1 Motivation

Für die Bearbeitung einer Aufgabenstellung, welche als Ziel die Kompression von visuellen Informationen besitzt, ist der erste Aspekt, mit dem man sich beschäftigen muß, die Frage nach der Repräsentation dieser Informationen. Die zu stellende Frage ist, wie sich 4-dimensionale visuelle Informationen (drei Dimensionen im Raum, eine Dimension in der Zeit) repräsentieren lassen, so daß man sie bearbeiten und damit auch komprimieren kann.

1. Repräsentation von Bildern und Bildsequenzen

Die erste Antwort auf diese Frage wurde mit der Einführung des Fernsehens gegeben. Die drei räumlichen Dimensionen wurden auf eine 2-dimensionale Bildebene projiziert und durch die Produktion einer Bilderserie wurde die zeitliche Dimension nachempfunden. Der Aufbau eines Bildes ist dabei durch eine Anzahl von Linien mit einer festen Anzahl von Werten für Helligkeits- und Farbinformationen gegeben. Dieses System arbeitet nun seit vielen Jahren erfolgreich, obwohl es diverse Nachteile besitzt. So wird z.B. das 4-dimensionale Signal in eine Ebene projiziert. Auch wenn man die zeitliche Dimension kontinuierlich machen könnte, bedeutet dies immer noch den Verlust von einer Dimension und damit von Informationen. Aus technischen Gründen ist eine kontinuierliche zeitliche Dimension allerdings nicht möglich, was zu einer ersten Diskretisierung und einem damit verbundenem Sampling in der zeitlichen Dimension führt. Die Erzeugung des Videosignals bedingt dann ein zweites vertikales Sampling in der Bildebene. Bei beiden Sampling-Vorgängen müßte das Sampling-Theorem berücksichtigt werden, was bei der heutigen Fernsehtechnik allerdings nicht passiert. Dies läßt sich z.B. an Fahrzeugrädern, die sich entgegen der Fahrtrichtung bewegen oder an Moiré-Mustern auf Kleidungsstücken von Kommentatoren erkennen.

In einer vollständig digitalen Umgebung kommt auch noch das Sampling horizontal in der Bildebene für die übrig gebliebene dritte Dimension hinzu. Alle drei Sampling-Vorgänge werden mit einer jeweils konstanten Sampling-Rate durchgeführt. Da das System vollständig digital sein soll, werden die Samples mit einer vorgegebenen

Anzahl von Bits quantisiert, was je nach Verfahren zu einer mehr oder weniger großen Menge von Bits für das ursprüngliche 4-dimensionale Signal führt. Dies führt desweiteren dazu, daß die originär 4-dimensionalen Informationen als stationäres 1-dimensionales Signal verarbeitet werden können. Für eine rein digitale Verarbeitung ist dies durchaus von Vorteil, der Nachteil allerdings besteht aus einem Informationsverlust, der schon bis zu diesem Verarbeitungsschritt aufgetreten ist. Diese verlorenen Informationen hätten eventuell für eine bessere Kompression der Daten von Vorteil sein können. Die Ausführungen machen deutlich, daß die Repräsentation der Informationen vollständig von den technischen Gegebenheiten bestimmt wird, während das HVS fast vollkommen außer acht gelassen wird. Die Repräsentation ist vollständig systembedingt, obwohl sie sinnvollerweise durch die Eigenschaften der Ausgangsinformationen bestimmt werden sollte. Als Folge daraus besitzt die Repräsentation natürlich Redundanzen und die Aufgabe der Datenkompression liegt nun darin, diese durch das System gegebenen Daten so zu kodieren, daß sie im Bezug auf das HVS möglichst wenig Redundanz enthalten. Der hier gewählte Ansatz unterscheidet sich von dem Ansatz der Rekonstruktion der Bildinformationen im Hinblick auf einzelne Pixel, da er eine Rekonstruktion im Hinblick auf das HVS verfolgt.

2. Pixel-basierte Methoden

Das Ziel dieses Abschnittes ist ein kurzer Abriß über die Methoden der ersten Generation, sowie eine Herausarbeitung ihrer Stärken und Schwächen. Anhand dieser Grundlagen können in den folgenden Kapiteln auch die Unterschiede zu einer Methode der zweiten Generation, der fraktalen Bildkodierung, aufgezeigt werden.

