

# 1 Kompressionsverfahren für Video und Audio

Jan Schulz

Kontinuierliche Medien wie Audio- und Videosequenzen stellen in Bezug auf erforderliche Datenrate und benötigten Speicherplatz hohe Anforderungen an die verarbeitenden Systeme. Beispielsweise besitzt eine unkomprimierte Bildsequenz mit der Auflösung von  $640 \times 480$  Pixeln bei einer Bildrate von 25 Bildern pro Sekunde und einer Farbtiefe von 8 Bit pro RGB-Kanal eine Datenrate von:

$$640 \cdot 480 \cdot 25 \text{ Hz} \cdot 8 \text{ Bit/Kanal} \cdot 3 \text{ Kanäle} = 184.320.000 \text{ Bit/s} \approx 184 \text{ MBit/s.}$$

10 Minuten einer solchen Bildsequenz benötigen Speicherplatz im zweistelligen Gigabyte-Bereich. Um also Audio- und Videosequenzen übertragen, verarbeiten und speichern zu können, ist der Einsatz effizienter Kompressionsverfahren fast immer unverzichtbar.

Dies gilt für den klassischen Fernsehbereich ebenso wie z. B. für die Wachstumsbranche der Mobilien Multimediadienste (vgl. Kap. 4). Unter der Sammelbezeichnung High Definition Television (HDTV) starten derzeit immer mehr Fernsehprogramme in höherer Auflösung, während Auflösung in der Mobilfunkbranche aufgrund der Größe der Endgeräte nur ein untergeordnetes Kriterium darstellt. Hier steht – mehr noch als bei HDTV – die Datenrate im Blickpunkt, die bei der ersten UMTS-Version z. B. bis zu 384 kBit/s beträgt. HDTV-Programme benötigen dagegen mit etwa 15 MBit/s (DVB-S) weitaus höhere Datenraten. Diese ausgeprägten Unterschiede beginnen mit der Annäherung von Kommunikations- und Unterhaltungsbranche auf dem Konsumentenmarkt zu schwinden, was sich auch bei neuen Verfahren der Datenkompression widerspiegelt. Während frühere Kompressionsstandards wie z. B. MPEG-1 stark auf einen bestimmten Einsatzzweck zugeschnitten wurden, sind neuere Standards wie bspw. H.264/AVC viel flexibler einsetzbar und werden in beiden oben genannten Branchen verwendet.

Trotz aller Annäherung bei der Auslieferung von Medieninhalten hat die Film- und Fernsehbranche jedoch nach wie vor Bedarf an

*Motivation für Datenkompression*

*Annäherung von Fernseh- und Kommunikationsbranche auf dem Konsumentenmarkt*

*Bleibender Bedarf an hoher Qualität*

Verfahren, mit denen Produktionen möglichst ohne Informationsverluste komprimiert werden können. Dies ist z. B. an den Fidelity Range Extensions ersichtlich, welche H.264/AVC u. A. um eine Möglichkeit ergänzen, die volle Farbinformation des Videomaterials zu erhalten.

*Inhalte des Kapitels*

Im Folgenden werden zunächst einige Konzepte der Informationstheorie und Begriffe aus dem Bereich der Datenkompression erläutert, bevor in Kap. 1.3 grundlegende Verarbeitungsschritte dargestellt werden, die sich in vielen Kompressionsverfahren wieder finden. Mit Interframe- und psychoakustischer Kompression werden in Kap. 1.4 zwei Verfahren für die Video- bzw. Audiokompression beleuchtet, welche medienspezifische Besonderheiten ausnutzen und in Kombination mit anderen Verarbeitungsschritten die Kompressionsrate stark erhöhen können.

Kapitel 1.5 gibt einen Überblick über zahlreiche Kompressionsverfahren für Audio- und Videodaten, wobei insbesondere Standards der ITU und der MPEG-Gruppe betrachtet werden.

Nachdem in Kap. 1.6 die JPEG-Kompression als Ausgangspunkt zahlreicher Verfahren zur Videokompression erläutert wurde, behandelt Kap. 1.7 ausführlich die Videokompression nach MPEG. Anhand von MPEG-1 wird hierbei die grundlegende Funktionsweise der MPEG-Standards dargestellt, so dass in den weiteren Unterkapiteln zu MPEG-2 und MPEG-4 darauf aufgebaut werden kann.

Kapitel 1.8 widmet sich der Audiokompression und stellt mit MPEG-1 und MPEG-4 ALS Verfahren vor, denen zwei unterschiedliche Funktionsweisen zu Grunde liegen. Während MPEG-1 Audio ein psychoakustisches Modell verwendet, basiert MPEG-4 ALS im Wesentlichen auf Prädiktion. Als Spezialfall der Audiokompression wird in Kap. 1.9 abschließend die Sprachkompression erläutert. Weitere Informationen hierzu finden sich auch in Kap. 3.

## 1.1 Einführung in die Informationstheorie

*Informationstheorie*

Die Informationstheorie wurde durch Shannon begründet, der die Übertragung von Zeichen zwischen Nachrichtenquelle und Nachrichtensenke betrachtete und daraus mathematische Methoden entwickelte, um Information quantitativ zu bestimmen [Shn48]. Erkenntnisse der Informationstheorie werden u. A. bei Entropiecodierungen (Kap. 1.3.2) eingesetzt, die Nachrichten möglichst kompakt darstellen und so Datenkompression erzielen.

*gedächtnislose Nachrichtenquelle*

Im Folgenden werden diskrete gedächtnislose Nachrichtenquellen mit endlichem Alphabet betrachtet, die in der Informationstheorie einen Spezialfall darstellen. Eine solche Quelle versendet zu diskreten

Zeitpunkten  $t = 0, 1, \dots$  die Symbole eines endlichen Alphabets  $A$ . Die Wahrscheinlichkeit für das Auftreten eines Zeichens  $z \in A$  zum Zeitpunkt  $t$  hängt dabei nicht vom Zeitpunkt  $t$  und nicht von den Zeichen ab, die zu den vergangenen Zeitpunkten  $0, \dots, t-1$  gesendet wurden. Gedächtnislose Quellen sind in der Realität selten zu finden, da hier häufig Abhängigkeiten zwischen einzelnen Quellsymbolen bestehen. Sie genügen jedoch, um grundlegende Konzepte und Begriffe der Informationstheorie zu erläutern, wie sie im Weiteren benötigt werden. Für eine Verallgemeinerung der Konzepte sei z. B. auf [Gra98] verwiesen.

Der Informationsgehalt  $I(z)$  eines Zeichens  $z \in A$ , das von einer diskreten gedächtnislosen Nachrichtenquelle mit endlichem Alphabet erzeugt wird, berechnet sich aus der Wahrscheinlichkeit, mit der die Nachrichtenquelle das Zeichen  $z$  produziert:

*Informationsgehalt*

$$I(z) = \log_2 \frac{1}{W(z)} [\text{bit}]$$

Mit abnehmender Auftrittswahrscheinlichkeit eines Zeichens steigt folglich sein Informationsgehalt. Tritt mit  $W(z) = 1$  stets dasselbe Zeichen auf, so ist sein Informationsgehalt gleich 0.

Betrachtet man eine Symbolfolge  $z_0, z_1, \dots, z_n$ , so lässt sich der mittlere Informationsgehalt dieser Folge berechnen, der auch als Entropie erster Ordnung bezeichnet wird. Unter der Voraussetzung, dass die Elemente der Zeichenfolge statistisch nicht voneinander abhängen, ergibt sich diese Entropie  $H$  aus der Summe der gewichteten Informationsgehalte der Einzelelemente:

*Entropie*

$$H = \sum_{i=0}^n W(z_i) \cdot I(z_i) [\text{bit/Symbol}]$$

Den höchsten Wert hat die Entropie demnach bei einer Gleichverteilung der Symbole. Je stärker sich die Auftrittswahrscheinlichkeiten um einen bestimmten Wert konzentrieren, desto geringer ist die Entropie.

Ist die gesamte Symbolfolge  $z_0, z_1, \dots, z_n$  schon im Voraus bekannt, so entspricht die Auftrittswahrscheinlichkeit eines Zeichens der relativen Häufigkeit des Zeichens in der Symbolfolge. Bei der Audio- und Videokompression ist dies jedoch selten der Fall, und die Wahrscheinlichkeiten müssen geschätzt werden.

