
Kompressionsverfahren für Multimediadaten

Markus Kuhn, Seminararbeit, IMMD IV, Universität Erlangen-Nürnberg, 1994-06-27

Multimediasysteme sind insbesondere dadurch gekennzeichnet, daß die zu behandelnden Informationsobjekte in ihrer Ausgangsform außerordentlich viel Speicherplatz belegen. Jedoch wird vom menschlichen Benutzer ein Großteil der präsentierten Datenmenge gar nicht aufgenommen. Daher ist es reizvoll, Verfahren zu entwickeln, die nach Möglichkeit nur den für den menschlichen Benutzer relevanten Informationsanteil als Daten ablegen (Irrelevanzreduktion). Darüber hinaus wird man bestrebt sein, regelmäßige und vorhersagbare Strukturen ohne Informationsverlust zu eliminieren (Redundanzreduktion), um mit maximaler Entropie den belegten Speicherbedarf optimal auszunutzen.

Als für Kompressionsalgorithmen geeignete Quelldaten im Multimediabereich kommen insbesondere in Frage:

Audiosignale (z.B. Musik, Sprache, Geräusche). Gängige Rohdatenraten sind beispielsweise 16 bit gleichförmig quantisiert \times 44.1 kHz Abtastrate für einen Monokanal und somit 1.4112 MBit/s für ein Stereosignal höchster Übertragungsqualität (Audio CD), sowie 8 bit ungleichförmig quantisiert \times 8 kHz Abtastrate = 64 kbit/s für noch sehr gut verständliche Sprache (digitale Telefonsysteme). Charakteristisch für Audiosignale ist, daß das menschliche Ohr im wesentlichen lediglich das Energiespektrum, nicht jedoch die Phaseninformation wahrnehmen kann. Darüber hinaus tauchen Frequenzen nahe der halben Abtastfrequenz im analogen Ausgangssignal wegen den einzusetzenden Antialiasingfiltern kaum auf und physiologische Maskierungseffekte des Ohres unterdrücken weitere Informationen.

Photos. Ein Dia mit 2000×4000 pel Auflösung und 24 bit Farbauflösung benötigt etwa 23 MByte, bei klinischen Röntgenaufnahmen liegt der Speicherbedarf noch deutlich darüber.

Volumendaten. Aufnahmen, die von CT- or MRI-Anlagen im klinischen Bereich anfallen, und die in Multimediasysteme zur Lagerung von Patientendaten aufgenommen werden müssen, lassen sich algorithmisch im wesentlichen als 3-dimensionale Äquivalente von Photos behandeln. Eine Magnetresonanzaufnahme eines Kopfes mit 512^3 pel und 12 bit/pel Auflösung benötigt im Ausgangszustand z.B. 192 MByte.

Video. Ein europäisches Farbfernsehsignal hat digita-

lisiert im CCIR 601 4:2:2 Studioformat eine Auflösung von 720×576 pels im Luminanzbereich (8 bit pro pixel) und noch einmal die gleiche Datenmenge fällt für das Farbsignal an, so daß sich bei 25 Vollbildern pro Sekunde eine Datenrate von 158 MBit/s ergibt. Mit einem guten Modem läßt sich aber zum Vergleich in einem terrestrischen Fernsehband nur etwa 20 MBit/s Übertragungsrate erreichen, so daß z.B. für digitale Fernsehübertragung leistungsfähige Kompressionsverfahren unabdingbar sind.

Zeitreihen. Im wissenschaftlichen und klinischen Bereich fallen große Mengen von oft mehrkanaligen Zeitreihenmeßdaten an (z.B. Seismogramm, EKG, EEG), die archiviert und zur Visualisierung in Multimediasystemen bereitgestellt werden müssen. In diesem Bereich ist oft nur eine Redundanzreduktion möglich, da die anzuwendenden Auswertemethoden oft nicht a priori bekannt sind, und daher der irrelevante Anteil schlecht zu bestimmen ist.

Digitalisierte Schrift. Neben Photos sind gescannte Texte und Diagramme mit nur 1 bit/pel (*bilevel image*) ein weiteres visuelles Medium, das mit geeigneten Algorithmen drastisch komprimiert werden kann. Eine mit 300 dpi gescannte A4 Seite benötigt roh beispielsweise etwa 1 MByte.

Daneben müssen in Multimediasystemen evtl. auch große Mengen an Texten und CAD-Objekten behandelt werden. Für diese Datentypen kommt jedoch ausschließlich eine Redundanzreduktion (*lossless compression*) in Betracht, die weitgehend unabhängig vom konkreten Datentyp implementiert werden kann, so daß in der Regel bereits generische Text-Kompressionsverfahren (Lempel-Ziv, RLE, etc.) mit geringeren Kompressionsverhältnissen am geeignetsten sind.

1 Entropiekodierung

Multimediaobjekte lassen sich in der Regel als lange Sequenzen von Elementen eines endlichen Alphabets darstellen (z.B. Abtastwerte, Fourierkoeffizienten, Zeiger auf Tabelleneinträge, etc.). Zur Erinnerung: Die Anzahl s der Bits, die man benötigt, um ein Zeichen aus einem Alphabet A darzustellen ist – da man mit s Bits 2^s verschiedene Bitfolgen erhält – bei einer einfachen Kodierung genau $s = \lceil \log_2 |A| \rceil$, wenn man für alle Symbole gleich viele Bits benutzt und in Kauf

nimmt, daß pro s -Bit-Folge einige Kombinationen unbenutzt bleiben. Wenn die Symbole des Alphabets jedoch ungleichmäßig verteilt sind, so läßt sich diese Technik deutlich verbessern, insbesondere wenn über die Auftretenswahrscheinlichkeiten der Symbole Informationen vorliegen.

Wenn bestimmte Symbole häufiger auftreten als andere, dann ist es günstig, diesen kürzere Bitfolgen zuzuweisen als den selteneren. Im Morsekode wird diese Technik verwendet. Dort benötigen die häufigen Zeichen E und T nur ein Bit (Punkt bzw. Strich), während seltener Zeichen länger sind. Jedoch ist der Morsekode nicht mehr eindeutig zu entschlüsseln, denn die Wörter „seen“ und „Eier“ ergeben beide im Morsekode „· · · · · – ·“. Man kann, da sie durch unterschiedlich lange Bitfolgen repräsentiert werden, die einzelnen Zeichen nicht mehr eindeutig in der gesamten Bitfolge trennen. Der Morsekode erfüllt daher die sogenannte *Fano-Bedingung* nicht – er ist kein *Präfixkode* (Funkler umgehen das Problem, indem sie zwischen den Buchstaben eine kurze Pause machen).