Als Eigenschaften, die allen Methoden der ersten Generation gemeinsam sind, lassen sich folgende zusammenfassend angeben:

1. Die Methoden versuchen auf der Dekoder-Seite ein Bild zu produzieren, welches dem Original in jedem Pixel möglichst ähnlich ist.
2. Beim Kodieren und Dekodieren wird das Bild in Bereiche zerlegt, die unabhängig vom Bildinhalt sind.
3. Räumliche Redundanz wird durch den Einsatz allgemeiner Techniken der Signalanalyse minimiert.
4. Zeitliche Redundanz wird durch Techniken zu Bewegungsanalyse und -vorhersage (*motion compensation* und *motion estimation*) zu minimieren versucht. Dies geschieht wiederum ohne Berücksichtigung des Bildinhaltes, auch wenn der Begriff “Bewegung” dies nahe legt.
5. Von den Eigenschaften und Besonderheiten des HVS wird nur ansatzweise Gebrauch gemacht.

Anhand dieser Eigenschaften wird deutlich, warum die Methoden der ersten Generation auch als *Pixel-basiert* oder *frequenzorientiert* bezeichnet werden [Li94]. Die Hauptkritikpunkte an diesen Methoden sind vor allem in der ersten und der letzten Eigenschaft begründet. Für ein System, was sich in erster Linie an das HVS richtet, ist ein Ansatz, der eine exakte Reproduktion bis auf Pixelebene fordert, zu stringent. Das HVS neigt

z.B. dazu, für verschiedene Kontexte Detailinformationen in einem Bild nicht wahrzunehmen. Die Forderung nach einer Rekonstruktion auf Pixelebene ist nur für solche Systeme sinnvoll, für die dies eine unabdingbare Forderung darstellt, wie z.B. medizinische Bildgebungs- und Bildverarbeitungssysteme.

3. Grundlegende Eigenschaften Pixel-basierter Methoden zur Kompression von Bilddaten

In diesem Abschnitt sollen die Hauptkonzepte von Pixel-basierten Methoden vorgestellt und anhand ihrer Eigenschaften und Funktionsweise klassifiziert werden.

Ein System zur Kompression und Dekompression von Bild- bzw. Videodaten läßt sich als ein Prozeß mit jeweils sechs Prozeßstufen für den Kodierer und den Dekodierer beschreiben:

Abb. 1: *Prinzip pixel-basierter Methoden zur Bildkompression*

Ist für die Übertragung eine Flußkontrolle (*rate control*) notwendig, so muß das Grundmodell um die gestrichelten Elemente zur Bandbreiten-Kontrolle der Ausgabe erweitert werden.

Farbraum-Transformation

Redundanzen in der Farbrepräsentation werden im allgemeinen schon im ersten Schritt minimiert. Bei den meisten Systemen bedeutet dies, daß eine Transformation aus dem RGB-Farbraum in einen Farbraum durchgeführt wird, in dem die Farbinformation von den Informationen über Helligkeit und Kontrast getrennt sind. Nach einer solchen

Transformation, z.B. in das YUV-Farbmodell, lassen sich die einzelnen Komponenten für Farbe, Helligkeit und Kontrast entsprechend ihrer Aufnahme durch das HVS weiterverarbeiten [ITU90].

Vor- und Nachverarbeitung

Durch eine Vorverarbeitung, die gewisse Eigenschaften der Eingabedaten hervorhebt oder bei Bedarf auch unterdrückt, läßt sich die Effizienz eines Kompressions-Schemas teilweise deutlich steigern. So arbeiten einige der bekannten Verfahren besser, wenn die Eingabebilder vor der Kompression low-pass gefiltert werden. Die Art und Weise dieser Vorverarbeitung bedarf natürlich einer Abstimmung und Optimierung im Hinblick auf das Gesamtsystem, um für das HVS eine möglichst optimale Qualität des gesamten Kompressions-Systems zu erreichen.

Auf der Empfängerseite kann das dekodierte Signal durch einen Nachverarbeitungsschritt qualitativ verbessert werden. Die verschiedenen Ansätze für eine Nachverarbeitung decken thematisch einen sehr großen Bereich ab. Zum einen existieren Methoden zur Unterdrückung von auftretenden Kompressions-Artefakten, wie z.B. die Block-Artefakte bei Methoden, die auf der Transformation von Pixelblöcke beruhen [Rama86], zum anderem werden Ansätze diskutiert, die das Originalbild aus den komprimierten Daten vorhersagen wollen [Ozcelik95].

