeitfähigkeit, Stimmzahl

technische Gesichtspunkte:

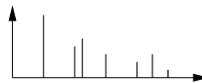
- Syntheseverfahren
- Implementierung
- uva.

verschiedene etablierte Verfahren:

- subtraktiv ("analoge" Synthese)
- additiv ("Orgeln")
- sampling (meistens kombiniert mit Filtern)
- Frequenzmodulation
- physical modelling
- ...

Subtraktive Synthese: Konzept

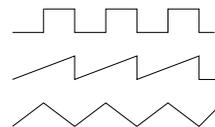
- additive Synthese wegen Kosten unmöglich



subtraktive Synthese:

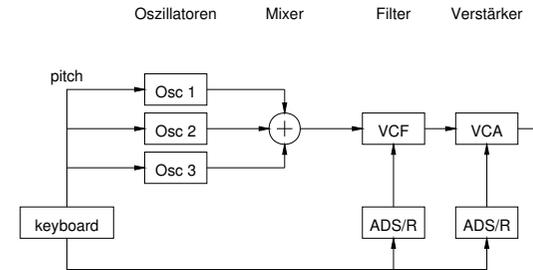
- obertonreiches Ausgangssignal
- gewünschten Klang durch Filterung erzeugen
- zusätzliche Modulation für "lebendigen" Klang

- Oszillatoren mit obertonreichen Klängen:
 - Rechteck, PWM Rechteck, Impulse
 - Sägezahn, Dreieck
 - Rauschen
 - oder unreiner Sinus



- musikalisch sinnvolle Filter, vor allem Tiefpaß
- Anschlagdynamik, Hüllkurven, Unisono, ...

Minimoog: Blockschaltbild

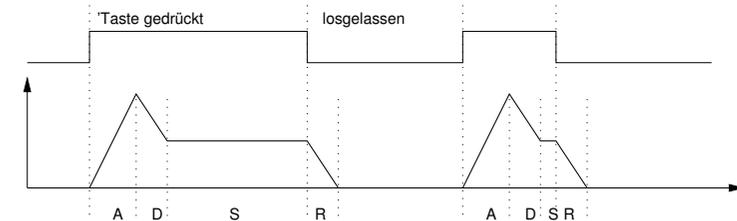


einfacher, aber ausreichend flexibler Signalfluß:

- Vorbild für praktisch alle späteren Synthesizer
- Erweiterungen über flexiblere Oszillatoren
- und weitere Modulationsmöglichkeiten



Hüllkurven: ADSR



- attack Anschlagphase Zeit
- decay Abfallzeit nach dem Anschlag Zeit
- sustain eigentlicher Klang Level
- release Ausklingen Zeit

- bei Bedarf mehrstufige / komplexere Hüllkurven
- attack/decay-Phase besonders wichtig für den Klang

Sampling

- Aufzeichnen von natürlichen Klängen / Geräuschen
- getriggerte Wiedergabe dieser "samples"

in Verbindung mit einem subtraktiven Synthesizer:

- Verwendung der Samples als Ausgangsmaterial
- statt der "einfachen" Oszillatoren
- erfordert ein/mehrere Samples pro Ton
- oder Frequenzanpassung (pitch-shifting) der Samples
- Klangformung via Filter und Verstärker
- Hüllkurven, Anschlagdynamik, usw.
- fast immer digital realisiert (Ausnahme: Mellotron)
- zusätzlich Digitaleffekte

Medientechnik | WS 2001 | 18.204

Additive Synthese

- Erzeugen eines Klangs durch Mischen von Sinussignalen
- zeitliche Modulation durch Amplitudenhüllkurve der Einzelsignale
- direkte Konstruktion des Obertonspektrums
- im Prinzip jeder Klang erzeugbar
- aber sehr hoher Aufwand (je ein Osz./ADSR/Amp pro Oberton)
- entsprechend hoher Kontrollaufwand

(Pfeifen-) Orgel:

- Grundklänge ("Register") nicht nur Sinussignale
- keine Parameter (Lautstärke / Hüllkurven) einstellbar

Medientechnik | WS 2001 | 18.204

Frequenzmodulation

- Ausgang eines Oszillators moduliert die Frequenz eines anderen
- "modulator" and "carrier" signals
- Ausgangssignal enthält Summen- und Differenzfrequenzen:

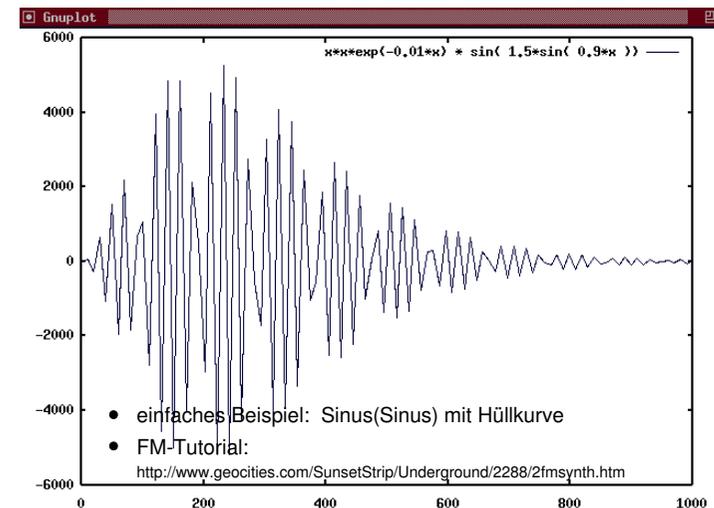
... C-3M C-2M C-M C C+M C+2M C+3M ...

=> eigenständige Syntheseform: FM-Synthese

- Hüllkurven zur Regelung der Oszillatorintensität
- völlig neuartige, z.B. "glockige" Klänge
- populär durch Yamaha DX7, erster rein digitaler Synthesizer
- kein direkter Zusammenhang zw. Parametern und Klang
- kaum zur Reproduktion "natürlicher" Klänge geeignet

Medientechnik | WS 2001 | 18.204

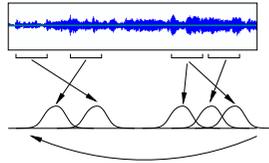
Frequenzmodulation: Beispiel



Medientechnik | WS 2001 | 18.204

Granularsynthese

- Zusammensetzen eines Klanges aus vielen kleinen "Körnchen"
- Ausgangsmaterial sind Samples / Sinuswellen
- Multiplikation mit Fensterfunktionen
- anschließend Aneinandersetzen und Looping
- "Körner": wenn sich einzelne Stückchen nicht mehr überlappen
- sehr viele Parameter zum "Herumspielen"
- daher auch komplex und wenig intuitiv
- Einführung und Literatur: keyboards 01/01 S.74ff
- GranuLab: <http://hem.passagen.se/rasmuse/Granny.htm>



Physical Modeling

- warum überhaupt vereinfachte Syntheseverfahren?
- "physical modeling synthesis"
- Modellierung realer / imaginärer Schallquellen
 - als physikalische Systeme: Saiten, Pfeifen, Resonanzkörper, ...
 - Aufstellen der Bewegungsgleichungen
 - numerische Lösung der Bewegungsgleichungen
 - möglichst in Echtzeit
 - im Prinzip beliebig genaue Simulation jedes Instruments
 - aber extremer Rechenaufwand
 - derzeit meistens auf 1D-Modelle beschränkt (Blasinstrumente)

Structured Audio: Konzept

- algorithmische Beschreibung von Klängen
- mit einer dazu angepassten Programmiersprache
- u.a. durch entsprechendes Zeitmodell (arate / krate)
- Übertragung der Algorithmen anstelle der Klangdaten
- jedes Syntheseverfahren, jeder Effekt möglich
- Decoder führt die Algorithmen aus
- erlaubt perfekte Rekonstruktion der Daten
- aber hoher Rechenaufwand
- bisher keine anwenderfreundlichen Tools verfügbar
- entsprechend wenig Material (Instrumente/Effekte/Musikstücke)
- keine Schutzmechanismen: Instrumente können kopiert werden
- derzeit virtuelle Instrumente fast nur im VST-Format

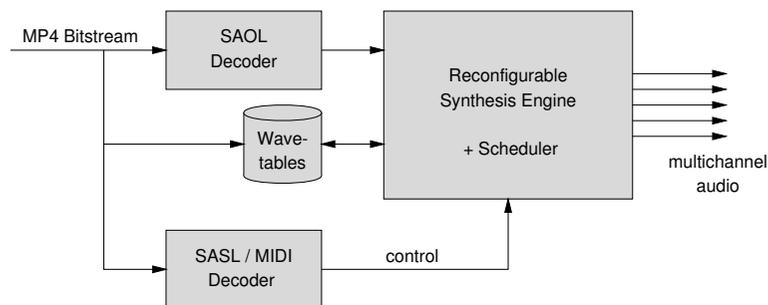
(Vercoe et.al., Structured Audio, Proc. IEEE 86-5, 1998)

Structured Audio: Begriffe

"instrument"	ein einzelner, klangerzeugender Algorithmus
"orchestra"	Gesamtheit aller Instrumente in einem Bitstream
"score"	("Partitur"): die Kontrollparameter für die Instrumente
"score time"	Zeitpunkt eines Kontrollevents (gemessen in Takten)
"control rate"	Aktualisierungsfrequenz der Kontrollparameter
"audio rate"	Abtast- und Aktualisierungsrate der Instrumente
"sample"	PCM-kodierte Audiodaten (z.B. für Wavetable-Synthese)
"opcode"	parametrisierte DSP-Funktion
"bus"	Speicherbereich zum Datenaustausch zw. Instrumenten
"route"	globale Einstellung des Signalflusses
...	