Der bekannte *Huffman-Algorithmus* konstruiert zu jeder gegebenen Auftretenswahrscheinlichkeitsverteilung von Symbolen sehr effizient und leicht implementierbar einen optimalen Präfixkode. Bei Wahrscheinlichkeiten der Form 2^{-n} wird dabei bereits der *ideelle Entscheidungsgehalt* $-\log_2 p(a_i)$ des Symbols a_i als Kodewortlänge erreicht, bei anderen Verteilungen kann man durch Blockbildung von Zeichen im Prinzip beliebig nahe an eine Wahrscheinlichkeit der Form 2^{-n} herankommen. Auf diese Weise ist es möglich, Huffman-Kodes zu erzeugen, deren im Mittel benötigte Anzahl von Bits pro Symbol beliebig nahe an die *Entropie*

$$H = - \sum_{i=1}^{|A|} p(a_i) \log_2 p(a_i)$$

einer Symbolquelle heranreicht, was bei einer *gedächtnislosen Quelle* – bei der die einzelnen Zeichen unabhängig von vorangegangenen Zeichen auftreten – bereits der optimalen Kodierung entspricht.

Obwohl Huffman-Kodes eine optimale Kodierung von gedächtnislosen Quellen mit bekannter Symbolverteilung erlauben, ist die Blockbildung von Symbolen zum asymptotischen Erreichen der von der Entropie vorgegebenen unteren Schranke in der Praxis nicht praktikabel. Daher werden Huffman-Kodes in der Regel nur ohne Blockbildung eingesetzt, insbesondere, da sie leicht verständlich und implementierbar sind. Als Alternative zur Huffman-Kodierung mit Blockbildung bietet sich die *Arithmetische Kodierung* an, bei der prinzipiell die ganze Nachricht aus einer beliebig langen Symbolfolge in einem einzigen Kodewort abgespeichert wird. Die Arithmetische Kodierung erlaubt es,

bis auf 1 bit für beliebig lange Symbolfolgen an die Entropie als Bitanzahl heranzukommen.

Satz: Intervalle der Länge p im Bereich $[0, 1)$ der rationalen Zahlen enthalten genau eine Zahl, die in Binärschreibweise mit

$$b \leq \lceil -\log_2 p \rceil$$

binären Nachkommastellen dargestellt werden kann.

Jede Zeichenkette wird jetzt wie folgt auf ein Intervall $I \subset [0, 1)$ abgebildet: Jedem einzelnen Symbol wird ein Teilintervall der Länge $p(a_i)$ aus $[0, 1)$ zugewiesen, was wegen $\sum_{i=1}^{|A|} p(a_i) = 1$ das ganze Einheitsintervall abdeckt. I liegt in dem Intervall I_1 , daß dem ersten Symbol der Nachricht zugeordnet wurde. Dieses Intervall wird jetzt wieder in $|A|$ Teilintervalle der Länge $|I_1| p(a_i)$ aufgeteilt, usw. Am Ende wurde also die gesamte Nachricht einem Intervall zugeordnet, dessen Länge exakt der Auftretenswahrscheinlichkeit dieser Nachricht entspricht. Obiger Satz sagt aus, daß nun eine Binärfolge ausgewählt werden kann, die eine Zahl aus diesem Intervall repräsentiert und die um nicht mehr als 1 bit länger ist als der Entscheidungsgehalt der Nachricht. Zur eindeutigen Dekodierbarkeit muß die Nachricht noch mit einem Ende-Marker versehen werden, und einige wenige zusätzliche Bits müssen eingefügt werden, damit die Kodierung effizient auf Prozessoren mit üblicher Wortgröße möglich ist (z.B. mit dem multiplikationslosen Rissanen-Mohiuddin-Algorithmus). Aber auch dann liegt bei längeren Nachrichten die Anzahl der benötigten Bits um kaum mehr als 1% über dem theoretischen Minimum.

2 Dekorrelation

Durch eine Entropiekodierung kann zwar bereits viel Redundanz eines Nachrichtenstromes eliminiert werden, jedoch gehen Entropiekoder (Huffman, arithmetische Kodierung) immer davon aus, daß Symbole völlig unabhängig voneinander auftreten. Es wird also nicht berücksichtigt, daß die Auftretenswahrscheinlichkeit eines Zeichens von z.B. seinen Nachbarsymbolen abhängig ist ($p(a_t) \neq p(a_t | a_{t-1} a_{t-2} \dots a_1)$, z.B. Markov-Quelle). Daher ist es zweckmäßig, vor der Entropiekodierung zunächst die Werte (Symbole) zu dekorrelieren, um die Entropie des Signales zu verringern.

2.1 Prädiktionskodierung

Allgemein wird bei der Prädiktionskodierung ein Modell benutzt, das den nächsten zu kodierenden Wert anhand der vorangegangenen Werte vorhersagt. Kodiert wird dann nur noch der Prädiktionsfehler. Auf

der Seite des Dekoders wird das gleiche Prädiktionsmodell benutzt, und der Prädiktionsfehler wird auf den Vorhersagewert addiert, um den Originalwert zu rekonstruieren. Im einfachsten Fall (DPCM) wird als Vorhersagewert für eine zu kodierende Zeitreihe nur der vorangegangene Wert benutzt, so daß die kodierten Werte die Differenzen zwischen den Werten des Ausgangssignales sind. Möglich ist auch, aus den letzten beiden Werten a_{t-2} und a_{t-1} den Folgewert durch lineare Extrapolation zu ermitteln: $a_t = 2a_{t-1} - a_{t-2} + \Delta_t$, wobei Δ_t der übertragene Prädiktionsfehler ist.

Dieses Prinzip läßt sich zur linearen Prädiktion verallgemeinern, bei der der nächste Wert einer Zeitreihe vom Dekoder als Linearkombination der letzten N Werte mit den N Prädiktionskoeffizienten c_i ($1 \leq i \leq N$) ermittelt wird:

$$a_t = \sum_{i=1}^N a_{t-i} c_i + \Delta_t.$$

Übertragen werden dann nur die Prädiktionsfehler

$$\Delta_t = a_t - \sum_{i=1}^N a_{t-i} c_i$$

und der Dekoder kann anhand der Anfangssequenz a_1, \dots, a_N , der c_i und der Δ_t die volle Sequenz wieder rekonstruieren. Die optimalen Prädiktionskoeffizienten können durch Lösen eines Gleichungssystems bestimmt werden. Oder man faßt die c_i als Impulsantwort eines kausalen (d.h. nur Werte aus der Vergangenheit berücksichtigenden) FIR (*finite impulse response*) Filters auf, das auf das Spektrum des übertragenen Signales abgestimmt wird. Ein Enkoder mit linearer Prädiktion ist nichts anderes als ein Filter, der alle Frequenzanteile so verstärkt bzw. abschwächt, daß alle Frequenzen mit der gleichen Amplitude am Ausgang auftreten. Das Ergebnissignal (der übertragene Prädiktionsfehler Δ_t) ist daher ein weißes Rauschen und hat damit maximale Entropie. Die geeigneten c_i sind also hochgradig vom Spektrum des Signales abhängig, weshalb bei Signalen mit schnell wechselnden Spektren regelmäßig neue Prädiktionskoeffizienten mitkodiert werden müssen. Bei einigen für Sprachübertragung optimierten verlustbehafteten Kompressionsverfahren geht man sogar soweit, daß der Prädiktionsfehler Δ_t garnicht mehr übertragen wird, sondern nur z.B. 20–50 mal pro Sekunde die neuen z.B. $N = 10$ Prädiktionskoeffizienten kodiert werden. Das erzeugte Spektrum ähndelt sehr dem des Ausgangssignales, und die verlorene Phaseninformation wird vom menschlichen Ohr ohnehin nicht wahrgenommen.

2.2 Transformationskodierung

Neben der Prädiktionskodierung, bei der immer ein Wert abhängig von seinen Vorgängern vorhergesagt wird und dann nur die viel unkorreliertere Vorhersagedifferenz übertragen wird, gibt es auch noch die Möglichkeit, gleich einen ganzen Werteblock durch eine invertierbare (vorzugsweise lineare) Transformation in einen weniger korrelierten Datenblock zu verwandeln.

Zur Erinnerung (siehe auch Bronstein, Kap. 5.1.4): Zwei Zufallsvariablen X und Y werden als *dekorreliert* bezeichnet, wenn

$$\text{cov}(X, Y) = E[(X - EX)(Y - EY)] = 0$$

ist. Zwei Zufallsvariablen werden als *unabhängig* bezeichnet, wenn

$$p(X = x \wedge Y = y) = p(X = x)p(Y = y), \quad \forall x, y.$$

Aus der Unabhängigkeit folgt die Dekorreliertheit, jedoch nicht umgekehrt. Kodierungstheoretisch ist es optimal, wenn die an den Entropiekoder gegebenen Werte unabhängig sind, da damit keine redundante Information mehr übertragen wird, jedoch ist es in der Praxis schwierig und zum Glück auch meist unnötig, Transformationen zu finden, die die Unabhängigkeit der Ausgangsgrößen erzielen. Daher begnügt man sich damit, die Dekorrelation als Näherung der Unabhängigkeit zu erreichen.

Sei eine Sammlung von $M \times N$ Pel großen Schwarz-Weiß-Bildern mit den Bildpunkthelligkeiten h_{ij} ($1 \leq i \leq M, 1 \leq j \leq N$) repräsentiert durch einen Zufallsvektor

$$X = (h_{11}, h_{12}, \dots, h_{MN}) \in \mathbb{R}^{MN}.$$

Sei weiterhin C die Kovarianzmatrix dieses Zufallsvektors, und sei

$$C = \Phi D \Phi^T$$

die Zerlegung der symmetrischen Matrix C in eine Orthonormalmatrix Φ und eine Diagonalmatrix D (Hauptachsentransformation). Dabei sind die Spaltenvektoren in Φ die Eigenvektoren und die Diagonalelemente d_{ii} in D die Eigenwerte. Sei darüber hinaus die Zerlegung so gewählt, daß die Eigenwerte der Größe nach sortiert sind: $d_{11} \geq d_{22} \geq \dots$ (durch Multiplikation mit geeigneter Permutationsmatrix). Dann gilt:

$$G = \Phi X$$

ist ein neuer Zufallsvektor, dessen Komponenten alle dekorreliert und der statistischen Signifikanz nach geordnet sind. Diese Transformation wird als *Karhunen-Loève Transformation* oder als Kovarianztransformation bezeichnet. Die Komponenten des Vektors G enthalten die gleiche Information wie die des Vektors X

($X = \Phi^{-1}G = \Phi^T G$), jedoch falls die Komponenten von X stark korreliert waren, haben die Komponenten von G eine deutlich geringere Entropie (d.h. weniger Bits werden vom Entropiekoder produziert). Außerdem sammelt sich die Energie des Signales in den ersten Komponenten von G , so daß die hinteren Komponenten nach der Quantisierung sehr oft nur Null sind und daher nicht übertragen werden müssen.

In der Praxis ist die Karhunen-Loève Matrix Φ mit ihren $M^2 N^2$ Elementen für ein ganzes Bild jedoch viel zu groß. Daher wird man das Bild in kleine Blöcke von z.B. 8×8 Bildpunkten zerlegen, was Φ auf die handhabbare Größe 64×64 reduziert.

Diese Methode ist aber immer noch mit einer Reihe von praktischen Nachteilen behaftet:

- Die Transformation hängt von der Auswahl von Bildern ab, die zur Bestimmung der Kovarianzmatrix herangezogen wurden.
- Die Matrix ist allgemein nicht separierbar in eine Nacheinanderausführung von kleineren Teiltransformationen.
- Da Φ im Prinzip eine fast beliebig aussehende Orthonormalmatrix ist, kann man keinen schnellen Transformationsalgorithmus angeben, wie er z.B. bei der Fourier- oder Hadamard-Transformation möglich ist.

Daher wurden eine Reihe von Ersatztransformationen, die nicht mit den obigen Nachteilen behaftet sind vorgeschlagen. Insbesondere die 1974 von Ahmet, Nataraajan und Rao vorgeschlagene *Diskrete Cosinus Transformation (DCT)* zeichnet sich durch eine große Ähnlichkeit zu anhand von vielen Beispielbildern ermittelten Karhunen-Loève Transformationen aus. Darüber hinaus existieren inzwischen schnelle auch in Hardware realisierbare Algorithmen für die DCT und eine 2-dimensionale 8×8 DCT läßt sich in 8 horizontale und 8 vertikale hintereinanderausgeführte 1-dimensionale 8-Punkte DCTs zerlegen. Die orthonormale Transformationsmatrix $T = (t_{ij})$ der gängigsten Form der 1-dimensionalen 8-Punkt DCT lautet:

$$t_{ij} = \frac{C_j}{2} \cos \frac{(2i+1)j\pi}{16}$$

mit $C_j = 1/\sqrt{2}$ für $j = 0$ und $C_j = 1$ sonst.

