

Großer Beleg

über das Thema

Suchverfahren für Konverterketten - Bewertung und Klassifikation

vorgelegt von

Marc Fiedler

email: marc.fiedler@inf.tu-dresden.de

angefertigt an der

Professur für Datenbanken
Institut für Systemarchitektur
Fakultät Informatik
Technische Universität Dresden

betreuende Hochschullehrer: Prof. Dr. Klaus Meyer-Wegener
Prof. Dr. Hermann Härtig
betreuender Mitarbeiter: Dipl.-Inf. Andreas März

Selbständigkeitserklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Ich bin damit einverstanden, dass die Arbeit veröffentlicht und dass in wissenschaftlichen Veröffentlichungen auf sie Bezug genommen wird.

Der Technischen Universität Dresden, vertreten durch die Professur Datenbanken, wird ein (nicht ausschließliches) Nutzungsrecht an dieser Arbeit sowie an den im Zusammenhang mit ihr erstellten Programmen eingeräumt.

Marc Fiedler
Dresden, den 31.05.2002

Inhaltsverzeichnis

1	Einleitung	1
2	Audio	2
2.1	Digitalisierung analoger Audiosignale	3
2.2	MPEG	4
2.2.1	MPEG-1 Audio	5
2.2.2	MPEG-2 Audio	5
3	Video	7
3.1	Farbmodelle	8
3.1.1	Das RGB-Modell	8
3.1.2	Das YUV-Modell	10
3.1.3	Das YCbCr-Modell	10
3.2	Interlaced / Noninterlaced	11
3.3	MPEG	13
3.3.1	MPEG-1 Video	13
3.3.2	MPEG-2 Video	15
4	Ausgewählte Konvertieroperationen	16
4.1	Transformationen	16
4.2	Spektrale Filter	16
4.2.1	Tiefpass-Filter	16
4.2.2	Hochpass-Filter	17
4.3	Interlaced-to-Noninterlaced Wandlung	17
4.3.1	Scan Line Duplikation	17
4.3.2	Scan Line Interpolation	17
4.3.3	Field Merging	17
4.4	Noninterlaced-to-Interlaced Wandlung	18
4.4.1	Abwechselndes Auslesen	18
4.4.2	Interpolation	18
4.5	Farbraumwandlung	18
4.6	Koordinatentransformation	19
4.6.1	Bildspiegelung	19
4.6.2	Bilddrehung	19
4.6.3	Bildverschiebung	20
4.7	Größenänderung	20
5	Klassifikation von Konvertern	22
5.1	Konvertermodell	22
5.2	Logische Klassifizierungsmerkmale von Konvertern	24
5.3	Logische Klassifikation	25

5.4	Physische Klassifizierungsmerkmale von Konvertern	29
5.5	Physische Klassifikation	30
5.6	Beispiele	33
5.6.1	Farbraumwandlung RGB zu YCbCr 4:2:2	33
5.6.2	Interlaced-to-Noninterlaced Wandlung	34
5.6.3	MPEG Konvertierung	34
6	Suchverfahren für Konverterketten	36
6.1	Umfassende Suche	39
6.2	Lokale Suche	40
6.2.1	Simulated Annealing	41
6.2.2	Tabusuche	42
6.3	Teile & Herrsche	42
6.4	Dynamisches Programmieren	43
6.5	Branch & Bound	44
6.6	A* Algorithmus	45
6.7	Evolutionistische Verfahren	46
7	Zusammenfassung & Ausblick	48

Abbildungsverzeichnis

1	Dimensionen eines Audios	2
2	Dimensionen eines digitalen Videos	7
3	RGB-Farbwürfel	9
4	YCbCr Abtastformate	11
5	Interlaced Bilddarstellung	12
6	Konvertermodell	22
7	Grobstrukturierung eines Medienobjektes	24
8	Logische Klassifikation	26
9	Physische Klassifikation	32
10	Beispiel Dynamisches Programmieren	44
11	Beispiel Branch & Bound	45

Tabellenverzeichnis

1	Bitraten der 3 MPEG-1 Layer	5
2	Standardauflösungen	20
3	Zusammenfassung Logische Klassifikation	29
4	Physische Klassifikation (deterministisch)	31
5	Physische Klassifikation (indeterministisch)	33
6	Beispiel Suchraumgröße	37
7	Beispiel Suchraumgröße	37

1 Einleitung

Im Rahmen des Projektes **memo.REAL**¹ wird eine Architektur für ein Medienobjektspeicherungssystem entwickelt. Dieser Medienserver soll in der Lage sein, verschiedene Medienobjekte wie z.B. Bild-, Video- und Audiodaten gemeinsam zu speichern und zu verwalten. Ziel ist es, die Medienobjekte geräte- und formatunabhängig zu speichern. Die Geräteunabhängigkeit, d.h. die abstrakte Sicht auf Geräte, ist Stand der Technik bei Datenbanken und Betriebssystemen. Dagegen ist Formatunabhängigkeit heute nur selten implementiert. Formatunabhängigkeit bedeutet die Unabhängigkeit des Datenzugriffs vom internen Speicherungsformat. Dadurch wird die Flexibilität eines solchen Systems erheblich vergrößert, so können eine Vielzahl von Parametern, insbesondere das Format zwischen Server und Client ausgehandelt werden. Das bedeutet, dass das interne Speicherungsformat und das von einem Client unterstützte Format unterschiedlich sein können.

Die Realisierung der Formatunabhängigkeit stellt eine Herausforderung dar, bedeutet sie doch, dass der Medienserver in der Lage sein muss, von einem internen Speicherungsformat entsprechende Konvertierungen zu einem geforderten Ausgabeformat vorzunehmen. Unter Umständen sind mehrere Konvertierungen, die über verschiedene Zwischenformate zum Ausgabeformat führen, nötig. Dafür ist eine Folge von Konvertern, eine sogenannte Konverterkette, zu durchlaufen.

Besonderes Augenmerk bei der Entwicklung wird auf die Echtzeitfähigkeit dieses Systems gelegt, was gerade im Hinblick auf die Verwaltung von Video- und Audiodaten von Interesse ist. Das Medienobjekt kann in einem solchen Fall nicht als Ganzes verarbeitet werden. Echtzeitfähigkeit erfordert die Kooperation des Medienservers mit anderen Systemen wie dem Betriebssystem. Auf der Ebene des Betriebssystems ist die Reservierung von Ressourcen wie Speicher und Prozessorzeit möglich. Insbesondere für die Suche von Konverterketten in Verbindung mit den Medientypen Audio und Video, die Gegenstand dieser Arbeit ist, gewinnt die Einhaltung von QoS-Garantien² an Bedeutung. Der Medienserver muss eine funktional korrekte und einplanbare Konverterkette finden, die es ermöglicht, das interne Speicherungsformat des Medienobjektes in ein gewünschtes Ausgabeformat zu überführen.

Dafür ist es notwendig, den Suchraum für Konverterketten sinnvoll einzuschränken. Dazu sollen in dieser Arbeit geeignete Klassifizierungsmerkmale für Konverter entwickelt werden.

In den folgenden 3 Abschnitten werden zuerst einige Aspekte der Audio- und Videokodierung betrachtet. Anschließend werden ausgewählte Konvertieroperationen vorgestellt. Im zweiten Teil dieser Arbeit werden klassische als auch alternative Algorithmen diskutiert und ihre Eignung zur Suche nach funktional korrekten und einplanbaren Konverterketten bewertet.

¹media object encoding by multiple operations in realtime

²Quality of Service

2 Audio

Der Begriff Audio umfasst alle Arten von hörbaren Signalen. Der Bereich vom Menschen wahrnehmbarer Frequenzen erstreckt sich von 30 Hz bis 15 kHz [Lut97]. Die obere und untere Grenze aller hörbaren Frequenzen variiert zwischen verschiedenen Individuen und in Abhängigkeit zur Lautstärke sehr stark. Deshalb decken technische Anwendungen zum Beispiel im HiFi Bereich einen größeren Frequenzbereich ab. Wodurch auch eine Verbesserung der Qualität bei der Reproduktion bzw. Übertragung solcher Signale erreicht wird.

Physikalisch gesehen wird Schall durch Luftdruckschwankungen, sogenannten Schallwellen unterschiedlicher Frequenz und Amplitude, repräsentiert. Diese Schwankungen des Luftdruckes werden vom Gehör wahrgenommen. Die Frequenz charakterisiert die Tonlage und die Amplitude die Lautstärke.

Um Signale wie gesprochenes Wort oder Musik über zeitliche oder räumliche Distanzen zu transportieren, müssen sie in eine andere Form umgewandelt werden. Bei vielen Anwendungen findet eine Wandlung der Schallwellen in elektrische Signale statt. Diese Arbeit leisten im Audiobereich z.B. Mikrophone. Man kann also Schall durch kontinuierliche elektrische Signale repräsentieren. Dies ist für viele Arten der Verarbeitung, Übertragung oder Speicherung unerlässlich. Eine häufige Repräsentation von Schall sind Werte elektrischer Spannungen über der Zeit. Eine äquivalente Darstellung ist die des Frequenzspektrums, welche bei der Fourier-Transformation entsteht. Diese Signale sind zunächst zweidimensional (Abbildung 1).

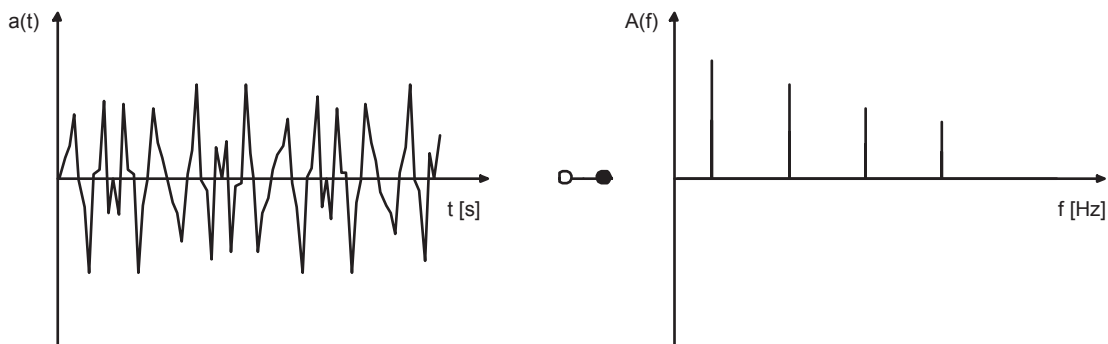


Abbildung 1: Dimensionen eines Audios

Die Verarbeitung von Audiosignalen im Computer setzt eine Digitalisierung voraus, da im Computer nur diskrete Signalwerte verarbeitet werden können. Dieser Vorgang wird Analog-Digital-Wandlung genannt. Umgekehrt setzt die Rekonstruktion digitaler Audiosignale in Schall eine Digital-Analog-Wandlung voraus.

2.1 Digitalisierung analoger Audiosignale

Die Analog-Digital-Wandlung hat zum Ziel, analoge Signale in eine von Rechnern verarbeitbare Darstellung zu bringen. Dazu werden Signale im Zeit- und Wertebereich diskretisiert, mit dem Ziel diese wieder fehlerfrei rekonstruieren zu können. Bei der Analog-Digital-Wandlung entsteht jedoch Informationsverlust. Die Grenze, wieviel Information mindestens erhalten werden muss, legt das Abtasttheorem nach Shannon fest. Es besagt, dass ein kontinuierliches Signal mindestens mit dem Zweifachen der grössten im Signal enthaltenen Frequenz f_G (obere Grenzfrequenz) abgetastet werden muss.

Beim Abtasten werden in äquidistanten Abständen Werte aus dem Signal entnommen. Das Abtasttheorem macht Angaben, wieviele Abtastwerte pro Sekunde mindestens benötigt werden. So gilt für die Abtastfrequenz f_A die Beziehung $f_A \geq 2 \cdot f_G$. Um z.B. ein Signal mit einer oberen Grenzfrequenz f_G von 22 kHz abzutasten, benötigt man nach Shannon eine Abtastfrequenz f_A von mindestens 44 kHz, d.h. es müssen mindestens 44000 Abtastwerte pro Sekunde entnommen werden.

Ist die Abtastfrequenz zu klein gewählt, kann das ursprüngliche Signal nicht fehlerfrei rekonstruiert werden. Diesen Effekt nennt man Aliasing³.

Um Aliasing zu vermeiden, ist eine vorherige Bandbegrenzung des Signals auf $f_G = \frac{f_A}{2}$ erforderlich. Dazu werden analoge Tiefpass-Filter mit steilen Flanken benötigt. Diese sind in der Praxis sehr schwer zu realisieren. Eine Möglichkeit, die Konstruktion von solchen genauen Filtern zu umgehen, ist das Oversampling, d.h. die Überabtastung. Hierbei wird das Signal mit einer wesentlich höheren Abtastfrequenz als nötig abgetastet. Gängige Abtastfrequenzen liegen dann 2 - 64 mal höher als nötig. Die Überabtastung erleichtert die Konstruktion von Antialiasingfiltern, da das Basisband des Signals und die beim Abtasten entstehenden periodischen Frequenzbänder um Vielfache der Abtastfrequenz einen größeren Abstand haben. Nach Abtastung und Quantisierung wird das Signal durch ein digitales Filter (Decimation-Filter) bandbegrenzt und auf die gewünschte Abtastfrequenz gebracht.

Durch die Abtastung entsteht ein PAM-Signal⁴. Es ist zeitdiskret und wertekontinuierlich. Der Zeitdiskretisierung muss sich noch eine Wertediskretisierung anschließen, um eine Verarbeitung mit heute gängiger Computertechnik zu ermöglichen. Dieser Vorgang wird als Quantisierung bezeichnet. Die Anzahl der Quantisierungsstufen bestimmt dabei den Signal-Rauschabstand. Im oberen Audiobereich (HiFi) sind 16 Bit zur Kodierung der Signalstufen ausreichend. Es können demnach $2^{16} = 65536$ Signalstufen unterschieden werden. Die Quantisierung kann mit linearer oder nicht linearer Kennlinie erfolgen. Bei der linearen

³Aliasing entsteht bei der Abtastung für jedes Signal, welches eine zu hohe obere Grenzfrequenz f_G besitzt. Dadurch kann auch ein Informationsverlust im Bereich unterhalb der oberen Grenzfrequenz durch Überlappung einsetzen.

⁴Ein PAM-Signal ist ein Pulse Amplituden Moduliertes Signal.

Quantisierung fallen in jedes Quantisierungsintervall gleich viele Signalwerte des ursprünglichen Signals. Die Quantisierungskennlinie ist eine Gerade.

Bei der nicht linearen Quantisierung werden, z.B. bei den häufig verwendeten logarithmischen Kennlinien, kleinere Signalwerte feiner aufgelöst als größere, d.h. für kleinere Werte stehen mehr Quantisierungsintervalle als für größere zur Verfügung. Beispiele für nicht lineare Quantisierungskennlinien sind die A- oder die 13-Segment-Kennlinie. Derartige Quantisierungskennlinien sind der menschlichen Wahrnehmung besser angepasst als lineare. Im Durchschnitt werden weniger Bit zur Kodierung pro Signalstufe benötigt. Bei der Quantisierung geht ebenfalls im Allgemeinen Information des ursprünglichen Signales verloren.

Die Signalstufen des nun zeit- und wertdiskreten Signals werden anschließend kodiert, es entsteht ein PCM-Signal⁵. Der Datenumfang eines PCM-Signals kann u.a. durch Differenzkodierung weiter reduziert werden. So werden beim DPCM⁶ nur noch die Differenzen zu vorhergehenden Amplitudenwerten übertragen. Dabei macht man sich den Umstand zunutze, dass Signalamplituden innerhalb kurzer zeitlicher Abstände häufig nur wenig variieren. Die Kodierung von Differenzen hat im Mittel einen geringeren Informationsgehalt und kann mit weniger Bit kodiert werden.

Eine Alternative zur Digitalisierung von Musik stellt MIDI⁷ dar. Hier wird die Musik durch entsprechende Hardware, z.B. Synthesizer, reproduziert. Gespeichert oder übertragen wird dann nur noch die Notenrepräsentation des Musikstückes. Dies führt zu einer drastischen Reduzierung des Datenumfanges. Auf diese Art der Musikrepräsentation wird aufgrund ihrer eingeschränkten Nutzbarkeit im Folgenden nicht näher eingegangen.

2.2 MPEG

Die Daten wie sie bei der Analog-Digital-Wandlung entstehen, sind wenig optimiert. In der nicht optimierten, also unkomprimierten Form haben Audiosignale im Allgemeinen einen höheren Informationsgehalt als der, der vom Menschen bei der Wiedergabe wahrnehmbar ist. So dass für den Audibereich eine Reihe von Methoden entwickelt wurden, die eine platzsparende Form der Speicherung ermöglichen. Eine gebräuchliche Art der Komprimierung von Audiodaten ist die Speicherung im mp3-Format⁸. Bei der Komprimierung von Audio macht man sich auch Kenntnisse aus der Psychoakustik zunutze. Das Ziel ist die Reduzierung des Datenumfanges ohne für den Menschen hörbare Verluste.

Einige Verfahren wurden im Rahmen der Definition der MPEG-Standards für Audio standardisiert und sind allgemein zugänglich.

⁵Ein PCM-Signal ist ein Pulse Code Moduliertes Signal.

⁶DPCM ist die Abkürzung für Differenzen-PCM.

⁷Die Abkürzung MIDI steht für Musical Instrument Digital Interface.

⁸Diese in der Praxis benutzte Bezeichnung ist ungenau und steht für MPEG-1 bzw. MPEG-2 Layer 3, nicht zu verwechseln mit MPEG-3.

2.2.1 MPEG-1 Audio – ISO/IEC 11172-3

MPEG-1 nutzt eine Familie von 3 Audio-Kodier-Schemata, die mit Layer 1, Layer 2 und Layer 3 bezeichnet werden. Sie unterscheiden sich in ihrer Komplexität und Leistungsfähigkeit. Dabei stellt der Layer 3 den Modus mit der höchsten Komplexität dar. Er ist optimiert, um höchstmögliche Qualität bei möglichst kleinen Bitraten zu erzielen. Tabelle 1 fasst die Bitraten der Layer 1 bis 3 zusammen.

Layer 1	32-448 kbit/s
Layer 2	8-384 kbit/s
Layer 3	8-320 kbit/s

Tabelle 1: Bitraten der 3 MPEG-1 Layer

Die drei Layer sind hierarchisch, d.h. sie umfassen sich gegenseitig. Beispielsweise kann ein Layer 3 Dekoder auch die Layer 1 und 2 dekodieren. Alle Layer unterstützen Abtastfrequenzen von 32, 44.1 oder 48 kHz mit 16 Bit Quantisierungsaufösung.