(ISO/IEC 14496-3, 1999)

SA: Blockschaltbild



Bitstream:

- enthält DSP-Algorithmen und Initialisierungen
- Sampledaten, soweit notwendig
- Steuersignale und zugehörige Zeitinformation

(Scheirer, MPEG-4 SA, 106th AES, Munich)

Medientechnik | WS 2001 | 18.204

SA: Bitstream

MPEG-4 SA definiert das zugehörige Binärformat

- "symtable" Symboltabelle
 - "orchestra" SAOL-Instrumentendefinitionen
 - "score" SASL-Kontrolldaten für das orchestra
 - "MIDI" MIDI-Kontrolldaten
 - "sample data" einzelne Audiosamples oder SASBF-Bänke
- einzelne Einträge gekennzeichnet durch "tokens"
 - direkte Navigation ("Vorspulen" usw.) nur eingeschränkt möglich

Auswertung des Bitstroms über globalen Scheduler:

- Initialisierung der globalen Daten (z.B. Abtastrate, Symboltabelle)
- Vorbereitung der "orchestra"-Algorithmen
- Parsen der "score" und "MIDI"-Steuerdaten
- Abspielen

Medientechnik | WS 2001 | 18.204

SA: Bitstream Header

```

class StructuredAudioSpecificConfig { // bitstream header
  bit(1) more_data = 1;
  while (more_data) { // shall have at least one chunk
    bit(3) chunk_type;
    switch (chunk_type) {
      case 0b000: orc_file orc; break;
      case 0b001: score_file score; break;
      case 0b010: midi_file SMF; break;
      case 0b011: sample single_sample; break;
      case 0b100: sbf sample_banK; break;
      case 0b101: symbtable sym; break;
    }
    bit(1) more_data; // 1 = more data, 0 = end-of-stream
  }
}

class SA_access_unit { // bitstream "event data"
  bit(1) more_data = 1;
  while (more_data) {
    bit(2) event_type;
    switch (event_type) {
      case 0b00: score_line score_evt; break;
      case 0b01: midi_event midi_evt; break;
      case 0b10: sample single_sample; break;
    }
    bit(1) more_data;
  }
} ...
  
```

Medientechnik | WS 2001 | 18.204

SAOL

"structured audio orchestra language":

- Sprache zur Beschreibung von Audio-DSP Algorithmen
 - basiert auf CSOUND und Netsound
 - Zeitmodell unterscheidet "arate" und "krate"
 - arate: Audio-Abtastrate, global einstellbar
 - krate: "Kontroll-Rate" für Parameteränderungen
 - Audio-Algorithmen werden einmal pro Abtastperiode berechnet
 - Kontroll-Algorithmen einmal pro K-Periode
- diverse "built-in" Funktionen, u.a.
- stückweise lineare/exp. Interpolation
 - elementare Synthesefunktionen (Sinus, Rauschen, ...)
 - elementare Filter, FFT/iFFT,

Medientechnik | WS 2001 | 18.204

SAOL: Codebeispiel

```
instr beep (pitch, amp) {
  ksig vibfreq, env;           // control signals
  asig sound, f1, f2;         // audio signals

  table vibshape(harm,128,1);  // sine wave table

  vibfreq = cpsmidi(pitch)     // convert MIDI pitch to freq.
    * (1 + koscil(vibshape,5)/40); // add vibrato

  sound = buzz( vibfreq, 0.1, 0.9 ); // predefined noise source
  f1 = bandpass( sound, 500, 100 ); // predefined filter: formants
  f2 = bandpass( sound, 700 + vibfreq*2, 100 ) / 10;

  env = kline( 0, 0.05, 1, dur, 0 ); // linear interpolated envelope

  output( (sound+f1+f2) * amp * env ); // mix together
}
```

(aus Scheirer, 106th AES)

Medientechnik | WS 2001 | 18.204

SAOL: Beispiel "piano"

```
instr piano(pitch, vel) preset 1 92 0 4 { // pitch, vel in MIDI
  ivar cps, scale, t, l, b; ksig off;
  asig samp, env;

  table pitchmap(step, -1, 0, 0, 28, 1, 38, 2, 46, 3, 55, 4, 64, 5, 69, 6, 200);
  table loop(data, -1, 197585, 151849, 137125, 132916, 104286, 59649, 50774);
  table base(data, -1, 21, 36, 41, 51, 61, 66, 71);

  imports table A0, C2, F2, Ds3, Cs4, Fs4, B4; // global definierte Samples
  tablemap pno(A0, C2, F2, Ds3, Cs4, Fs4, B4);
  ...
  if (released) { extend(0.01);
    env = aexpon(1, 0.01, 0); // exp. release envelope
  }
  else { env = 1; } // play sample during attack

  pitch = pitch -12;
  t = tableread(pitchmap, pitch); // which sample for this note?
  l = tableread(loop, t); // read loop settings
  b = cpsmidi(tableread(base, t)); // get sample base freq.
  cps = cpsmidi(pitch); // convert MIDI pitch to freq.
  samp = loscil(pno[t], cps, b, l); // wavetable synth

  scale = (vel/100)*(vel/100); // simple velocity mapping
  env = env * scale;
  output(samp * env * (1-(pitch/100)), samp*env*pitch/100);
}
```

(elpelele.saol)

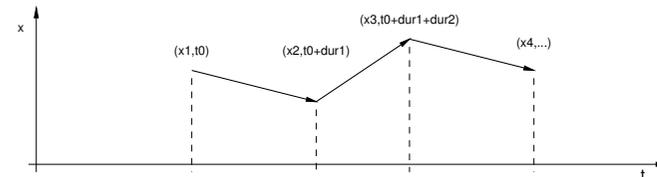
Medientechnik | WS 2001 | 18.204

SAOL: Operatoren, Funktionen

- Datentypen int, float, table, ...
- Arithmetik, mathematische Funktionen (sin, cos, exp, log, ...)
- universelle FIR-/IIR-Filter, diverse vordefinierte Filter
- FFT-/iFFT-Transformation, Fensterfunktionen
- Wavetablesynthese, Unterstützung für DLS-2 (SASBF)
- Granularsynthese, ...
- elementare Signalgeneratoren (Sinus, Rauschen, ...)
- delaylines, Reverb, Chorus, div. Effekte
- universelles Signal-Routing ("bus routing")
- MIDI-Integration und Konvertierungsfunktionen
- stückweise lineare/exp. Interpolation für Audio-/Kontrollsignale
- benutzerdefinierte Opcodes

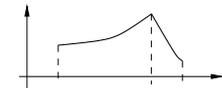
Medientechnik | WS 2001 | 18.204

SAOL: aline, kline



```
kline( ivar x1, ivar dur1, ivar x2 [, ivar dur2, ivar x3, ...]);
aline( ivar x1, ivar dur1, ivar x2 [, ivar dur2, ivar x3, ...]);
```

- built-in Funktion für stückweise lineare Interpolation
- sowohl für Audio- als Controllerdaten
- einfache Beschreibung von Zeitsignalen
- aexpon für Interpolation mit exp()-Funktion
- arand, alinrand, ...: Zufallssignale



Medientechnik | WS 2001 | 18.204

SASL

"structured audio score language":

- separate "minimale" Sprache
- einfaches Textformat, effizientes Binärformat
- Beschreibung von "Kontrollereignissen":
 - globale Tempoangabe (bzw. -änderungen)
 - Erzeugen, Aktivieren, Löschen von Instrumenten
 - Aktivieren von "Noten"
 - Kontrollparameter für Instrumente
- Konzept wie MIDI / SMF
- aber explizite Zeitangaben / keine vordefinierten Controller

Medientechnik | WS 2001 | 18.204

SASL: Beispiel

```
0.000 control ls_freq 100
0.000 control ls_gain 1
0.000 control bf_freq 500
0.000 control bf_gain 1
0.000 control bf_Q 1
0.000 control fad_value 0.4
0.000 control pp_value 0.5
0.000 oscillator 3 440
0.200 control fad_index 0
0.200 control fad_value 0.4
0.300 control fad_index 1
0.300 control fad_value 0.45
1.200 control fad_index 0
1.200 control fad_value 0.3
...
0.200 control pp_index 0
0.200 control pp_value 0.4
0.300 control pp_index 1
0.300 control pp_value 0.6
...
6.000 oscillator 3 220
7.000 control ls_index 0
7.000 control ls_freq 250
7.000 control ls_gain 1.6
9.100 end
```

für jedes einzelne Event:

```
<Zeitangabe>
<Instrument>
<Kontrollparameter>
<neuer Wert des Parameters>
```

Medientechnik | WS 2001 | 18.204

MIDI: Motivation und Aufgabe

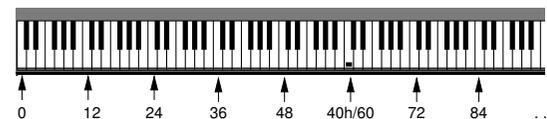
MIDI := "Musical Instruments Digital Interface"

Hardware und Protokoll:

- Spielen von Synthesizern
- Gerätesteuerung / Parametereinstellung / Automation
- "Sequencer" zur MIDI-Aufzeichnung / Editieren / Wiedergabe
- derzeit praktisch alle Musik-Geräte mit MIDI-Schnittstelle
- z.B. Kombination beste Tastatur / bester Klangerzeuger
- auch für weitere Geräte, z.B. Lichtanlagen
- billig und robust (Hardware: RS232 8N1 31250baud optisch)
- extrem kompaktes Dateiformat: Musik mit 1 kbps (!)

Medientechnik | WS 2001 | 18.204

MIDI: Tonhöhe



Konzept der Tonhöhe (pitch):

- Tasteninstrument mit festen Tonhöhen
- wohltemperierte Stimmung
- kodiert als ein Byte, 0 .. 127
- "Schlüssel-C" = 60, C# = 61, D = 62 usw.
- andere Skalen möglich: "tuning"-Meldungen
- Feintuning via "pitch wheel" Meldungen (14 bit Auflösung)

Medientechnik | WS 2001 | 18.204

MIDI: Channel messages

8K	note	dyn	note off	K	channel 0 .. F	
9K	note	dyn	note on	note	pitch 0 .. 127	
AK	note	dyn	poly after touch	dyn.	attack / release	
BK	ID	value	control change	ID	controller-ID	
BK	00	BN	CK	PN	BN	bank number
CK	PN	bank select				
CK	PN	program change				
DK	value	channel pressure				
EK	val	val	pitch wheel change, 14 bit			
F0	Hersteller ID	7 bit data	parity	F7	sysex data	

Medientechnik | WS 2001 | 18.204

MIDI: note on, note off

90	40	7F	80	40	7f	90	41	52	80	41	06
90h	note on		80h	note off		90h	note on		80h	note off	
40h	middle C (=64)		40h	middle C		41h	c#1		41h	c#1	
7Fh	very hard (forte)		7Fh	very hard		52h	medium		06h	very soft	
(hier: alle Meldungen für Kanal 0)											

"keyboard"-orientiert:

- feste Tonhöhen
- Anschlagstärke (attack/release)
- Modulation via "controllers"
- "all notes off" für Paniksituationen

BK	78	00
----	----	----

Medientechnik | WS 2001 | 18.204

SASL: vs. MIDI

SASL:

- keine Beschränkung der max. Datenrate
- gleichzeitige Events möglich
- kontinuierliche, beliebig feine Kontrollmöglichkeiten
- "lesbarer" Quelltext und kompakter Bitstrom
- bisher nicht in Musikinstrumenten implementiert

MIDI:

- der de-facto Standard, riesiger Fundus an MIDI-Dateien
- in allen Geräten/Instrumenten implementiert
- zur Echtzeit-Kontrolle
- aber schlechte Zeitauflösung, geringe Datenrate

=> SA nutzt SASL und/oder MIDI zur Steuerung

Medientechnik | WS 2001 | 18.204

SA: Tools

bisher nur wenige SA-Programme verfügbar:

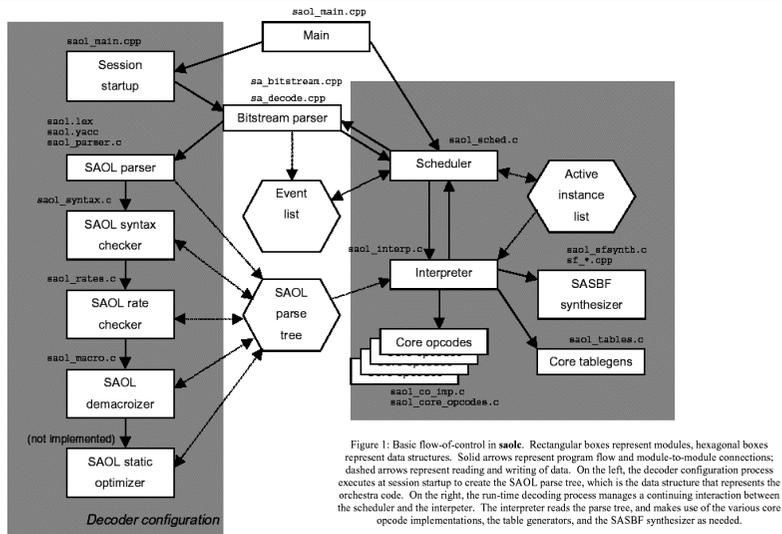
saolc	SAOL/SASL-Compiler, MIT Media Lab Referenzsoftware für den MPEG-Standard
sfront	SAOL/SASL-Compiler, UC Berkeley nutzt C als Zwischenformat und gcc als Compiler
saint	SA INTERpreter, EPFL Lausanne direktes Ausführen von SAOL/SASL-Code

- interaktive Steuerung jeweils nur stark eingeschränkt möglich
- meistens nur WAV/AIFF-Ausgabe, Direktausgabe problematisch
- für Musiker bisher kaum brauchbar
- SA Webseite mit Tools: www.saol.net

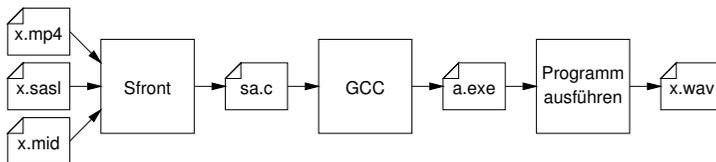
(sound.media.mit.edu/mpeg4, www.cs.berkeley.edu/~lazarro/sa, isiwww.epfl.ch/saog)

Medientechnik | WS 2001 | 18.204

SA: SAOLC



SA: Sfront



- parst SA-Binärformat, erzeugt SAOL/SASL-Quelltext
- übersetzt SASL/SAOL Quellcode nach C
- anschliessend Aufruf des C-Compilers (gcc)
- fertiges Programm berechnet die originalen DSP-Algorithmen
- portabel, Versionen für Unix/Linux/Windows verfügbar
- aber: kein interaktives Arbeiten
- `sfront -orc piano.saol -sco prelude.sasl -bitout prelude.mp4`

SASBF:

"SA sample bank format"

- Beschreibung von Sampledaten für Wavetable-Synthese
- entspricht dem DLS-2 Format (Microsoft DLS, Creative SoundFont2)
- "opaque" Einbettung in MPEG-4 Bitstrom:
 - RIFF-Dateiformat, diverse Chunk-Typen definiert, u.a.:
 - einzelne Audiosamples, zugehörige Loop-Einstellungen
 - Kombination zu Multisamples / Velocity-Layers
 - Filter und Amp-Hüllkurven
 - diverse Effekteinstellungen
- erlaubt Zugriff auf DLS2/SF2 Samplelibraries
- Parametersteuerung über SASL oder MIDI

SoundFont:

- Dateiformat für Samplebanken, Creative Labs 1997
- ursprünglich für Soundblaster 64 AWE
- RIFF-Struktur, insgesamt 22 Chunk-Typen:

RIFF

SFBK

INFO-list

header chunks: name, version, ...

sdta-list

sample data

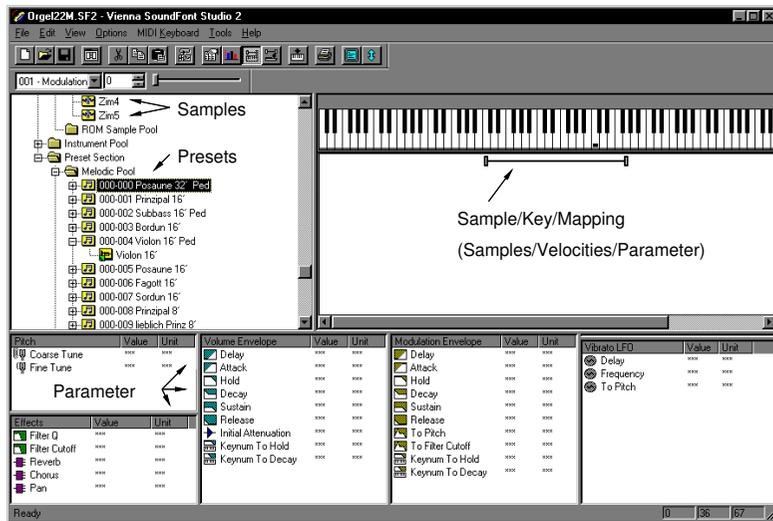
pdta-list

presets (headers, list)

instruments (headers, list)

sample headers

SoundFont: Vienna Editor



Medientechnik | WS 2001 | 18.204

Steinberg: VST

"virtuelle Studio-Technologie":

- Schnittstelle / API für virtuelle Audiogeräte
- Audio- und MIDI-Funktionen, samplegenau
- Integration in Cubase / andere Hostapplikationen
- realisiert als C++ Basisklasse, implementiert für PC/Mac/SGI
- 32-bit Gleitkomma für alle Datenoperationen
- ISSE-Optimierung
- zusätzliche GUI-Wrapper für Oberfläche der Plugins
- minimaler Overhead, optimale Performance
- als Standard etabliert, Dutzende Effekte/Synths/Tools erhältlich

siehe Beispiel:

(www.steinberg.de)

Medientechnik | WS 2001 | 18.204

VST: Beispiel AGain-Plugin

```
#include "AGain.hh"

AGain::AGain( audioMasterCallback audioMaster )
    : AudioEffectX( audioMaster, 1, 1 ) // 1 program, 1 parameter
{
    fGain = 1.0; // default gain 0 dB
    setNumInputs( 2 ); // stereo in
    setNumOutputs( 2 ); // stereo out
    setUniqueID( "AGain" ); // unique name for this plugin
    canMono(); // ok to feed with input with same values
    canProcessReplacing(); // accumulate / overwrite
    strcpy( programName, "default" );
}

void AGain::setParameter( long index, float value )
{
    fGain = value;
}

...

void AGain::process( float **inputs, float **outputs, long n_samples )
{
    float *in1 = inputs[0]; float *out1 = outputs[0];
    float *in2 = inputs[1]; float *out2 = outputs[1];

    while( --n_samples >= 0 ) {
        (*out1++) += (*in1++) * fGain; // accumulating: Mixer
        (*out2++) += (*in2++) * fGain; // should use ISSE/3Dnow
    }
}

...
```

Medientechnik | WS 2001 | 18.204

SA vs. VST

SA:

- Sprache für DSP-Algorithmen, viele Operationen vordefiniert
- standardisiertes, portables Binärformat
- bisher keine (für Musiker) brauchbaren Tools verfügbar
- entsprechend geringe Verbreitung

VST:

- sehr einfache Schnittstelle, nur eine C++ Klasse
- "nackte" Schnittstelle: alle Algorithmen, Synchronisation "per Hand"
- direkter Datentransfer, geringer Overhead, geringe Latenz
- direkte Integration in wichtige Tools (Cubase, Logic, ...)
- volle Echtzeitkontrolle
- Dutzende von Freeware und kommerziellen Synths

=> zukünftige Rolle von SA?

Medientechnik | WS 2001 | 18.204

AudioBIFS:

"Binary Format for Scene (descriptions)":

- hierarchische Beschreibung von Objekten im 3D-Raum
- AudioBIFS für Schallquellen
- mit dem "Sound" Knoten
- 3D-Position und Ausrichtung
- basiert auf VRML "Sound"-Objekt
- aber deutlich erweiterte Möglichkeiten
- zusätzliche Knoten für AudioMix, AudioFX, usw.
- Beschreibung von Sound-Effekten und Mischpult
- Schwerpunkt auf Musikproduktion, nicht nur für 3D-Audio

(Huopaniemi, 106th AES workshop, Munich, 1999)

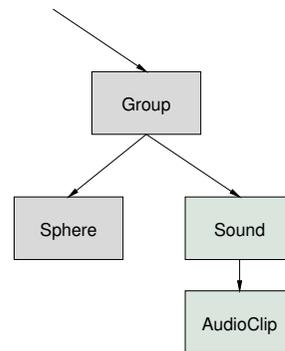
Medientechnik | WS 2001 | 18.204

AudioBIFS: Features

- Sound Schallquelle im 3D-Raum, wie VRML
- Sound2D Schallquelle in 2D
- ListeningPoint Position der Hörers
- TermCap Anpassung an vorhandene Geräte (z.B. 5.1 vs. stereo)
- AudioSource Datenquelle für Sound(2D), inkl. Streaming
Anschluss an jeden MPEG-4 Decoder möglich
- AudioDelay Verzögerung eines Audiosignals
- AudioBuffer Zwischenspeichern von Audiodaten
- AudioMix universelles "Mischpult", M-auf-N Kanäle
- AudioFX Download beliebiger Effekte
Beschreibung in SAOL

Medientechnik | WS 2001 | 18.204

AudioBIFS: VRML Codebeispiel



```

Group {
  children [
    Sphere {
      radius 3
      position 0,0,0
    }
    Sound {
      position 0,0,0
      source
      AudioClip {
        url "http://.../demo.wav"
        loop = 1
      }
    }
  ]
}
  