Für die DCT wurde 1988 von Arai, Agui und Nakajima ein sehr schneller Algorithmus vergleichbar der FFT veröffentlicht. Bei einer 8-Punkte Transformation benötigt dieses Verfahren bis auf Skalierung des Ergebnisses nur 5 Multiplikationen und 29 Additionen. Die Faktoren, mit denen die Ergebnisse noch skaliert sind können gleich in die Quantisierungsfaktoren mit eingebracht werden. Zum Vergleich dazu sind bei einer

klassischen Matrixmultiplikation 64 Multiplikationen und 56 Additionen notwendig.

Die bekannte diskrete Fouriertransformation (DFT) ist der DCT sehr ähnlich, hat jedoch den Nachteil, daß bei ihr der erste und letzte Wert des zu transformierenden Vektors benachbart sind, da die DFT aus der kontinuierlichen Fouriertransformation durch periodische Fortsetzung des endlichen Signales entstanden ist. Dies kann bei der Bildkodierung zu sichtbaren Blockeffekten führen. Die DCT entsteht aus der DFT dadurch, daß die 8 Werte gespiegelt werden, so daß nun 16 Werte vorhanden sind. Diese werden mittels einer DFT transformiert. Wegen der Symmetrie fallen alle imaginären Koeffizienten weg, so daß die 8 reellen Koeffizienten genau dem Ergebnis der DCT entsprechen. Durch die Spiegelung ist jetzt der erste und der letzte Wert zu sich selbst benachbart, was bei Veränderungen der Koeffizienten im Frequenzbereich (z.B. durch Quantisierung) nicht mehr zur zyklischen Faltung führt.

3 Der JPEG Standard

Ende der 80er Jahre wurde basierend auf den eben beschriebenen Techniken ein Standard für ein hocheffizientes verlustbehaftetes Kompressionsverfahren für Photos entwickelt. Diese Arbeiten wurden bei den internationalen Normungsgremien ISO, IEC und ITU durchgeführt und die entsprechende Expertengruppe wurde unter der Bezeichnung *Joint Photographic Expert Group (JPEG)* bekannt. Ziel von JPEG war es, einen Standard für Algorithmus und Datenformat zu entwickeln, der in seinem Kompressionsverhältnis für qualitativ hochwertige Ergebnisse dem derzeitigen Stand der Technik entspricht. Die Anforderungen waren unter vielen anderen insbesondere:

- Farbige Bilder (digitale Fernsehstudioqualität) sollten bei 0.25 bit/pel (Kompressionsverhältnis 1:100) noch gut erkennbar, bei 0.75 bit/pel (1:30) von sehr hoher Qualität und bei 3.00 bit/pel (1:8) mit dem geschulten Auge nicht mehr vom Original zu unterscheiden sein.
- Der Algorithmus sollte so effizient sein, daß ein 16 MHz i386 PC etwa mit 8 kbyte/s JPEG Daten dekomprimiert kann.
- Der Algorithmus und das Datenformat sollten so universell wie möglich die Anforderungen einer Vielzahl von Anwendungen erfüllen.

Der Grundalgorithmus für den man sich dann nach vielen Tests entschied beinhaltet folgende Einzelschritte:

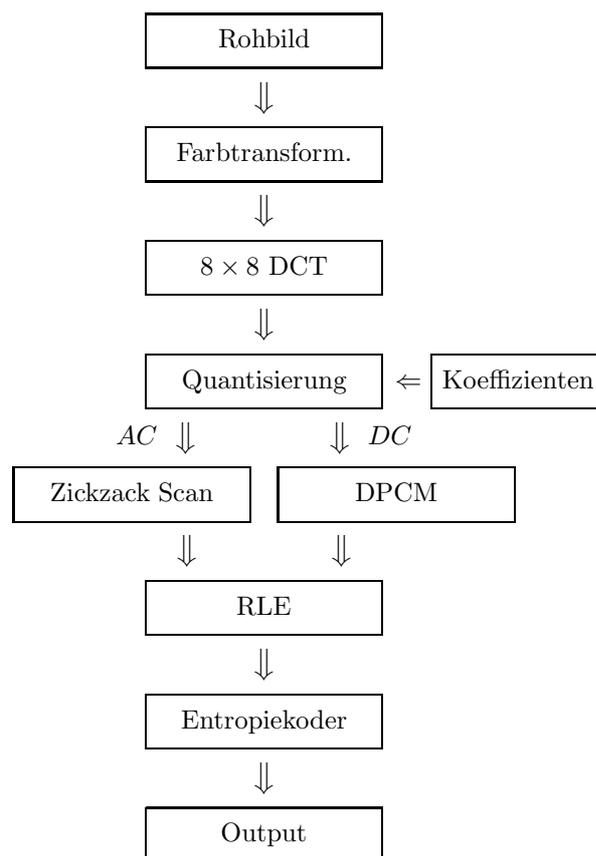
- Zunächst wird wie auch beim Farbfernsehen ausgenutzt, daß das menschliche Auge Helligkeitskontraste mit etwa einer doppelt so hohen

Auflösung wahrnimmt wie Farbkontraste. Daher werden die Farbkoordinaten falls sie noch im RGB Raum vorliegen in den (beim Digitalfernsehen gebräuchlichen) YCrCb Farbraum transformiert. Das Helligkeitsbild Y wird in der vollen, die Farbkomponentenbilder Cr und Cb in der halben Auflösung kodiert. Hier findet der erste Informationsverlust statt. Der eigentlich erst anschließend folgende JPEG Algorithmus betrachtet die Farbkomponenten völlig unabhängig voneinander wie einzelne Grauwertbilder. JPEG wird daher als farbenblind bezeichnet.

- Anschließend werden die Komponentenbilder in Blöcke der Größe 8×8 zerlegt. Diese Blöcke werden mit einer DCT in den Frequenzbereich transformiert. Die Ausgangsdaten der DCT können in 8- oder 12-bit Auflösung vorliegen, die DCT ist so spezifiziert, daß bei 8-bit Auflösung mit 16-bit Integerarithmetik gearbeitet werden kann.
- Nun folgt der Schritt, bei dem wirklich die Informationen verloren gehen. An dieser Stelle kann auch das Kompressionsverhältnis frei gewählt werden. Die 64 Ergebniskoeffizienten jedes DCT Blockes werden durch einen Quantisierungskoeffizienten geteilt und auf eine ganze Zahl gerundet. Es werden 64 Quantisierungskoeffizienten eingesetzt, so daß jede Frequenz mit einer eigenen Auflösung übertragen werden kann. Die Auswahl dieser 8×8 Quantisierungskoeffizientenmatrix – die im Datenstrom am Anfang mit übertragen wird – entscheidet über das Kompressionsverhältnis und die resultierende Bildqualität. Durch die Division tauchen häufiger sehr kleine Werte auf, die Entropie wird also verringert. Bei einer Division durch 4 gehen z.B. 2 Bits verloren.
- Zur weiteren Reduktion der Entropie wird zusätzlich der DC („Gleichstrom“) Koeffizient, der bei jedem Block die mittlere Helligkeit angibt mit einer eindimensionalen DPCM Kodierung verarbeitet. Es wird also nur die Differenz zum vorangegangenen DC Koeffizienten verwendet.
- Um von einer 8×8 Matrix zu einem sequentiellen Symbolstrom für den Entropiekoder zu kommen werden die Blöcke nicht etwa zeilenweise, sondern in einem Zickzack-Muster ausgelesen. Dadurch stehen am Anfang jeder Blocksequenz die DC Koeffizienten und die niedrigen Frequenzen mit in der Regel hohen Werten. Am Ende der Blocksequenz sammeln sich dagegen die hohen Frequenzanteile, die oftmals nur Null sind.
- Die Zickzack-Anordnung der Koeffizienten erreicht, daß im Symbolstrom oft viele Nullen aufeinanderfolgen. Bei der *run-length encoding* werden aufeinanderfolgende Nullen durch ein Symbol