Subsampling

Subsampling reduziert die zu komprimierende Datenmenge, in dem die zugrunde liegende Sampling-Struktur der diskreten Bild- oder Videodaten in eine neue Sampling-Struktur mit einer reduzierten Sampling-Rate überführt wird. Die Reduzierung der Sampling-Rate kann im Raum, in der Zeit oder im Raum und in der Zeit vorgenommen werden [Dubois95].

Transformation

Der Transformationsschritt führt das Signal in eine Beschreibung über, die effizienter zu bearbeiten und aussagekräftiger im Bezug auf die Signaleigenschaft ist, als die Beschreibung durch Pixelwerte. Dies bedeutet meist die Transformation in einen anderen Raum, wie z.B. die Abbildung mit Hilfe der DCT (DCT = *Discrete Cosine Transform*) in einen Frequenzraum [ISO94b], in welchem dann Aussagen über den Inhalt des Signals getroffen werden können. Mit Hilfe dieser Aussagen verfolgt man bei Pixel-basierten Techniken das Ziel, statistische Redundanzen aus dem Signal zu entfernen. Dieser Transformationsschritt ist der zentrale Kern eines jeden Kompressions-Systems der ersten Generation und wird deshalb in folgenden Abschnitten noch weiter betrachtet.

Neben der Minimierung der Redundanz ist dieser Verarbeitungsschritt allerdings auch wesentlich für jene Eigenschaften verantwortlich, die das Signal nach der Kodierung besitzt. So wird auch die Aufarbeitung und die Strukturierung der Daten für das Versenden über einen Kanal stark von der Wahl der Transformation beeinflußt. Für

Puffer und Datenfluß-Kontrolle

chen Dimensionen vollständig zu beschreiben. Das Interlaced-Video-Format, wie es z.B. beim Fernsehen verwendet wird, benötigt mehrere Zeiteinheiten für die vollständige Beschreibung der beiden räumlichen Dimensionen [ITU90]. Ein vollständiges Bild wird in beiden Fällen *frame* genannt. Im Falle der Interlaced-Verfahren heißt ein Teilbild *field*.

Abb. 2: *Interlaced Video-Format für zwei Zeiteinheiten*

5. Transformation

Für die Eliminierung statistischer Redundanzen im Transformationsschritt existieren im wesentlichen zwei Hauptansätze:

– Prädiktive Verfahren (*predictive approaches*)
– Transformations basierende Verfahren (*transformational approach*)

Die prädiktiven Verfahren benutzen für die Kompression einen direkten Zugang zu den Daten, indem sie von dem zu komprimierenden Signalteil jenen Teil subtrahieren, den sie aus dem aktuellen Kontext vorhersagen können.

Verfahren, die auf der Anwendung von Transformationen beruhen, bearbeiten das Signal nicht direkt, sondern bilden das Signal oder einen Teil davon vor einer weiteren Bearbeitung in einen anderen Raum ab. Als Resultat dieser Transformation erhält man eine Darstellung des Signals, die eine möglichst weitgehende Entfernung der Redundanz erlaubt. Die Transformationsverfahren kann man dann noch in die folgenden Hauptansätze unterscheiden:

– Transformation von Pixelblöcken (*block transformation*)
– Pyramiden Schemata (*pyramidal schemes*)
– Frequenzband / Wavelet Analyse (*subband / wavelet analysis*)

Prädiktive Verfahren

Prädiktive Kodierungsverfahren beruhen auf der Kodierung eines Samples auf der Basis einer Menge schon kodierter Samples. Das zu kodierende Sample wird aus den vorhergegangenen Samples berechnet und nur der Vorhersagefehler, also die Differenz

zwischen Berechnung und dem tatsächlichen Wert, wird kodiert und übermittelt. Die Berechnung des aktuellen Vorhersagewertes kann sich dabei nur auf den direkten oder auf eine Menge von Vorgängern beziehen.

Abb. 3: *Schema für Prädiktive Verfahren*

Ordnet man obiges Diagramm in das allgemeine Kompressionsschema gemäß der Abbildung *Prinzip pixel-basierter Methoden zur Bildkompression* auf Seite 9 ein, so entspricht der Kodierer den Komponenten Transformation und Quantisierung, die für prädiktive Verfahren nicht getrennt werden können. Beim Kodierer wird das Eingabe-Signal $x(n)$ von der Vorhersage (*prediction*) $p(n)$ abgezogen und der resultierende Fehler (*prediction error*) $d(n)$ wird quantisiert und übermittelt.