```

(siehe Huopaniemi, AES 106, Munich)

Medientechnik | WS 2001 | 18.204

AudioBIFS: Codebeispiel

```

instr low_shelving(chan) { // instrument low-pass shelving filter
  ivar pi = 3.1415;
  ksig kk, rgain, freq, dgain, a0, a1, a2, b0, b1, b2;
  ksig first, ki, curr_gain, curr_freq;

  imports ivar ls_freq_init, ls_gain_init; // initial values
  imports ksig ls_freq, ls_gain, ls_index; // control signals
  asig ls_out[inchannels], this_chan[inchannels]; // audio ports
  oparray port[2]; // controller inputs

  ... // initialization code

  // get controller input values, update gain and frequency
  dgain = port[0](curr_gain, 0.02);
  freq = port[1](curr_freq, 0.02);

  // update filter parameters
  kk = sin(pi * freq / s_rate) / cos(pi * freq / s_rate);
  if(dgain >= 1) { // boost case
    a0 = 1 + sqrt(2 * dgain) * kk + (kk * kk * dgain);
    a1 = 2 * (kk * kk * dgain - 1);
    a2 = 1 - sqrt(2 * dgain) * kk + (kk * kk * dgain);
    b0 = 1 + sqrt(2) * kk + (kk * kk);
    b1 = 2 * (kk * kk - 1);
    b2 = 1 - sqrt(2) * kk + (kk * kk);
  } else { ... } // cut case

  // call IIR filter with current parameters, mix
  ls_out[chan] = iir(input[chan], dgain*a0/b0, b1/b0, dgain*a1/b0, b2/b0, dgain*a2/b0);
  this_chan[chan] = input[chan];
  output(this_chan + ls_out);
}
  
```

Medientechnik | WS 2001 | 18.204

Miniaturisierung, MEMS, SmartDust

- Motivation für Miniaturisierung
- Moore's Gesetz und Trends bis ca. 2010
- Mikrosystemtechnik (MEMS)
- Beispiele für aktuelle Mikrosysteme
- mobile Geräte und Stromverbrauch
- Zusammenhänge bei CMOS-Technologie
- theoretische und praktische Limits
- Beispiel Smart-Dust
- Ziele und aktueller Status des Projekts
- Ausblick

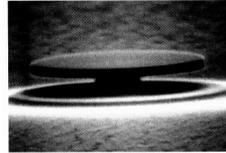


Abb. 1: Einzelne Quantenpunkte in diesem Mikrosensormodul koppeln an diesen Moden und lassen sich zur Emission einzelner Photonen anregen. Die Scherbe hat einen Durchmesser von 5 µm.

Literatur

Introduction to the World of Microsystems (Multimedia CD-ROM), FSRM, 2000

Proc. of the IEEE, special issue on microsystems, 03/1999

Proc. of the IEEE, special issue on semiconductor technology, 03/2001

U. Mescheder, Mikrosystemtechnik, Teubner 2000

U. Hilleringmann, Mikrosystemtechnik auf Silizium, Teubner 1995

Berkeley Smart-Dust Projekt, robotics.eecs.berkeley.edu/

"Grenzen": robotics.eecs.berkeley.edu/~pister/SmartDust/

Compaq Itsy, research.compaq.com/wrl/projects/itsy/

Wearable Computers, IEEE Micro, 03/2001 und 04/2001

Sandia Labs (Mikromechanik), www.sandia.gov/mems/

Skalierung

Table 1 Technology Scaling Rules for Three Cases

Physical parameter	Constant-Electric Field Scaling Factor	Generalized Scaling Factor	Generalized Selective Scaling Factor
Channel length, Insulator thickness	$1/\alpha$	$1/\alpha$	$1/\alpha_d$
Wiring width, channel width	$1/\alpha$	$1/\alpha$	$1/\alpha_w$
Electric field in device	1	ϵ	ϵ
Voltage	$1/\alpha$	ϵ/α	ϵ/α_d
On-current per device	$1/\alpha$	ϵ/α	ϵ/α_w
Doping	α	$\epsilon\alpha$	$\epsilon\alpha_d$
Area	$1/\alpha^2$	$1/\alpha^2$	$1/\alpha_d^2$
Capacitance	$1/\alpha$	$1/\alpha$	$1/\alpha_w$
Gate delay	$1/\alpha$	$1/\alpha$	$1/\alpha_d$
Power dissipation	$1/\alpha^2$	ϵ^2/α^2	$\epsilon^2/\alpha_w\alpha_d$
Power density	1	ϵ^2	$\epsilon^2\alpha_w/\alpha_d$

α is the dimensional scaling parameter, ϵ is the electric field scaling parameter, and α_D and α_W are separate dimensional scaling parameters for the selective scaling case. α_D is applied to the device vertical dimensions and gate length, while α_W applies to the device width and the wiring.

(Proc. IEEE, 03/2001)

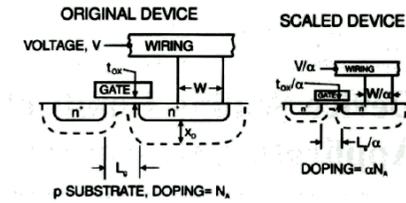


Fig. 1. Schematic illustration of the scaling of Si technology by a factor alpha. Adapted from [5].

Skalierung

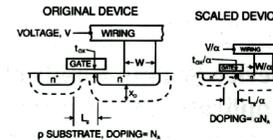


Fig. 1. Schematic illustration of the scaling of Si technology by a factor alpha. Adapted from [5].

Table 1 Technology Scaling Rules for Three Cases

Physical parameter	Constant-Electric Field Scaling Factor	Generalized Scaling Factor	Generalized Selective Scaling Factor
Channel length, Insulator thickness	$1/\alpha$	$1/\alpha$	$1/\alpha_d$
Wiring width, channel width	$1/\alpha$	$1/\alpha$	$1/\alpha_w$
Electric field in device	1	ϵ	ϵ
Voltage	$1/\alpha$	ϵ/α	ϵ/α_d
On-current per device	$1/\alpha$	ϵ/α	ϵ/α_w
Doping	α	$\epsilon\alpha$	$\epsilon\alpha_d$
Area	$1/\alpha^2$	$1/\alpha^2$	$1/\alpha_d^2$
Capacitance	$1/\alpha$	$1/\alpha$	$1/\alpha_w$
Gate delay	$1/\alpha$	$1/\alpha$	$1/\alpha_d$
Power dissipation	$1/\alpha^2$	ϵ^2/α^2	$\epsilon^2/\alpha_w\alpha_d$
Power density	1	ϵ^2	$\epsilon^2\alpha_w/\alpha_d$

α is the dimensional scaling parameter, ϵ is the electric field scaling parameter, and α_D and α_W are separate dimensional scaling parameters for the selective scaling case. α_D is applied to the device vertical dimensions and gate length, while α_W applies to the device width and the wiring.

(Proc. IEEE, 03/2001)

- MOS-Prinzip weitgehend langenenunabhangig
- Verkleinerung ist moglich
- bei Anpassung der Parameter
- kleinere Transistoren
- schnellere Transistoren
- hohere elektr. Felder
- daher anfalliger fur Defekte
- hohere Leistungsdichte
- begrenzt durch Leckstrome
- bessere Transistor-Designs notig

Skalierung: DRAM ...

Year of 1st DRAM Shipment	1997	1999	2002	2005	2008	2011	2014
Minimum Feature Size	250 nm	180 nm	130 nm	100 nm	70 nm	50 nm	35 nm
Isolated Gate Length	-	100 nm	70 nm	50 nm	35 nm	25 nm	18 nm
DRAM Bits/Chip	256M	1G	(3G)	8G	(24G)	64G	(192G)
DRAM Chip Size (mm ²)	280	400	460	530	630	710	860
Equivalent Physical Gate	3-5	1.9-2.5	1.5-1.9	1.0-1.5	0.8-1.2	0.6-0.8	0.5-0.6
Oxide Thickness (nm)							
Dielectric Constant of DRAM Capacitor		22	50	250	700	1500	1500
Max Gate Electrode Resistivity (μΩ cm)		60	43	33	23	16	11
Max Silicide/Si Contact Resistivity ρ _c (Ω cm ²)		30x10 ⁴	17x10 ⁴	10x10 ⁴	5x10 ⁴	2.5x10 ⁴	1.5x10 ⁴
S/D Extension Sheet Resistance (Ω/sq)		350-800	250-700	200-625	150-525	120-450	100-400
S/D Extension x _s (nm)	50-100	42-70	25-43	20-33	16-26	11-19	8-13
S/D Extension Lateral Abruptness (nm/decade)		14	8.5	6.5	4.5	3.2	2.2
Minimum Supply Voltage (volts)	1.8-2.5	1.5-1.8	1.2-1.5	0.9-1.2	0.6-0.9	0.5-0.6	0.5

(Proc. IEEE, 03/2001)

Skalierung: Fab-costs ...

Table 2 Major Characteristics of a Modern Baseline Wafer Fab

Wafer Diameter	200-mm
Throughput	25K wafers/month
Cleanroom Area	10K m ²
Equipment Capital	\$0.5-2B
Support Systems Capital	\$200M
Building Capital	\$150M
Personnel	1000

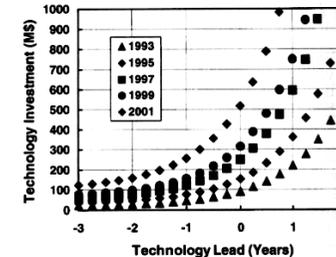


Fig. 6. Annual R&D investments required for technology lag/parity/leadership.

Angaben für eine "mittlere" (baseline) Fabrik:

- Anlagekosten von 750M\$.. 2000M\$
- Betriebszeit ca. 3 .. 5 Jahre, dann Nachrüstung/Neubau erforderlich
- erforderlicher Gewinn >400M\$ / Jahr zum Abdecken der Kapitalkosten
- bei 300K wafers/yr: mehr als 1000\$/wafer

(Proc. IEEE, 03/2001)

Moore's Law: bessere Transistoren

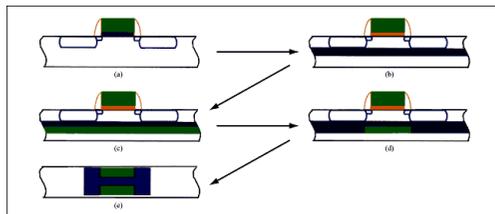


Figure 4

Plausible evolution in transistor structure toward a more symmetric structure that results in better control of the fields in the gate region, regulating device condition. The FETs pictured are: (a) bulk Si, (b) silicon-on-insulator (SOI), (c) ground plane, counter-electrode, (d) vertical double gate, and (e) fully symmetric double gate.