Wenn man die Elemente einer Zeichenfolge auf binäre Codewörter abbildet, wird die mittlere Codelänge interessant. Ist  $l_i$  die Länge des Codeworts, mit dem das Zeichen  $z_i$  codiert wird, errechnet sich die mittlere Codelänge  $\bar{l}_i$  aus der Summe gewichteter Codelängeln:

*mittlere Codelänge*

$$\bar{l}_i = \sum_{i=0}^n W(z_i) \cdot l_i [\text{Bit/Symbol}]$$

*Codierungsredundanz*

Die Differenz zwischen Entropie und mittlerer Codelänge wird Codierungsredundanz genannt. Shannon zeigte, dass immer ein Code existiert, dessen mittlere Codelänge kleiner als  $H+1$  ist. Hohe Redundanz ist bei der Datenkompression kontraproduktiv. Kompressionsverfahren zielen daher unter anderem auf die Minimierung der Codierungsredundanz ab.

Bei Nachrichten bzw. Symbolfolgen, die im Rahmen der Audiokompression betrachtet werden, handelt es sich um diskrete Samples eines digitalen Tonsignals. Bei Bilddaten sind es die Helligkeits- oder Farbwerte von Pixeln, die bei sequenzieller Betrachtung der Zeilen ebenfalls diskrete Symbolfolgen darstellen. Daher werden die Begriffe Zeichenfolge, Symbolfolge, Signal und Nachricht im Folgenden synonym verwendet.

## 1.2 Grundlagen der Datenkompression

*Quellencodierung*

Datenkompression ist eine Form der Codierung, welche die Symbolfolgen einer Nachrichtenquelle möglichst platzsparend abspeichert. Codierung zum Zwecke der Datenkompression wird häufig auch als Quellencodierung bezeichnet, um sie von Kanalcodierungen bei der Datenübertragung und kryptographischen Codierungen im Bereich der Datensicherheit abzugrenzen. Im Folgenden wird diese Unterscheidung nicht mehr getroffen, und der Begriff Codierung ist immer als Quellencodierung zu verstehen.

*verlustfreie und verlustbehaftete  
Kompression*

Mit verlustfreien und verlustbehafteten Verfahren lassen sich zwei grundlegende Arten der Datenkompression bzw. Quellencodierung unterscheiden. Die verlustfreie Kompression verwirft ausschließlich redundante Informationen und ist daher voll reversibel. Durch Redundanzreduktion verlustfrei komprimierte Informationen können fehlerfrei wiederhergestellt werden. Die verlustbehaftete Kompression nimmt dagegen den Verlust von irrelevanten Informationen in Kauf. Während sich Redundanz eindeutig definieren lässt, ist Irrelevanz stark subjektiv geprägt. Irrelevant sind jene Informationen, die den Empfänger nicht interessieren oder nicht von ihm wahrgenommen werden können. Die Irrelevanzreduktion ist irreversibel. Einmal verworfene Informationen lassen sich nicht wieder zurückgewinnen.

*symmetrische und asymmetrische  
Kompression*

Ein weiteres Kriterium für die Klassifizierung von Kompressionsverfahren erhält man, indem der zeitliche bzw. rechnerische Aufwand für die Kompression  $A_K$  mit dem Aufwand für die Dekompression  $A_D$  in Beziehung gesetzt wird. Ist der Aufwand identisch, spricht man von symmetrischer Kompression. Dagegen handelt es sich um asymmetrische Kompression, wenn  $A_K > A_D$  ist. Bei der Kompression von Audio-

und Videodaten ist häufig ein geringer Aufwand bei der Decodierung erwünscht, da diese für eine flüssige Wiedergabe des Mediums in Echtzeit erfolgen muss. Verfahren für die Audio- und Videokompression sind daher zumeist asymmetrischer Natur.

Ein Algorithmus oder Schaltkreis, der eine Codierung oder Datenkompression vornimmt, wird als Coder oder Encoder bezeichnet. Umgekehrt macht ein Decoder die Kompression wieder rückgängig. Spricht man von einem Kompressionsverfahren als Ganzes, so verwendet man den zusammengesetzten Begriff Codec.

*Coder/Encoder, Decoder, Codec*

Als Maß für die quantitative Bewertung eines Kompressionsverfahrens kann die Kompressionsrate herangezogen werden, bei der die ursprüngliche Datenmenge zur komprimierten Datenmenge ins Verhältnis gesetzt wird. Durch Irrelevanzreduktion lassen sich weitaus höhere Kompressionsraten erzielen als mit Verfahren, die ausschließlich auf Redundanzreduktion beruhen.

*Kompressionsrate*

Die Kompressionsrate als rein objektives Kriterium ist bei der verlustbehafteten Irrelevanzreduktion nur bedingt aussagekräftig. Daher unterscheidet man bei der Bewertung von verlustbehafteten Kompressionsverfahren häufig zwischen objektiver und subjektiver Qualität. Als einfaches Kriterium für die Beurteilung der subjektiven Qualität hat sich der Mean Opinion Score (MOS) etabliert, der aus dem Bereich der telefonischen Sprachübertragung stammt. Für dieses Einsatzgebiet existieren Empfehlungen der International Telecommunication Union (ITU), einem internationalen Zusammenschluss zur Koordination von Entwicklungen im Telekommunikations- und Fernsehbereich. Der MOS wird durch Versuchspersonen ermittelt, die anhand einer fünfstufigen Skala qualitative Bewertungen zu unterschiedlich komprimierten Signalen abgeben. Im einfachsten Fall wird anschließend der arithmetische Mittelwert dieser Bewertungen gebildet [ITU96d] [ITU03b].

*objektive und subjektive Qualität,  
Mean Opinion Score (MOS)*

Die Ermittlung der subjektiv empfundenen Qualität durch Versuchspersonen ist ein aufwändiges Verfahren. Daher wurde im Videobereich die Video Quality Experts Group (VQEG) ins Leben gerufen, die sich mit der Standardisierung objektiver Messmethoden zur Bewertung subjektiver Qualität bei Videokompressionsverfahren befasst.

*Video Quality Experts Group (VQEG)*

## 1.3 Elementare Verarbeitungsschritte

Bei der Audio- und Videokompression verwendet man Verfahren, die sequentiell verschiedene grundlegende Verarbeitungsschritte durchführen, um besonders hohe Kompressionsraten zu erzielen. Meist entstehen dadurch hybride Methoden, die Irrelevanz- und Redundanzreduktion kombinieren.

*hybride Kompressionsverfahren*

### 1.3.1 Unterabtastung und Quantisierung

*Unterabtastung* Durch Unterabtastung lässt sich die Datenrate eines Signals auf einfache Weise reduzieren. Bei digitalen Signalen wird dabei eine bestimmte Anzahl der diskreten Signalwerte (engl.: Samples) verworfen, und nur jeder  $n$ -te Wert mit  $n = 2, 3, \dots$  wird beibehalten. Dies resultiert in einer Reduktion der Datenrate um den Faktor  $1/n$ . Die verworfenen Samples lassen sich nicht fehlerfrei wiederherstellen, weshalb es sich hierbei um einen verlustbehafteten Vorgang handelt.

*Abtasttheorem, Aliasfrequenz* Durch die Unterabtastung können bei der Rekonstruktion des Signals Aliasfrequenzen entstehen, die sich in Videosignalen beispielsweise durch störende Muster äußern. Diese Aliasfrequenzen treten nach dem fundamentalen nachrichtentheoretischen Abtasttheorem [Kot33][Shn48] immer dann auf, wenn die Frequenz der diskreten Signalwerte  $f_a$  durch die Unterabtastung kleiner wird als die doppelte Bandbreite des Signals. Um die Entstehung von Aliasfrequenzen zu verhindern, kann das Signal vor der Unterabtastung einer Bandpassfilterung mit der Grenzfrequenz  $f_g = f_a/2n$  [Hz] unterzogen werden.

*Chrominanz und Luminanz, YCrCb-Farbraum* Bei digitalem Video findet man häufig eine Unterabtastung der Farbinformation (Chrominanz), da diese vom menschlichen Auge schlechter aufgelöst werden kann als Helligkeitsinformation (Luminanz). Um Luminanz und Chrominanz zu trennen, werden die Videobilder in den YCrCb-Farbraum konvertiert. Dieser ist in der ITU-Empfehlung BT.601 [ITU95a] für digitale Fernsehstudioteknik definiert und wird aus den RGB-Kanälen durch folgende Produktmatrix gewonnen:

$$\begin{pmatrix} Y \\ Cr \\ Cb \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ 0,500 & -0,419 & -0,081 \\ -0,169 & -0,331 & 0,500 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

*4:2:2-Subsampling* In dieser weitgehend historisch bedingten Formel wird die Luminanz von der Y-Komponente, und die Chrominanz von den Komponenten Cr und Cb repräsentiert. Bei einer Unterabtastung der Chrominanz nach BT.601 werden für die Komponenten Cr und Cb jeweils nur halb so viele Pixelwerte aufgewendet wie für die Luminanz, was als 4:2:2-Subsampling bekannt ist. Noch geringer ist die Abtastung der CrCb-Komponenten bei der Kompression nach dem MPEG-1-Standard (Kap. 1.7.1), wo 4:2:0-Subsampling zum Einsatz kommt (Abb. 1.1).