ersetzt, das sowohl die Anzahl der Nullen als auch den Wert des nächsten Nichtnullsymbols enthält. Ein weiteres Blockendesymbol erlaubt es, sehr effizient zu kennzeichnen, wenn in der Sequenz bis zum Ende nur noch Nullen folgen.

- Die Koeffizienten und die speziellen RLE-Symbole werden abschließend an einen Entropiekoder gegeben. Es stehen sowohl ein Huffmankoder mit fester Tabelle die mit abgespeichert werden muß, als auch ein adaptiver arithmetischer Koder zur Auswahl. Hauptsächlich auf Grund von U.S. Patenten für arithmetische Koder wird heute fast ausschließlich der etwa 10% schlechtere Huffmankoder eingesetzt. Der Huffmankode besteht nicht aus einem eigenen Kodewort für jedes Symbol (das würde die entsprechende Tabelle sehr lang werden lassen). Die Kodewörter die mit dem Huffman-Algorithmus gewonnen werden sind Präfixe, die ein Werteintervall auswählen, die letzten Bits bestimmen dann, welcher Wert im Intervall gemeint ist.



Für Anwendungen, bei denen es nötig ist, daß vor Betrachtung eines hochauflösenden Bildes in voller Qualität schnell mit wenig Datenübertragungsaufwand eine Grobansicht des Bildes erzeugt werden kann, stellt

der JPEG Standard neben dem normalen sequentiellen (*baseline*) Modus noch zwei besondere Modi zur Verfügung, die es erlauben, ein Bild etappenweise mit kontinuierlich besserer Auflösung/Qualität zu übertragen:

- Im *progressive mode* haben die einzelnen Etappen (*scans*) des Bildaufbaues die gleiche Auflösung, mit jedem Scan werden nur die DCT Koeffizienten besser approximiert. Dabei kann ausgewählt werden, ob die Aufteilung der Informationen auf die einzelnen Scans nach Koeffizienten (*spectral selection*) oder nach Bits pro Koeffizient (*successive approximation*) erfolgen soll. Dabei wird man zuerst die tiefen Frequenzen, bzw. die höchstwertigen Bits übertragen. Im progressive mode ist JPEG nicht oder kaum weniger effizient als im baseline mode.
- Im *hierarchical mode* wird zuerst ein komplettes Bild in niedrigerer Auflösung übertragen. Bei jedem nachfolgenden Scan wird dann das zuerst übertragene Bild auf die neue Auflösung vergrößert und anschließend nur noch die Differenz zu diesem Bild übertragen, die dann eine geringere Entropie hat als das gesamte Bild in der neuen Auflösung. Dennoch sind insgesamt im hierarchical mode mehr Bits nötig als im baseline oder progressive mode.

Darüber hinaus definiert der JPEG Standard noch einen *lossless mode* mit bis zu 16-bit/pel Auflösung. Im lossless mode wird garantiert, daß jedes Bit des wiederhergestellten Bildes identisch ist mit dem des Ausgangsbildes. Dafür werden bei typischen Photos aber nur etwa Kompressionsverhältnisse von 1 : 2 erzielt. Der lossless mode wird insbesondere im wissenschaftlichen und medizinischen Bereich eingesetzt, wo die Ergebnisbilder nicht nur von menschlichen Augen, sondern auch von Bildverarbeitungssystemen untersucht werden. Da sich die DCT nicht ohne weiteres hinreichend exakt (d.h. das genaue Rundungsverhalten) spezifizieren läßt, wird statt ihr im lossless mode eine einfache Auswahl von linearen Prädiktoren benutzt, die anhand von drei bereits zuvor dekodierten Nachbarbildpunkten einen Schätzwert liefern. Zur Entropiereduktion wird dann nur noch die Differenz zu diesem Schätzwert zusammen mit dem benutzen Prädiktortyp kodiert.

Hier wurde ein JPEG Encoder beschrieben. Der Dekoder wendet exakt alle Schritte rückwärts an (Multiplikation statt Quantisierung, inverse DCT, etc.) und unterscheidet sich in der Komplexität und im Rechenaufwand kaum vom Encoder. Daher wird JPEG auch als symmetrisches Kompressionsverfahren bezeichnet.

4 Der MPEG Video Standard

Ermutigt durch den Erfolg der JPEG Arbeitsgruppe beschloß ISO, auch ein Normierungsprojekt für Bewegtbild- und Audiokompression zu starten. So wurde die *Motion Pictures Expert Group (MPEG)* ins Leben gerufen. Als Ergebnis existiert bereits ein Videokompressionsstandard MPEG-1 mit VHS Qualität für normale CD-ROM Datenübertragungsrate (1.5 Mbit/s). Im Herbst 1994 wird voraussichtlich der MPEG-2 Standard verabschiedet, der digitale Fernsehstudioqualität im Bereich 8 Mbit/s liefert und der auch für hochauflösendes Fernsehen einsetzbar ist. Alle derzeit weltweit (Europa, USA, Japan) laufenden Projekte zur Einführung neuer digitaler Fernsehstrahlungsnormen setzen – trotz aller anderen Inkompatibilitäten bei Modems, Frequenzen und Auflösungen – MPEG-2 als Quellkodieralgorithmus ein, so daß diese Technik in naher Zukunft sicher sehr große Bedeutung auch bei Multimedia-Computeranwendungen gewinnen wird.