MPEG-1 Audio arbeitet mit Mono- oder Stereosignalen. Es können bis zu 2 Kanäle benutzt werden. Die resultierenden Operationsmodi sind:

- Einkanal
- Zweikanal, zwei voneinander unabhängige Kanäle, die beispielsweise zwei Sprachversionen transportieren können
- Stereo
- Joint Stereo⁹

Alle Layer kodieren nach dem psychoakustischen Prinzip, welches darauf beruht, dass Töne mit nahe beieinanderliegenden Frequenzen und wesentlich unterschiedlicher Lautstärke sich gegenseitig maskieren, d.h. vom Menschen nicht mehr unterschieden werden können.

2.2.2 MPEG-2 Audio – ISO/IEC 13818-3

MPEG-2 Audio stellt eine abwärtskompatible Erweiterung des MPEG-1 Standards für Audio dar. Es wurde erweitert, um mehrere Audiokanäle zu unterstützen. So können bis zu 5 Audiokanäle verwendet werden. Des Weiteren sieht der Standard zusätzlich die Verwendung der halben Abtastfrequenzen 16, 22.05 und 24 kHz vor. Die Audio-Kodier-Schemata sind von MPEG-1 übernommen.

⁹Diese Technik wird dazu benutzt, ein Stereosignal effizienter zu kodieren. Dazu werden nur die Unterschiede zwischen dem linken und dem rechten Kanal kodiert. Informationen über Phasenverschiebungen zwischen linken und rechten Kanal (z.B. bei Dolby Sourround) gehen dabei im Allgemeinen verloren.

Durch weitergehende Untersuchungen hat sich gezeigt, dass durch den Verzicht auf vollständige Abwärtskompatibilität eine weitere Effizienzsteigerung erreicht werden kann. Diese Erweiterungen sind unter dem Begriff MPEG-2 Advanced Audio Coding (AAC) zusammengefasst. Durch Anwendung von MPEG-2 AAC wird eine weitere Reduktion der Bitraten bei gleichzeitig hoher Qualität des Audios erreicht.

Für alle MPEG-Layer und Versionen gilt, dass die Komprimierung verlustbehaftet in Abhängigkeit des Frequenzspektrums ist.

3 Video

Ein Video wird als eine Sequenz von mehreren aufeinanderfolgenden Bildern betrachtet. In vielen Standards, z.B. MPEG (Moving Pictures Expert Group), sind jeweils Verfahren zur Kodierung von Audio und Video vorgesehen. Obwohl umgangssprachlich zu einem Video zusätzlich auch die Audio-Information zählt, widmet sich dieser Abschnitt nur der visuell wahrnehmbaren Information, also der Behandlung von bewegten Bildern.

Ein Video hat einen wesentlich komplexeren Aufbau als ein Audio. Abbildung 2 veranschaulicht die verschiedenen Dimensionen eines digitalen Videos. Die Para-

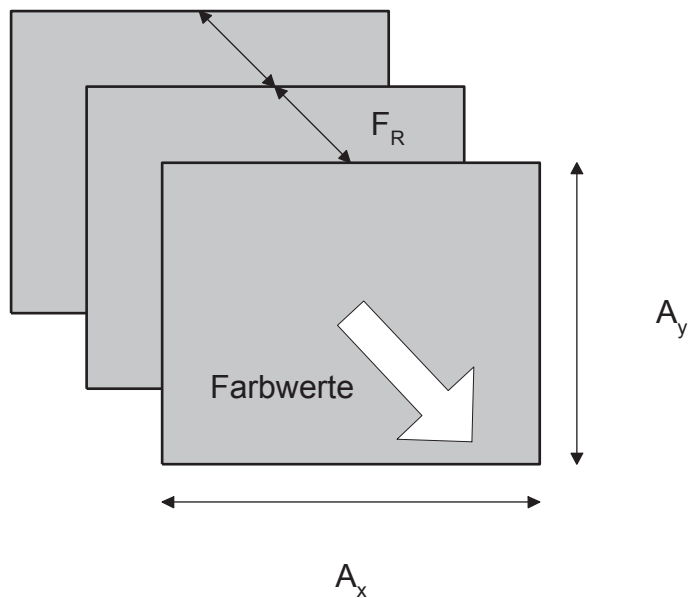


Abbildung 2: Dimensionen eines digitalen Videos

meter sind:

- A_X Auflösung in horizontaler Richtung
- A_Y Auflösung in vertikaler Richtung
- F_R Framerate oder Bildwiederholfrequenz
- Farbkomponenten, die mit jedem Bildpunkt dargestellt werden

Ein digitales Video besitzt demnach (mindestens) vier Dimensionen.

3.1 Farbmodelle

Eine wesentliche Rolle bei der Digitalisierung von Bildern spielen die verwendeten Farbmodelle. Diese sind aus ganz unterschiedlichen Überlegungen und Zielstellungen entstanden. Man unterscheidet im Großen und Ganzen additive und subtraktive Farbmodelle und solche, die sich an der menschlichen Farbwahrnehmung orientieren.

Aus physikalischen Überlegungen sind das additive RGB-Farbmodell und das subtraktive CMY-Farbmodell¹⁰ entstanden. Farbmodelle, die der menschlichen Farbwahrnehmung entlehnt sind, sind beispielsweise das YUV- und das YCbCr-Farbmodell.

Die Farbmodelle werden aufgrund ihrer unterschiedlichen Interpretation bzw. Darstellungsweise von Farbe in verschiedenen Anwendungen angewandt. So ist das RGB-Modell für die Bildschirmdarstellung bei CRT-Monitoren angepasst. Das YUV- und YCbCr-Farbmodell bilden die Grundlage der analogen bzw. digitalen Fernseh- und Videoübertragung.

Aufgrund der großen Anwendungsvielfalt können Farbraumwandlungen relevant zur Herstellung der Formatunabhängigkeit eines Medienobjektspeichersystems sein. Deshalb werden im Folgenden zunächst einige Farbmodelle erläutert.

3.1.1 Das RGB-Modell

Das RGB-Modell als ein additives Modell folgt der Überlegung, dass durch Überlagerung der drei Grundfarben Rot, Grün und Blau alle anderen Farben erzeugt werden können. Diese drei Farben definieren ein dreidimensionales kartesisches Koordinatensystem (Abbildung 3), in dem jeder Punkt charakterisiert durch das Tripel (r, g, b) einer anderen Farbe entspricht. Dabei spiegelt jeder Wert des Tripels die Intensität der entsprechenden Farbe wider.

Die „Eckfarben“ des RGB-Farbwürfels sind in Abbildung 3 angegeben. Zwischen 0 % und 100 % Intensität¹¹, jeweils repräsentiert durch 0 und 1, kann jeder Wert angenommen werden.

Das RGB-Modell findet speziell Anwendung bei der Braunschen Röhre (Fernsehgeräte und Monitore), bei der im Farbfall die möglichen Farben als eine Überlagerung eines roten, grünen und blauen Anteils dargestellt werden (additive Farbmischung). Aufgrund der gewählten Farbdarstellung sind manche Manipulationen wie das Verändern der Helligkeit nicht direkt an nur einer Komponente ausführbar. Derartige Verarbeitungsprozesse haben Auswirkungen auf die drei RGB-Komponenten. Demgegenüber wird die Realisierung von Rot-, Grün- und Blaufiltern direkt unterstützt, da die jeweiligen Farbanteile direkt zugreifbar sind.

¹⁰Auf das CMY-Farbmodell wird hier nicht näher eingegangen, da es vorrangig in der Drucktechnik eine Rolle spielt und direkt komplementär zum RGB-Modell ist.

¹¹Die hier angegebenen Werte spiegeln jeweils nur die Intensität einer Farbe in Prozent wider. In der Praxis wird man bei der Kodierung eines Farbanteils mit 8 Bit Werte zwischen 0 und 255 vorfinden.

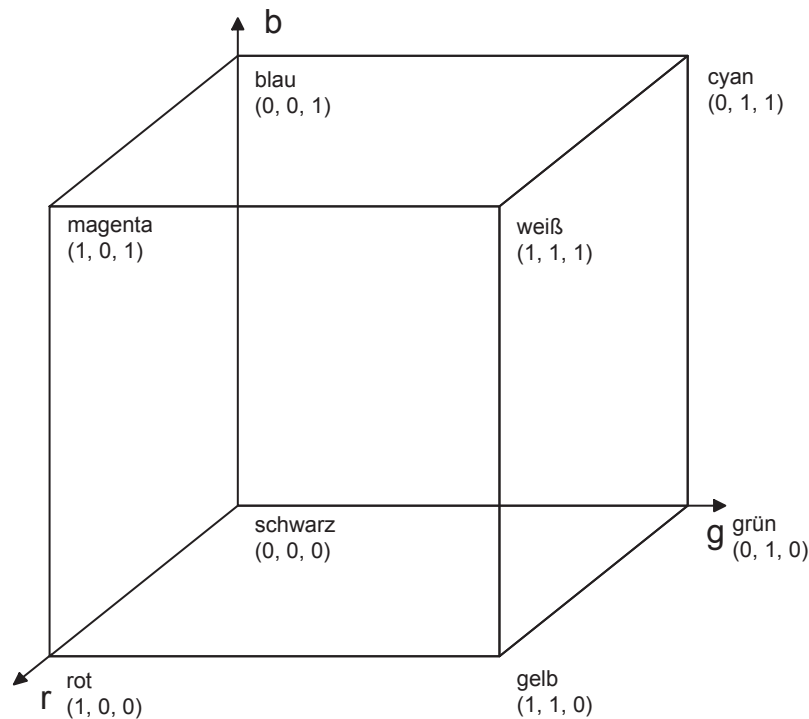


Abbildung 3: RGB-Farbwürfel

Im Zusammenhang mit Video spielt das RGB-Format nur bedingt eine Rolle. Hier findet man das R'G'B'-Format vor. Man spricht dann von gamma-korrigierter Darstellung. Diese Art der Darstellung orientiert sich an der menschlichen Helligkeitswahrnehmung und an der damit verbundenen Anpassung an entsprechende Darstellungsgeräte, z.B. an CRT-Monitoren. In einem System mit 8 Bit Kodierung der R'-, G'- und B'-Komponenten steht jeweils ein Intervall $[0, 255] \in \mathbb{N}$ zur Verfügung. Damit können insgesamt 2^{24} Werte kodiert werden. Davon kann nur ein Teil vom Menschen unterschieden werden, so dass in der ITU-R BT. 601 jeweils ein Intervall $[16, 235] \in \mathbb{N}$ zur Farbkodierung vorgesehen ist. Dies entspricht einer Verringerung der Farbstufen um ca. 38 %.

Diese Reduktion der Farbtiefe hat keine Auswirkung auf die menschliche Wahrnehmung, da der Mensch empfindlicher auf Helligkeitsschwankungen als auf Farben reagiert. Diesen Umstand machen sich die nachfolgenden Farbmodelle zunutze, wobei das YCbCr-Modell zusätzlich verschiedene Abtastformate zulässt.

RGB kann direkt unter Anwendung von Umrechnungsformeln in das YUV- und das YCbCr-Modell umgewandelt werden. Im Abschnitt 4.5 wird eine Farbraumwandlung anhand eines Beispiels einer Wandlung von gamma-korrigiertem RGB zu YCbCr 4:2:2 erläutert.

3.1.2 Das YUV-Modell

Im YUV-Modell wird in der Komponente Y ¹² die Helligkeit und in den Komponenten U und V die Farbinformation kodiert. Es ist das Farbmodell, welches vorrangig bei Fernsehsignalen im PAL-, NTSC- und SECAM-Format Anwendung findet. Die Trennung von Helligkeits- und Farbinformation geht auf die menschliche Farbwahrnehmung zurück. Sie bietet darüber hinaus den Vorteil, dass Farbfernsehsignale auch mit herkömmlichen Schwarzweißfernsehgeräten empfangen werden können. Diese werten nur die Y -Komponente aus. Bei der U - und V -Komponente handelt es sich um Farbdifferenzsignale.

Die drei Komponenten Y , U und V können aus den drei Komponenten des RGB-Modells berechnet werden. Hierzu wird häufig gamma-korrigiertes RGB verwendet. Eine Berechnung der korrespondierenden R -, G - und B -Werte ist ebenfalls möglich.

Das YUV-Modell hat darüber hinaus den Vorteil, dass hier Helligkeits- und Farbinformation getrennt vorliegen und sich Änderungen nicht zwingend auf alle Komponenten auswirken müssen. So kann beispielsweise sehr vorteilhaft die Helligkeit durch direkte Manipulation der Y -Komponente geändert werden.

3.1.3 Das YCbCr-Modell

Das YCbCr-Modell wurde im Rahmen der Entwicklung weltweiter digitaler Videostandards durch die ITU-R BT.601 entwickelt. Bei 8 Bit Kodierung ist für Y ein Intervall von $[16, 235] \in \mathbb{N}$, für Cr und Cb jeweils ein Intervall von $[16, 240] \in \mathbb{N}$ vorgesehen. Ein Wert von 128 für die Cb - und Cr -Komponente kennzeichnet den Nullpunkt. Die Y -Komponente kodiert wieder die Helligkeitsinformation und in den Komponenten Cb und Cr steckt die Farbinformation. Daneben besteht die Möglichkeit, Farbinformation zu reduzieren. YCbCr unterstützt dazu die Abtastformate:

- YCbCr 4:4:4 jeder Bildpunkt enthält einen Wert für Y , Cb und Cr .
- YCbCr 4:2:2 jeder Bildpunkt enthält einen Wert für Y , die Farbdifferenzsignale (Cb , Cr) werden innerhalb einer Zeile nur für jeden zweiten Bildpunkt übertragen.
- YCbCr 4:1:1 mit jedem Bildpunkt wird ein Y -Wert übertragen, Farbinformation steckt innerhalb einer Zeile nur in jedem vierten Bildpunkt.
- YCbCr 4:2:0 wie die vorangegangenen Kodierungen, für die Werte Cb und Cr wird eine 2:1 Reduktion in horizontale und vertikale Richtung vorgenommen.

¹²In [Poy01] wird darauf hingewiesen, dass es sich hierbei nicht um die von der CIE definierten Helligkeit handelt. Die Y -Komponente des YUV- und des nachfolgenden YCbCr-Farbmodells wird als gewichtete Summe von gamma-korrigierten R -, G - und B -Komponenten berechnet.

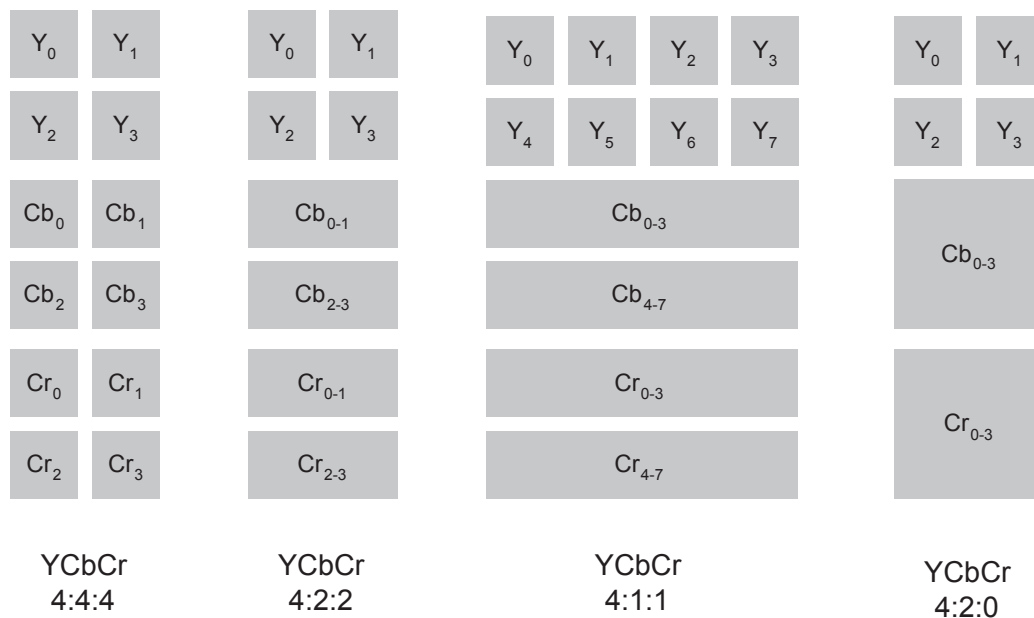


Abbildung 4: Gezeigt werden jeweils die Y-, Cb- und Cr-Komponenten eines Bildpunktes. Komponenten mit gleichen Indices beziehen sich auf einen dadurch kodierten Bildpunkt. Beim Format YCbCr 4:4:4 wird für jeden Bildpunkt ein Y-Wert und die korrespondierenden Cb- und Cr-Werte übertragen. Bei YCbCr 4:2:0 findet eine 2:1 Reduktion in horizontaler und in vertikaler Richtung statt. Das bedeutet, dass jeweils eine Cb- und Cr-Komponente für jeweils vier korrespondierende Y-Komponenten übertragen wird. Die Komponenten Cb₀₋₃ und Cr₀₋₃ beziehen sich jeweils auf vier Bildpunkte.

Durch die Abtastformate ist es möglich, den Anteil von übertragener Farbinformation und damit die Datenrate je nach Anwendung weiter zu reduzieren. Dabei macht man sich die Tatsache zunutze, dass die menschliche Wahrnehmung empfindlicher gegenüber Helligkeit als gegenüber Farben ist. Die Reduktion des Farbanteils wird als Subsampling bezeichnet. Das YCbCr-Modell findet u.a. bei der MPEG Kodierung Anwendung.

Eine Wandlung von YCbCr-Modell in andere Farbmodelle ist möglich. Für weitergehende Details wird auf [Jac96] verwiesen.

3.2 Interlaced / Noninterlaced

Für die Bilddarstellung am Monitor oder bei einem Fernsehgerät gibt es zwei prinzipielle Wege den Bildaufbau zu realisieren. Man unterscheidet die interlaced

und die noninterlaced Darstellung. Wobei Erstere bei heutiger Fernsehtechnik¹³ die gebräuchlichste Darstellung ist. Die noninterlaced Darstellung findet man heute häufig bei Computermonitoren.

Bei der interlaced Darstellung wird jedes Bild in t Teilbilder zerlegt. Jedes Teilbild enthält nur noch den $\frac{1}{t}$ -ten Teil der Information des Gesamtbildes. Die Teilbilder werden nacheinander mit der Teilbildwiederholfrequenz $f_T = t \cdot f_{refresh}$ angezeigt. Dabei ist $f_{refresh}$ die vertikale Bildwiederholfrequenz, welche identisch mit der Framerate F_R ist. Der Vorteil dieser Vorgehensweise ist, dass ein so zusammengesetztes Bild wesentlich weniger flimmert als ein Bild, welches als Ganzes mit der gleichen Bildwiederholfrequenz $f_{refresh}$ angezeigt wird. Dieser Effekt ist besonders bei niedrigen Bildwiederholfrequenzen (25 Hz, 30 Hz) feststellbar. Durch die Trägheit des menschlichen Auges erscheint ein derart zusammengesetztes Bild wieder als ein Ganzes.