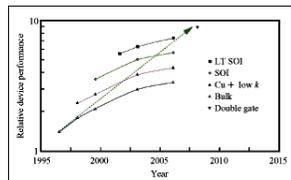


Figure 5

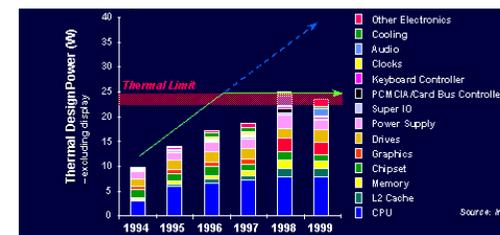
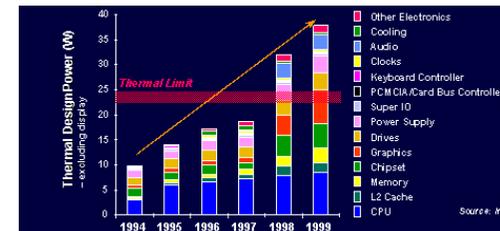
Application of new structures and materials to continue the trend (dashed line) of exponential improvement in device performance vs. time. The transistor structures indicated are bulk Si and double gate. The labels SOI and LT SOI refer to the use of silicon-on-insulator at room temperature or low temperature, while Cu + low k refers to the use of copper metal interconnections with low-dielectric-constant insulators.

Evolution des MOS-Transistors:

- "bulk"
 - "silicon on insulator", SOI
 - "ground plane"
 - "double gate"
- Transistor direkt im Si-Substrat
 dünnes Substrat, darunter Isolationsschicht
 Substrat, Isolator, leitende Schicht: Spiegelladungen
 optimale, symmetrische Anordnung

(IBM JR&D 44-3, 2000)

Leistungsverbrauch: Intel Roadmap

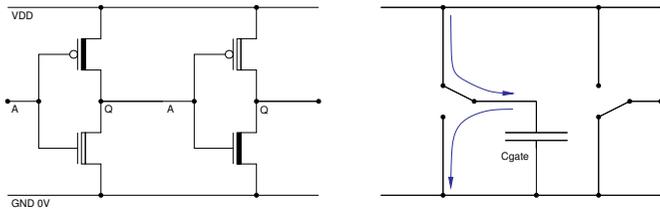


Stromverbrauch von Notebook-Komponenten [Intel IDF 98]

- thermisches Limit 25 W
- Batteriegewicht vs. Laufzeit
- Daten ohne Display, ca. 5..10 W zusätzlich
- Powermanagement
- bessere Batterien?

(developer.intel.com/mobile)

Leistungsverbrauch: CMOS



- (fast) kein statischer Stromverbrauch
- Kurzschluss-Strom beim Umschalten
- Umladen der Gate-Kapazität dominiert, also:

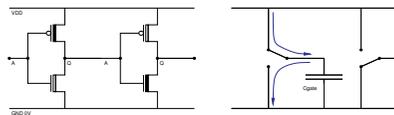
$$Q = C \cdot U$$

$$I = dQ/dt \sim f \cdot C_{gate} \cdot U$$

$$P = U \cdot I \sim f \cdot U^2$$

PC-Technologie | SS 2001 | 18.214

Leistungsverbrauch: CMOS



- kein Ruhestrom
- Stromaufnahme $P \sim f \cdot U^2$

- => Taktfrequenz reduzieren
- => unbenutzte Rechenwerke abschalten
- => Versorgungsspannung reduzieren

(Rechenleistung sinkt)
(Powermanagement)

- Rücksicht auf Systemumgebung
- erfordert angepasste Si-Technologie
- Signallaufzeiten spannungsabhängig:
- reduzierte Betriebsfrequenz

(split voltage μ Ps)
(0.35 -> 0.25 -> ...)

- Intel "SpeedStep", Transmeta "LongRun", AMD "PowerNow"

PC-Technologie | SS 2001 | 18.214

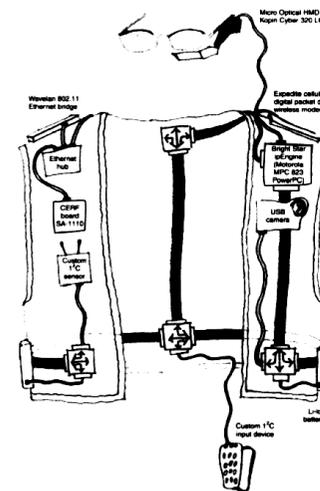
Wearable Computers: Power ...

wearable != portable:

- Vergleich: Armbanduhr vs. Taschenuhr
- jederzeit direktes Ablesen, kein "Suchen und Öffnen"
- erfordert entsprechende Miniaturisierung
- und passende I/O-Komponenten (Tastatur, Displays, ...)
- Gesamtgewicht wird durch Batterien dominiert
- möglichst geringer Stromverbrauch
- stromsparende Prozessoren (StrongArm, Crusoe, ...)
- Bsp: TMS320VC33: 140 MFLOPS @ 200 mW
- intelligentes "context aware" Powermanagement

Medientechnik | WS 2001 | 18.204

MIThril: Komponenten



StrongARM CPU, 206 MHz
mit Chipsatz (Intel SA1110/1111)
128 MByte DRAM
2 MB Flash, 1 GB Microdrive
TMS 320VC33 DSP

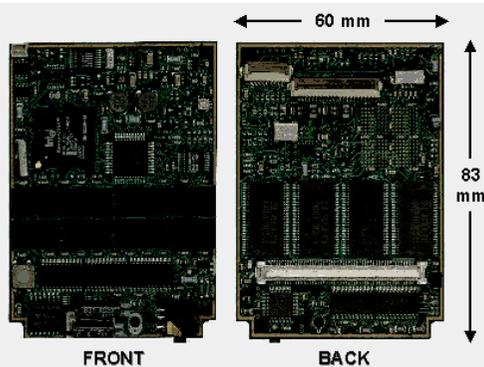
Schnittstellen:
"MicroOptical" Displaybrille
Mikrophon, Lautsprecher
2 Kameras, 256x256 Pixel
IrDA, Netzwerk, Modem, I2C
Tastatur in die Weste integriert
oder separat

(IEEE Micro 06/2001, www.media.mit.edu/wearables/mithril)

Medientechnik | WS 2001 | 18.204

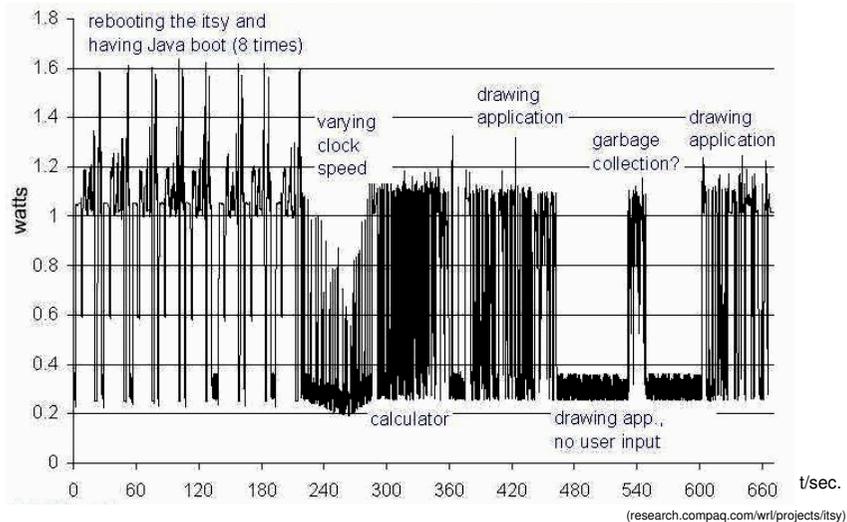
Itsy: Hauptplatine

- ◆ 206 MHz StrongARM
- ◆ 32 MB DRAM
- ◆ 32 MB Flash
- ◆ 2-axis accelerometer
- ◆ Microphone
- ◆ Jacks:
 - headset
 - docking: USB, power, serial
- ◆ Daughtercard connector



Medientechnik | WS 2001 | 18.204

Itsy: Stromverbrauch



Medientechnik | WS 2001 | 18.204

iButton

- Mikrorechner (CPU, RAM, ROM, I/O)
- in kleinem Metallgehäuse / als Fingerring
- bidirektionale 1-Draht Kommunikation
- entsprechend einfache Kontakte ("blue dot")
- sehr robuste Bauform
- optional mit Sicherheitsfunktionen / Verschlüsselung
- "intrusion detection" / Verlust interner Daten beim Öffnen
- Demo: ibutton "James Bond" Video



(www.ibutton.com)

Medientechnik | WS 2001 | 18.204

iButton: Java-Features

Computation Resources of the Java-powered iButton

- 0.6-micron CMOS process operates from supplies of 3V to 5V
- 800,000 transistors speed up the computationally intensive operations of public key cryptography
- High-speed processor core is self-clocked to prevent tampering with program execution
- 32kbyte (64k for R1.1) ROM stores firmware for Java VM and operating system
- 6V SRAM (from 6Kbytes in R1.1 to 134Kbytes in R2.0) writes in 100 nanoseconds to create a high performance operating environment for the Java Virtual Machine
- Math accelerator performs RSA encryption in less than one second using 1024-bit modulus and exponent (R1.1)
- Tamper-resistant True Time Clock used to time stamp transactions and impose expiration dates
- 1-Wire® Net controller allows many iButtons to multi-drop on the same wire and communicate at rates up to 115,200 bits per second
- 1-Wire scratchpad buffer keeps each transaction with the iButton atomic even if the contact with a Blue Dot receptor is an intermittent hot connection

Physical Security

- Armored with stainless steel for the hard knocks of everyday use
- Wear tested for 1 million insertions and more than 10 years of life
- ESD protection is more than 25,000 volts for wash-and-wear dependability
- Three-layer metal technology and flip-chip bonding form barricades to protect data
- Opening of the physical perimeter generates a tamper response
- Tamper response causes rapid zeroization of NV SRAM to prevent disclosure of secure data

Medientechnik | WS 2001 | 18.204

Smart Dust: in the year 2010 . . .