4.1 MPEG Video

Neben den Eigenschaften von einzelnen Photos besteht auch zwischen aufeinanderfolgenden Bildern eines Filmes eine sehr hohe Korrelation. Daher kann man bei Videodaten deutlich mehr erreichen, als nur JPEG auf Einzelbilder anzuwenden (eine als Motion-JPEG bezeichnete Technik). Zusätzlich zur Korrelation mit den zeitlichen Vorgänger- und Nachfolgebildpunkten kommt beim Film noch hinzu, daß durch Bewegungen und Kameraschwenks sehr viel Entropie dadurch reduziert werden kann, daß Bewegungen erkannt und berücksichtigt werden.

Der MPEG Algorithmus basiert im wesentlichen auf den gleichen Grundtechniken wie JPEG (YCrCb Farbraum, 8×8 DCT, Quantisierung, etc.), jedoch wird nur der Huffman-Algorithmus statt dem arithmetischen Koder benutzt, und dieser hat nur eine feste Kodeworttabelle. Darüber hinaus ist es bei MPEG möglich, pro 8×8 Block zwischen mehreren Quantisierungstabellen zu wählen (*adaptive quantization*). Da alleine die adaptive quantization einen bedeutenden Kompressionsgewinn ermöglicht hat, wird derzeit auch der JPEG Standard um diese erweitert, was dann jedoch den Encoder komplexer als den Dekoder macht, da die richtige Tabelle ausgesucht werden muß.

Mit diesem leicht modifizierten JPEG Algorithmus werden drei verschiedene Arten von Einzelbildern kodiert:

- Die *I-frames* (intra) sind völlig eigenständige Bilder, die unabhängig von anderen Bildern ähnlich wie einzelne JPEG Bilder kodiert werden. *I-frames* tauchen regelmäßig in MPEG Datenströmen auf, damit nach einem Sprung mitten in

einen Film der Dekoder an einem *I*-frame aufsetzen kann, ohne die vorangegangenen Bilder bearbeiten zu müssen. *I*-frames werden auch bei Szenenwechseln benötigt.

- Die *P*-frames (predicted) sind Einzelbilder, die lediglich die Differenz zum vorangegangenen Bild enthalten.
- Die *B*-frames (bidirectional) werden ausgehend von einem vorangegangenen und einem zukünftigen Bild vorhergesagt. Sie enthalten also nur die Differenz zu einem interpolierten Schätzbild.

Ein MPEG Datenstrom besteht in erster Linie aus einer Sequenz von *groups-of-pictures* (*GOP*). Jede *group-of-pictures* enthält genau einen *I*-frame als erstes Bild. Eine typische *GOP* besteht etwa aus der folgenden Anordnung von Bildtypen:

$$I_0 B_1 B_2 B_3 P_4 B_5 B_6 B_7 P_8 B_9 B_{10} B_{11} P_{12}$$

Der *I*-frame ist unabhängig dekodierbar, die *P*-frames beziehen sich auf den vorangegangenen *I*- oder *P*-frame und die *B*-frames beziehen sich auf den vorangegangenen und/oder nachfolgenden *I*- oder *B*-frame. Da ein Dekoder *B*-frames nur dekodieren kann, wenn er den nachfolgenden referenzierten *P*-frame kennt, müssen die Bilder in einer geeigneten Reihenfolge abgespeichert werden, die nicht der Reihenfolge der Darstellung entspricht:

$$I_0 P_4 B_1 B_2 B_3 P_8 B_4 B_5 B_6 B_7 P_{12} B_8 B_9 B_{10} B_{11}$$

Die Prädiktion in *P*- und *B*-frames erfolgt Bewegungskompensiert. Dazu wird jedes Bild in Macroblocks bestehend aus 16×16 Bildpunkten, d.h. 1 Cr, 1 Cb und 4 Y Blöcke, eingeteilt. Bei jedem Macroblock in *P*- und *B*-frames ist getrennt vermerkt, ob er völlig neu kodiert ist oder nur die Differenz zum Vor- und/oder Nachfolger-*P*/*I*-frame enthält. Sowohl für das Vorgänger- als auch für das Nachfolgebild kann angegeben werden, um wieviel dort der Macroblock verschoben werden muss, bevor interpoliert wird. Durch diese Verschiebung kann ein entsprechend leistungsfähiger Encoder Bewegungen kompensieren.

Bei der Auswahl der Bewegungskompensationsvektoren, bei der Auswahl der Quantisierungstabelle und bei der Festlegung pro Macroblock, relativ zu welchem Frame interpoliert werden soll kann der Encoder sehr viel mehr Aufwand investieren als für einen Dekoder notwendig ist. MPEG ist daher ein asymmetrisches Verfahren, und die Bildqualität hängt stark vom im Encoder getroffenen Aufwand ab. Der Standard definiert nur die Funktionalität eines Dekoders, so daß bei den Encodern große qualitative Freiräume für den

Hersteller bestehen. Ein Encoder kann in einem *variable bit rate* Modus arbeiten, bei dem die Quantisierungstabellen konstant sind und die anfallende Datenrate von der Komplexität des Bildes abhängt. Da dies bei einer vorgegebenen Sendebandbreite oder Magnetbandgeschwindigkeit nicht praktikabel ist, kann ein Encoder auch im *constant bit rate* Modus iterativ die Quantisierungskoeffizienten solange variieren, bis die für eine Anzahl von Frames zur Verfügung stehende Datenmenge gerade ausgeschöpft ist.

Beim MPEG-2 Standard kommen jetzt u.a. noch Möglichkeiten der getrennten Behandlung von geraden und ungeraden Zeilen hinzu, um mit *interlaced* Quelldaten zurechtzukommen. Ferner besteht ähnlich dem sequential/hierarchical mode in JPEG die Möglichkeit, auf einem Kanal ein (z.B. besser gesichertes) grobes Bild zu übertragen und auf einem zweiten Kanal mit einer anderen *quality of service* das Differenzsignal zum qualitativ hochwertigen Bild (Skalierbarkeit). Dadurch wird es z.B. möglich, bei digitalem Fernsehen zusätzlich zu den normalen Fernsehsignalen noch die Signale für ein hochauflösendes $16 : 9$ Format (HDTV) auszustrahlen, ohne daß Bandbreite doppelt genutzt wird. Einfache und preiswerte Empfangsgeräte können dann nur die normale Auflösung darstellen, während teurere Geräte die volle Auflösung mit entsprechend mehr Speicher und Verarbeitungsaufwand dekodieren können.