4:2:2-Subsampling nach ITU-R BT.601



◇ Luminanz-Wert

◆ Luminanz-Wert und Chrominanz-Wert

4:2:0-Subsampling nach MPEG-1



× Chrominanz-Wert

◆ Luminanz-Wert und Chrominanz-Wert

Abb. 1.1:

4:2:2- und 4:2:0-Subsampling

Quantisierung ist eine weitere, sehr einfache Methode der Datenreduktion. Anstatt einzelne Samples zu verwerfen, wird hierbei die Genauigkeit reduziert, mit der sie abgespeichert werden. Man unterscheidet in skalare Quantisierung und Vektorquantisierung.

*Quantisierung*

Die skalare Quantisierung ordnet jedem Signalwert einen quantisierten Wert aus einer endlichen Wertemenge zu. Die Zuordnung erfolgt dabei im einfachsten Fall linear auf Basis eines Rasters mit Intervallen fester Länge. Alle Samples innerhalb eines bestimmten Intervalls werden dabei auf denselben quantisierten Wert abgebildet, wodurch verlustbehaftete Datenkompression entsteht. Anstelle eines festen Rasters können auch unterschiedliche Intervallbreiten gewählt werden, um bestimmte Werte stärker zu quantisieren als andere. Dies kann beispielsweise Sinn machen, wenn sich dadurch Einschränkungen der menschlichen Wahrnehmung ausnutzen lassen und wird als nichtlineare Quantisierung bezeichnet. Die Intervallbreiten bei nichtlinearer Quantisierung werden in Form von sog. Quantisierungskennlinien festgelegt.

*skalare Quantisierung, nichtlineare Quantisierung*

Die Vektorquantisierung berücksichtigt  $n$  Signalwerte gleichzeitig, die als Vektor des  $n$ -dimensionalen Raums aufgefasst werden. Wie die skalare Quantisierung stellt auch die Vektorquantisierung einen verlustbehafteten Vorgang dar. Ein Vektorquantisierer der Dimension  $n$  und Größe  $s$  bildet Eingavektoren auf eine endliche Menge  $C$  ab, die aus  $s$  Ausgabevektoren besteht. Für die Wahl der Ausgabevektoren aus  $C$  können verschiedene Kriterien herangezogen werden. Im einfachsten Fall kommt das euklidische Abstandsmaß der Vektoren zum Einsatz. Die Menge  $C$  der Ausgabevektoren wird als Codebuch bezeichnet. Die größte Herausforderung bei der Vektorquantisierung ist die Wahl eines geeigneten Codebuchs. Dieses muss in einer Trainingsphase mit Hilfe

*Vektorquantisierung, Codebuch*

charakteristischer Signalvektoren optimiert und so an typische Signalstatistiken angepasst werden. Ein verbreiteter Algorithmus zur Codebuch-Erstellung ist der LBG-Algorithmus [LBG80].

Vektorquantisierung in der Videokompression teilt Einzelbilder in quadratische Blöcke auf, wobei gleichartige Blöcke durch einen generischen Block aus einem Codebuch ersetzt werden. Der Einsatz von Vektorquantisierung in Codecs wie Sorenson oder Cinepak trug in den neunziger Jahren zur Verbreitung von digitalem Video auf Heimcomputern bei [WNW94]. Ein Audio-Codec auf Basis von Vektorquantisierung ist z. B. SoundVQ, der unter der Bezeichnung TwinVQ entwickelt [IMM95] und inzwischen als Teil von MPEG-4 Audio standardisiert wurde (Kap. 1.8).

### 1.3.2 Entropiecodierung

Entropiecodierungen nutzen die statistische Verteilung von Symbolen innerhalb einer Zeichenkette aus. So lässt sich die mittlere Codelänge der Entropie annähern und damit die Codierungsredundanz mindern. Da Entropiecodierungen ausschließlich redundante Informationen verwerfen, arbeiten sie verlustfrei.

#### 1.3.2.1 Huffman-Codierung

Die Huffman-Codierung [Huf52] geht davon aus, dass die einzelnen Symbole einer Nachricht statistisch unabhängig voneinander sind, und codiert diese separat. Symbole, die eine hohe Auftrittswahrscheinlichkeit besitzen, werden mit kürzeren Codes abgespeichert als Werte mit geringer Auftrittswahrscheinlichkeit.

grafische Ermittlung

Um eine Huffman-Codierung auf grafische Weise zu ermitteln, werden die Quellsymbole als Codebäume mit einem Knoten betrachtet. In jedem Knoten wird die Auftrittswahrscheinlichkeit des jeweiligen Symbols eingetragen. Die verschiedenen Knoten werden dann wie folgt zu einem einzelnen Baum zusammengefügt:

1. Die zwei Knoten mit der geringsten Auftrittswahrscheinlichkeit werden zu einem neuen Knoten zusammengefasst.
2. Dem neu entstandenen Knoten wird die Summe der beiden Auftrittswahrscheinlichkeiten zugewiesen.
3. Die neu entstandenen Zweige erhalten als Beschriftung 0 bzw. 1.
4. Man beginnt von vorne, wenn die Wurzel des Baumes mit der Auftrittswahrscheinlichkeit 1 noch nicht erreicht ist.



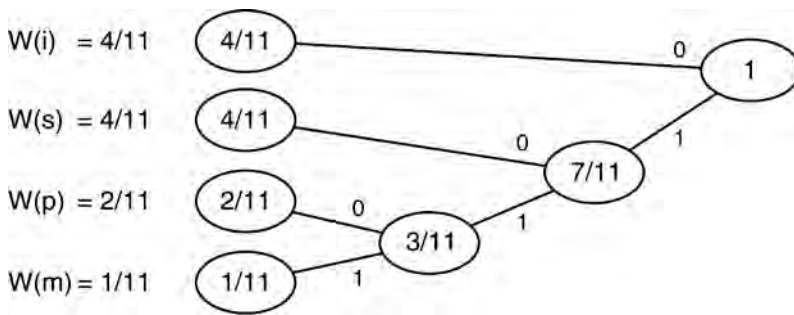


Abb. 1.2:  
Grafische Entwicklung einer  
Huffman-Codierung

Ausgehend von den Blattknoten kann am Ende das Codewort für jedes Symbol abgelesen werden. Die grafische Entwicklung einer Huffman-Codierung für die Zeichenkette *mississippi* ist in Abb. 1.2 dargestellt. Die Huffman-Codierung besitzt eine Entropie von  $1,82 \text{ bit/Symbol}$  und eine mittlere Codelänge von  $1,91 \text{ Bit/Symbol}$ . Die Codierungsredundanz ist demnach nahe null. Dabei wurde jedoch nicht berücksichtigt, dass der Codebaum für die Decodierung bekannt sein muss und daher zusätzlich abgespeichert wird.

Da die Huffman-Codierung einzelne Symbole codiert, und diese immer durch eine ganzzahlige Folge von Bits dargestellt werden müssen, ist die Codierung nicht in allen Fällen optimal. Eine Codierungsredundanz von  $0 \text{ Bits}$  wird nur dann erreicht, wenn der Informationsgehalt eines Symbols und die zugeordnete Codelänge übereinstimmen. Dies ist der Fall, wenn die Auftretswahrscheinlichkeiten der Bedingung  $W(z_i) = 2^{-n}$  mit  $n = 1, 2, 3, \dots$  genügen. Ein Vorteil der Huffman-Codierung ist, dass sie einfach implementiert werden kann, und es sich nur um ein geringfügig asymmetrisches Verfahren handelt.

Eigenschaften

### 1.3.2.2

#### Arithmetische Codierung

Die arithmetische Codierung rückt von der separaten Symbolcodierung ab und erzeugt einen einzigen Code für die gesamte Nachricht. So wird Redundanz auch dort effizient vermindert, wo sich eine Huffman-Codierung als ungünstig erweisen würde. Trotzdem lässt sich die arithmetische Codierung sequentiell durch Abarbeiten einzelner Symbole ermitteln. Wie die Huffman-Codierung geht die arithmetische Codierung davon aus, dass die Nachrichtensymbole statistisch unabhängig voneinander sind.