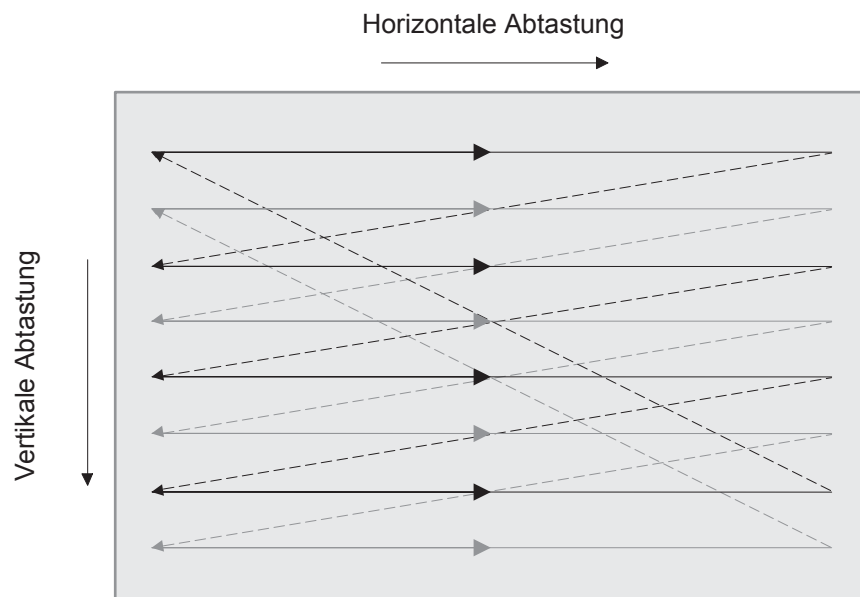


Abbildung 5: Interlaced Bilddarstellung: Die durchgezogenen horizontalen Pfeile deuten den Aufbau des Gesamtbildes aus zwei Halbbildern an. Dabei stellen die schwarzen Pfeile die Bildschirmzeilen des ersten und die grauen Pfeile die des zweiten Halbbildes dar. Die gestrichelten Pfeile deuten den Sprung zur jeweils folgenden Bildschirmzeile an.

Dazu soll ein Beispiel für $t = 2$ betrachtet werden, d.h. das Gesamtbild zerfällt

¹³Die interlaced Darstellung wird selten bei der heute weit verbreiteten 100 Hz Technik eingesetzt. Vielmehr werden beide Halbbilder gespeichert und dann mit einheitlicher Wiederabgabefrequenz dargestellt.

in zwei Halbbilder (Abbildung 5). Dabei enthält ein Halbbild die Information der geraden und eines die der ungeraden Bildschirmzeilen. Geht man von einem System mit einer Bildwiederholfrequenz $f_{refresh}$ von 25 Hz aus, innerhalb dieser $\frac{1}{25}$ s werden beide Teilbilder dargestellt, so ergibt sich eine Teilbildwiederholfrequenz f_T von 50 Hz. Ein derart zusammengesetztes Bild vermittelt den Eindruck, dass es weniger flimmert.

Die noninterlaced Darstellung ist das genaue Gegenteil der interlaced Darstellung. Jede Zeile des Bildes wird sequentiell, eine nach der anderen, am Bildschirm gezeichnet. Dies erfordert wesentlich höhere Bildwiederholfrequenzen verglichen mit der interlaced Darstellung, um dem Problem des Flimmerns zu begegnen.

Es kann vorkommen, dass verschiedene Anpassungen an die jeweiligen Nutzerbedürfnisse vorgenommen werden müssen. Zwischen beiden Darstellungsweisen gibt es Möglichkeiten der Konvertierung. Auf entsprechende Verfahren wird unter den Gliederungspunkten 4.3 und 4.4 eingegangen.

3.3 MPEG

MPEG umfasst eine Vielzahl von Standards zur Bewegtbildkodierung. MPEG-1 wurde entwickelt, um Audio und Video mit hoher Qualität zu speichern und zu verteilen.

MPEG-2 erweitert MPEG-1, um eine größere Anzahl von Anwendungen abzudecken. Es wurde speziell für die Bedürfnisse von digitaler Videoübertragung mit Bitraten von 4 - 9 Mbit/s konzipiert. Eine Hauptanwendung von MPEG-2 ist die digitale Fernsehübertragung.

3.3.1 MPEG-1 Video – ISO/IEC 11172-2

Das MPEG-1 Verfahren arbeitet mit dem YCbCr 4:2:0 Farbmodell zur Kodierung von Bilddaten. MPEG-1 erreicht eine Reduktion der Videodatenrate durch die Zerlegung eines Videodatenstromes in vier Framearten.

- I-Frames (Intra-Frames) enthalten die vollständige Bildinformation. Durch sie wird der wahlfreie Zugriff auf die Videodaten möglich. Sie sollten mindestens $2 \times$ pro Sekunde oder an Szenenwechseln vorkommen.
- P-Frames (Predicted-Frames) nehmen Bezug auf vorhergehende I- oder P-Frames. Sie enthalten lediglich die Differenz zu ihren referenzierten Vorgängern.
- B-Frames (Bidirectional-Frames) referenzieren die am nächsten liegenden I- oder P-Frames, sowohl vorhergehende als auch zukünftige. Durch die bidirektionale Kodierung wird die höchste Kompressionsrate erreicht.
- D-Frames ermöglichen einen schnellen Bildsuchlauf.

Es besteht kein Zwang, P-, B- bzw. D-Frames zu generieren. Das Fehlen solcher prediktiven Frames erhöht jedoch im Allgemeinen die Datenrate.

Eine Gruppe von Bildern (GOP¹⁴) ist durch eine Reihe von einem oder mehreren Bildern definiert. Je weniger Bilder eine Gruppe bilden, desto geringer ist die Kompressionsrate, umso besser werden Bewegungen dargestellt. Jede Gruppe muss mit einem I-Frame beginnen, gefolgt von einer Anzahl von I-, P- oder B-Frames. Man unterscheidet geschlossene und offene GOP-Formate. Beim geschlossenen GOP-Format kann eine Gruppe von Bildern unabhängig von der vorhergehenden Gruppe dekodiert werden. Entsprechend wird bei einem offenen GOP-Format die Verfügbarkeit der vorhergehenden Gruppe vorausgesetzt.

Motion Compensation Die Kompression von P- und B-Frames wird durch Bewegungskompensation (Motion Compensation) noch zusätzlich verbessert. Dieses Verfahren basiert auf der Feststellung, dass innerhalb einer kurzen Sequenz von Bildern ähnlichen Inhalts die Lage von dargestellten Objekten nur geringfügig variiert. Diese Art von Bewegung wird durch einen zweidimensionalen Bewegungsvektor ausgedrückt, der festlegt, wie ein Makroblock¹⁵ des aktuellen aus einem eines vorhergehenden oder zukünftigen Frames gewonnen werden kann. Ein solcher Makroblock enthält die Bewegungsvektoren und die inhaltliche Differenz zum referenzierten Makroblock. Es gibt Situationen, in denen keine Vorhersage eines Makroblockes aus vorhergehenden Bildern abgeleitet werden kann. Betrifft das einen P-Frame, so wird der betreffenden Makroblock wie bei I-Frames kodiert. Bei B-Frames, die sowohl vorhergehende als auch zukünftige I- oder P-Frames referenzieren, ergeben sich vier denkbare Kodierungen:

1. Intra-Kodierung, keine Bewegungskompensation
2. Vorwärts-Voraussage, nächste vorhergehende I- oder P-Frames als Referenz
3. Rückwärts-Voraussage, nächste zukünftige I- oder P-Frames als Referenz
4. Bidirektionale-Voraussage, kombiniert 2. und 3.

Jeder Block eines Frames wird der Diskreten Kosinus Transformation (DCT) unterworfen, die jeden Block in den Frequenzbereich transformiert. Der dabei entstehende Block von 8×8 Frequenzkoeffizienten wird quantisiert. Die Quantisierung ist nicht linear, d.h. höhere Frequenzen werden gröber aufgelöst als niedrige. Dieses Vorgehen ist der menschlichen Wahrnehmung besser angepasst. Bei der Quantisierung werden insbesondere die Frequenzkoeffizienten der höheren Frequenzen zu Null gerundet. Das bedeutet, dass diese Frequenzanteile herausgefiltert werden.

¹⁴group of pictures

¹⁵Ein Makroblock besteht aus einer 16×16 Pixel großen Gruppe von Y-Komponenten und deren korrespondierenden 8×8 großen Gruppe von Cb- und Cr-Komponenten. Ein Block besteht aus einer 8×8 Pixel großen Gruppe von Y-, Cb- oder Cr-Komponenten.

Anschließend werden die Frequenzkoeffizienten der Lauflängen- und Huffman-Kodierung unterzogen, was den Datenumfang weiter reduziert. Für weitere Details wird auf [Jac96] verwiesen.

3.3.2 MPEG-2 Video – ISO/IEC 13818-2

Eine Erweiterung von MPEG-1 stellt MPEG-2 dar. Es unterstützt im Gegensatz zu MPEG-1 auch die Farbformate YCbCr 4:4:4 und YCbCr 4:2:2. Durch die erhöhte Farbauflösung der Formate YCbCr 4:4:4 und YCbCr 4:2:2 verbessert sich die Bildqualität. Um die Abwärtskompatibilität mit MPEG-1 herzustellen, kann auch YCbCr 4:2:0 verwendet werden. Zur Reduktion des Datenumfangs werden Bilder wie bei MPEG-1 als I-, P- und B-Frames kodiert. D-Frames werden für die Kompatibilität mit MPEG-1 zwar dekodiert, aber im MPEG-2 Format nicht mehr kodiert, so dass ein reiner MPEG-2 Videodatenstrom nur noch bis zu 3 Framearten enthält. Wie schon bei MPEG-1 wird zur weiteren Reduzierung der Datenrate Bewegungskompensation eingesetzt.

4 Ausgewählte Konvertieroperationen

Die hier vorgestellten Konvertieroperationen stellen nur einen kleinen Ausschnitt von vielen Möglichen dar. In diesem Abschnitt werden an Beispielen die Auswirkungen von Konvertierungen auf den Inhalt von Audios und Videos betrachtet. Die Konvertierungen im Audio- und Videobereich werden hier zusammengefasst. Der Schwerpunkt der Konvertieroperationen in diesem Abschnitt betrifft jedoch Videos.

4.1 Transformationen

Ohne näher auf die konkreten Verfahren der Transformation einzugehen, spielen diese bei Verarbeitungsprozessen im Audio- und Videobereich eine große Rolle. Bekannte Transformationen sind z.B. die Fourier-Transformation oder die Diskrete Kosinus Transformation (DCT).

Letztere spielt bei der Videokodierung für MPEG oder H.26x eine Rolle. Die Fourier-Transformation findet u.a. im Audibereich Anwendung. Derartige Transformationen erlauben eine andere Sichtweise auf die transformierten Daten. Dazu werden die Daten in den Frequenzbereich übertragen. Man spricht in diesem Zusammenhang auch von einem Frequenzspektrum, welches sich aus Spektrallinien zusammensetzt. Eine Spektrallinie ist durch die zugeordnete Frequenz und ihren Betrag eindeutig bestimmt.

Jede der oben genannten Transformationen ist umkehrbar, so dass geeignete Modifizierungen im Frequenzbereich vorgenommen werden können. Ein typischer Verarbeitungsprozess im Frequenzbereich ist das Filtern. Hierfür stehen verschiedene Filter mit jeweils unterschiedlichen Charakteristiken zur Verfügung.

4.2 Spektrale Filter

Auf zuvor transformierten Daten lassen sich sehr vorteilhaft Filter anwenden. Filter sind in verschiedenen Ausführungen, z.B. als Hochpass-, Tiefpass- oder Bandpass-Filter, in der Analogtechnik verfügbar. In der Digitaltechnik existieren entsprechende Komplemente. Diese simulieren die Eigenschaften entsprechender Analogfilter und werden als Digitale Filter bezeichnet. Sie bestehen im Wesentlichen aus Speichern und Rechenwerken und arbeiten auf Folgen von diskreten Signalwerten.

4.2.1 Tiefpass-Filter

Ein Tiefpass-Filter ist ein Filter, welches die Spektralanteile eines Signals oberhalb einer definierten Grenzfrequenz f_G unterdrückt.

Tiefpass-Filter sind gleichermaßen für die Audio- und Videobearbeitung interessant. Bei Audios filtern sie hohe Frequenzanteile aus dem Signal heraus. In

Digital-Analog-Wandlern dienen Tiefpässe der Rückgewinnung des Originalsignals, indem die bei der Abtastung periodisch um Vielfache der Abtastfrequenz entstehenden Signalspektren unterdrückt werden.

Angewendet auf Bilddaten bewirken sie, dass Detailinformationen des Bildes verloren gehen. Das Bild wird somit für einen Betrachter unscharf [KZ95].

4.2.2 Hochpass-Filter

Ein Hochpass-Filter ist nur für Spektralanteile oberhalb einer Grenzfrequenz f_G durchlässig. Bei Audios werden entsprechend die Töne unterhalb von f_G herausgefiltert. Wird ein Hochpass-Filter auf Bilddaten angewendet, so führt dies zur Hervorhebung von Detailinformationen. Einzelne Punkte oder feine Linien werden besser erkennbar [KZ95].

4.3 Interlaced-to-Noninterlaced Wandlung

4.3.1 Scan Line Duplikation

Bei diesem Verfahren wird die vorhergehende Bildschirmzeile eines Teilbildes dupliziert und anstelle der korrespondierenden Bildschirmzeile des anderen Teilbildes angezeigt. Dadurch wird die Anzahl der vertikalen Bildschirmzeilen verdoppelt, jedoch nicht die vertikale Auflösung. Diese Vorgehensweise ist akzeptabel für die Farbdifferenzsignale wie CbCr und UV.

4.3.2 Scan Line Interpolation

Auch bei diesem Verfahren wird die vertikale Zeilenanzahl verdoppelt. Der Unterschied zur vorhergehenden Vorgehensweise ist, dass interpolierte Bildschirmzeilen zwischen die originalen Bildschirmzeilen eingefügt werden. Um das System zu vereinfachen, werden meist die Helligkeitskomponenten interpoliert und die Farbkomponenten nach vorhergehenden Verfahren verdoppelt.

4.3.3 Field Merging

Das aus Teilbildern bestehende Bild wird wieder zusammengesetzt, indem die jeweils korrespondierenden Bildschirmzeilen zu einem Gesamtbild fusioniert werden. Jedoch kann es in Regionen mit schnellen Bewegungen zu Bildfehlern kommen, da die vormals getrennten Teilbilder eine zeitliche Differenz aufweisen. Ein bewegtes Objekt kann sich in jedem Teilbild an einer anderen Position befinden, d.h. vom Objekt können Doppelbilder entstehen. Deshalb eignet sich dieses Verfahren für Regionen mit wenig Bewegung.

4.4 Noninterlaced-to-Interlaced Wandlung

4.4.1 Abwechselndes Auslesen

Das Gesamtbild wird in Teilbilder zerlegt, indem die Bildschirmzeilen ausgelesen werden, um die entsprechenden Teilbilder zu erzeugen.

4.4.2 Interpolation

Auch eine Interpolation zwischen zwei oder mehreren Bildschirmzeilen, um eine interlaced Bildschirmzeile zu generieren, ist denkbar.

4.5 Farbraumwandlung

Wie schon bei der Vorstellung der Farbmodelle angedeutet, verlangen verschiedene Anwendungen unterschiedliche Farbmodelle. Ebenfalls schon angedeutet wurde, dass Wandlungen zwischen den unterschiedlichen Farbmodellen möglich sind. Hier wird beispielhaft die Wandlung von gamma-korrigiertem RGB (R'G'B') zu YCbCr 4:2:2 vorgestellt.

Die Umrechnung von R'G'B' zu YCbCr kann durch eine 3×3 Matrix Multiplikation gemäß nachfolgender Formel geschehen [Jac96].

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.301 & 0.586 & 0.113 \\ -0.172 & -0.340 & 0.512 \\ 0.512 & -0.430 & -0.082 \end{pmatrix} \cdot \begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} \quad (1)$$

Die Berechnungsvorschrift ist geeignet, um R'G'B'-Werte in einem Bereich von $[16, 235] \in \mathbb{N}$ zu konvertieren, entsprechende Rundung vorausgesetzt. Um gamma-korrigiertes RGB in einem Intervall von $[0, 255] \in \mathbb{N}$ in das YCbCr-Format zu konvertieren, ist eine andere Berechnungsvorschrift notwendig. Für diese Formel wird auf [Jac96] verwiesen.

Zunächst ist für jeden Bildpunkt jeweils eine Y-, Cb- und Cr-Komponente vorgesehen, d.h. es entsteht YCbCr 4:4:4. Um YCbCr 4:2:2 zu erzeugen, muss für jeden zweiten Bildpunkt jeweils die Cb- und Cr-Komponente entfernt werden. Das Verfahren wird als Subsampling bezeichnet.

Bei der Darstellung von YCbCr 4:2:2, 4:1:1 und 4:2:0 müssen für jeden Bildpunkt die Cr- bzw. Cb-Komponenten wieder rekonstruiert werden. Dies kann z.B. durch Kopieren der Farbanteile vorhergehender vollständig kodierter Bildpunkte geschehen.

Abschließend wird noch angemerkt, dass Farbraumwandlungen nicht unproblematisch sind. Es ist zwar jederzeit möglich, Wandlungen zwischen den verschiedenen Modellen vorzunehmen, jedoch gilt zu beachten, dass allgemein bei der Wandlung von RGB zu YCbCr Farbinformation verloren geht. Mit RGB im Intervall $[16, 235] \in \mathbb{N}$ können $220^3 = 10648000$ Farbwerte kodiert werden. Bildet man

dies auf YCbCr ab, so werden nur noch $\frac{1}{4}$ der insgesamt 11137500¹⁶ zur Verfügung stehenden Kodeworte für Farbwerte genutzt. Es können nur noch 2784375 Farbwerte kodiert werden. Bei einer Rückwandlung in das RGB-Modell führt dies zum Verlust von 74 % der ursprünglich darstellbaren Farben [Poy98].