Neben dem MPEG Standard existiert noch der $p \times 64$ Standard, der für Bildtelefon- und Videokonferenzanwendungen vorgesehen ist. Er ist vergleichbar mit einer einfachen Version des MPEG Standards. Die Ausgangsdaten liegen im *Common Intermediate Format* (*CIF*) mit einer Luminanzauflösung von 352×288 Bildpunkten, der halben Chrominanzauflösung und einer Bildrate von etwa 10 Hz vor. Der Encoder vergleicht mittels einer zusätzlichen inversen DCT permanent das übertragene verlustbehaftete Bild mit den Ausgangsdaten und überträgt die Differenz. Durch diese Rückkopplung konvergiert bei ruhigem Bild das Dekodierergebnis in mehreren Schritten, während bei häufigen Bildänderungen nur der grobe Bildinhalt übertragen werden kann. Der $p \times 64$ Standard ist wie der Name andeutet für die Bandbreite eines oder mehrerer 64 kbit/s ISDN B-Kanäle ausgelegt. Durch den Verzicht auf die bidirektionale Interpolation wird eine kürzere Verzögerung im Vergleich zu MPEG erreicht, was bei Bildtelefonanwendungen wichtig ist. Wie bei MPEG wird auch eine 16×16 -Macroblock-Bewegungskompensation eingesetzt.

4.2 MPEG Audio

Da MPEG für komplette Fernsehsignale konzipiert wurde, müssen neben Bewegtbildern auch noch Au-

diosignale und Zusatzdaten (z.B. Videotext) mit in den Datenstrom kodiert werden.

Der MPEG-1 Standard definiert drei verschiedene als Levels bezeichnete Audiokodierverfahren. Sie unterscheiden sich im Implementationsaufwand und im erreichbaren Kompressionsverhältnis. Die Ausgangsdaten liegen dabei gleichförmig mit 16-bit quantisiert in den Abtastraten 48 kHz, 44.1 kHz oder 32 kHz vor.

Bei der Audiokodierung wird eine physiologische Eigenschaft des menschlichen Ohres ausgenutzt. Die Hörschwelle, d.h. der Schalldruck, ab dem ein Ton einer bestimmten Frequenz wahrgenommen werden kann, hängt nicht nur von der Frequenz ab, sondern auch von der Lautstärke der Töne in benachbarten Frequenzen. Neben einem lauten Ton kann aufgrund der lateralen Hemmung der Neuronen in der Cochlea (Gehörschnecke) ein leiserer Ton nicht mehr wahrgenommen werden, der alleine noch deutlich zu hören wäre.

Daher wird bei den MPEG Audiokodierverfahren jeweils das Signal zunächst in einer Filterbank mit digitalen FIR Filtern in 32 Frequenzteilbänder zerlegt. Ein psychoakustisches Modell schätzt für jedes Teilband die Maskierungsschwelle ab, unter der Töne in diesem Frequenzband nicht mehr wahrgenommen werden können. Für jedes Band werden die Quantisierungskoeffizienten dann entsprechend den Maskierungsschwellen festgelegt, so daß das durch die Quantisierung entstehende Störsignal unter der Maskierungsschwelle liegt. Vor der Quantisierung werden die Teilbänder noch so auf gleiche Pegel skaliert, daß die notwendige Dynamik erhalten bleibt. Diese Skalierungsfaktoren müssen ebenfalls mit den Quantisierungskoeffizienten als „Seiteninformation“ zusammen mit den quantisierten Teilbanddaten übertragen werden.

Das Layer I Verfahren ist das einfachste Verfahren, das bei 384 kbit/s (1:4) für ein Stereosignal eine der Compact Disc vergleichbare Qualität liefert. Das Eingangssignal wird in Blöcken zu je 384 Abtastwerten (bei 48 kHz sind das 8 ms) verarbeitet. Die Filterbank zerlegt das Signal in 32 Teilbänder, die jeweils 750 Hz breit sind. Dabei entstehen 12 zeitliche Abtastwerte pro Block und Teilband. Der maximale dieser 12 Abtastwerte bestimmt den 6-bit Skalierungsfaktor für diesen Block in diesem Teilband. Anschließend wird auf die skalierten Werte je nach Auswahl durch das psychoakustische Modell einer von 14 verschiedenen gleichförmigen Quantisierern angewandt. Das Ergebnis wird ohne Entropiekodierung zusammen mit der Seiteninformation aus Skalierungsfaktoren und Quantisierertyp sowie mit einer CRC Prüfsequenz zur Fehlermaskierung abgespeichert.

Der Layer II Encoder ist dem Layer I Encoder sehr ähnlich, jedoch komplexer und hat eine bessere Kodie-

rungeffizienz. CD Stereoqualität wird noch bei 256 kbit/s erreicht, und bei 192 kbit/s kommt er immer noch der CD Qualität sehr nahe, aber es sind dann bereits bei sehr kritischen Signalen Störungen wahrnehmbar. Die Abtastblöcke sind $3\times$ so lange wie im Layer I (1152 Werte). Die Skalenfaktoren werden immer noch in Gruppen zu 12 Abtastwerten bestimmt, aber diese werden in Abhängigkeit von den anderen beiden Gruppen eines Blockes effizienter kodiert. Das psychoakustische Modell ist aufwendiger als das des Layer I und benutzt intern eine 1024-Punkte Fouriertransformation statt einer 512-Punkte FFT bei Layer I. Es wird auch bei Layer II kein Entropiekoder eingesetzt, aber die quantisierten Abtastwerte werden durch Blockbildung und Bitumsortierung effizienter kodiert.

Beim Layer III Verfahren wird der größte Aufwand betrieben, aber auch das beste Kompressionsverhältnis erzielt. Er verhält sich bei den Bitraten 256 kbit/s und 192 kbit/s dem Layer II Verfahren vergleichbar, erzielt aber bei 128 kbit/s (entsprechend 64 kbit/s mono, 1 : 12) die beste heute auf ISDN B-Kanälen erreichbare Audioqualität. Bei dieser Bitrate ist die Qualität der der CD meistens sehr nahe, jedoch sind bei einigen kritischen Testsignalen die Störgeräusche deutlich wahrnehmbar.

Die Blockgröße entspricht der des Layer II. Anschließend werden jedoch nicht die 36 zeitlichen Abtastwerte der Filterbank direkt skaliert, sondern eine modifizierte DCT sorgt noch für eine verbesserte Spektralzerlegung (18 Spektralkoeffizienten pro Filterband, d.h. insgesamt $32 \times 18 = 576$ Werte). Bei perkussiven Signalen wird jedoch statt der besseren Spektralzerlegung eher eine bessere zeitliche Auflösung benötigt, was durch eine Umschaltung auf nur 6 Spektralkoeffizienten aus der DCT erreicht werden kann. Bei der Quantisierung werden nicht einfache gleichmäßige Quantisierer eingesetzt, sondern eine Analyse-durch-Synthese Technik. Dabei überprüft das psychoakustische Modell das entstandene Störsignal, und paßt die Quantisierung entsprechend an. Die quantisierten Werte, nicht aber die Seiteninformation werden mit einem Huffmankoder entropiekodiert. Wie bei den anderen Layers erlaubt eine CRC Sequenz eine Fehlerkorrektur oder -maskierung. Das psychoakustische Modell des Layer III Encoders ist das leistungsfähigste der drei Verfahren. Es basiert auf einer 1024-Punkte Fouriertransformation, die überlappend angewandt wird.