Der Dekodierer addiert den rekonstruierten Fehler $s(n)$ zu der Vorhersage $p(n)$ und erhält so das rekonstruierte Sample $r(n)$ als Ausgabe. Außerdem wird $r(n)$ als Eingabe für die Berechnung der Prädiktion $p(n+1)$ des Samples $x(n+1)$ benutzt.

Eine Erweiterung dieses Basis-Schemas stellt die Einführung adaptiver Parameter in die Berechnung der Prädiktion dar. Eine Optimierung des Gesamtsystems im Hinblick auf das HVS kann so z.B. durch eine Adaption der Quantisierung an die Sichtbarkeit des Vorhersagefehlers in den verschiedenen Bildbereichen erreicht werden. [Jain81a] [Netra80]

Beim Kodieren mit prädiktiven Verfahren können im wesentlichen drei Haupttypen von Artefakten auftreten: [Jayant84]

– *Slope Overhead*:
  Slope Overhead tritt auf, wenn bei der Kodierung von Kanten in einem Bild der zu kodierende Vorhersagefehler den Wertebereich des Quantisierers überschreitet. Das Artefakt äußert sich in einer Glättung der Kanten.
– *Granular Noise*:
  Ist dagegen der zu kodierende Vorhersagefehler kleiner als der kleinste Quantisierungsschritt, so erhält man in sehr gleichförmigen Bildbereichen ein Rauschen, das sog. *granular noise*.

– *Edge Busyness*:
Das letzte auftretende Artefakt ist das sog. *edge busyness*, was sich in einer zufälligen Verschiebung von Kanten äußert. Dies ist das Resultat einer zufälligen Zuordnung einer Kante zwischen zwei Quantisierungsschritten, z.B. aufgrund einer zu geringen Auflösung des Quantisierers.

Die Leistung der prädiktiven Verfahren wird von den im folgenden besprochenen Transformationsverfahren bei weitem übertroffen. Das Paradigma der Vorhersage spielt bei den Pixel-basierten Verfahren in Bezug auf zeitliche Vorhersagen (*motion estimation* und *motion compensation*) in Video-Kompressionssystemen immer noch eine wichtige Rolle.

Zwei internationale Standards basieren auf prädiktiven Verfahren. Der Standard ITU-T J.80 [ITU93b] verwendet eine räumliche Prädiktion für die Kompression von Fernsehsignalen auf eine Übertragungsrate von 140 MBit/s. Auch der ISO Standard JPEG [ISO94b] besitzt einen Modus zur verlustlosen Kompression basierend auf einer Prädiktion. Für ein Pixel x wird die beste von acht möglichen Prädiktionen P gewählt:

Pixel:

	c	b
	a	x

Prädiktion:

Modus	Berechnung
0	$P = 0$
1	$P = a$
2	$P = b$
3	$P = c$

Modus	Berechnung
4	$P = a + b - c$
5	$P = a + (b - c)/2$
6	$P = b + (a - c)/2$
7	$P = (a + b)/2$

Abb. 4: *Prädiktion für ein Pixel x im JPEG-Standard*

Durch die Verwendung von Pixeln zur Berechnung der Prädiktion, die schon vor der Kodierung von x kodiert wurden, haben sowohl der Kodierer als auch der Dekodierer Zugriff auf die Pixel a, b und c. Nach der Kodierung des Vorhersagefehlers $d = P - x$ ist damit gewährleistet, daß der Dekodierer den Wert von x wieder berechnen kann.

Transformation von Pixelblöcken (*block transforms*)

Eine Alternative zur Entfernung von Redundanz stellen Verfahren dar, die auf der Transformation von sich nicht überlappenden Pixelblöcken beruhen. Diese Verfahren basieren auf einer Segmentierung des Ausgangsbildes in einzelne Pixelblöcke. Die Segmentierung ist im allgemeinen unabhängig vom Inhalt des Bildes und zerlegt das Bild in gleichgroße, disjunkte Pixelblöcke. Die Anwendung einer linearen invertierbaren Transformation wird dann für jeden Pixelblock unabhängig durchgeführt und wurde zuerst in [Huan63] vorgeschlagen.