4.6 Koordinatentransformation

Jedem Bild liegt ein Koordinatensystem zu Grunde. Unter Angabe der Koordinaten in horizontaler und vertikaler Richtung kann ein Bildpunkt aus einer Menge von vielen eindeutig referenziert werden. Er repräsentiert dabei einen oder mehrere Farbwerte. Ein Bildpunkt kann demnach eindeutig durch das Tripel

$$\left(\underbrace{x, y}_{\text{Koordinaten}}, \underbrace{f^n(x, y)}_{\text{Farbwerte}} \right)$$

beschrieben werden [KZ95]. Dabei sind für n die Werte 1 und 3 üblich, wobei mit $n = 1$ ein Grauwert und $n = 3$ ein Farbwert kodiert werden kann. Man spricht für $n > 1$ auch von mehrkanaligen Bildern. Ein Bild besteht aus einer Menge von Bildpunkten.

Auf diesen Bildpunkten sind eine Reihe von Koordinatentransformationen [KZ95] wie die Bildspiegelung, die Bildrotation und die Bildverschiebung definiert.

4.6.1 Bildspiegelung

Eine Bildspiegelung kann um seine mittlere waagerechte und/oder senkrechte Achse erfolgen. Auch eine diagonale Spiegelung ist möglich, jedoch mit der Einschränkung, dass das Bild in horizontaler und vertikaler Richtung die gleiche Auflösung besitzen muss.

Eine horizontale Spiegelung ergibt sich, wenn der neue Bildwert $f'(x, y)$ aus dem Bildwert $f(X - x + 1, y)$ des ursprünglichen Bildes entsteht. Entsprechend ist die vertikale Spiegelung definiert mit $f'(x, y) = f(x, Y - y + 1)$. Dabei bezeichnet der Wert X die horizontale und der Wert Y die vertikale Auflösung des gesamten Bildes. Die dargestellte Koordinatentransformation setzt gerade Werte für X und Y voraus.

4.6.2 Bildrotation

Die Bildrotation um 180° ist durch die Koordinatentransformation $f'(x, y) = f(X - x + 1, Y - y + 1)$ definiert, gerade Werte für X und Y vorausgesetzt. Mit der Einschränkung von $X = Y$ oder einer vorausgehenden Skalierung sind auch Drehungen von z.B. 90° ohne Verlust realisierbar.

¹⁶YCbCr sieht für Y ein Intervall von $[16, 235] \in \mathbb{N}$ und für Cb und Cr jeweils ein Intervall von $[16, 240] \in \mathbb{N}$ vor. Daraus ergeben sich $220 \cdot 225^2 = 11137500$ Kodeworte.

4.6.3 Bildverschiebung

Eine achsenparallele Bildverschiebung ist in der Bildverarbeitung oft nützlich, um beispielsweise Bilder aus verschiedenen Teilen zusammenzustellen. Denkbar ist auch, dass Bilder eines Videos zur Anpassung an eine Anwendung in horizontale und/oder vertikale Richtung verschoben werden. Eine solche Anwendung könnte die Bild-in-Bild-Darstellung mehrerer vorher skaliertes Videos sein.

Die entsprechenden Werte der Bildpunkte $f'(x, y)$ ergeben sich aus den Werten $f(x - d_x, y - d_y)$ für $1 \leq x + d_x \leq X$ und $1 \leq y + d_y \leq Y$. Dies entspricht einer Verschiebung nach links in horizontale und einer Verschiebung nach oben in vertikale Richtung. Werden keine gesonderten Vorkehrungen getroffen, z.B. eine vorhergehende Skalierung, so gehen die Bildpunkte, die über ein vorher definiertes Bildraster hinausragen, verloren. Entsprechend gelten die Grenzen: $-X \leq d_x \leq X$ und $-Y \leq d_y \leq Y$ für ein $X \times Y$ großes Bild.

4.7 Größenänderung

Eine häufige Anwendung ist die Anpassung der Größe eines Videos. Hier gibt es unterschiedliche Standardauflösungen, die in Tabelle 2 zusammengefasst sind.

Computer-Auflösungen	PAL-Auflösungen	
	Standard	MPEG
640×480	720×576	352×288
800×600	768×576 ¹⁷	384×288 ¹⁷
1024×768		720×576
1280×1024		

Tabelle 2: Standardauflösungen

Anpassungen sind beispielsweise vorstellbar, wenn der Medienserver ein gespeichertes Video in eine Standard-PAL-Auflösung konvertieren muss. Auch hier gibt es mehrere Möglichkeiten der Konvertierung, beispielhaft wird hier eine Möglichkeit diskutiert. Die vorgestellte Vorgehensweise ist nicht auf Vektorgrafiken anwendbar. Bei Vektorgrafiken verringert sich das Problem auf die Einführung eines Skalierungsfaktors für alle Elemente.

Die einfachste Form der Verkleinerung ist das Löschen von m ausgewählten Bildpunkten in einer Gruppe von n , jeweils in horizontale wie vertikale Richtung. Die Auswahl der zu erhaltenden Bildpunkte kann mit einer abgewandelten Form des Bresenham-Algorithmus¹⁸ erfolgen. Bei Anwendung des Bresenham-

¹⁷Square-Pixel-Format: Das Seitenverhältnis eines Bildpunktes ist 1:1, der Bildpunkt ist quadratisch.

¹⁸Der Bresenham-Algorithmus wird in seiner ursprünglichen Form zum Zeichnen von geraden Linien zwischen einem Anfangs- und Endpunkt auf gerasterten Ausgabegeräten genutzt.

Algorithmus bedeutet ein Skalierungsfaktor von $\frac{1}{3}$, dass in einer Gruppe von 3 Bildpunkten jeweils 2 verworfen werden. Dies führt zu einem Bild, welches nur noch ein $\frac{1}{9}$ der ursprünglichen Bildfläche einnimmt. Eine Verbesserung zum reinen Löschen von Pixeln stellt das Verfahren der Linearen Näherung dar. Bei der Linearen Näherung wird der Farbwert eines Pixels aus den linear genäherten Anteilen der benachbarten Pixel berechnet. Jedoch führt auch bei diesem Verfahren das Verkleinern zu einem Informationsverlust und zum Verwerfen von Pixeln.

Ein Vergrößerungsverfahren für Rasterbilder ist die Duplizierung schon bestehender Bildpunkte. So bedeutet ein Skalierungsfaktor von 3, dass die Bildfläche auf das 9 fache der Ausgangsfläche anwächst. Für jeden Bildpunkt werden 2 Duplikate erzeugt.

Abschließend wird noch auf einige Probleme, die bei der Skalierung von Pixelbildern auftreten, hingewiesen. So sind der Vergrößerung und der Verkleinerung Grenzen gesetzt. Eine zu starke Vergrößerung führt zur Erzeugung von Blöcken im Bild, eine zu starke Verkleinerung zieht den Verlust von Details nach sich.

Auf den Erkenntnissen der drei vorangegangenen Abschnitte (2 Audio, 3 Video, 4 Ausgewählte Konvertieroperationen) baut der nun folgende Abschnitt 5 Klassifikation von Konvertern auf. Anhand von Eigenschaften dieser Konvertieroperationen werden Klassifizierungsmerkmale entwickelt. Unter dem Punkt 5.6 Beispiele werden im Anschluss einige Konverter klassifiziert.

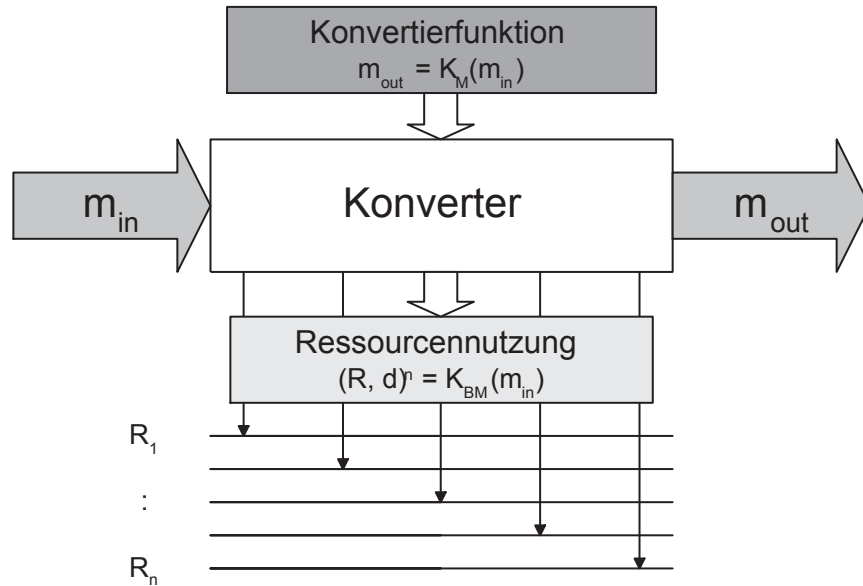


Abbildung 6: Konvertermodell

5 Klassifikation von Konvertern

5.1 Konvertermodell

In [Mär00] wird ein Modell zur Konverterbeschreibung entwickelt. Ein Konverter ist allgemein eine virtuelle Verarbeitungseinheit, die Daten mit einem Eingangsformat f_{in} unter Anwendung einer Konvertierfunktion k in ein Ausgangsformat f_{aus} überführt. Wobei nicht notwendigerweise gilt $f_{in} \neq f_{out}$.

Ein Konverter kann in Software oder Hardware realisiert werden. Er nutzt zur Ausführung seiner Konvertierfunktion Ressourcen des darunterliegenden Systems wie Speicher, CPU oder Datenträger.

In Verbindung mit zeitkritischen Medien (Audio und Video) müssen Konverter Zeitrestriktionen einhalten. Zur Modellierung dienen Schwankungsbeschränkte Mediendatenströme, welche jeweils am Eingang und Ausgang eines Konverters auftreten. Ein Mediendatenstrom m ist definiert als:

$$m = ((T, D, \tau, t_0), (S, M, \sigma, s_0), f) \in M$$

Er besteht aus einem Zeit- und Volumenstrom. Der Parameter f bezeichnet die Formatbeschreibung, die in der Formatmenge F enthalten ist. Es gilt $f \in F$. Der Zeitstrom

$$t = (T, D, \tau, t_0)$$

beschreibt Ereignisse, welche mit einer Periode T auftreten. Die Periode kann

in festen Grenzen schwanken, ausgedrückt durch τ . Die Parameter D und t_0 kennzeichnen den minimalen Abstand zweier Ereignisse und den Startzeitpunkt.

Derartige Ereignisse treten auf, wenn Quanten eines Mediendatenstromes verarbeitet werden, beispielsweise bei einem Video, bei dem zu den Zeitpunkten $i \cdot T_R = i \cdot \frac{1}{F_R}$ Frames auftreten. Schwankungen von T_R können u.a. durch Zugriffe auf einen Externspeicher oder durch unterschiedliche Verarbeitungszeiten entstehen.

Der Volumenstrom

$$s = (S, M, \sigma, s_0)$$

beschreibt Datenpakete mit einer durchschnittlichen Größe S und einer maximalen kumulierten Abweichung σ von S . Die Parameter M und s_0 bezeichnen die minimale Größe und die Anfangsgröße.

Die Kombination aus Zeit- und Volumenstrom bildet einen Schwankungsbeschränkten Datenstrom d mit

$$d = ((T, D, \tau, t_0), (S, M, \sigma, s_0)) \in D$$

Ergänzt man einen Schwankungsbeschränkten Datenstrom noch um das Format f , so erhält man einen Schwankungsbeschränkten Mediendatenstrom. Der Inhalt des Parameters f eines Mediendatenstromes ist noch nicht endgültig festgelegt, er enthält aber mindestens die beschreibenden Daten (Metadaten) des Medienobjektes. Dazu zählen u.a. der Mediendatentyp $typ(f)$ sowie gegebenenfalls weitere beschreibende Informationen, z.B. auch Angaben zum Inhalt.

Ein Konverter nach dem in [Mär00] entwickelten Modell bildet einen eingehenden Mediendatenstrom m_{in} auf einen ausgehenden Mediendatenstrom m_{out} ab. Es gilt die Abbildung:

$$K_M : M \rightarrow M \text{ mit } m_{out} = K_M(m_{in})$$

Dabei wird K_M als logische Abbildung bezeichnet. Eine Klassifikation, die sich an der Wirkung der Konvertierfunktion k auf das Eingangsformat f_{in} orientiert, soll hier als Logische Klassifikation bezeichnet werden.

Des Weiteren benötigt ein Konverter Ressourcen R_i des darunterliegenden Systems zur Realisierung seiner Konvertierfunktion. Dabei erzeugt er eine Menge von Datenströmen d auf den benötigten Ressourcen R_i ausgedrückt durch die Abbildung:

$$K_{BM} : M \rightarrow \wp(BM \times D) \text{ mit } (R, d)^n = K_{BM}(m_{in})$$

Die Funktion K_{BM} beschreibt die physische Abbildung. Die Klassifizierung ausgehend vom Ressourcenverbrauch $|R_i|$ soll in dieser Arbeit entsprechend als Physische Klassifikation bezeichnet werden.

Ein Konverter ist durch seine Ein- und Ausgangsdaten sowie durch seine Konvertierfunktion und den Ressourcenbedarf charakterisierbar (Abbildung 6).

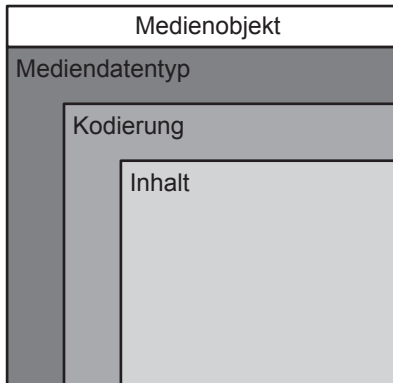


Abbildung 7: Grobstrukturierung eines Medienobjektes

5.2 Logische Klassifizierungsmerkmale von Konvertern

Ausgehend von dem in [Mär00] entwickelten Konvertermodell bietet sich zunächst ein Vergleich der Ein- und Ausgangsdaten eines Konverters an. Dabei wird die Wirkung der Konvertierfunktion auf die Eingangsdaten untersucht. Ausgangspunkt der Untersuchungen ist die Formatbeschreibung f , wobei in dieser Arbeit f_{in} das Eingangs- und f_{out} das Ausgangsformat des Konverters bezeichnen.

Ein Medienobjekt $mobj \in MObj$ kann grob durch 3 Eigenschaften charakterisiert werden, wobei $MObj$ die Menge aller Medienobjekte darstellt. Diese Eigenschaften sind:

- Mediendatentyp $mtyp \in MTyp$
 $MTyp$ ist die Menge aller Mediendatentypen, der nach [MW] u.a. die Elemente Grafik, Rasterbild, Text und Video angehören.
- Kodierung / Darstellung
- Inhalt $|mobj|$

Mit dem Inhalt $|mobj|$ eines Medienobjektes $mobj$ ist in dieser Arbeit der tatsächliche Informationsgehalt des Medienobjektes gemeint, welcher durch die in einer Darstellungsform (Kodierung) vorliegenden Rohdaten eines Medienobjektes beschrieben wird. D. h. der Inhalt ändert sich, wenn bei gleichbleibender Kodierung die Rohdaten geändert werden. Dahingegen bleibt der Inhalt gleich, wenn sich ausschließlich die Kodierung der Rohdaten ändert. Zwei Medienobjekte $mobj$ und $mobj'$ sind also inhaltsgleich ($|mobj| = |mobj'|$), wenn sie durch eine Änderung der Darstellungsform ineinander überführt werden können. Wirkt sich jedoch die Änderung der Darstellung zusätzlich auf den Inhalt eines Medienobjektes aus (verlustbehaftete Verfahren), so verringert sich der Informationsgehalt des Medienobjektes und es gilt $|mobj| > |mobj'|$.

Die Parameter Medientyp, Darstellung und Inhalt bedingen sich gegenseitig. Für einem Medientyp gibt es eine medientypabhängige Kodierung. Wobei die Kodierung Auswirkungen auf den Inhalt eines Medienobjektes haben kann.

Eine Konvertierfunktion kann allgemein Änderungen am Medientyp, an der Darstellung und am Inhalt vornehmen. Dies hängt von der konkreten Zielstellung der Konvertierung ab. So sind Medientypkonverter, Formatkonverter oder Inhaltskonverter denkbar.

Insbesondere bei den im Multimediabereich angewandten Kodiervverfahren ist eine Unterscheidung in verlustbehaftete und verlustlose Verfahren sinnvoll. Ein Verfahren ist verlustlos, wenn für die Konvertierfunktion:

$$k : MObj \rightarrow MObj \text{ mit } mobj' = k(mobj)$$

gilt:

$$\forall mobj \in MObj \exists k^{-1} : MObj \rightarrow MObj \text{ mit } mobj = k^{-1}(mobj')$$

Das heißt, für eine verlustlose Konvertierfunktion k existiert eine Umkehrfunktion k^{-1} , die in der Lage ist, den Inhalt $|mobj|$ von $mobj$ aus dem Inhalt $|mobj'|$ von $mobj'$ zu rekonstruieren. Im Umkehrschluss gilt für ein verlustbehaftetes Verfahren für die Konvertierfunktion k der Zusammenhang:

$$\exists mobj \in MObj \nexists k^{-1} : MObj \rightarrow MObj \text{ mit } mobj = k^{-1}(mobj')$$

Eine Vielzahl von Konvertierverfahren arbeitet verlustbehaftet mit jeweils unterschiedlichen Auswirkungen auf das menschliche Wahrnehmungsvermögen. Auch wenn verlustbehaftete Verfahren sehr gut an die menschliche Wahrnehmung angepasst sind, spielt der Informationsverlust eine entscheidende Rolle, geht doch dadurch Information unwiederbringlich verloren.

Für die Inhaltsbearbeitung steht eine große Anzahl von Verarbeitungsroutinen zur Auswahl. Hier nehmen Filter eine Sonderstellung ein, da sie ihre Effekte durch die gezielte Extraktion von Inhalten erzielen. Bei der Anwendung eines Filters auf ein Medienobjekt $mobj$ durch die das Medienobjekt $mobj'$ entsteht, gilt demnach $|mobj| \geq |mobj'|$. Der Informationsgehalt des Medienobjektes ändert sich nur, wenn sein Inhalt die Filterkriterien erfüllt.

Die Logische Klassifizierung untersucht was, wie und zu welchem Zweck an einem Medienobjekt verändert wird.

5.3 Logische Klassifikation

Die logische Klassifikation erfolgt gemäß Abbildung 8. In einem ersten Schritt wird die Reichweite der Konvertierfunktion betrachtet, d.h. welche Merkmale des Medienobjektes sind von der Konvertierung betroffen.