5 Kompression von Bilevel-Bildern

Wie bereits Eingangs erwähnt, besteht auch ein großer Bedarf für die effiziente und verlustfreie Speicherung von 1-bit Rastergrafiken, wie sie z.B. bei gescannten

Texten und Zeichnungen anfallen. Das derzeit verbreitetste Verfahren ist der T.4 Standard, der in jeder Fax-Maschine eingesetzt wird. Die Seiten werden beim Telefax mit 100 oder 200 dpi (*dots per inch*) digitalisiert, bei anderen Anwendungen wie etwa Dokumentenarchiven auf optischen Platten werden auch 300 und 600 dpi benutzt. Beim T.4 Fax-Standard wird das Bild zeilenweise abgetastet und laulängenkodiert, d.h. die dabei entstehenden Symbole haben Bedeutungen wie etwa *12-weiße-Punkte* und *3-schwarze-Punkte*. Diese Laulängensymbole werden anschließend mit einer fest vordefinierten Huffmantabelle entropiekodiert. Die Huffmantabelle wurde dabei für 8 Testseiten (kurzer Geschäftsbrief, technische Zeichnung, Wetterkarte, usw.) bei einer Auflösung von 100 dpi optimiert.

Inzwischen wurde (was ursprünglich im Rahmen des JPEG Projektes geplant war) von ISO eine neue Gruppe mit dem Namen *Joint Bilevel Image Group (JBIG)* für die Entwicklung eines Nachfolgers des Fax-Standards ins Leben gerufen. Obwohl das T.4 Verfahren sehr effizient ist, existieren inzwischen bessere Verfahren, insbesondere für höhere Auflösungen, und für geditherte Bilder, bei denen das Fax-Verfahren große Schwierigkeiten hat.

Auch der JBIG Standard (ISO 11544) tastet das Bild zeilenweise ab. Für einen Bildpunkt wird während der Kompression statistisch die bedingte Wahrscheinlichkeit dafür ermittelt, daß er schwarz ist. Die bedingte Wahrscheinlichkeit hängt dabei von 10 Nachbarpels überhalb und links des zu kodierenden neuen Bildpunktes ab. Es stehen verschiedene Anordnungen von 9 Nachbarpunkten zur Auswahl. Ein 10. Nachbarpunkt ist frei in gewissen Grenzen verschiebbar. Er dient in erster Linie dazu, zu ermöglichen, daß das Nachbarschaftsmuster sich über eine Periode eines Dithermusters ausdehnen kann. Bei 10 Bildpunkten, die zum Nachbarschaftskontext gehören gibt es $2^{10} = 1024$ verschiedene Kontexte und entsprechend viele bedingte Wahrscheinlichkeiten dafür, daß dieses Pixel schwarz ist. Ausgehend von der aktuellen Wahrscheinlichkeit wird jedes weitere Pixel mit einem arithmetischen Koder kodiert. Die dabei verwendeten Intervallgrößen für Schwarz und Weiß sind dem Dekoder ja aus der auch auf dessen Seite geführten Statistik und aus dem auch dort vorhandenen Kontext bekannt.

Der Gruppe 4 ISDN Fax-Standard ist etwa 30% besser als der Gruppe 3 Standard (T.4, etwa 1 : 20), und das JBIG Verfahren ist bei durchschnittlichen Seiten nochmal um etwa 30% effizienter. Dies alleine mag vielleicht noch keinen neuen Standard rechtfertigen, aber JBIG erreicht immer noch etwa eine Kompression von 1 : 8 bei gerasterten Photos, die bei den Fax-Verfahren in der Regel sogar zu einer Expansion statt

Kompression führen.

Obwohl primär für die Kompression von bilevel Bildern gedacht, kann JBIG auch für die Kodierung von Graustufen oder color lookup table Bildern mit wenigen Bit Tiefe benutzt werden. Es empfiehlt sich dazu, die Grauwerte nicht als normale z.B. 4-bit Binärzahlen, sondern im Graykode darzustellen. Im Graykode ändert sich beim Übergang zum nächsten Wert immer nur genau ein Bit. Anschließend werden die einzelnen Bitebenen völlig unabhängig voneinander mit JBIG kodiert. Für Grauwertbilder kann alternativ auch der JPEG lossless mode eingesetzt werden. Bei 4–6 Bit Auflösung sind dabei JPEG und JBIG etwa gleich effizient, bei geringerer Auflösung schneidet JBIG besser ab, bei mehr Bits pro Bildpunkt empfiehlt sich das JPEG Verfahren.

Literatur:

- [1] Pennebaker, William B.; Mitchell, Joan L.: *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993.
- [2] Wallace, Gregory K.: *The JPEG Still Picture Compression Standard*, CACM, April 1991, Vol. 34, No. 4, pp 30-44.
- [3] Liou, Ming: *Overview of the P×64 kbit/s Video Coding Standard*, CACM, April 1991, Vol. 34, No. 4, pp 59-63.
- [4] Oppenheim, Alan V.: *Applications of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, 1978.
- [5] Girod, B.: *Bildkommunikation I/II*, Skriptum zur Vorlesung, Universität Erlangen-Nürnberg, 1993/94.
- [6] Topsøe, F., *Informationstheorie*, Teubner Verlag, Stuttgart 1974.
- [7] Press, William H.; et. al.: *Numerical Recipes in C – The Art of Scientific Computing*, Cambridge University Press, New York, 1992.
- [8] Musmann, H.-G.; Werner, O.; Fuchs, H.: *Kompressionsalgorithmen für interaktive Multimedia-Systeme*, in: B. Girod (Hrsg.): *it+ti 2/93*, Informationstechnik und Technische Informatik 35 (1993) 2, Oldenburg Verlag.
- [9] ISO 10918/ITU-T T.81: *Information Technology – Digital Compression and Coding of Continuous-Tone Still Images*, International Organization for Standardization, Geneva, 1993.
- [10] ISO 11172: *Information Technology – coding of moving pictures and associated audio for digital storage media at up to about 1.5 MBit/s*, International Organization for Standardization, Geneva.
- [11] ITU-T: *Video Codec for Audiovisual Services at p × 64 kbit/s*, Recommendation H.261, International Telecommunication Union, Geneva.