Das Funktionsprinzip der arithmetischen Codierung ist es, die gesamte Zeichenfolge  $z_0, z_1, \dots, z_n$  durch eine rationale Zahl aus dem Einheitsintervall  $[0,1)$  zu codieren. Dies geschieht wie folgt:

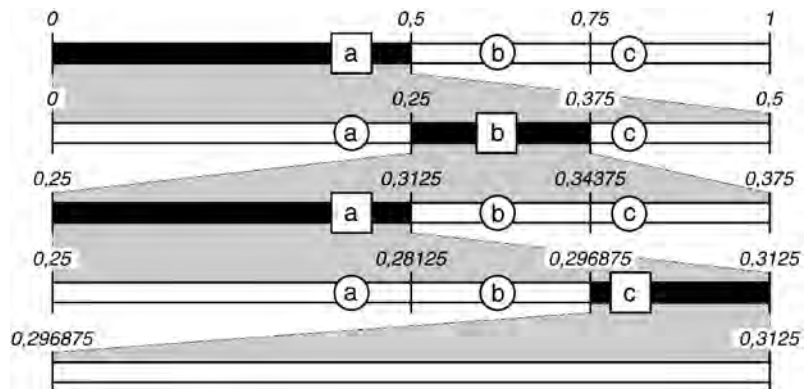
Funktionsprinzip

1. Das Einheitsintervall  $[0,1)$  wird ausgewählt.
2. Das aktuell gewählte Intervall wird so in Teilintervalle eingeteilt, dass die Größen der Teilintervalle mit den Auftretswahrscheinlichkeiten  $W(z_i)$  der Symbole  $z_0, z_1, \dots, z_n$  korrespondieren.
3. Das Teilintervall, das dem aktuellen zu codierenden Symbol  $z_i$  entspricht, wird ausgewählt.
4. Ist  $z_i$  das letzte Symbol der Zeichenfolge, so kann diese durch eine beliebige Zahl aus dem Teilintervall codiert werden. Ansonsten setzt man bei Schritt 2 fort.

*Beispiel*

Das beschriebene Verfahren wird in Abb. 1.3 am Beispiel der Zeichenfolge *abac* mit den Auftretswahrscheinlichkeiten  $W(a) = 0,5$  sowie  $W(b) = W(c) = 0,25$  dargestellt. Die Zeichenkette kann in diesem Fall durch eine Zahl aus  $[0,296875, 0,31275)$  codiert werden. Die Decodierung wendet das eben beschriebene Verfahren rückwärts an und muss daher nicht explizit erläutert werden.

Abb. 1.3:  
Arithmetische Codierung der  
Zeichenkette *abac*



*Entstehung*

Die theoretische Grundlage der arithmetischen Codierung wurde durch [Shn48] gelegt. Trotz des relativ einfachen Prinzips scheiterte eine algorithmische Umsetzung lange Zeit an der endlichen Genauigkeit von Fließkommazahlen auf Computern. Mit zunehmender Nachrichtenlänge werden die Teilintervalle sehr klein und können mit endlicher Fließkomma-Genauigkeit u. U. nicht mehr eindeutig unterschieden werden. Schließlich gelang der Nachweis, dass bestimmte endlich lange Zahlendarstellungen noch die benötigte Unterscheidbarkeit bieten. Einer der ersten Algorithmen für die technische Umsetzung der arithmetischen Codierung wurde von IBM veröffentlicht [RiL79].

### 1.3.3 Prädiktion

Die Prädiktion zieht vorhergegangene und damit bekannte Symbole einer Nachricht heran, um Voraussagen über unbekannte Folgesymbole treffen zu können. Dies ist sinnvoll, da zwischen benachbarten Symbolen einer Nachricht häufig Abhängigkeiten bestehen. So ist in der deutschen Sprache beispielsweise eine hohe Korrelation zwischen den Buchstaben *Q* und *U* zu finden. Auch die Pixel eines Bildes können korrelieren, wenn sie demselben abgebildeten Objekt zugeordnet sind. Die Prädiktion stammt ursprünglich aus dem Bereich der Sprachverarbeitung, wo sie zur Kompression und Spracherkennung verwendet wird [AtS67].

Die Abweichung zwischen vorausgesagtem Prädiktionswert und tatsächlich auftretendem Folgesymbol wird als Prädiktionsfehler bzw. Residuum bezeichnet. Seien  $\hat{s}(n)$  ein Prädiktionswert und  $s(n)$  das tatsächlich auftretende Folgesymbol, so ist der Prädiktionsfehler  $e(n) = s(n) - \hat{s}(n)$ . Mit Hilfe von Prädiktionsfehlern lassen sich Nachrichten eindeutig decodieren, indem der Decoder ebenfalls Prädiktionswerte errechnet und diese durch Addition der Prädiktionsfehler korrigiert.

*Prädiktionsfehler bzw. Residuum*

Durch die Symbolkorrelation bedingt, entsteht bei der Prädiktion eine starke Ungleichverteilung der Prädiktionsfehler, was wiederum eine geringere Entropie bedeutet. Daher ist die Prädiktion ein Verarbeitungsschritt, welcher die Bedingungen für eine nachfolgende Entropiecodierung verbessert.

Die Prädiktion von Folgesymbolen erfolgt im einfachsten Fall als lineare Prädiktion durch das unmittelbar vorhergehende Symbol. Dieses kann entweder direkt verwendet oder vorher einer Gewichtung unterzogen werden. Der Gewichtungsfaktor wird als Prädiktorkoeffizient  $a_k$  bezeichnet und muss meist fortlaufend an das Signal angepasst werden. Wenn mehr als ein Vorgängersymbol zur Prädiktion herangezogen wird, so handelt es sich um eine Prädiktion höherer Ordnung. Dies kann z. B. bei der Prädiktion von Bildinhalten sinnvoll sein, da sich ein Bildpixel genauer vorhersagen lässt, wenn mehrere umliegende Pixel, das sog. Template, berücksichtigt werden.

*lineare Prädiktion*

Die lineare Prädiktion  $p$ -ter Ordnung erfolgt mit  $p$  Prädiktorkoeffizienten  $a_k$ , die auch als Linear Prediction Coefficients (LPC) bezeichnet werden. Der Prädiktionswert  $\hat{s}(n)$  errechnet sich dann aus  $p$  vorangegangenen Symbolen als FIR-Filter nach:

*Linear Prediction Coefficients (LPC)*

$$\hat{s}(n) = \sum_{k=1}^p a_k \cdot s(n-k)$$

Es muss folglich ein FIR-Filter mit den Filterkoeffizienten  $a_k$  entwickelt werden, das die zukünftigen Symbole möglichst gut auf Basis vergangener Symbole vorhersagen kann. Dieses Filter wird Analyse- oder Prädiktionsfilter genannt, während das korrespondierende Gegenstück des Decoders als Synthese- oder LPC-Filter bezeichnet wird. Gibt man das Prädiktionsfehler-Signal, das vom Analysefilter errechnet wurde, unverändert in das Synthesefilter hinein, so kann das ursprüngliche Signal ohne Fehler rekonstruiert werden.

Eine etwas andere, wenn auch verwandte Aufgabe stellt sich mit der AR-Modellierung auf dem Gebiet der parametrischen Sprachkompression (Kap. 1.9.2). Dort wird ein Synthese-Filter gesucht, das ein stationäres Anregungssignal, wie z. B. ein weißes Rauschen, möglichst gut an das zu codierende Sprachsignal annähert. Zwischen linearer Prädiktion und AR-Modellierung besteht ein sehr enger Zusammenhang, der in [KaK02] ausführlich erläutert wird.

Prinzipiell arbeitet die Prädiktion verlustfrei, sofern die errechneten Prädiktionsfehler nicht quantisiert werden. Quantisierung bietet sich an, wenn die Prädiktionsfehler rational sind, was eine weitere Verarbeitung erschweren würde.

In vielen Verfahren zur Bildkompression wird die Prädiktion nicht direkt auf Bildpunkte angewendet. Statt dessen wird das Bild zunächst in eine Repräsentation überführt, die sich besser für den Einsatz einer prädiktiven Codierung eignet. Dies geschieht bspw. durch Signaltransformationen, die im Folgenden erläutert werden.

### 1.3.4 Signaltransformationen

Wie die Prädiktion zielen auch Signaltransformationen darauf ab, die Korrelation zwischen Nachrichtensymbolen zu verringern, und so die Effizienz nachfolgender Verarbeitungsschritte zu steigern. Um eine anschaulichere Darstellung zu ermöglichen, liegen den folgenden Erläuterungen Bilddaten zugrunde. Signaltransformationen sind jedoch nicht auf solche Daten beschränkt.