Dabei hat eine Medientypkonvertierung die größte Reichweite, beeinflusst sie doch neben dem Medientyp auch die Darstellung und in manchen Fällen den

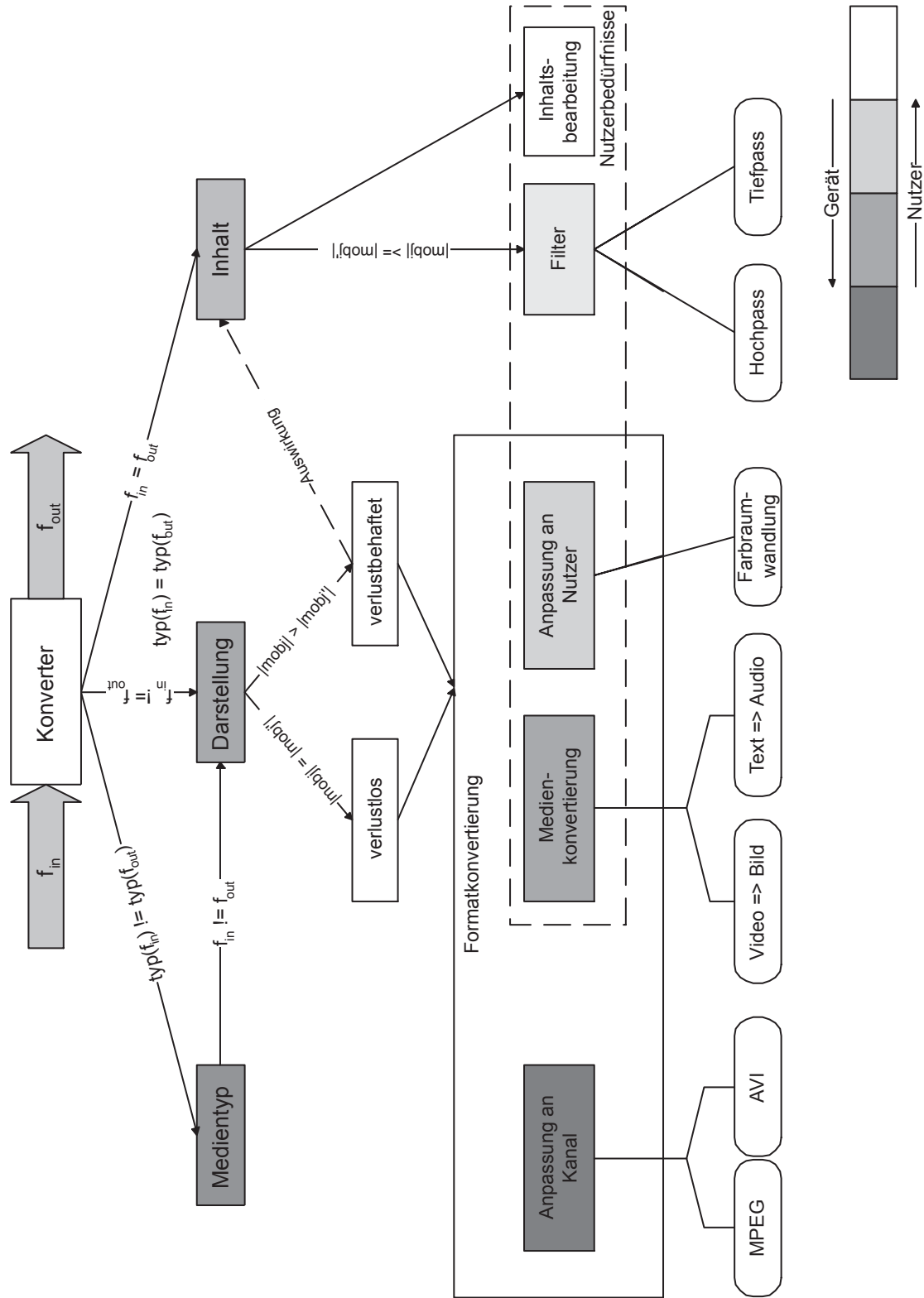


Abbildung 8: Logische Klassifikation

Inhalt eines Medienobjektes. In jedem Fall ist eine Medientypkonvertierung mit einer Darstellungsänderung verbunden.

Eine Darstellungswandlung ändert den Medientyp nicht. Sie kann sich jedoch auf den Inhalt eines Medienobjektes auswirken. Die Reichweite ist geringer als bei einer Medientypwandlung. Es kann in Abhängigkeit von der konkreten Konvertierfunktion neben dem Format auch der Inhalt mit betroffen sein. Ein Konverter, der die Darstellung eines Medienobjektes bearbeitet, soll im Weiteren als Koder bezeichnet werden. Da im Multimediabereich eine Vielzahl inhaltsrelevanter Kodierverfahren eingesetzt werden, ist eine Unterscheidung in verlustbehaftet und verlustlos angebracht.

Ein Kodierverfahren ist verlustbehaftet, wenn sich der Informationsgehalt eines Medienobjektes verringert, d.h. es gilt $|mobj| > |mobj'|$. Dies ist beispielsweise bei den verschiedenen Abtastformaten des YCbCr-Farbmodells, speziell bei 4:2:2, 4:1:1 und 4:2:0 der Fall. Auch Verfahren, die auf Interpolation beruhen, zählen zu den verlustbehafteten. Jede verlustbehaftete Darstellungskonvertierung hat Auswirkungen auf den Inhalt eines Medienobjektes $|mobj|$. Ob ein Verfahren verlustbehaftet ist oder nicht, ist relevant für die Einordnung eines entsprechenden Converters in eine Converterkette. Hier können Erkenntnisse abgeleitet werden, ob ein Converter am Anfang oder eher am Ende einer Converterkette eingeordnet werden soll oder ob die Reihenfolge der Operationen irrelevant ist. Wobei für Letztere am ehesten verlustlose Verfahren in Frage kommen. Die Vermutung liegt nahe, verlustbehaftete Verfahren vorrangig am Ende einer Converterkette einzuordnen.

Eine Inhaltskonvertierung hat eine direkte Manipulation des Inhaltes, d.h. der Information selbst, zum Ziel. Zu einer Inhaltskonvertierung zählen z.B. die Extraktion von Farbkomponenten oder Filterung von Frequenzanteilen eines Audios. Aber auch verschiedene Bearbeitungsroutinen wie Einblenden von Grafiken oder Entfernen bzw. Hinzufügen von Szenen bei Video zählen mit dazu. Diese Routinen sind allgemein unter dem Begriff Inhaltsbearbeitung zusammengefasst.

Eine Sonderstellung bei der Inhaltskonvertierung nehmen Filter ein. Diese erzielen ihre Effekte durch die gezielte Extraktion von Inhalten eines Medienobjektes. Man könnte auch den Begriff verlustbehaftet gebrauchen. Die Unterscheidung in verlustbehaftet oder verlustlos wird bei ausschließlich inhaltsbearbeitenden Convertern bewusst vermieden. Der Grund dafür ist, dass der Inhalt gezielt manipuliert wird, d.h. etwaige Verluste in Kauf genommen werden. Eine Inhaltskonvertierung dient im hohem Maße der Befriedigung von Nutzerbedürfnissen.

Im Gegensatz zu einer Inhaltskonvertierung läuft eine Darstellungskonvertierung mindestens auf die Veränderung des Formates eines Mediendatentypes hinaus, d.h. es ist das Format f mit seinen Kenngrößen betroffen. Derartigen Formatkonvertierungen können viele Zielsetzungen zu Grunde liegen. In einem ersten Schritt kann man feststellen, dass diese der Anpassung an ein Gerät (spezielle Hardware) oder an Nutzerbedürfnisse dienen. Die Grenzen sind jedoch fließend und von jedem Einzelfall abhängig. In [Abbildung 8](#) ist dieser Sachverhalt

durch die Legende angedeutet. Man kann die Tendenz ableiten, dass von links nach rechts zunehmend Nutzerbedürfnisse eine Rolle spielen. Wohingegen von rechts nach links die Anpassung an ein Gerät oder eine spezielle Anwendung im Vordergrund stehen.

Formatkonvertierungen können zunächst der Anpassung an einen Übertragungskanal (Kanalkodierung) dienen. Ein wichtiger Vertreter dieser für die Übertragung optimierter Formate ist MPEG-2. MPEG-2 wurde speziell für die Anforderungen der digitalen Massenfernsehübertragung entwickelt. Die speziellen Verfahren der Prediktion, Bewegungskompensation und der Quantisierung nach der DCT führen zu einer Reduktion der Bitraten. Dadurch wird die Übertragung von MPEG-2 Datenströmen, z.B. über Netzwerke unter Einhaltung von Zeitrestriktionen, unterstützt. Im weiteren Sinne kann unter Kanalkodierung auch die Speicherplatzoptimierung¹⁹ verstanden werden. Anpassung an einen Kanal bedeutet hier nicht zwangsläufig die Vorbereitung für einen Übertragungsweg.

Wie schon bemerkt, führt eine Medientypkonvertierung zu einer Darstellungsänderung und damit zu einer Formatkonvertierung. Sie wird unter dem Begriff Formatkonvertierung separat aufgeführt. Das ermöglicht eine genauere Zuordnung von konkreten Beispielen. Medientypkonvertierungen können verlustlos oder verlustbehaftet sein. So entsteht bei der Wandlung von Text zu Audio im Allgemeinen kein Verlust. Es ändern sich lediglich der Medientyp und seine Darstellung. Anders verhält es sich bei der Wandlung Video zu Rasterbild. Hier wird nur die Information, die in einem einzelnen Frame steckt, erhalten. Der Rest des Videos geht typischerweise verloren. Es ändern sich der Medientyp, die Darstellung und der Inhalt. Medientypkonvertierungen dienen sowohl der Anpassung an Nutzerbedürfnisse als auch der Anpassung an Geräte.

Unter dem Punkt Anpassung an den Nutzer sind alle Konvertierfunktionen zusammengefasst, die vorrangig von Nutzern initiiert werden. Das können Farbraumwandlungen, Skalierungen oder Koordinatentransformationen sein. An dieser Stelle wird noch einmal erwähnt, dass sich zwischen Anpassung an Nutzer und Anpassung an Gerät keine scharfe Grenze ziehen lässt. Es ist lediglich eine Tendenz erkennbar, wie in der Legende in Abbildung 8 dargestellt. So kann beispielsweise eine Farbraumwandlung auch der Anpassung an einen Monitor dienen und nicht vordergründig der Befriedigung von Nutzerinteressen.

In Tabelle 3 sind die sich unterscheidenden Merkmale von Medientyp-, Darstellungs- und Inhaltskonvertierung zusammengefasst. Wobei die Funktion *typ* den Medientyp eines Medienobjektes liefert. Sie ist definiert als:

$$typ : F \rightarrow MTyp \text{ mit } mtyp = typ(f)$$

Für die Formatunabhängigkeit lässt sich an Abbildung 8 eine Tendenz für die Relevanz der Konvertierfunktion für einen Medienserver erkennen. So sollte der

¹⁹Reduktion des Datenumfanges mit Hilfe von Komprimierungsverfahren

Konverter		
Medientyp	Darstellung	Inhalt
$typ(f_{in}) \neq typ(f_{out})$ $f_{in} \neq f_{out}$	$typ(f_{in}) = typ(f_{out})$ $f_{in} \neq f_{out}$	$typ(f_{in}) = typ(f_{out})$ $f_{in} = f_{out}$

Tabelle 3: Die Tabelle fasst die sich unterscheidenden Merkmale von Medientyp-, Darstellungs- und Inhaltskonvertierung zusammen.

Medienserver mindestens die Funktionalität der Medientyp- und Darstellungs-konvertierung anbieten, um Formatunabhängigkeit zu gewährleisten. Bei den inhaltstbearbeitenden Routinen ist in jedem Einzelfall zu prüfen, inwieweit die Funktionalität von einem Medienserver bereitgestellt werden muss. Der Medienserver wird demnach vorrangig Konvertierfunktionen für den Medientyp und die Darstellung anbieten müssen. Inhaltsbearbeitende Konverter besitzen eine geringere Relevanz für die Realisierung in einem Medienserver. Ausgenommen davon sind einige Filter wie z.B. Filter für das Subsampling für das YCbCr-Farbmodell.

5.4 Physische Klassifizierungsmerkmale von Konvertern

Die Physische Klassifikation beschäftigt sich mit dem Ressourcenverbrauch eines zunächst einzelnen Konverters. Physische Klassifizierungsmerkmale charakterisieren den Verbrauch an Betriebsmitteln und bieten damit eine Grundlage für die Einplanung von Konvertern mit definierten Konvertierfunktionen.

Wesentlich für eine Einplanung ist die Frage, ob der Ressourcenverbrauch anhand von Eigenschaften des Medienobjektes zuverlässig bestimmbar, in guter Näherung abschätzbar oder nicht bestimmbar ist. Kann der Verbrauch nicht bestimmt werden, spielen evtl. neben den Medientypeigenschaften noch andere Einflüsse eine Rolle.

Eine Voraussetzung, die die Bestimmung des Ressourcenverbrauches und damit die Einplanung wesentlich vereinfacht, ist die Determiniertheit. Demgegenüber sind auch Konverter denkbar, bei denen im besten Fall lediglich eine Abschätzung getroffen werden kann. Die konkrete Auslastung von Betriebsmitteln variiert bei wiederholter Durchführung der Konvertierung mit dem selben Medienobjekt. Ein solches Verhalten ist indeterministisch, weil nicht eindeutig vorhersehbar.

Ein weiterer Aspekt ist die Beschreibung, inwieweit sich der Ressourcenverbrauch während der Ausführung der Konvertieroperation verändert. Im günstigsten Fall ist dieser unter allen Umständen konstant. Demgegenüber kann dieser auch nicht konstant sein und während der Bearbeitungszeit variieren. Ein solches Verhalten macht eine exakte Einplanung schwieriger.

Die Physische Klassifikation strukturiert die dargestellten Klassifizierungsmerkmale und leitet Erkenntnisse für die Machbarkeit (Einplanbarkeit) einer Konvertierfunktion ab.

5.5 Physische Klassifikation

Ein wesentliches Klassifizierungsmerkmal der Physischen Klassifikation ist Determinismus bzw. Indeterminismus. Wobei Determinismus eine wichtige Voraussetzung für die Angabe eines funktionalen Zusammenhanges zwischen bekannten beschreibenden Merkmalen eines Medienobjektes und dem Ressourcenverbrauch, im Folgenden als $|R_i|$ bezeichnet, ist. Die Formatbeschreibung f_{in} lässt Rückschlüsse auf die Betriebsmittelnutzung zu. Kann ein funktionaler Zusammenhang abgeleitet werden, ist eine exakte Bestimmung der Auslastung einer jeden beanspruchten Ressource möglich. Damit kann sehr elegant die traditionelle Worst-Case-Analyse vermieden werden. Dies führt zu einer besseren Auslastung der Betriebsmittel, da der Ressourcenverbrauch in seinen wesentlichen Eigenschaften bekannt ist.

Ist kein funktionaler Zusammenhang ableitbar bzw. variiert der Ressourcenverbrauch bei wiederholter Anwendung ein und derselben Konvertierfunktion auf einem Medienobjekt ständig, liegt Indeterminismus vor. Genaue, den Ressourcenverbrauch vorhersagende Angaben sind nicht möglich. Dies kann darin begründet liegen, dass genaue Abhängigkeiten nicht bekannt bzw. nicht ableitbar sind. Ebenso können medienunabhängige Einflüsse eine Rolle spielen. Unter medienunabhängigen Einflüssen werden in dieser Arbeit sämtliche Einflüsse, die außerhalb der Formatbeschreibung f liegen, verstanden.

Indeterminismus allein macht eine Einplanung eines Konverters noch nicht unmöglich. Eine traditionelle Worst-Case-Analyse kann hier weiter helfen, wobei sämtliche Wiederholungen der Konvertierung zur Bestimmung einer oberen Schranke (so vorhanden) maximal oft durchgeführt werden müssen. Das Vorhandensein einer oberen Schranke für die Nutzung einer Ressource, die zuverlässig bestimmt werden muss, ist dabei ein entscheidendes Kriterium. Diese muss dann als Worst-Case-Annahme bei der Ressourcenreservierung berücksichtigt werden. Ressourcen werden zwar verschwendet, jedoch ist die Durchführung der Konvertierung unter Echtzeitbedingungen möglich.

Ergibt eine entsprechende Analyse keine obere Schranke für den Ressourcenverbrauch, kann die Konvertierung nicht unter Echtzeitbedingungen durchgeführt werden, da diese nicht im Voraus reserviert werden können [HRW⁺99]. Dies ist der ungünstigste Fall und wird im Folgenden nicht weiter betrachtet.

Ein weiteres Klassifizierungsmerkmal ergibt sich aus der Charakterisierung des funktionalen bzw. gemessenen Zusammenhanges für die Betriebsmittelnutzung. Die beschreibende Funktion, soweit bestimmbar, kann absolut konstant, zeitlich konstant oder nicht konstant sein.

Die absolute Konstanz stellt einen Sonderfall dar. Ein Konverter, welcher einen absolut konstanten Ressourcenverbrauch aufweist, benötigt unabhängig von anderweitigen Einflüssen immer die gleiche konstante Bandbreite auf der Ressource. Seine Konvertierfunktion zeichnet sich dadurch aus, dass sie unabhängig von den beschreibenden Medienobjektmerkmalen in f arbeitet.

Wesentlich allgemeiner ist die zeitliche Konstanz des Ressourcenverbrauches.

Konverter		
<i>deterministisch</i>		
$f_{ R_i } : F \rightarrow R $ mit $ R_i = f_{ R_i }(f_{in})$		
konstant		nicht konstant
absolut konstant	zeitlich konstant	
$ R_i = \text{konst.}$	$ R_i = f_{ R_i }(f_{in}) = \text{konst.}$	$ R_i = f_{ R_i }(f_{in}) = f(t)$

Tabelle 4: Physische Klassifikation (deterministisch)

Die den Ressourcenverbrauch charakterisierende Funktion ist über die Laufzeit der Konvertierfunktion konstant. Die konkrete Auslastung eines Betriebsmittels kann von Medienobjekt zu Medienobjekt variieren. Der funktionale Zusammenhang ist über der Zeit immer konstant und lässt sich aus f_{in} ableiten.

Schließlich kann die Funktion noch variieren, also nicht konstant sein. Allen drei genannten Eigenschaften (absolut konstant, zeitlich konstant, nicht konstant) ist gemein, dass sich ein funktionaler Zusammenhang ableiten lässt, der Ressourcenverbrauch demnach determiniert ist.

Für den Fall eines indeterministischen Verbrauches eines Betriebsmittels ist Beschränktheit eine Voraussetzung für eine Einplanbarkeit eines Konverters unter Echtzeitbedingungen. Auch hier kann Konstanz oder Nichtkonstanz vorliegen. Entscheidend ist hier, dass sich durch entsprechende Analysen eine obere Grenze zur Abschätzung der Ausnutzung eines jeden benötigten Betriebsmittels finden lässt. Die obere Schranke kann mit der Zeit variieren oder konstant sein.