In 2010 radios will cost \$0.10. They will be short range (1-100m), low power (10nJ/bit), and low bit rate (1-100kbps). They will be truly single-chip radios, requiring no external components at [all]. MEMS devices will be used liberally for filters and tuneable oscillators.

In 2010 scanning 3 color laser projection systems will be no larger than a grain of rice, and cost under a dollar. They will be in augmented reality displays that appear to others as regular glasses. They will be in laser pointers, turning any wall into an electronic whiteboard. They will be in large arrays on walls, forming a truly staggering 3D display with brightness, contrast, and viewing angle unparalleled by any technology available or predicted today.

In 2010 everything you own that is worth more than a few dollars will know that it's yours, and you'll be able to find it whenever you want it. Stealing cars, furniture, stereos, or other valuables will be unusual, because any of your valuables that leave your house will check in on their way out the door, and scream like a troll's magic purse if removed without permission (they may scream at 2.4 GHz rather than in audio).

(robotics.eecs.berkeley.edu/~pister/SmartDust/2010.txt)

Medientechnik | WS 2001 | 18.204

Smart Dust: . . . 2010 . . .

. . . In 2010 a speck of dust on each of your fingernails will continuously transmit fingertip motion to your computer. Your computer will understand when you type, point, click, gesture, sculpt, or play air guitar.

In 2010 infants will not die of SIDs, or suffocate, or drown, without an alert being sent to the parents. How will society change when your neighbors pool calls your cell phone to tell you that Johnny is drowning and you're the closest adult that could be located?

In 2010 hunting SCUD TELs in Iraq or T-72s in Yugoslavia will consist of firing up a web browser and proving your authorization.

In 2010 everything of any value that you own will have it's own set of sensors, letting you know when your tire pressure is low, the bridge ahead is out (or unsafe), your milk is going bad, or your water heater is about to die.

In 2010 MEMS sensors will be everywhere, and sensing virtually everything. Scavenging power from sunlight, vibration, thermal gradients, and background RF, sensors motes will be immortal, completely self contained, single chip computers with sensing, communication, and power supply built in. Entirely solid state, and with no natural decay processes, they may well survive the human race. Descendants of dolphins may mine them from arctic ice and marvel at the extinct technology.

Medientechnik | WS 2001 | 18.204

Smart Dust: . . . limits . . .

Acquiring a digital data sample from many sensors requires on the order of 1 nJ. Threshold detection at discrete time periods will require substantially less energy in most cases. Higher performance sensors will require more energy per sample, but the nJ/sample number is applicable to, for example: whisper-to-chainsaw acoustic, sub-degree accuracy temperature, milli- to kilo- gravity acceleration sensing (Which also provides tilt and vibration information), magnetic field to 0.1% of earth's field, barometric pressure to 5m, wind flow to 1 m/s, relative humidity to 2%, ambient light level and spectrum.

Transmitting a bit of data over 10-100 meters by RF today takes approximately 100nJ with Bluetooth, Wavelan, and other local area RF networks. Transmitting a kilometer takes at 10 to 100 microJoules. These numbers are not likely to fall much, since they are often pushed up close to the fundamental physical limits. Another order of magnitude may be available by sacrificing immunity to unintentional jamming from nearby transmitters. If the dynamic range of the radio receivers is reduced, substantial improvements in power can be realized.

Collimated line-of-site optical communication systems will transmit 10m with an energy cost of 10pJ/bit, more than 10,000 times lower than existing radio technology. We have demonstrated 1nJ/bit in the lab already. This incredible gain over RF is due entirely to an antenna gain of roughly 7 orders of magnitude when going from an isotropic radiator to a 1 mrad divergence beam.

32 bit computation currently costs around 1nJ/instruction on power-optimized microprocessors. Engineering limits in the next 5 years or so are approximately 1pJ/instruction for dedicated hardware.

Medientechnik | WS 2001 | 18.204

Smart Dust: . . . limits

Good batteries provide roughly 1 J/mm³. Solar cells provide approximately 100uW/mm² in full sunlight, more than 100nW/mm² in average room lighting. Vibrational energy available in an office setting is in the nW/mm³ range. RF power in a simple antenna is generally not useful, unless there is a cell phone in use in the room, or a dedicated RF power source, in which case microWatts can easily be generated. Conversion is difficult, but feasible.

Assuming a simple task of sampling a sensor, performing some relatively simple processing (threshold, FIR/IIR filtering, statistical analysis, or FFT), listening for incoming messages, and transmitting a simple outgoing message, the energy cost will be a few nanoJoules.

Combining this with the power source information, a cubic millimeter battery will provide enough power to perform such a simple task once a second for 10 years. A cubic millimeter vibrational energy rectifier will operate at that rate forever. Indoors a square millimeter solar cell will provide enough power to perform 100 tasks/second, or in full sunlight 100,000 tasks/second.

For indoor optical line of sight communication, a cubic millimeter battery will provide enough energy to transmit 50 billion bits (roughly half a dozen full-length movies).

Medientechnik | WS 2001 | 18.204

MEMS: Mikrosysteme

MEMS := "Micro- Electro- Mechanical- System"

- Kombination aller Einzelkomponenten:
 - digitale Elektronik (CPU, DSP, ASIC, Speicher, I/O)
 - analoge Elektronik (A/D- D/A-, RF- Komponenten)
 - mikromechanische Sensoren (Druck, Temperatur, Licht, Magnetfeld, ...)
 - mikromechanische Aktoren (Motoren, Pumpen, Spiegel)
- zu einem integrierten Gesamtsystem
- auf Basis der Halbleiter (CMOS-) Technologie
- dramatisch gesenkte Kosten möglich (Beispiel: Airbag für Autos)
- extreme Miniaturisierung (gegenüber diskretem Aufbau)

Medientechnik | WS 2001 | 18.204

MEMS: Bulk- vs. Surface-Technologie

Surface-Micromachining:

- Aufbau von Mikromechanik auf der Materialoberfläche
- z.B. Membranen über flachen Hohlräumen für Drucksensoren
- aufstellbare, bewegliche, flache Spiegel
- flache Hebel- und Zahnradgetriebe
- oft direkt mit bestehenden Technologien / CMOS kombinierbar

Bulk-Verfahren:

- Erzeugen von voll-dreidimensionalen Strukturen
- durch tiefes, anisotropes Ätzen
- oder Manipulation der Wafer-Rückseite (Bsp: Mikropumpe)
- Schichtaufbau aus mehreren Wafern / Kombination mit anderen Materialien
- LIGA-Verfahren - auch für Metallbearbeitung

Medientechnik | WS 2001 | 18.204

MEMS: CMOS-Prozesse (1)

- drei Beispiele für Integration von Sensoren / Aktoren auf CMOS
- jeweils Querschnitt durch digitale Schaltungen (links)
- und mikromechanische Komponenten (rechts)
- UCB, Berkeley:

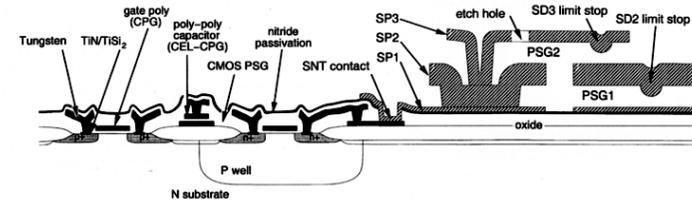


Fig. 4. Cross-sectional illustration of the UC Berkeley Modular Integration of CMOS and Micro-Structures technology. Post-IC MEMS fabrication requires using tungsten interconnect metallization. Electrical connections to the mechanical structure rely on poly-to-poly contacts at the SNT region [23].

Medientechnik | WS 2001 | 18.204

MEMS: CMOS-Prozesse (2)

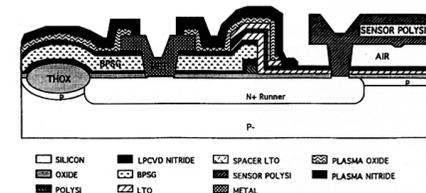


Fig. 5. Cross-section illustration of the Analog Devices BiMEMS integrated MEMS technology. Electrical connections to the structural device are made using the diffused n⁺ runner [25].

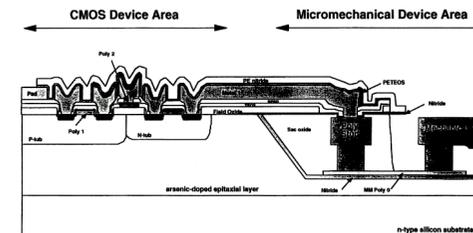
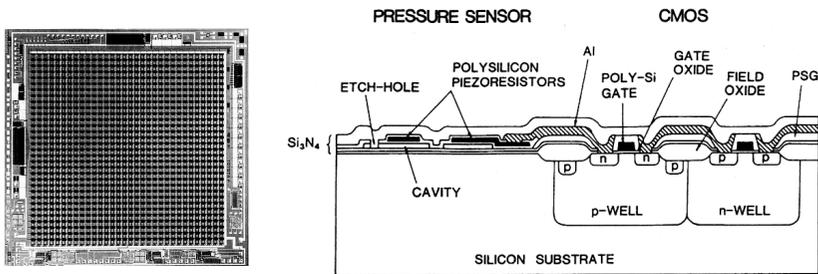


Fig. 6. Cross-section illustration of the Sandia MMCMOS integrated MEMS technology. The mechanical structure fabricated prior to the electronics is embedded down inside a planarized well [54]. Electrical connections are made down through the poly stud and over via the MMCPoly 0 layer.

Medientechnik | WS 2001 | 18.204

MEMS: Drucksensor



- Sensorfeld mit 32x32 integrierten Drucksensoren
- Fläche ca. 10x10 mm²
- integrierte "Fingerspitze", oder auch als Mikrophon-Array, ...
- Druckänderung verbiegt die integrierte Mikromembran
- Messung als Widerstandsänderung in poly-Si Bahnen

(Zitat)

MEMS: Gyroskop

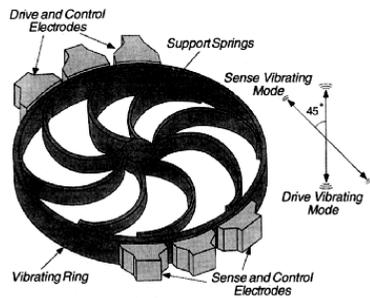


Fig. 15. Structure of a vibrating ring gyroscope.