#### 1.3.4.1 Diskrete Kosinustransformation

Die diskrete Kosinustransformation (DCT) als Sonderfall der Fouriertransformation existiert in verschiedenen Ausprägungen, die von [Wan84] erstmals in vier Transformationstypen eingeteilt wurden. Bei der Bilddatenkompression wird die DCT-II nach [ANR74] verwendet. Die DCT-II und die zugehörige inverse Transformation werden im Folgenden nur noch als DCT bzw. IDCT bezeichnet. Die DCT zur

Transformation zweidimensionaler Bilder wird auf quadratische Teilbereiche des Bildes angewendet und arbeitet auf Basis einzelner Bildpixel. Bei einem quadratischen Bildausschnitt von  $N \times N$  Pixeln wird jedes Pixel  $f(x,y)$  wie folgt transformiert:

$$F(u,v) = \frac{2}{N} \cdot C_u C_v \cdot \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cdot \cos\left[\frac{(2x+1) \cdot u \cdot \pi}{2N}\right] \cdot \cos\left[\frac{(2y+1) \cdot v \cdot \pi}{2N}\right]$$

Die IDCT ist:

$$f(x,y) = \frac{2}{N} \cdot \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C_u C_v \cdot F(u,v) \cdot \cos\left[\frac{(2x+1) \cdot u \cdot \pi}{2N}\right] \cdot \cos\left[\frac{(2y+1) \cdot v \cdot \pi}{2N}\right]$$

Dabei gilt für die beiden Gleichungen:

$$C_u = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } u = 0 \\ 1 & \text{für } u \neq 0 \end{cases} \quad \text{und} \quad C_v = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } v = 0 \\ 1 & \text{für } v \neq 0 \end{cases}$$

Die DCT transformiert das Bildsignal vom Orts- in den Frequenzbereich und zerlegt es dazu in Kosinus-Basisfunktionen unterschiedlicher Frequenz. Jeder Bildausschnitt mit  $N \times N$  Pixeln resultiert in  $N \times N$  Koeffizienten, die als zweidimensionale Frequenzangaben interpretiert werden können. Hohe Frequenzen repräsentieren im Ortsbereich feine Bildstrukturen, niedrige Frequenzen dagegen homogene Flächen.

*Orts- und Frequenzbereich*

In Abb. 1.4 ist der Ausschnitt eines Graustufenbildes zu sehen. Es handelt sich dabei um einen Bildbereich aus  $8 \times 8$  Pixeln, wie er z. B. auch vom JPEG-Standard verwendet wird (Kap. 1.6.1). Das Graustufenbild ist mit 8 Bit pro Pixel quantisiert, weshalb die einzelnen Pixel Graustufenwerte von 0 bis 255 besitzen können.

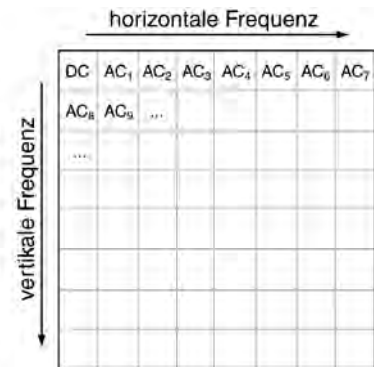
*Beispiel*



*Abb. 1.4:  
Quadratischer Pixelblock  
eines Graustufenbildes*

Abb. 1.5:  
DCT-Koeffizienten eines Pixelblocks

|        |        |       |       |       |      |      |       |
|--------|--------|-------|-------|-------|------|------|-------|
| 806,8  | -136,5 | 56,6  | -59,3 | -94,5 | -2,7 | 7,9  | -8,8  |
| -137,6 | -71,0  | 35,7  | 20,4  | 12,3  | 24,4 | 12,2 | 3,3   |
| 47,6   | 9,8    | -67,8 | 46,9  | 88,6  | 12,0 | -2,1 | -4,3  |
| 1,9    | 15,3   | -3,5  | -21,2 | -8,2  | 19,0 | 18,8 | -11,5 |
| -33,5  | 43,8   | 34,6  | -66,5 | -31,3 | 37,5 | 22,2 | 1,7   |
| -17,6  | 0,4    | -37,8 | -52,1 | 7,1   | 13,8 | 1,0  | 0     |
| 1,1    | 8,4    | 0,4   | 15,6  | 14,7  | 3,9  | 5,3  | -9,6  |
| -5,5   | 10,1   | 15,6  | 1,3   | -11,0 | -2,0 | 2,0  | -8,6  |



#### DC-Koeffizient und AC-Koeffizienten

Abbildung 1.5 zeigt die gerundeten DCT-Koeffizienten des transformierten Pixelblocks. In Anlehnung an die englischen Bezeichnungen für Gleich- und Wechselstrom lassen sich die Koeffizienten in einen DC-Koeffizient und 63 AC-Koeffizienten unterscheiden. Der DC-Koeffizient, der häufig auch als Gleichanteil bezeichnet wird, entspricht dem Anteil der Frequenz 0 Hz in beiden Bildachsen und repräsentiert den mittleren Farbton bzw. Grauwert des  $8 \times 8$  Pixelblocks. Die AC-Koeffizienten entsprechen Frequenzen größer 0 Hz in horizontaler und vertikaler Richtung. So entspricht der Koeffizient AC<sub>7</sub> beispielsweise der höchsten Frequenz, die im Pixelblock innerhalb der horizontalen Bildachse auftritt. Im Ortsbereich ist dies das dichtest mögliche Muster senkrechter Streifen innerhalb des Pixelblocks.

#### Weiterverarbeitung durch Entropie- und Lauflängencodierung

Da viele Bilder hauptsächlich aus flächigen Bereichen und nur zu einem geringen Teil aus scharfen Kanten bestehen, werden DCT-Koeffizienten, die höhere Frequenzen repräsentieren, sehr häufig niedrige Werte annehmen, wie auch in Abb. 1.5 beispielhaft zu erkennen ist. Die DCT-Koeffizienten sind also oft ungleich verteilt, und weisen somit eine geringe Entropie auf. Dies machen sich Verfahren zur Bild- und Videokompression zu Nutze und unterziehen die Koeffizienten einer nachfolgenden Entropiecodierung. Indem die Koeffizienten zuvor quantisiert werden, lässt sich die Kompression weiter optimieren. Durch die Quantisierung nehmen viele der kleineren Koeffizienten den Wert 0 an und können durch eine Lauflängencodierung zusammengefasst werden.

DCT-Koeffizienten können durch die IDCT theoretisch ohne Informationsverluste in die ursprünglichen Graustufenwerte zurück transformiert werden. In der Praxis lassen sich DCT und IDCT jedoch nur annähernd genau berechnen und die diskrete Kosinustransformation ist verlustbehaftet.

### 1.3.4.2

#### Wavelet-Transformation

Obwohl ein erstes Wavelet schon sehr früh im Rahmen der Haar-Transformation [Haa10] beschrieben wurde, entstand eine zusammenhängende Wavelet-Theorie erst in den 80er Jahren, nachdem Grossmann u. Morlet [GrM84] den Wavelets ihren Namen gegeben und eine genauere Beschäftigung mit dem Thema ausgelöst hatten.

Wie die DCT wandelt die Wavelet-Transformation das Bildsignal in den Frequenzbereich. Anstelle einer Kosinusfunktion werden dabei jedoch die Wavelets einer Wavelet-Familie als Transformationskern verwendet. Diese Wavelets erhält man durch einfache Modifikationen eines Mutter-Wavelets, von denen einige in Abb. 1.6 zu sehen sind. Das Haar-Wavelet ist sehr einfach aufgebaut und besitzt nur eingeschränkte Funktionalität, da es u. A. nicht differenzierbar ist. Aufgrund ihrer mathematischen Eigenschaften werden in der Signalverarbeitung oft Daubechies-Wavelets [Dau88] eingesetzt. Diese Wavelets lassen sich nicht funktionsanalytisch darstellen, sondern werden iterativ erzeugt. Durch die Wahlmöglichkeit eines Mutter-Wavelets und durch die Verwendung verschiedener Wavelets einer Wavelet-Familie ist die Wavelet-Transformation flexibler als die DCT mit ihrer festgelegten Basisfunktion.