Im Allgemeinen benötigt ein Konverter mehrere Ressourcen für die Durchführung einer Konvertierung. Der Ressourcenverbrauch eines Konverters kann als determiniert charakterisiert werden, wenn für jede benötigte Ressource ein funktionaler Zusammenhang abgeleitet werden kann. Sonst ist er indeterministisch, d.h es existiert mindestens eine Ressource für die keine Aussage getroffen werden kann. Ebenso verhält es sich mit den Klassifizierungsmerkmalen konstant und beschränkt. Dies gilt jeweils nur, wenn das für alle benötigten Betriebsmittel festgestellt wird. Sonst gilt das Gegenteil, nicht konstant bzw. nicht beschränkt. Treten Mischformen für die beteiligten Ressourcen auf, z.B. konstant / nicht konstant oder beschränkt / nicht beschränkt, so wird zur Charakteristik des Gesamtverbrauchs die jeweils ungünstigere Variante angenommen. D.h. die Gesamtcharakteristik orientiert sich jeweils am Ressourcenverbrauch, der am weitesten rechts in Abbildung 9 liegt.

Die gewonnenen Erkenntnisse werden in Abbildung 9 und in den Tabellen 4 und 5 zusammengefasst. In den Tabelle 4 und 5 bezeichnen $|R_i|$ den Verbrauch auf Ressource R_i und $f_{|R_i|}$ den entsprechenden funktionalen Zusammenhang. Die Komplexität der Ressourcenplanung unter Echtzeitbedingungen nimmt in Abbildung 9 von links nach rechts zu. Besonders im indeterministischen Fall ist es schwierig, eine obere Schranke für den Ressourcenverbrauch zu bestimmen.

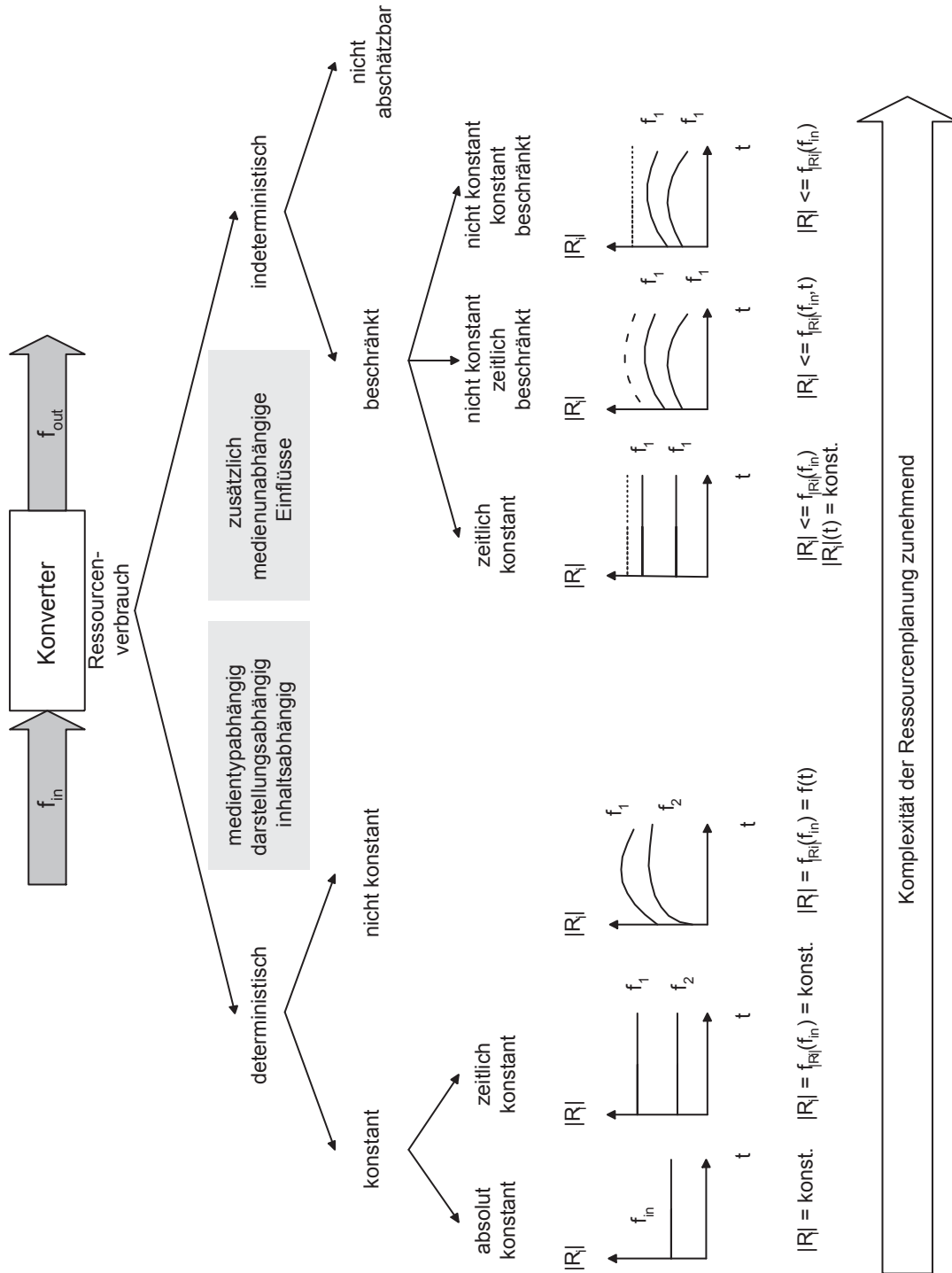


Abbildung 9: Physische Klassifikation

Konverter			
<i>indeterministisch</i>			
keine Funktion ableitbar			
beschränkt			nicht abschätzbar
zeitlich konstant	nicht konstant zeitlich beschränkt	nicht konstant konstant beschränkt	-
obere Schranke als Worst-Case			
$ R_i \leq f_{ R_i }(f_{in})$ $ R_i = \text{konst.}$	$ R_i \leq f_{ R_i }(f_{in}, t)$	$ R_i \leq f_{ R_i }(f_{in})$	

Tabelle 5: Physische Klassifikation (indeterministisch)

5.6 Beispiele

Abschließend werden einige Konverter in das Schema der Logischen und Physischen Klassifikation eingeordnet.

5.6.1 Farbraumwandlung RGB zu YCbCr 4:2:2

Eine Farbraumwandlung kann im Allgemeinen auf eine 3×3 Matrixmultiplikation zurückgeführt werden. Ein Konverter, der gamma-korrigiertes RGB (R'G'B') in YCbCr umwandelt, rechnet 3 Farbkomponenten gemäß der Berechnungsvorschrift 1 in YCbCr um. Es ändert sich das Farbmodell und damit liegt eine Darstellungsänderung vor. Die Interpretation der 3 Komponenten zur Farbkodierung eines Bildpunktes ändert sich.

Wie schon im Abschnitt 4.5 dargelegt, geht bei einer derartigen Farbraumwandlung Farbinformation verloren. Von RGB zu YCbCr gehen 74 % der darstellbaren Farben verloren. Eine solche Konvertierung ist verlustbehaftet. Der Informationsverlust wird durch Subsampling noch zusätzlich verstärkt.

Die Farbraumwandlung fügt sich damit in das Schema der Logischen Klassifikation als verlustbehaftete Darstellungsänderung ein. In Abbildung 8 wird dies der Anpassung an Nutzerbedürfnisse zugerechnet. Eine Farbraumwandlung, insbesondere zu YCbCr, kann aber auch einer MPEG-Kodierung vorausgehen und damit eine Kanalkodierung vorbereiten.

Die Umrechnung der 3 Komponenten des RGB-Modells zu YCbCr erfolgt für jeden einzelnen Bildpunkt. Daraus folgt, dass die Farbraumwandlung medienabhängig, im Speziellen bildgrößenabhängig ist. Anhand der Bildgröße und der Zahlenbereiche in die abgebildet wird, kann der Verbrauch an Ressourcen exakt errechnet werden. Dieser bleibt während der Verarbeitung konstant.

Nimmt man vereinfachend an, dass neben der Ressource Speicher nur noch die Ressource CPU benutzt wird, fügt sich diese spezielle Farbraumwandlung recht einfach in das Schema der Physischen Klassifikation als deterministisch

und zeitlich konstant ein. Da für jede RGB-Komponente eines Bildpunktes eine Berechnung gemäß Formel 1 vorgenommen wird, steht fest, dass die CPU bei wiederholter Ausführung immer gleich belastet wird, das Verfahren also deterministisch und zeitlich konstant ist.

Werden mehr als eine Ressource betrachtet (z.B. Speicherbus etc.), ist zu prüfen, inwieweit der Ressourcenbedarf konstant ist. In dem speziellen Fall der Farbraumkonvertierung kann davon ausgegangen werden, dass abhängig von der Bildgröße die zur CPU übertragenen Informationen ebenfalls zeitlich konstant sind.

5.6.2 Interlaced-to-Noninterlaced Wandlung

Eine Interlaced-to-Noninterlaced-Wandlung wird im Kontext der Logischen Klassifikation einer Darstellungsänderung zugeordnet. Sie wirkt sich auf die Formatbeschreibung f aus. Je nach konkretem Verfahren gemäß 4.3 ist diese Konvertierung verlustlos oder verlustbehaftet.

Die Verfahren der Scan Line Duplikation und der Scan Line Interpolation sind als verlustbehaftet einzustufen, da hier nur Informationen aus einem Teilbild zur Rekonstruktion des Gesamtbildes herangezogen werden. Wohingegen das Verfahren Field Merging verlustfrei ist. Das Gesamtbild entsteht aus der Fusion der beiden zeitlich getrennten Teilbilder.

Bei der Interlaced-to-Noninterlaced Wandlung steht die Anpassung an ein Gerät im Vordergrund. So kann eine Wandlung zur Darstellung eines Videos, welches für die Fernsehdarstellung gedacht war, auf einem Computerbildschirm ermöglicht werden.

In das Schema der Physischen Klassifikation ordnet sich diese Konvertierung als deterministisch und zeitlich konstant ein. Im Beispiel wird davon ausgegangen, dass die Ressourcen CPU und Speicher beansprucht werden. Die Auslastung der Ressourcen kann anhand der Bildauflösung, die Bestandteil der Formatbeschreibung f ist, genau bestimmt werden.

5.6.3 MPEG Konvertierung

Eine MPEG-Konvertierung ist ein weiteres Beispiel einer Darstellungsänderung. Das MPEG-Format ist eines der gebräuchlichsten Kodierungsverfahren im Videobereich²⁰.

Um hohe Kompressionsraten zu erzielen, arbeitet die MPEG-Kodierung verlustbehaftet. Der Verlust entsteht durch Runden von Frequenzanteilen auf Null, die nach der DCT entstehen. Zusätzlich werden noch Verfahren, die auf Prediktion und Bewegungskompensation beruhen, eingesetzt. Eine ausführliche Darstellung der MPEG-Videokodierung ist in Abschnitt 3.3.1 enthalten.

²⁰Der MPEG-Standard umfasst darüber hinaus auch Kodierverfahren für Audiodaten.

Eine MPEG Kodierung dient im hohem Maße einer Kanalkodierung. Meist ermöglicht erst eine erhebliche Reduktion der Datenrate eine Übertragung über Kanäle mit geringer bzw. begrenzter Bandbreite. MPEG-2 ist Grundlage von HDTV²¹, einem Standard für digitale Massenfernsehübertragung mit hoher Qualität.

In das Schema der Physischen Klassifikation ist die MPEG Konvertierung als determiniert und zeitlich nicht konstant einzuordnen. An dieser Stelle wird jedoch darauf hingewiesen, dass die Ressourcenplanung nur dann zuverlässig möglich ist, wenn die Formatbeschreibung f Rückschlüsse auf den Inhalt (z.B. Bewegungshäufigkeit, Szenen) zulässt.

²¹high definition television

6 Suchverfahren für Konverterketten

Im Folgenden werden Verfahren zur Suche nach Lösungen für verschiedene Problemstellungen diskutiert und ihre Relevanz bzw. Eignung für die Suche nach Konverterketten bewertet.

Häufig ist es schwierig, für ein gestelltes Problem eine Lösung zu finden. Das kann den Grund haben, dass das Problem zu kompliziert ist und man auf vereinfachende Modelle zurückgreifen muss. Wobei sich dann immer die Frage stellt, ob eine gefundene Lösung, die das Modell erfüllt, auch auf die Wirklichkeit, also auf das ursprüngliche Problem, anwendbar ist. Des Weiteren ist die Anzahl möglicher Lösungen, der sogenannte Suchraum \mathcal{S} , häufig sehr groß. So dass es nicht möglich ist, durch Ausprobieren aller möglichen Lösungen in einer angemessenen Zeitspanne zu einem Ergebnis zu gelangen. So spielt die Größe des Suchraumes $|\mathcal{S}|$ eine wesentliche Rolle bei der Auswahl eines geeigneten Suchverfahrens. Diese lässt sich je nach Anwendungsfall, z.B. durch kombinatorische Verfahren, abschätzen.

Eine Konverterkette ist eine Sequenz der Länge l von Konvertern aus der Menge K aller Konverter des Medienservers. Im Fall der Suche nach Konverterketten ist die genaue Bestimmung der Größe des Suchraumes $|\mathcal{S}|$ schwierig. Hier spielen eine Reihe unterschiedlicher Einflüsse eine Rolle. So müsste man die maximale Länge einer Konverterkette l_{max} , d.h. die Maximalanzahl von Konvertern in einer Konverterkette, kennen. Diese abzuschätzen, ist schwierig und vom konkreten Einzelfall abhängig. Weiterhin müsste eine genaue Abhängigkeit bekannt sein, die angibt, wieviele Konverter als Nachfolger nach Auswahl eines speziellen in Frage kommen. Das wird von den jeweiligen Aus- und Eingangsformaten und der geforderten Konvertierfunktion abhängen. Deshalb kann hier nur eine Abschätzung in Form einer oberen Grenze $\max(|\mathcal{S}|)$, basierend auf vereinfachenden Annahmen, angegeben werden. Der Abschätzung der oberen Grenze liegen folgende Annahmen zu Grunde:

1. Die Reihenfolge der Konverter in einer Konverterkette ist relevant.
2. Ein und derselbe Konverter kommt nur einmal in einer Konverterkette vor.
3. Es stehen insgesamt $n = |K|$ Konverter zur Verfügung.
4. Für die Länge l einer Konverterkette gilt $1 \leq l \leq l_{max}$.

Basierend auf diesen Annahmen ergibt sich die maximale Größe $\max(|\mathcal{S}|)$ des Suchraumes \mathcal{S} als Summe der Variationen V_n^l ohne Wiederholung gemäß nachfolgender Formel 2:

$$\max(|\mathcal{S}|) = \sum_{l=1}^{l_{max}} V_n^l = \sum_{l=1}^{l_{max}} \frac{n!}{(n-l)!} \text{ für } l_{max} \leq n \quad (2)$$

Maximalgröße des Suchraumes	
$l_{max} = 50$	
$n = 50$	$\max(\mathcal{S}) = 8,267 \cdot 10^{64}$
$n = 100$	$\max(\mathcal{S}) = 3,130 \cdot 10^{93}$
$n = 150$	$\max(\mathcal{S}) = 1,961 \cdot 10^{104}$

Tabelle 6: Abschätzung der Suchraumgröße mit variabler Konverteranzahl

Maximalgröße des Suchraumes	
$n = 150$	
$l_{max} = 50$	$\max(\mathcal{S}) = 6,183 \cdot 10^{104}$
$l_{max} = 100$	$\max(\mathcal{S}) = 1,916 \cdot 10^{198}$
$l_{max} = 150$	$\max(\mathcal{S}) = 1,553 \cdot 10^{263}$

Tabelle 7: Abschätzung der Suchraumgröße mit variabler Maximallänge einer Konverterkette

Aus den Tabellen 6 und 7 geht hervor, dass bereits für eine relativ geringe Anzahl von Konvertern n und einer kleinen Maximallänge l_{max} der Konverterkette der Suchraum eine beträchtliche Größe annimmt.

Je größer der Suchraum ist, desto mehr muss man diesen durch geeignete Maßnahmen einschränken, um zu einem (wenn auch nicht immer optimalen) Ergebnis zu gelangen. Es sind geeignete Zusatzbedingungen zu formulieren, die der Lösung des gestellten Problems dienen und zusätzlich den Suchraum einschränken. Im Fall einer Konverterkette ist eine solche Nebenbedingung, dass das Eingangsformat des i -ten Konverters $f_{in,i}$ mit dem Ausgangsformat des $i-1$ -ten Konverters $f_{out,i-1}$ übereinstimmen muss. Der Teil des Suchraumes, welcher alle passenden Kombinationen von Konvertern enthält, wird in dieser Arbeit als $\mathcal{F} \subseteq \mathcal{S}$ bezeichnet.

Für eine weitere Einschränkung des Suchraumes kommen auch die Erkenntnisse der Logischen Klassifikation in Betracht. So kann je nach Anwendungsfall gezielt die Menge der Medientyp-, Darstellungs- oder Inhaltskonverter durchsucht werden. Auch ist für eine Konvertierung, die den Mediendatentyp nicht ändert, nur die Menge der Konverter relevant, die Operationen für diesen Mediendatentyp bereitstellt. Beides führt zu einer weiteren erheblichen Einschränkung des Suchraumes, da von vornherein Lösungen, die nicht zum Ziel führen, ausgeschlossen werden.

Wenn eine gefundene Konverterkette alle Zusatzbedingungen erfüllt und die geforderte Konvertierung leistet, so ist diese funktional korrekt. Da \mathcal{F} nur alle passenden Kombinationen von Konvertern enthält, können demnach auch Konverterketten enthalten sein, die nicht funktional korrekt im Sinne einer geforderten Konvertierung sind. Ein anderer Aspekt ist die Frage, wann eine gefunde-

ne Lösung optimal ist. Zur Beurteilung wie gut eine gefundene Konverterkette ist, benötigt man eine Möglichkeit der Evaluierung, eine sogenannte Evaluierungsfunktion. Diese kann sich z.B. am Ressourcenverbrauch orientieren. Unter diesem Gesichtspunkt kann eine Konverterkette als optimal angesehen werden, wenn diese leicht einplanbar ist und/oder nur wenig Bandbreite auf der jeweiligen Ressource benötigt. Alle gefundenen, funktional korrekten Konverterketten werden anschließend einer Machbarkeitsanalyse unterzogen [MWM02]. Somit wird jede Konverterkette unter diesem Gesichtspunkt evaluiert. Eine Evaluierung, die sich am Ressourcenverbrauch orientiert, kann zusätzlich zur Einschränkung des Suchraumes benutzt werden, indem Konverterketten mit unerfüllbaren Ressourcenbedarf ausgeschlossen werden.