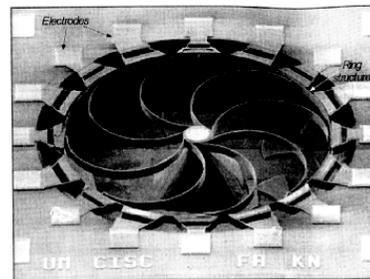
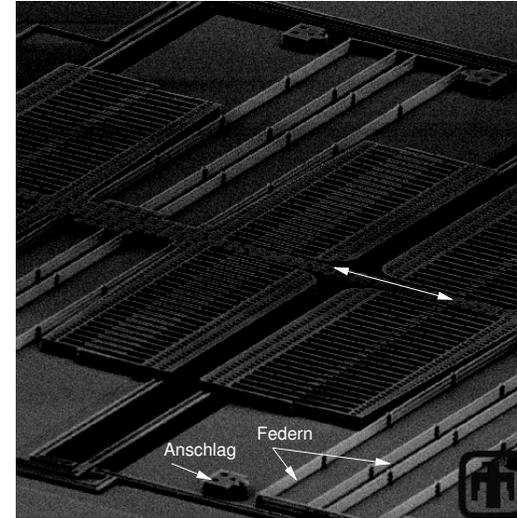


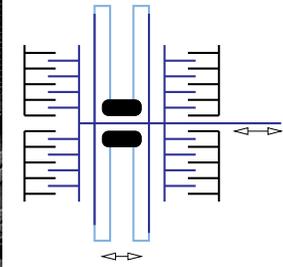
Fig. 16. SEM view of a PRG [123]. The polysilicon ring is 1 mm in diameter, 3 μm wide, and 35 μm tall.

- beweglich aufgehängter poly-Si Ring
- außen angebrachte Sensoren messen Drehbewegungen
- linear bewegte Masse für Beschleunigungssensoren
- Messung von Beschleunigung / Geschwindigkeit / Position

MEMS: Comb Actuator

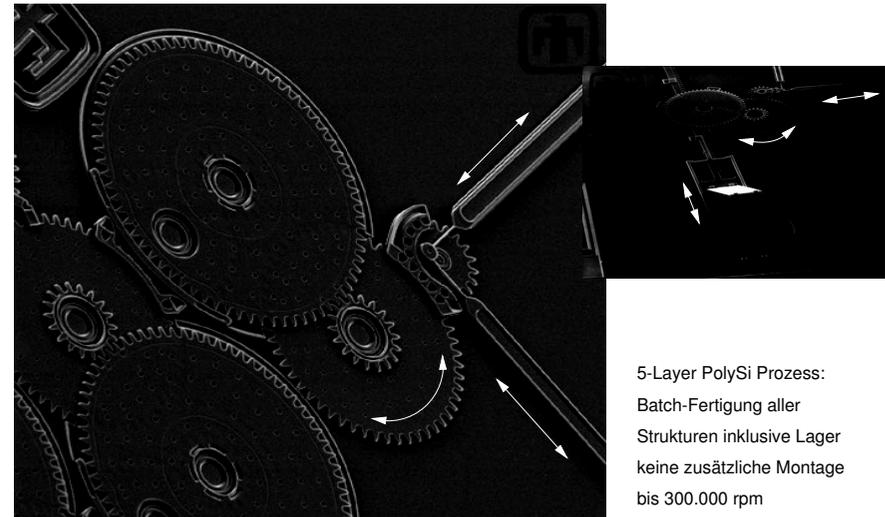


elektrostatische Auslenkung
Kraftvervielfachung durch
die Kammstruktur
Kamm und Federn aus PolySi.



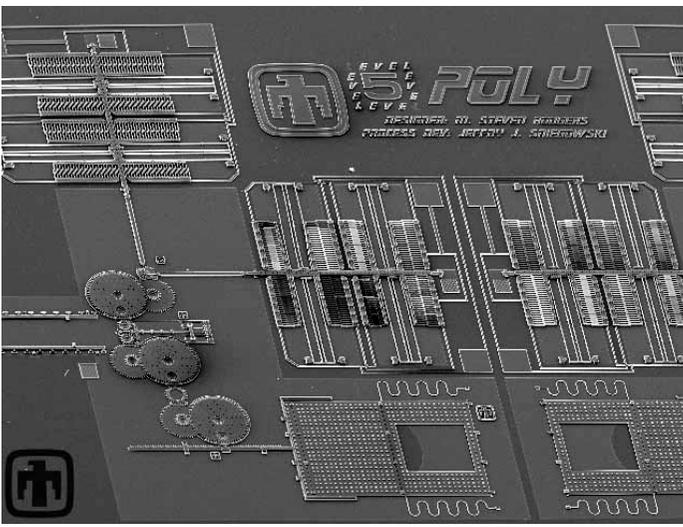
(www.sandia.gov/mems/)

MEMS: Mikrogetriebe



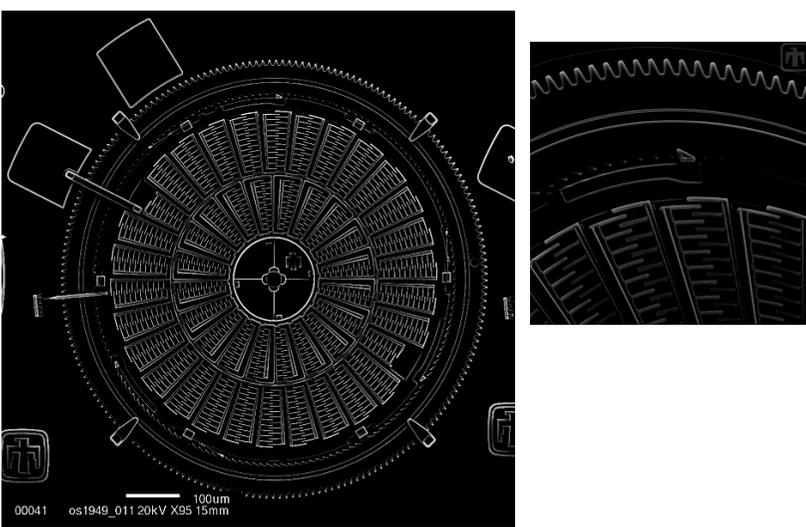
5-Layer PolySi Prozess:
Batch-Fertigung aller
Strukturen inklusive Lager
keine zusätzliche Montage
bis 300.000 rpm

MEMS: Spiegelantrieb, 5L Poly



Medientechnik | WS 2001 | 18.204

MEMS: Mikromotor



Medientechnik | WS 2001 | 18.204

MEMS: optical switch

Flash
Datei Ansicht Steuerung Hilfe

Optical Switch

How it works

Sercalo Microtechnology Ltd.

In optical data communication systems, light is guided by optical fibres.

The switch consists of a double sided mirror that is positioned in the light pathway at the intersection of two pairs of fibres. The mirror's motion is controlled by an electrostatic comb drive actuator.

The complete device consists of

- a silicon structure
- the mirror with the actuator
- the 4 fibres
- an electronic module
- an aluminium housing

30 V

Description How it works Fabrication Applications Economics

FSRM

Examples Technologies Glossary Credits Courses Links Navigation Quit

Medientechnik | WS 2001 | 18.204

MEMS: Spektrometer

Flash
Datei Ansicht Steuerung Hilfe

VIS Spectrometer

FSRM course: Polymer Microfabrication
Forschungszentrum Karlsruhe
STEAG microParts GmbH

How It Works

The light to be measured is coupled into a planar waveguide formed by a three-layer plastic part.

The light waves propagate in the middle layer and reflect on a grating mirror which separates the wavelengths and focuses them on a linear photodiode array.

The video signal provided by the photodiode array gives the spectral information of the incoming light.

100µm

Description How it works Fabrication Applications Economics

FSRM

Examples Technologies Glossary Credits Courses Links Navigation Quit

Medientechnik | WS 2001 | 18.204

Smart Dust

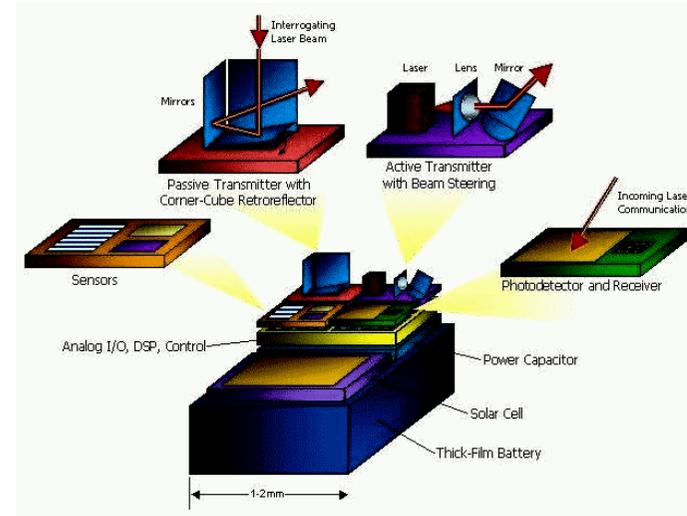
Berkeley "Smart Dust" Projekt:

- Integration kompletter Rechensysteme auf 1 mm³
- vollständiger Digitalrechner (CPU, RAM, ROM, I/O)
- inklusive Betriebssystem (Tiny OS)
- inklusive autonome Vernetzung
- inklusive Sensoren (z.B. Photodioden, Magnetfeld, Gyro.)
- inklusive Kommunikation (Funk oder optisch, s.u.)
- inklusive Stromversorgung (Photodioden, Batterie, Vibration, μ Generatoren)
- Tausende vollständig autonome Mikrorechner
- "Ausstreuen" in der Umgebung
- minimale Kosten, riesiges Anwendungsspektrum

(Berkeley Sensor & Actuator Center, eecs.berkeley.edu)

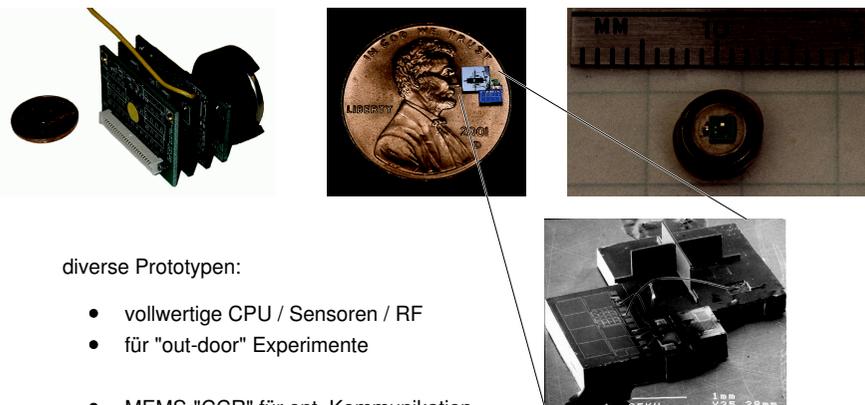
Medientechnik | WS 2001 | 18.204

Smart Dust: Konzept für das Gesamtsystem



Medientechnik | WS 2001 | 18.204

Smart Dust: Prototypen



diverse Prototypen:

- vollwertige CPU / Sensoren / RF für "out-door" Experimente
- MEMS-"CCR" für opt. Kommunikation
- Stromversorgung via Batterie oder Photodioden

Medientechnik | WS 2001 | 18.204

Smart Dust: Grenzen der Miniaturisierung?