Mutter-Wavelets können durch Verschiebungen auf der Abszisse und durch Skalierungen modifiziert werden. Durch Dehnung des Wavelets lassen sich stationäre Signalabschnitte erfassen, für die man eine gute Frequenzauflösung erhält, während sich gestauchte Wavelets mit hoher Frequenz für wechselhafte Signalabschnitte eignen, in denen eine gute zeitliche bzw. örtliche Auflösung gewünscht ist. Wavelets ermöglichen demnach eine Kombination aus Frequenzanalyse und zeitlicher bzw. örtlicher Analyse. Funktionswerte von Wavelets sind außerhalb eines engen Intervalls 0, während Kosinusfunktionen unendlich periodisch verlaufen. Durch diesen kompakten Träger wirken Wavelets immer nur auf einen Teilbereich des Bild- oder Tonsignals, und eine Aufteilung in Blöcke, wie z. B. bei der DCT, ist nicht mehr erforderlich.

*Funktionsprinzip, Wavelet-Arten*

*Modifikation von Mutter-Wavelets*

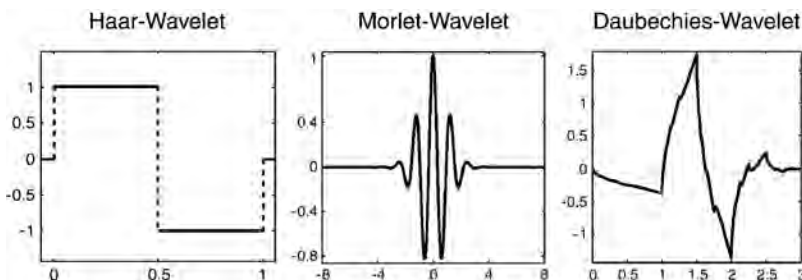


Abb. 1.6:  
Wavelets

### Multiskalen-Analyse

Die Multiskalen-Analyse [Mal89] führt eine Skalierungsfunktion als Basis der Wavelet-Familie ein und ermöglicht so eine effiziente algorithmische Umsetzung der Transformation. Diese wird als schnelle Wavelet-Transformation (FWT) bezeichnet und lässt sich als Kombination spezieller Hoch- und Tiefpass-Filter auffassen.

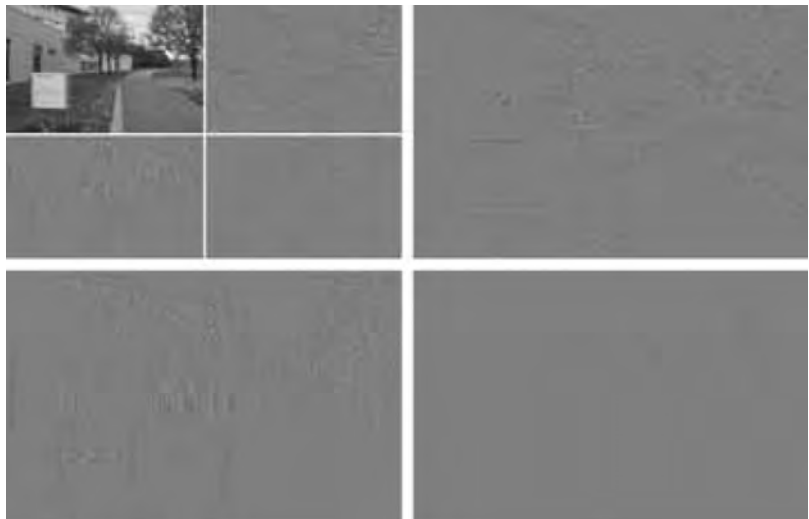
### schnelle Wavelet-Transformation (FWT)

In der Bilddatenkompression wird die FWT meist nacheinander auf die Zeilen und Spalten des Bildes angewendet. Zunächst werden die Zeilen des Ausgangsbildes transformiert. Durch den Tiefpass-Filter erhält man eine geglättete, unscharfe Version des Bildes, der Hochpass resultiert in feinen Detailinformationen. Zusätzlich wird die Bildauflösung durch Subsampling reduziert, indem jedes zweite Pixel verworfen wird. Derselbe Vorgang wird nun auf die Spalten angewendet, wodurch man insgesamt vier verkleinerte Versionen des Ausgangsbildes erhält. Die Bildversion, deren Spalten und Zeilen Tiefpass-gefiltert wurden, kann nun mehrere Male auf dieselbe Weise weiterverarbeitet werden, wobei in jedem Verarbeitungsschritt vier weitere verkleinerte Bildversionen entstehen. Die drei anderen Bildversionen enthalten Detailinformationen, die mit senkrechten, waagerechten bzw. diagonalen Kanten innerhalb des Bildinhaltes korrespondieren. Durch die schrittweise Verarbeitung durch Filter erhält man eine unscharfe Repräsentation des Originalbildes in sehr geringer Auflösung, sowie mehrere Bilder mit Detailinformationen in unterschiedlichen Auflösungen.

### Beispiel

In Abb. 1.7 ist dies anhand der ersten zwei Verarbeitungsschritte mit einem Haar-Wavelet zu sehen. Bei der Dekompression wird die gering aufgelöste Bildrepräsentation sukzessive durch die Detailinforma-

Abb. 1.7:  
Zwei Verarbeitungsschritte der  
schnellen Wavelet-Transformation





tionen angereichert, was als hierarchische Kompression bezeichnet wird. Hierarchische Kompression ist auch schon im JPEG-Standard (Kap. 1.6.1) als spezieller Modus verfügbar, allerdings auf Basis einer DCT. Die Dekompression des Bildes muss nicht komplett erfolgen, sondern kann gestoppt werden, wenn eine ausreichend hohe Auflösung erreicht wurde.

Die Wavelet-Transformation kann in der Bildverarbeitung anstelle der DCT eingesetzt werden, wodurch sich die eingangs geschilderten Vorteile der Wavelets nutzen lassen. Wie bei der DCT werden die resultierenden Koeffizienten durch eine Quantisierung und Entropiecodierung weiterverarbeitet. Die FWT ist bei Wahl eines geeigneten Wavelets und zugehöriger Skalierungsfunktion ein verlustfreier Vorgang. Trotz des Subsamplings lässt sich das Bild ohne Informationsverlust rekonstruieren. Erst durch eine entsprechende Quantisierung wird die Wavelet-Kompression zum verlustbehafteten Vorgang.

*Eigenschaften*

## 1.4 Medienspezifische Verarbeitungsschritte

Neben elementaren Verarbeitungsschritten, die sich für die Datenkompression sehr vielseitig einsetzen lassen, existieren auch medien-spezifische Verarbeitungsschritte, bei denen die Charakteristiken bestimmter Medientypen ausgenutzt werden.

### 1.4.1 Interframe-Kompression

Bei der Kompression von Videodaten lassen sich besonders hohe Kompressionsraten erzielen, wenn zusätzlich zu örtlichen Redundanzen innerhalb eines Einzelbildes auch zeitliche Redundanzen zwischen aufeinander folgenden Bildern berücksichtigt werden. In vielen Fällen sind Folgebilder einer Videosequenz sehr ähnlich, da sich von Bild zu Bild nur begrenzte Bereiche unterscheiden.

Werden auch zeitliche Redundanzen für die Kompression ausgenutzt, so spricht man von Interframe-Kompression. Im Gegensatz dazu verarbeitet die Intraframe-Kompression einzelne Videobilder separat und berücksichtigt somit nur örtliche Redundanz.

*Interframe- und Intraframe-Kompression*

Ein möglicher Ansatz für die Interframe-Kompression ließe sich realisieren, indem die Prädiktion nach Kap. 1.3.3 um eine Dimension für die Zeit erweitert wird. Um jedoch exaktere zeitliche Vorhersagen zu erhalten, werden Bewegungen innerhalb des Bildinhalts abgeschätzt. Dabei beschränkt man sich in der Regel auf translatorische

Bewegungen und verzichtet auf die Prädiktion von Rotationen, Skalierungen, Verzerrungen usw.

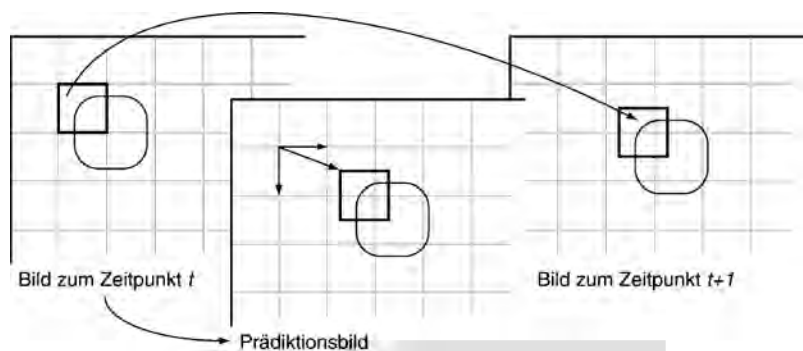
*Bewegungsschätzung*

Im einfachsten Fall werden bei der Bewegungsschätzung ausschließlich Luminanzwerte von Pixeln betrachtet, und Voraussagen über die Bewegung erfolgen ohne Berücksichtigung einer Semantik. Die Bedeutung der Bildinhalte spielt also keine Rolle, und es wird ausschließlich die Ähnlichkeit unterschiedlicher Bildausschnitte bewertet. Ein solches Verfahren kommt bspw. für die MPEG-1-Kompression (Kap. 1.7.1) zum Einsatz, verarbeitet dort quadratische Pixelblöcke und wird als Blockmatching bezeichnet. Das Funktionsprinzip basiert auf einer Publikation von [RoZ72] und wird im Folgenden näher erläutert.