Die Suche nach Konverterketten kann demnach in zwei Teilaufgaben gegliedert werden.

1. Suche nach einer funktional korrekten Konverterkette in der Menge \mathcal{F} , die alle passenden Kombinationen von Konvertern aus der Menge K , der im Medienserver zur Verfügung stehenden Konverter enthält.
2. Suche einer optimalen Konverterkette aus der Menge $K_{kor} \subseteq \mathcal{F}$ der funktional korrekten Konverterketten für die geforderte Konvertierung. Die Optimierung orientiert sich dabei am Ressourcenverbrauch. Hierfür ist eine geeignete Evaluierungsfunktion zu entwickeln.

Für die Suche nach funktional korrekten Konverterketten scheint es nicht sinnvoll, alle möglichen Kombinationen von Konvertern aus K in die Suche einzubeziehen. Deshalb sollte zuerst die Menge K_r , der für diese Konvertierung relevanten Konverter gebildet werden. Das kann ausgehend vom Zielformat geschehen, indem alle Konverter, die entsprechende Ein- bzw. Ausgangsformate besitzen, in die Suche einbezogen werden. Somit wird vermieden, dass sämtliche Konverter des Medienservers, insbesondere nicht relevante bei der Suche berücksichtigt werden. Es ist demnach möglich, den Suchraum der passenden Kombinationen von Konvertern \mathcal{F} zu verkleinern. Die Frage, ob eine Kombination von Konvertern in \mathcal{F} die geforderte Konvertierung leistet, muss dann mit Hilfe einer geeigneten Evaluierungsfunktion entschieden werden. Erst im Anschluss kann aus der Menge K_{kor} der funktional korrekten Konverterketten eine optimale Konverterkette mittels einer anderen Evaluierungsfunktion gesucht werden.

Prinzipiell kann man die Suchverfahren in zwei große Gruppen einteilen. Es gibt Verfahren, die

- mit vollständigen Lösungen oder
- mit Teil- bzw. Näherungslösungen

arbeiten. Immer wenn ein Verfahren mit vollständigen Lösungen arbeitet, steht zu jedem Zeitpunkt wenigstens eine potentielle Lösung für das gestellte Problem fest,

ganz im Gegensatz zu Verfahren, die mit Teillösungen arbeiten. Eine vollständige Lösung liegt immer dann vor, wenn alle problemspezifischen Variablen spezifiziert sind.

Viele der folgenden Verfahren arbeiten nach dem Prinzip des Vergleichens von gefundenen vollständigen Lösungen mit Hilfe einer geeigneten Evaluierungsfunktion und der Manipulation dieser Lösung, um eine potentiell bessere zu finden.

Teillösungen sind nicht vollständige Lösungen des ursprünglichen Problems oder vollständige Lösungen für hoffentlich einfachere Teilprobleme. Nicht vollständige Lösungen decken eine Teilmenge des Suchraumes mit einer bestimmten Eigenschaft ab. Damit ist die Hoffnung verbunden, dass diese Eigenschaft dann auch auf die gesuchte Lösung zutrifft.

Alternativ ist das Vorgehen, ein Problem in mehrere einfachere Teilprobleme zu zerlegen. Jedes dieser Teilprobleme wird für sich gelöst, um anschließend aus der Kombination dieser Teillösungen die Gesamtlösung zu finden. Problematisch bei derartigen Verfahren ist die Festlegung einer geeigneten Evaluierungsfunktion, die die Teillösungen entsprechend ihrer Relevanz für das Gesamtproblem bewertet.

6.1 Umfassende Suche

Das Verfahren der Umfassenden Suche überprüft jede der potentiellen Lösungen des Suchraumes. Für die Suche nach einer funktional korrekten Konverterkette bedeutet diese Vorgehensweise, dass alle Kombinationen von Konvertern der Menge \mathcal{F} gebildet und auf funktionale Korrektheit getestet werden müssen. Das Verfahren arbeitet mit vollständigen Lösungen, d.h. jede Kombination von Konvertern stellt eine potentielle Lösung dar.

Hier entsteht zunächst das Problem alle potentiellen Lösungen in \mathcal{F} , d.h. alle möglichen Kombinationen von Konvertern aus K_r , aufzuzählen. Unterstellt man weiterhin, dass diese beliebig kombinierbar sind, so ergibt sich eine obere Grenze zur Abschätzung der Größe des Suchraumes $\max(|\mathcal{F}|)$ mit Hilfe der Formel:

$$\max(|\mathcal{F}|) = \sum_{l=1}^{l_{max}} \frac{|K_r|^l}{(l_{max}-l)!} \text{ für } l_{max} \leq |K_r| \quad (3)$$

Auch hier ist eine erhebliche Anzahl von Kombinationen aus der Menge K_r der relevanten Konverter zu erwarten, wenn $|K_r|$ und l_{max} hinreichend groß sind. Daher ist die Umfassende Suche, die auf der Aufzählung aller potentiellen Lösungen beruht, wenig geeignet, eine funktional korrekte Konverterkette zu finden. Schon bei Werten von $|K_r| = 50$ und $l_{max} = 10$ und einer unterstellten Bearbeitungszeit von 1 ms pro Lösung ergibt sich eine Rechenzeit von ca. $1,2 \cdot 10^6$ Jahren. Es bleibt demnach festzustellen, dass die Umfassende Suche für große Suchräume nicht praktikabel ist.

Sind für eine geforderte Konvertierung alle funktional korrekten Konverterketten bekannt, so kann die Umfassende Suche unter Einbeziehung einer Evaluierungsfunktion zum Finden einer optimalen Konverterkette eingesetzt werden. Der Vorteil des Verfahrens liegt darin, dass immer die global optimalste Konverterkette aus K_{kor} gefunden wird. Das Problem der Suchraumgröße bleibt, was die generelle Eignung des Verfahrens einschränkt.

6.2 Lokale Suche

Im Gegensatz zur Umfassenden Suche konzentriert sich die Lokale Suche nur auf einen kleinen Teil des Suchraumes. Ausgehend von einer potentiellen Lösung für das Problem wird die Nachbarschaft dieser Lösung untersucht. Dazu wird in jedem Schritt eine vollständige Ausgangslösung leicht verändert. Eine solche Transformation einer Lösung führt auf eine neue usw. Jede so erzeugte Lösung wird evaluiert. Die jeweils bessere Lösung dient als neue Ausgangslösung. Das Verfahren terminiert, wenn keine Verbesserung mehr erzielt werden kann.

Bezogen auf das Problem der Suche nach Konverterketten startet dieses Verfahren bei einer beliebigen Kombination von Konverten. Das Evaluierungskriterium ist zuerst funktionale Korrektheit. Es werden nun sukzessive alle Konverterketten erzeugt, die in der Nachbarschaft der Ausgangskonverterkette liegen. Der Begriff Nachbarschaft drückt aus, dass sich die jeweiligen Konverterketten nur in wenigen Merkmalen unterscheiden. Welche Konverterketten benachbart sind, hängt von der Transformationsvorschrift zur Erzeugung von Nachbarschaften ab. So bilden beispielsweise Konverterketten, die sich an einer oder an wenigen Stellen unterscheiden, eine Nachbarschaft. Konverterketten, die ein und dieselbe Konvertierung vornehmen, können unterschiedliche Längen haben. Daraus folgt, dass es nicht ausreichend ist, wenn die Transformation lediglich an einer oder an wenigen Stellen Konverter durch andere ersetzt. Bei diesem Vorgehen würde man nur Konverterketten fester Länge auf funktionale Korrektheit untersuchen. Das ist im Allgemeinen nicht ausreichend.

Bei Anwendung dieses Verfahrens ist es fraglich, ob eine funktional korrekte Konverterkette gefunden wird, selbst wenn eine existiert. Das ist dann der Fall, wenn ein Teil des Suchraumes inspiziert wird, der keine funktional korrekte Konverterkette enthält. Somit hängt der Erfolg der Lokalen Suche im Wesentlichen von der gewählten Ausgangskonverterkette und der Transformationsvorschrift zur Bildung neuer Konverterketten ab.

Die entsprechende Transformationsvorschrift müsste auf Regeln basieren, die auf eine funktional korrekte Konverterkette führen. Allein die Bedingung, dass die jeweiligen Ausgangs- und Eingangsformate zusammenpassen müssen, sagt noch nichts über die funktionale Korrektheit aus.

Für die Suche nach einer optimalen Konverterkette aus der Menge K_{kor} eignet sich dieses Verfahren nur bedingt. Auch hier benötigt man eine geeignete Transformation, die eine Ausgangskonverterkette aus K_{kor} auf eine benachbar-

te Konverterkette aus K_{kor} abbildet. Existieren keine Nachbarschaften oder sind diese sehr klein, so ist die Chance gering, die optimalste Lösung zu finden. Das ist immer dann der Fall, wenn die Konverterketten aus K_{kor} sehr unterschiedlich sind (unterschiedliche Länge etc.).

Im Falle großer Ähnlichkeiten im Sinne einer Transformation können Nachbarschaften sehr groß werden, so dass wieder ein zeitliches Problem entsteht. Das Verfahren birgt weiterhin den Nachteil, dass man sich bei Abschluss der Suche nicht sicher sein kann, die global beste Lösung gefunden zu haben. Man findet mitunter nur lokale Optima.

Es existieren Verfahren, die einen größeren Bereich des Suchraumes abdecken, z.B. Simulated Annealing und Tabusuche. Für die Suche nach funktional korrekten Konverterketten ergeben sich jedoch dieselben Probleme wie bei der Lokalen Suche. Diese Verfahren werden deshalb unter dem Aspekt der Optimierung von Konverterketten aus K_{kor} vorgestellt.

6.2.1 Simulated Annealing

Sind funktional korrekte Konverterketten bekannt, so ist der nächste Schritt diese unter dem Gesichtspunkt des Ressourcenverbrauches zu optimieren.

Bei Simulated Annealing²² wird zunächst eine Konverterkette aus K_{kor} gewählt. Der Unterschied zur Lokalen Suche liegt darin, dass nur eine Lösung aus der Nachbarschaft untersucht wird und diese mit der Wahrscheinlichkeit p akzeptiert wird. Es wird demnach nicht zwangsläufig die bessere Lösung für den nächsten Durchgang ausgewählt. Die Wahrscheinlichkeit mit der eine Konverterkette akzeptiert wird, ist:

$$p = \frac{1}{1 + e^{\frac{-\Delta_{eval}}{T}}} \quad (4)$$

Dabei sind:

- $-\Delta_{eval}$ ist die Differenz der evaluierten Werte von neuer und Ausgangskonverterkette.
- T ist die Temperatur, die die Relevanz der relativen Evaluierungswerte angibt. Je größer T , desto höher die Wahrscheinlichkeit, dass eine neue Lösung akzeptiert wird und umgekehrt.

Die Optimierung beginnt zuerst mit großen Werten für T , was einer zufälligen Auswahl gleicht. Der Parameter T wird dann sukzessive verkleinert, was bedeutet, dass besser evaluierte Lösungen mit hoher Wahrscheinlichkeit gewählt werden.

²²Der Begriff stammt aus der Thermodynamik und heißt soviel wie Simuliertes Ausglühen. Um Kristalle zu erzeugen, werden dort verschiedene Materialien stark erhitzt und anschließend langsam abgekühlt, bis sich die Kristallstrukturen ergeben.

Simulated Annealing greift anfangs beliebige Lösungen aus K_{kor} heraus. Gegen Ende wird lokal (innerhalb einer Nachbarschaft) nach der optimalsten Konverterkette gesucht. Durch diese Vorgehensweise deckt das Verfahren einen größeren Bereich des Suchraumes ab. Die Chancen, die optimalste Lösung zu finden, steigen.

6.2.2 Tabusuche

Die Tabusuche basiert im Gegensatz zu Simulated Annealing nicht auf Wahrscheinlichkeiten. Hier kommt ein Speicher zum Einsatz, der sicherstellt, dass die Suche möglichst viele Bereiche des Suchraumes abdeckt. Dadurch wird vermieden, dass immer wieder ähnliche Lösungen ausgewählt werden. Die Suche wird also möglichst vielseitig gestaltet.

Wieder werden potentielle Lösungen in der Nachbarschaft einer Ausgangslösung gebildet. Dabei hängt die Auswahl von benachbarten Lösungen von in einem Speicher protokollierten, vorhergehenden Schritten ab. Häufig durchgeführte Transformationen werden für eine gewisse Zeitspanne verboten, daher der Name Tabusuche. Es besteht in Erweiterung des Verfahrens die Möglichkeit, eine verbotene Lösung auszuwählen, falls diese wesentlich besser als alle nicht verbotenen Lösungen ist. Wann das der Fall ist, wird von einem festzulegenden *aspiration criterion* bestimmt.

Für die Optimierung von Konverterketten aus K_{kor} bedeutet das, dass möglichst viele verschiedene Konverterketten ausgewählt und bewertet werden. Dabei bildet immer die jeweils bessere Konverterkette den Ausgangspunkt für den nächsten Durchgang. Wieder ist festzulegen, welche Konverterketten benachbart sind. Es ist demnach die Frage zu klären, wie eine entsprechende Transformationsvorschrift gebildet werden kann.

6.3 Teile & Herrsche

Wie der Name andeutet, basiert das Verfahren Teile & Herrsche auf der Zerlegung des Gesamtproblems in einfachere Einzelprobleme. Diese können meist rekursiv weiter zerlegt werden. Es wird das Ziel verfolgt, durch die weitere Zerlegung die Teilprobleme einfacher lösen zu können. Die Lösung für das Gesamtproblem ergibt sich durch Integration aller Teillösungen.

Zur Identifikation der Teilprobleme bei der Suche nach funktional korrekten Konverterketten kann die Logische Klassifikation einen ersten Anhaltspunkt geben. So kann eine Konvertierung zunächst in eine Format- oder Inhaltskonvertierung zerfallen. Eine Formatkonvertierung kann weiter in eine Kanalkodierung und/oder Medientypkonvertierung aufgespalten werden. Inhaltsrelevant sind z.B. Filter. Die gewünschte Konvertierung zerfällt somit in Teilabschnitte, was einer Verkleinerung des Suchraumes gleichkommt. Dort werden zuerst die entsprechenden Teile der noch zu findenden Konverterkette konstruiert. Im nächsten Schritt

wird versucht, die Teilketten sinnvoll zu verbinden, bis schließlich eine funktional korrekte Konverterkette entsteht.

Weiterhin können die Teillösungen bezüglich des Ressourcenverbrauches bewertet werden und somit bei unerfüllbarem Ressourcenverbrauch schon frühzeitig ausgeschlossen werden. Unter der Voraussetzung, dass jede Konvertierung in geeignete Teilabschnitte zerlegt werden kann, ist der Einsatz von Teile & Herrsche zur Suche nach einer funktional korrekten Konverterkette durchaus interessant.

Es bietet sich die Möglichkeit, die Suche nach funktionaler Korrektheit mit der Optimierung zu kombinieren. Das erste Kriterium ist funktionale Korrektheit, das zweite Optimalität.

6.4 Dynamisches Programmieren

Beim Dynamischen Programmieren wird die Gesamtlösung stufenweise zusammengestellt. Das Gesamtproblem wird in eine Sequenz von Teilentscheidungen unterteilt. Auf jeder Stufe stehen verschiedene Alternativen mit definierten Kosten bereit. Das Verfahren ist rekursiv. Jede Zwischenstufe hängt von den vorher besuchten Stufen ab. Die Einzelentscheidungen auf jeder Stufe sind unabhängig voneinander.

Für die Suche nach funktional korrekten Konverterketten müssen hier ausgehend vom geforderten Ausgangsformat Regeln über Kosten konstruiert werden, wie vom geforderten Ausgangsformat zum Format im Medienserver gelangt werden kann. Das setzt zusätzliches Wissen voraus und nimmt die eigentliche Suche vorweg.

Besser eignet sich das Verfahren zur Optimierung von bekannten funktional korrekten Konverterketten. Gelingt es einen Graphen aufzubauen, in dem die funktional korrekten Konverterketten aus K_{kor} als Pfade vom Zielformat zum Format im Medienserver modelliert werden, so kann mit Hilfe des Dynamischen Programmierens die optimalste Konverterkette bezüglich des Ressourcenverbrauches gefunden werden. Die Kosten für einen Pfad stellen dabei den Ressourcenverbrauch dar. Dazu soll ein einfaches Beispiel für eine MPEG Kodierung betrachtet werden (Abbildung 10). Die Konvertierung zerfällt in Einzelabschnitte, z.B. Farbraumwandlung, Subsampling und MPEG Kodierung. Im Beispiel wurden 3 funktional korrekte Konverterketten gefunden. Zum Optimieren werden alle Pfade vom Ziel- bis hin zum Ausgangsformat zurückverfolgt und die jeweiligen Kosten protokolliert. Am Ende gilt die Konverterkette als optimal, die die geringsten Gesamtkosten hat.

Mit diesem Verfahren werden alle Konverterketten aus K_{kor} getestet. Somit wählt man immer die optimalste. Ist $|K_{kor}|$ jedoch sehr groß, so kann das Optimieren viel Zeit beanspruchen. Somit ist die jeweilige Suchraumgröße für den Einsatz des Dynamischen Programmierens ausschlaggebend.

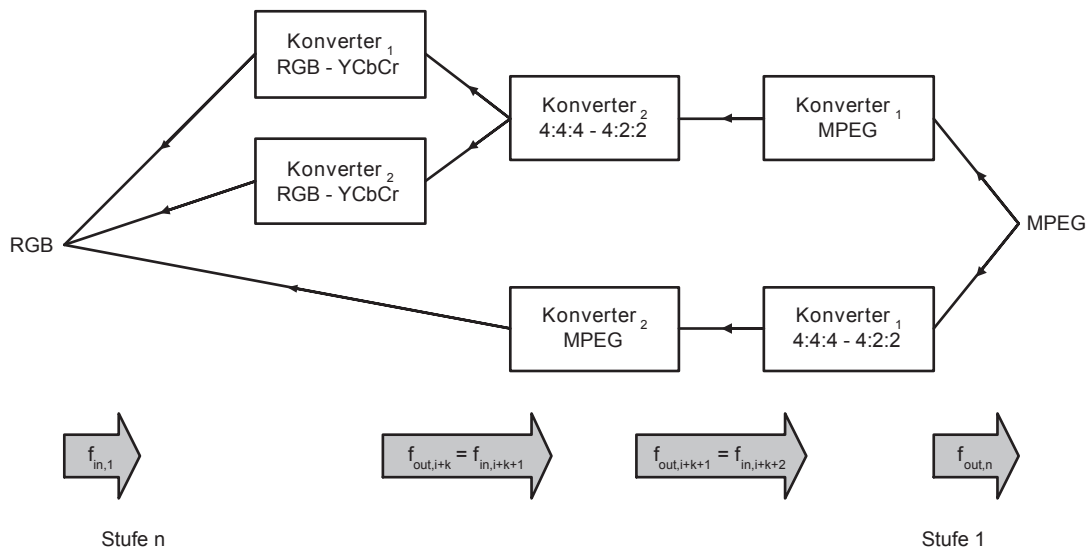


Abbildung 10: Suche einer optimalen Konverterkette durch Dynamischen Programmieren

6.5 Branch & Bound

Wie schon bei dem Verfahren der Umfassenden Suche in 6.1 erwähnt, ist die Größe des Suchraumes ein Problem. Vorteilhaft ist es, wenn eine Möglichkeit besteht, den Suchraum zu partitionieren. D.h. für die Suche nach Konverterketten bereits an dem Punkt zu stoppen, an dem erkennbar ist, dass keine funktional korrekte bzw. optimale Konverterkette konstruiert werden kann und anschließend zu einem Punkt zurückzukehren, an dem das noch möglich war.