Miniatur-Solarzellen
Wirkungsgrad ca. 3%
26 μ W/mm² in vollem Sonnenlicht



- Batterien: ~ 1 J/mm³
- Kondensatoren: ~ 10 mJ/mm³
- Solarzellen: ~ 0.1 mW/mm² ~ 1 J/mm²/day (außen, Sonne)
 ~ 10 μ W/mm² ~ 10 mJ/mm²/day (innen)
- Digitalschaltung: 1 nJ/instruction (StrongArm SA1100)
- Analoger Sensor: 1 nJ/sample
- Kommunikation: 1 nJ/bit (passive transmitter, s.u.)
- opt. digitale ASICs: ~ 5 pJ/bit (LFSR Demonstrator, 1.4V)

Medientechnik | WS 2001 | 18.204

Smart Dust: Kommunikation

- welche Art der Datenübertragung ?!
- möglichst bidirektionale Kommunikation

Funktechnik ("RF radio"):

- ausgereifte Technologie
- sehr effizient, kaum weitere Steigerung möglich -
Bsp. Satellitenkommunikation: nahe am physikalischen Limit
- Richtantennen in Mikrosystemen aus Platzgründen nicht möglich
- hoher Verlust durch isotrope Abstrahlung

optische Wellenlängen:

- wesentlich bessere Richtwirkung / längere Reichweite
- Photodioden leicht integrierbar / Kombination mit Spannungsversorgung
- Laserdioden und "active beam steering"
- passive Modulation eines eingestrahlichten starken Signals (CCR)



Medientechnik | WS 2001 | 18.204

Smart Dust: Laser-Kommunikation

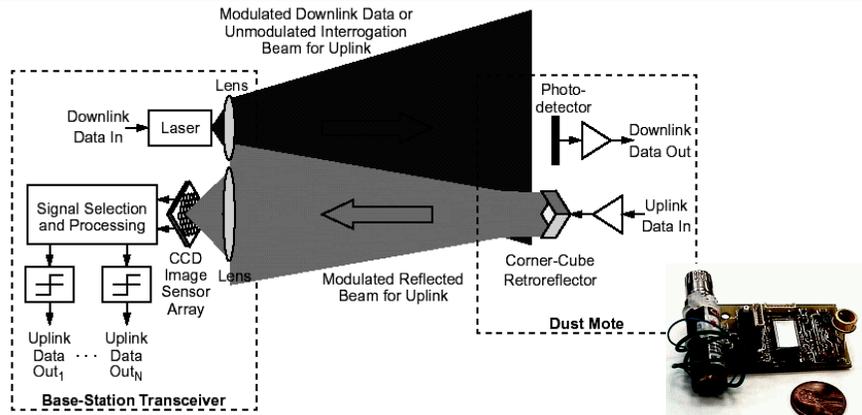
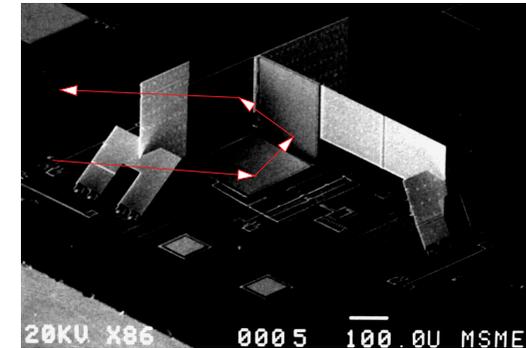
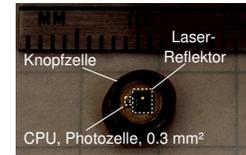


Fig. 3. Design of a free-space optical network in which a base-station transceiver communicates simultaneously with a collection of many dust motes (only one dust mote is shown). A single laser at the base station supplies optical power for the downlink and the uplink.

- asymmetrische Aufteilung: starke Basisstation, schwache Mobilgeräte

Medientechnik | WS 2001 | 18.204

Smart Dust: Corner Cube Reflektor



CCR- Retroreflektor: ("Katzenauge")

- 2 starre Spiegel (Gold auf Si), Aufstellen "per Hand"
- untere Spiegelfläche beweglich (elektrostatisch, ca. 30V)
- gezielte Modulation von eingestrahlt (Laser-) Licht
- Reichweiten >100m demonstriert

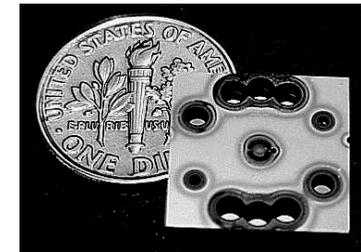
(robotics.eecs.berkeley.edu/~pister/SmartDust/)

Medientechnik | WS 2001 | 18.204

Smart Dust: Energieversorgung

mobile und autonome Geräte:

- (Netzbetrieb)
- Batterien / Akkus
- Photozellen / Photodioden
- Vibrationsenergie
- Thermoelemente
- Mikroturbinen, Brennstoffzellen



(Mikroturbine für Wasserstoff-Verbrennung)

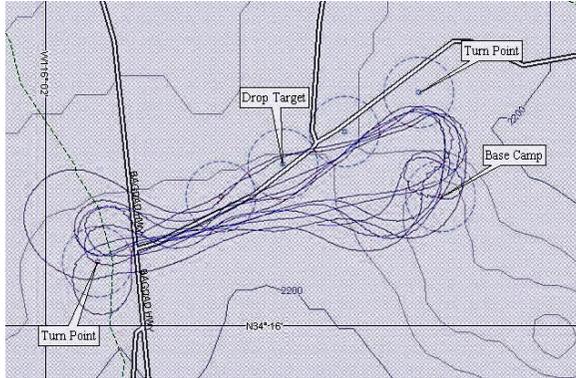
- evtl. das grösste Problem vor weiterer Miniaturisierung
- vgl. Notebooks/PDAs: Laufzeit vs. Gewicht

Einsparpotential ?

- Logikoperationen: Reduktion um mehrere Größenordnungen denkbar
- Datenübertragung (Funk): bereits nahe am physikalischen Limit

Medientechnik | WS 2001 | 18.204

SmartDust: Demonstration



- Abwurf von fünf Demonstrationsrechnern vom Modellflugzeug
- pro Rechner: Magnetfeldsensor, Radiosender/emfänger, Batterie (5h)
- autonome Vernetzung, Messung von LKW-/Panzerbewegungen

- Leerseite

- Leerseite

Cyborgs . . .

Why not build a TV camera with zoom lens into our eyes, allowing magnification of distant scenes or vision through infrared, the better to see at night? Why not amplifiers in our ears? And why not recorders capable of saving all that we have heard, seen, or even felt for later recall? Never again does inattentiveness have to mean missing something. Linger over the interesting parts of life, fast-forward through the boring parts. Add some pattern recognition and never again would we be at a loss for words or forget a name or face. "Why John, good to see you again. Your wife Elaine had knee surgery; how did it go? And your son Peter is about to have a birthday..."

These enhancements are apt to be impressive. Memory aids, calculators, decision aids, implanted dictionaries and translators. Arithmetic calculators. Recognize that athletes enhance their bodies to add strength. If it is possible to increase muscle power, why not brain power? Today, we test athletes in an ever-more difficult attempt to eliminate drug-enhanced performance. Some day we may have to do full X-ray (3D tomographic) scans in an ever-more difficult attempt to detect artificial implants. Why? Because it is possible. Today, we have not yet achieved the necessary miniaturization, but we can see how to get there. Power is still a problem, but it will be solved. At first, enhancements are apt to be implanted for medical reasons, but when people discover

that an artificial eye is better than the real one, or that the memory chip that remembers events, names, and facts can be ever-present, the trend will be unstoppable. Order your implant today. The major remaining hurdle is the control circuitry. How does one communicate with an implanted circuit? Imagine it and it shall come to pass! Not really.

Linger over the interesting parts of life, fast-forward through the boring parts.

The brain communicates with itself through a complex, highly parallel communication process involving the firing patterns of neural impulses, biochemical stimulation that bathes the brain structures with highly tuned molecular structures, and methods as yet unknown. Just how information is stored, regenerated, and interpreted within brain circuits remains a major mystery, one unlikely to be solved soon. Sure, we can record neural firings from tens of neurons. Sure we can stimulate neurons to create crude sensory images and coarse motor control. But full-fledged, precise control of the expert performer eludes understanding—which means the behavior of all of us,

for we are all experts in doing our everyday activities like walking, talking, seeing, and feeling. Note how easy it is for computers to perform tasks we find difficult, such as arithmetic calculations and precise memory, how difficult to perform tasks we find trivial, such as walking and talking, throwing and seeing, understanding and creating. The future holds major changes in computers as they become specialized devices with specialized forms, meaning we will interact with them in an increasingly rich set of natural ways. At the same time, the future promises major changes in humans, as technologies are embedded within our bodies, altering our capabilities. Not only will the devices we use have increased power and dynamic response, but the way they interact with people will be more natural, more complex, and more powerful. An incredible variety of new devices will emerge, most not yet conceived. Many will find their way into the human body, to amplify and transform our sensory, mental, and motor capacities. Do you think the current concerns over privacy violations and personal autonomy are large and complex? You haven't seen anything yet. ■

DONALD A. NORMAN (don@ind.org) is a co-founder of the Nielsen Norman group (www.nngroup.com) and president of UNext Learning Systems, a distance education company (www.unext.com). Copyright held by author.

Why not build a TV camera with zoom lens into our eyes, allowing magnification of distant scenes or vision through infrared, the better to see at night? Why ...

...
Do you think the current concerns over privacy violations and personal autonomy are large and complex? You haven't seen anything yet. ■

DONALD A. NORMAN (don@ind.org) is a co-founder of the Nielsen Norman group (www.nngroup.com) and president of UNext Learning Systems, a distance education company (www.unext.com).