*Blockmatching,  
Bewegungsvektoren*

Zunächst wird die Y-Komponente des Originalbilds in Makroblöcke aus  $16 \times 16$  Pixeln zerlegt. Diese Blockgröße resultiert aus einem Kompromiss zwischen exakter Bewegungsschätzung durch möglichst kleine Blöcke und geringem Codierungsaufwand durch eine möglichst geringe Block-Anzahl. Da die CrCb-Komponenten meist geringer aufgelöst sind als die Y-Komponente, werden sie nicht für die Bewegungsschätzung herangezogen. Für jeden Makroblock muss nun ein korrespondierender Block im Folgebild gefunden werden, dessen Bildinhalt möglichst genau übereinstimmt. Dazu empfiehlt der MPEG-1-Standard verschiedene Suchstrategien, die jedoch nicht verbindlich sind. Hersteller von Encodern besitzen so die Freiheit, individuelle Optimierungen bei der Blocksuche einzusetzen. Für die Beurteilung der Übereinstimmung von Blöcken können unterschiedliche Abstandsmaße verwendet werden, wie z. B. die mittlere absolute Differenz. Die Positionen der gefundenen Blöcke werden als zweidimensionale Bewegungsvektoren abgespeichert. Diese geben an, wie die ursprünglichen Blöcke verschoben werden müssen, um an die Positionen der korrespondierenden Blöcke im Folgebild zu gelangen (Abb. 1.8). Vektoren für die Chrominanzkanäle werden unter Berücksichtigung des Subsamplings aus den Bewegungsvektoren der Y-Komponente errechnet.

Abb. 1.8:  
Funktionsprinzip der  
vorwärtsgerichteten Prädiktion



Bewegungsvektoren werden bei MPEG-1 durch lineare Interpolation mit einer Genauigkeit von  $\frac{1}{2}$  Pixel ermittelt. Es ist jedoch auch eine einfache, ganzzahlige Pixelgenauigkeit erlaubt. Der Wertebereich der Vektoren-Elemente ist eingeschränkt, wobei je nach Verschiebungsgenauigkeit zwei verschiedene Wertebereiche definiert sind. Vektoren, die über die Bildränder hinauszeigen, sind unzulässig.

Nachdem durch die Bewegungsschätzung alle Vektoren errechnet wurden, erzeugt die Bewegungskompensation ein Prädiktionsbild. Dazu werden alle Makroblöcke des Originalbilds so verschoben, wie von Bewegungsvektoren vorgegeben. Man erhält dadurch ein Prädiktionsbild, das mosaikartig aus Blöcken zusammengesetzt ist. Der Unterschied zwischen diesem Prädiktionsbild und dem Originalbild wird als Prädiktionsfehler codiert.

Abbildung 1.9 zeigt zwei aufeinander folgende Videobilder, in denen der Regler des abgebildeten Mischpults nach oben bewegt wird. Darunter sind Prädiktionsbild und Prädiktionsfehlerbild zu sehen.

*Bewegungskompensation*

*Beispiel*



aufeinanderfolgende Videobilder

Abb. 1.9:  
Vorwärtsgerichtete Prädiktion am  
Beispiel



Prädiktionsbild



Prädiktionsfehlerbild

Das oben geschilderte Blockmatching wird als vorwärtsgerichtete Prädiktion bezeichnet, da ein zeitlich folgendes Bild vorhergesagt wird. Vorwärtsgerichtete Prädiktion kann jedoch nicht immer erfolgreich sein, da in Folgebildern häufig neue Elemente auftauchen, die in vorhergehenden Bildern verdeckt waren oder sich außerhalb des Bildrahmens befanden. Daher setzt MPEG-1 zusätzlich bidirektionale Prädiktion ein, wobei die Bewegungsschätzung auch im vorhergehenden Bild nach ähnlichen Blöcken sucht. So ergeben sich für jeden Makroblock zwei Bewegungsvektoren, von denen derjenige in den

*vorwärtsgerichtete und  
bidirektionale Prädiktion*

Datenstrom eincodiert wird, der den kleineren Prädiktionsfehler erzeugt. Als weitere Variante können auch beide Vektoren im Datenstrom verwendet werden, wobei der Decoder zur Prädiktion einen Mittelwert der Vektoren heranzieht.

Da die Blockstruktur des Blockmatchings nur in den seltensten Fällen mit der Struktur von Bildinhalten übereinstimmt, sind immer relativ große Prädiktionsfehler zu erwarten. Dies gilt auch für Bewegungen, die sich nicht durch das Verschieben von Blöcken erfassen lassen, wie z. B. Veränderungen der Lichtverhältnisse oder Skalierungen. Trotzdem ist Blockmatching ein relativ leistungsfähiges Verfahren und lässt sich einfach implementieren.

*Korrelation bei benachbarten  
Makroblöcken*

Insgesamt können Bewegungsvektoren bis zu 40% der gesamten Information eines komprimierten Video-Datenstroms ausmachen [ChP89]. Darüber hinaus sind die Vektoren benachbarter Makroblöcke stark korreliert. So stimmen insbesondere bei Kameraschwenks viele der Vektoren weitgehend überein. Daher ist auch für die Codierung der Vektoren der Einsatz von Prädiktion sinnvoll. Für MPEG-1 verwendet man bspw. eine einfache Delta-Codierung und sagt Bewegungsvektoren für die bidirektionale Prädiktion nur aus Vektoren derselben Richtung vorher.

#### 1.4.2

### Psychoakustische Kompression

*psychoakustisches Modell,  
Perceptual Coder*

Unter psychoakustischer Kompression versteht man die verlustbehaftete Kompression von Audiodaten mit Hilfe eines psychoakustischen Modells. Ein solches Modell beschreibt Grenzen des menschlichen Gehörs und ermöglicht dadurch effektive Irrelevanzreduktion. Da sie das subjektive Hörempfinden betreffen, lassen sich psychoakustische Modelle nur durch umfangreiche Testreihen mit Versuchspersonen empirisch ermitteln. Encoder, die ein psychoakustisches Modell verwenden, werden als Perceptual Coder bezeichnet.

*MP3*

Weltweite Bekanntheit erlangte die Kompression unter Einsatz eines psychoakustischen Modells durch MP3, das am Erlanger Fraunhofer Institut für Integrierte Schaltungen (IIS) entwickelt wurde [BrS94]. MP3 ist Teil des internationalen Standards MPEG-1 (Kap. 1.7.1) für die Kompression von Video- und Audiodaten. Im MPEG-1-Standard werden 3 Audio-Layer definiert, die sich bezüglich Komplexität und Kompressionsleistung unterscheiden. MP3 bezeichnet Audiodateien, die gemäß Layer III codiert sind.

*Schalldruckpegel*

Psychoakustische Modelle nutzen mit Simultanverdeckung und zeitabhängiger Verdeckung zwei Eigenschaften des menschlichen Gehörs, die im Folgenden nach [ZwF99] dargestellt werden. Der

Schalldruckpegel  $L_p$  eines Schallereignisses mit dem Schalldruck  $p_1$  ergibt sich, indem  $p_1$  zum Bezugsschalldruck  $p_0 = 2 \cdot 10^{-5} [Pa]$  ins Verhältnis gesetzt, und das Resultat nach folgender Formel logarithmiert und gewichtet wird:

$$L_p = 20 \cdot \lg \frac{p_1}{p_0} [Pa]$$

Zeichnet man den benötigten Schalldruckpegel, um einen Sinuston  $T$  gerade wahrzunehmen, als Funktion der Frequenz von  $T$  auf, so erhält man die Ruhehörschwelle. Schallereignisse unterhalb der Ruhehörschwelle sind für das menschliche Gehör nicht wahrnehmbar und können daher durch Irrelevanzreduktion eliminiert werden. Das Gehör weist nicht für alle Frequenzen die gleiche Empfindlichkeit auf, sondern ist für Frequenzen von 2 bis 5 kHz am sensibelsten. Demnach ist die wahrgenommene Lautstärke und damit auch die Ruhehörschwelle nicht nur vom Schalldruckpegel sondern auch von der Frequenz des Schalls abhängig (Abb. 1.10).