Zum Verständnis des Verfahrens ist es hilfreich, sich den Suchraum als einen Baum vorzustellen (Abbildung 11). Eine Konverterkette ist ein Weg von der Wurzel (Speicherungsformat des Medienservers) über die Zwischenknoten²³ zum Zielformat, sofern ein solcher existiert. Auf dem Weg über die verschiedenen Zwischenknoten wird mittels einer geeigneten Evaluierungsfunktion entschieden, ob dieser weiterverfolgt oder von der weiteren Suche ausgeschlossen wird. Hier besteht die Möglichkeit der kombinierten Suche nach funktionaler Korrektheit und Optimierung. Dazu sind zwei Evaluierungsfunktionen nötig, eine zur Überprüfung der funktionalen Korrektheit und eine zur Optimierung. Wird ein Pfad verfolgt, der eine potentiell korrekte Konverterkette bildet, so kann gleichzeitig der Ressourcenbedarf protokolliert werden. Ist dieser unerfüllbar, so braucht diese Konverterkette nicht weiter betrachtet zu werden.

Problematisch ist die Konstruktion einer geeigneten Evaluierungsfunktion für

²³Die Zwischenknoten sind die Konverter.

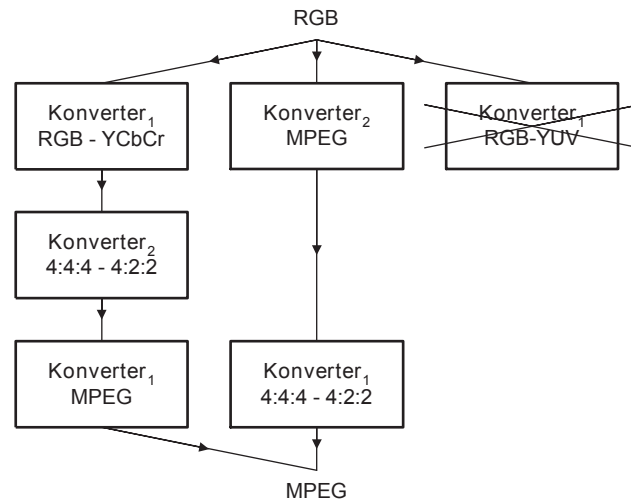


Abbildung 11: Die Abbildung zeigt ein Beispiel, in dem das Vorhandensein einer Evaluierungsfunktion für die funktionale Korrektheit vorausgesetzt wird. Somit ist es möglich, vorzeitig Pfade von der Inspektion auszuschließen.

die funktionale Korrektheit. Für Branch & Bound müsste eine Evaluierungsfunktion geschaffen werden, die in der Lage ist, sinnvolle von nicht sinnvollen Kombinationen von Konvertern frühzeitig zu unterscheiden. Ob dies möglich ist, müsste an anderer Stelle näher untersucht werden.

Zur Optimierung ist das Verfahren gut geeignet, da eine derartige Evaluierungsfunktion lediglich den Ressourcenverbrauch summiert. Beim Erreichen einer Schranke kann negativ evaluiert werden.

6.6 A* Algorithmus

Zum Verständnis des A* Algorithmus ist es hilfreich, sich den Suchraum ebenfalls als einen Baum vorzustellen. Dieser kann auf unterschiedlichste Weise durchsucht werden. Hier wird die Möglichkeit *Best-first* vorgestellt.

Dabei werden zuerst die viel versprechendsten Knoten im Sinne einer Evaluierungsfunktion ausgewählt. Dazu werden zwei Listen verwaltet, in der einen finden sich alle bereits besuchten Knoten (Geschlossene Liste) und in der anderen die noch nicht besuchten (Offene Liste). Wird ein Knoten besucht, so wird er in die Liste der geschlossenen Knoten und alle seine Kinder in die Liste der offenen Knoten eingeordnet. Diese werden evaluiert und die beste Alternative wird für die Weiterverarbeitung ausgewählt usw.

Für die Suche nach funktional korrekten Konverterketten muss wieder eine geeignete Evaluierungsfunktion konstruiert werden. Auch hier ist die Frage zu klären, ob und wann anhand einer Teilkonverterkette entschieden werden kann,

inwieweit die Möglichkeit besteht, diese funktional korrekt zu ergänzen.

Bei dem A* Algorithmus hat die Evaluierungsfunktion die Form:

$$\text{eval}_{kor}(kon_i) = c(kon_i) + h(kon_i)$$

Wobei $c(kon_i)$ die Bewertung für die bereits besuchten Konverter auf dem Pfad und $h(kon_i)$ eine Bewertung für die nachfolgenden Knoten auf dem gleichen Pfad angibt. Dabei ist $h(kon_i)$ eine Vorhersage des Potentials der Folgeentscheidungen. Im Kontext der Suche nach funktional korrekten Konverterketten würde eine solche Funktion eine Bewertung über die funktionale Korrektheit der Folgeschritte abgeben. Entsprechend $c(kon_i)$ für die funktionale Korrektheit, die bisher erreicht wurde.

Existiert eine derartige Evaluierungsfunktion eval_{kor} , so kann mit einer zweiten eval_{res} gleichzeitig der Ressourcenverbrauch auf gleiche Weise evaluiert werden. Diese müsste jeweils den Verbrauch an Ressourcen, der bisher benötigt wird, aufsummieren und eine Abschätzung für die Folgeschritte treffen. Somit ist auch hier die kombinierte Suche nach funktionaler Korrektheit und Optimalität einer Konverterkette möglich.

6.7 Evolutionistische Verfahren

Die Evolutionistische Vorgehensweise beginnt die Suche mit einer Menge von potentiellen Lösungen. Auch hier spielt eine geeignete Evaluierungsfunktion, die den Wert einer Lösung gegenüber anderen bewertet, eine Rolle. Basierend auf dieser Auswahl werden Lösungen gewählt, aus denen mittels einer geeigneten Transformation neue Lösungen geschaffen werden. Diese werden wieder evaluiert und pflanzen sich weiter fort. Daher auch der Name Evolutionistisches Verfahren, da eine solche Vorgehensweise den Vorgängen in der Natur ähnlich ist. Auch hier pflanzen sich Individuen, die im Überlebenskampf²⁴ mit anderen stehen, fort. Mit der Zeit ist durch Mutation eine immer bessere Anpassung an die Umwelt erkennbar.

Evolutionistische Verfahren lassen sich aufgrund ihrer Vielseitigkeit auf eine Vielzahl von Problemstellungen anwenden. Die Schwierigkeit Evolutionistische Verfahren bei der Suche nach funktional korrekten Konverterketten anzuwenden, liegt wieder darin, eine geeignete Evaluierungsfunktion zu finden. Diese muss quasi in der Lage sein, sinnvolle Kombinationen von Convertern von nicht sinnvollen zu unterscheiden. Die Einschränkung, das $f_{out,i-1} = f_{in,i}$ gelten muss, ist für eine solche Entscheidung nicht ausreichend. An dieser Stelle sind weitere Zusatzbedingungen zu definieren, die schon sehr früh erkennen lassen müssen, ob funktionale Korrektheit noch erreicht werden kann.

²⁴Nur die am besten angepassten Individuen überleben und können sich fortpflanzen. Diese übertragen ihre positiven Eigenschaften auf ihre Nachkommen. Daher ist für Evolutionistische Verfahren auch der Begriff Genetische Verfahren gebräuchlich.

Eine weitere Schwierigkeit liegt darin, auf die gewählten Ausgangslösungen geeignete Transformationen anzuwenden, um eine nächste viel versprechendere Generation von Lösungen zu erzeugen. Hier müssen Regeln zur Anwendung kommen, die am Schluss, die Bildung einer funktional korrekten Konverterkette, wenn vorhanden, aus der Menge von Vorgängerlösungen sicherstellen.

Auch ist die Frage zu beantworten, wie die potentiellen Ausgangslösungen, mit denen das Verfahren startet, aussehen sollen. Werden beliebige Kombinationen von Konvertern generiert oder startet die Suche bei bereits bekannten, die Aufgabenstellung erfüllenden Konverterketten. Letzteres würde die eigentliche Suche vorwegnehmen und schließlich auf eine Optimierung hinauslaufen.

Auch bei der Optimierung stellt sich die Frage, wie eine Transformation vorgenommen werden muss, um aus zwei oder mehreren funktional korrekten Konverterketten bessere zu generieren. Diese müssten natürlich auch das Kriterium der funktionalen Korrektheit erfüllen.

Die Bewertung der Eignung von Evolutionistischen Verfahren für die Suche nach funktional korrekten sowie optimalen Konverterketten ist aus diesem Gründen hier nicht endgültig möglich.

7 Zusammenfassung & Ausblick

In der Arbeit wurden zuerst ausgewählte Verfahren zur Audio- und Videokodierung und -bearbeitung vorgestellt. Konverter, die ein Medienobjekt in ein anderes überführen, nutzen diese und andere Verfahren. Dabei hat sich gezeigt, dass gerade im Audio- und Videobereich eine Reihe von Optimierungsmöglichkeiten bestehen. Diese zielen im Allgemeinen darauf ab, den Datenumfang durch verlustbehaftete Komprimierungstechniken zu reduzieren. Dabei finden Modelle wie die Psychoakustik Anwendung. Diese basieren auf z.T. empirischen Erkenntnissen, die den Informationsverlust vor dem Menschen verbergen.

Ausgehend von den beispielhaft vorgestellten Verfahren wurden Klassifizierungsmerkmale für Konverter entwickelt. Unter Beachtung des Konvertermodells nach [Mär00] entstand ein Schema für die Logische und Physische Klassifikation.

Die Logische Klassifikation beschäftigt sich mit den Auswirkungen einer Konvertierung auf ein Medienobjekt. Hier spielt der Informationsverlust eine entscheidende Rolle. Schränkt er doch die Weiterverarbeitungsmöglichkeiten ein. Ausgehend vom Schema der Logischen Klassifikation wurde insbesondere die Relevanz von Medientyp- und Darstellungskonvertern für die Realisierung von Medienservern festgestellt.

Der Physischen Klassifikation liegt der Ressourcenverbrauch eines Konverters zu Grunde. Hier wurden Erkenntnisse abgeleitet, die die Einplanbarkeit eines Konverters betreffen. Dabei spielt Determinismus für die Vorhersagbarkeit eine herausragende Rolle. Es ist die Frage zu klären, inwieweit der Ressourcenverbrauch anhand von Eigenschaften des Medienobjektes bestimmt werden kann oder, ob andere Einflüsse (medienunabhängige Einflüsse) eine Rolle spielen. Ein weiteres Kriterium der Physischen Klassifikation war die Konstanz / Nichtkonstanz bzw. Beschränktheit / Nichtbeschränktheit des Ressourcenverbrauches. Kann keine obere Grenze für den Ressourcenverbrauch eines Konverters ermittelt werden, so ist die Durchführung dieser Konvertierung unter Echtzeitbedingungen nicht möglich. In Abbildung 9 deutet der Pfeil die Zunahme der Komplexität der Ressourcenplanung an.

Die hier vorgestellten Schemata der Logischen und Physischen Klassifikation bilden eine Möglichkeit der Klassifizierung von Konvertern. Sie erheben nicht den Anspruch eine vollständige Klassifikation von Medienkonvertern zu sein. Im Umfeld des Projektes *RETAVIC* (Universität Erlangen-Nürnberg) wird an einer Konverterklassifikation gearbeitet, die sich an der eigentlichen Funktion eines Konverters orientiert.

Abschließend wurden Suchverfahren vorgestellt und ihre Relevanz zur Suche nach Konverterketten bewertet. Dabei hat sich herausgestellt, dass die Suche in zwei Abschnitte gegliedert werden muss. Zuerst müssen funktional korrekte Konverterketten gefunden werden. Anschließend kann eine Optimierung, die sich am Ressourcenverbrauch orientiert, stattfinden. Bei einigen Verfahren (z.B. Teile & Herrsche, Branch & Bound) besteht die Möglichkeit, die Suche nach funktionaler

Korrektheit mit der Optimierung zu kombinieren. Somit kann der Ressourcenverbrauch schon bei der Suche als Evaluierungskriterium eingebracht werden.

Es wird deutlich, dass Kriterien zu formulieren sind, die funktionale Korrektheit gewährleisten. Diese Kriterien müssen in die Suchverfahren einbezogen werden. Bei Verfahren, die mit Teillösungen arbeiten, z.B. Teile & Herrsche, Branch & Bound, muss zur effizienten Suche frühzeitig erkannt werden, ob eine noch unvollständige Konverterkette funktional korrekt erweitert werden kann.

Inwieweit die vorgestellten Verfahren zur Suche nach funktional korrekten Konverterketten geeignet sind, hängt wesentlich davon ab, ob diese Kriterien algorithmisch zu formulieren sind. Aus diesen Gründen kann in dieser Arbeit nicht endgültig beurteilt werden, ob ein Verfahren besser geeignet ist als ein anderes. Bei den vorgestellten Verfahren wurde deshalb immer auf die Probleme hingewiesen, die sich bei der Anwendung ergeben.

In dieser Arbeit wird an mehreren Stellen auf die Formatbeschreibung f Bezug genommen. So basieren die Logische und die Physische Klassifikation auf der Formatbeschreibung. Zum einen wird in dieser Arbeit davon ausgegangen, dass eine Konvertierung das Format beeinflusst und zum anderen wird davon ausgegangen, dass die Formatbeschreibung Informationen, die relevant für den Ressourcenverbrauch sind, enthält. Auch bei den Suchverfahren spielen die jeweiligen Ein- und Ausgangsformate (f_{in}, f_{out}) als ein Kriterium für funktionale Korrektheit eine Rolle. Da der Inhalt der Formatbeschreibung f noch nicht eindeutig definiert ist, müsste in einem nächsten Schritt eine Formatdefinition für die verschiedenen Mediendatentypen erarbeitet werden. Dabei ist zu prüfen, welche Informationen mindestens enthalten sein müssen und welche aus anderen abgeleitet werden können.

Literatur

- [Bra99] BRANDENBURG, KARLHEINZ: *MP3 AND ACC EXPLAINED*. In: *AES 17th International Conference on High Quality Audio Coding*, September 1999.
- [DKS96] DOMSCHKE, WOLFGANG, ROBERT KLEIN und ARMIN SCHOLL: *Taktische Tabus – Tabu Search: Durch Verbote schneller optimieren*. c't magazin für computer technik, 96(12):326–332, 1996.
- [HMMW01] HAMANN, CLAUDE-JOACHIM, ANDREAS MÄRCZ und KLAUS MEYER-WEGENER: *Buffer Optimization in Realtime Media Servers Using Jitter-constrained Periodic Streams*. Technischer Bericht, Technische Universität Dresden, January 2001.
- [HRW+99] HÄRTIG, HERMANN, LARS REUTHER, JEAN WOLTER, MARTIN BORRIS und TORSTEN PAUL: *Cooperating Resource Managers*. In: *Proceedings of the Fifth IEEE Real-Time Technology and Applications Symposium (RTAS)*, June 1999.
- [Jac96] JACK, KEITH: *Video Demystified: A Handbook for the Digital Engineer*. HighText Interactive, Inc., San Diego, CA 92121, 1996.
- [KZ95] KLETTE, REINHARD und PIERO ZAMPERONI: *Handbuch der Operatoren für die Bildbearbeitung*. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, 1995.
- [Lut97] LUTHER, ARCH C.: *Principles of Digital Audio and Video*. Artech House, Inc., 685 Canton Street Norwood, MA 02062, 1997.
- [Mar00] MARDER, ULRICH: *Transformation Independence in Multimedia Database Systems*. Technischer Bericht, Universität Kaiserslautern, Fachbereich Informatik, November 2000. Forschungsbericht, Sonderforschungsbereich 501.
- [MF00] MICHALEWICZ, ZBIGNIEW und DAVID B. FOGEL: *How to Solve It: Modern Heuristics*. Springer-Verlag, 2000.
- [Mär99] MÄRCZ, ANDREAS: *Leistungs- und Engpassanalyse eines H.263-Codierers*. Großer Beleg, Technische Universität Dresden, Fakultät Informatik, 1999.
- [Mär00] MÄRCZ, ANDREAS: *Entwurf eines Modells zur Konverterbeschreibung*. Diplomarbeit, Technische Universität Dresden, Fakultät Informatik, 2000.

- [MW] MEYER-WEGENER, KLAUS: *Multimedia-Datenbanken*. Vorlesungsmanuskript, Technische Universität Dresden, Fakultät Informatik, Institut für Systemarchitektur, Professur Datenbanken.
- [MWM02] MEYER-WEGENER, KLAUS und ANDREAS MÄRCZ: *Bandwidth-based converter description for realtime scheduling at application level in media servers*. In: *Proceedings of 4th SDA Workshop*, April 2002.
- [Poy96] POYNTON, CHARLES: *A Technical Introduction to Digital Video*. John Wiley and Sons, 1996.
- [Poy98] POYNTON, CHARLES: *Merging computing with studio video: Converting between R'G'B' and 4:2:2*, 1998. www.inforamp.net/~poynton.
- [Poy01] POYNTON, CHARLES: *YUV and luminance considered harmful: A plea for precise terminology in video*, 2001. www.inforamp.net/~poynton.
- [SKKD97] SCHOLL, ARMIN, GABRIELA KRISPIN, ROBERT KLEIN und WOLFGANG DOMSCHKE: *Besser beschränkt – Clever optimieren mit Branch and Bound*. c't magazin für computer technik, 97(10):336–345, 1997.

Ich danke...

Prof. Dr. Klaus Meyer-Wegener und Prof. Dr. Hermann Härtig für die freundliche Unterstützung.

Meinem Betreuer Andreas März für viele wertvolle Hinweise, Anregungen und aufschlussreiche Diskussionen.

Meinen Eltern.