*Ruhehörschwelle*

Die Simultanverdeckung betrifft Verdeckungseffekte durch stationäre Schallereignisse, die zeitgleich auftretende Signalkomponenten mit gleicher und abweichender Frequenz verdecken. Solche verdeckenden Schallereignisse werden als Maskierer bezeichnet. Verdeckungseffekte lassen sich als Anhebung der Ruhehörschwelle für die verdeckten Komponenten interpretieren (Abb. 1.11). Simultan verdeckende Maskierer bewirken folglich, dass die Hörschwelle für andere, zeitgleiche Schallereignisse angehoben wird. Fallen Schallereignisse dabei nur teilweise unter die Hörschwelle, so handelt es sich um Teilverdeckung, die zur Lautstärkereduktion der teilverdeckten Signalkomponente führt.

*Simultanverdeckung, Maskierer*

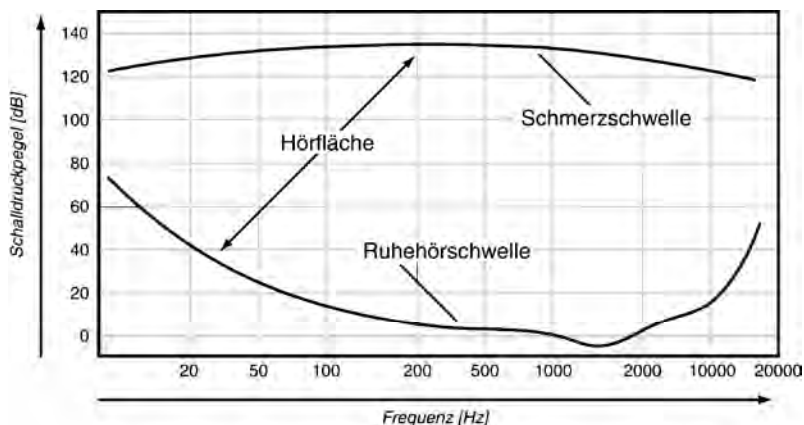
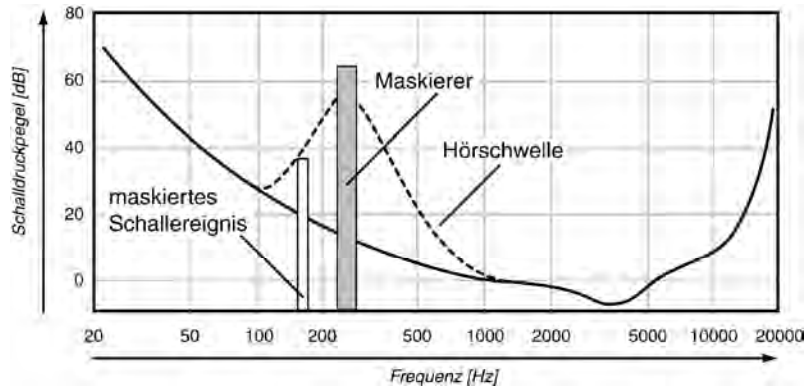


Abb. 1.10:  
Ruhehörschwelle des menschlichen Gehörs

Abb. 1.11:  
Simultanverdeckung



zeitliche Verdeckung

Zeitliche Verdeckungseffekte betreffen zeitlich aufeinander folgende Schallereignisse von kurzer Dauer und können in Vor- und Nachverdeckung unterschieden werden. Nachverdeckung besagt, dass das Gehör nach einem maskierenden Schallereignis eine Zeit von etwa 150 ms benötigt, bis sich die Ruhehörschwelle wieder normalisiert hat. Es kommt dabei zu Verdeckungseffekten, obwohl der Maskierer physikalisch nicht mehr existiert, da bereits angeregte Nervenzellen weniger sensitiv reagieren als ruhende Zellen. Vorverdeckung wirkt sich aus, bevor der Maskierer auftritt, was logisch vielleicht schwerer zu fassen ist. Vorverdeckung entsteht, da das Gehör eine begrenzte zeitliche Auflösung besitzt, und ein unvollständig verarbeitetes Schallereignis von der nachfolgenden Signalkomponente verdeckt wird.

Quantisierungsrauschen

Perceptual Coder quantisieren Audiosignale sehr grob, so dass das rekonstruierte Signal dem ursprünglichen Signalverlauf nur noch schlecht folgen kann. Dadurch entsteht mit dem sog. Quantisierungsrauschen ein rauschartiges Störgeräusch, das vom Audiosignal moduliert wird und sich folglich mit dem Signalverlauf ändert. Im Vergleich zu konstantem Hintergrundrauschen wird Quantisierungsrauschen subjektiv als besonders störend empfunden. Die beschriebenen Verdeckungseffekte werden nun genutzt, um dieses Quantisierungsrauschen geschickt zu maskieren. So können teilweise sehr grobe Quantisierungsstufen eingesetzt und dadurch hohe Kompressionsraten erzielt werden. Bei der psychoakustischen Kompression handelt es sich also im Wesentlichen um variable Quantisierung, die durch ein psychoakustisches Modell gesteuert und als Rauschformung bezeichnet wird.

Durch Versuchsreihen wurde ermittelt, dass der Schneckenang des Innenohrs (lat.: Cochlea) Schallereignisse in Teilbänder zerlegt, und das Gehör nur ein eingeschränktes Auflösungsvermögen für Frequenzen besitzt. Das Auflösungsvermögen ist nicht linear, sondern beträgt in den tiefen Frequenzen etwa 50 Hz und verschlechtert sich

mit zunehmender Frequenz bis etwa 4 kHz. Dabei ist es keinesfalls so, dass Frequenzen innerhalb eines Teilbands nicht unterschieden werden können. Die Hörschwelle für Quantisierungsrauschen kann jedoch separat für jedes Teilband ermittelt werden und hängt nur von Schallereignissen innerhalb des Teilbands ab. Die Teilbänder des menschlichen Gehörs werden als kritische Frequenzgruppen bezeichnet. Alle psychoakustischen Kompressionsverfahren zerlegen das Eingangssignal in Frequenzbänder und bestimmen die Quantisierung mit Hilfe des psychoakustischen Modells für jedes Subband separat. Die quantisierten Koeffizienten werden abschließend meist einer Lauflängen- und Huffman-Codierung unterzogen.

Audiodaten, die mit einem Perceptual Coder komprimiert wurden, eignen sich nicht für die Nachbearbeitung. Eine Veränderung der Klangcharakteristik, z. B. durch Anpassung von Höhen oder Bässen, verändert nachträglich auch die Entscheidungsgrundlage des psychoakustischen Modells, wodurch das Quantisierungsrauschen wieder hörbar werden kann.

*Nachbearbeitung*

## 1.5 Codecs im Überblick

Die Joint Photographic Experts Group (JPEG), die Moving Picture Experts Group (MPEG) und die ITU sind prominente Gremien im Bereich der Mediendaten-Kompression. Die JPEG-Gruppe entwickelte mit dem gleichnamigen JPEG-Format einen der ersten Standards [ISO94] [ITU92b] für die Standbildkompression, dessen grundlegende Verarbeitungsschritte auch in Video-Kompressionsverfahren der ITU und MPEG-Gruppe verwendet werden.

*Standardisierungsgremien*

Während die ITU Video- und Audiokompression getrennt betrachtet, definiert die MPEG-Gruppe Methoden, um Audio- und Videodaten in einem einzigen Datenstrom unterzubringen. Da die ITU hauptsächlich im Telekommunikationsbereich tätig ist, sind ihre Empfehlungen traditionell durch Anwendungen wie Bildtelefonie und Videokonferenzen motiviert. Die MPEG-Gruppe, deren Verfahren durch die International Organization for Standardization und International Electrotechnical Commission (ISO/IEC) standardisiert werden, hat ihren Schwerpunkt stärker in den Bereichen Unterhaltungselektronik und Fernsehen. Während man diese zwei Lager anhand der ersten Veröffentlichungen noch relativ gut unterscheiden konnte, sind neuere Kompressionsverfahren der beiden Gremien sehr universell einsetzbar.

*MPEG-Gruppe und ITU im Vergleich*

Um patentrechtlichen Schwierigkeiten bei der Implementierung der MPEG-Standards entgegenzutreten, wurde die MPEG Licensing Authority (MPEG-LA) gegründet, die mit der Lizenzierung der Stan-

*MPEG Licensing Authority  
(MPEG-LA)*