Nach einer Umordnung eines Pixelblockes in einen Spaltenvektor x läßt sich die Transformation als eine Matrixmultiplikation schreiben:

$$y = Ax$$

Diese Transformation läßt sich auch als ein Wechsel der Basis im linearen Raum aller Pixelblöcke auffassen. Der Vektor y mit den transformierten Koeffizienten ist somit die Darstellung des Original-Vektors x im Bezug auf eine neue Basis. Von besonderem Interesse sind in diesem Kontext orthonormale Transformationen. Deren Transformationsmatrix A ist orthonormal und enthält als Spaltenvektoren die Koeffizienten der Einheitsvektoren bezüglich der neuen Basis. Eine wichtige Eigenschaft der orthonormalen Transformationen ist die Erhaltung von Signaleigenschaften bezüglich der quadratischen Norm.

Eine Transformation für ein 1-dimensionales Signal kann durch die Definition einer separablen linearen Transformation, welche unabhängig in beide Richtungen arbeitet, zu einer Transformation für einen 2-dimensionalen Pixelblock erweitert werden:

$$Y = AXA^T$$

X und Y sind Matrizen und repräsentieren den Original- bzw. den transformierten Pixelblock. Ist einer der transformierten Koeffizienten von Y proportional zu einem Mittelwert der Original-Pixelwerte aus X, so wird dieser der DC-Koeffizient (*DC coefficient*) und die anderen Koeffizienten werden AC-Koeffizienten (*AC coefficients*) genannt.

Der Sinn des Transformationsansatzes liegt darin, daß eine um so effizientere Kodierung erreicht werden kann, je Informationen aus dem Original-Pixelblock in ein paar transformierte Koeffizienten konzentriert werden kann. Die in diesem Sinne optimale Transformation ist die Karhunen-Loeve Transformation (KLT) [Huan63]. Die KLT ist eine orthonormale Transformation, deren Basisvektoren die Eigenwerte der Autokorrelationsmatrix des Signals sind. Für ein Kompressionsschema, das auf dem “Aussortieren” von Koeffizienten mit den wenigsten Informationen bzw. der geringsten Energie beruht, ist die KLT optimal. Aufgrund der Abhängigkeit der KLT von den statistischen Eigenschaften des Signals und dem Fehlen von effektiven Algorithmen zur Berechnung ist eine direkte Anwendung der KLT zu Dekorrelation leider nur sehr schwer möglich.

Als Annäherung an diese optimale Transformation wird heute im allgemeinen die Diskrete-Kosinus-Transformation (DCT = *discrete cosine transform*) verwendet [Rao90]. Die DCT ist für stationäre Markov-Sequenzen erster Ordnung eine gute Annäherung an die KLT mit einem asymptotischen Verhalten für den Fall, daß der Korrelations-Koeffizient gegen 1 geht [Ahmed74]. Desweiteren existieren für die Berechnung der DCT schnelle Algorithmen, so daß dieser Nachteil der KLT hier nicht zum Tragen kommt. Eine Erweiterung der DCT auf mehrere Dimensionen kann durch die Definition einer entsprechenden multidimensionalen separablen Transformation erreicht werden. Modelliert man das HVS als lineares räumlich invariantes System, dann kann durch eine Quantisierung der Transformations-Koeffizienten aufgrund ihrer Sichtbarkeit durch das HVS ansatzweise auch eine Adaption an das HVS vorgenommen werden. In diesem Sinne werden bei diesen DCT-basierten Systemen auch schon Ansätze und Verfahren von Kompressionssystemen der zweiten Generation angewendet.

Die Kodierung der transformierten Koeffizienten wird im allgemeinen durch die Anwendung einer Lauflängenkodierung für Null-Koeffizienten und einer adaptiven Kodierung für die restlichen Koeffizienten optimiert. Die wohl bekannteste und auch

verbreiteste Methode zur Kompression von Bildern, welche auf der Transformation von Pixelblöcken beruht, ist der JPEG Algorithmus (Siehe *Überblick über die JPEG-Technik* auf Seite 18) [ISO94b].

Die Vorteile einer Transformations-Kodierung gegenüber einer prädiktiven Kodierung treten vor allem bei einer praktischen Anwendung zu Tage. Während der theore-

Überblick über die JPEG-Technik

Eine JPEG-Kodierung läuft prinzipiell in drei aufeinanderfolgenden Schritten ab:

Abb. 5: *JPEG-Kodierung*

Diese drei Schritte zusammen ergeben einen leistungsfähigen Komprimierer, der in der Lage ist, Bilder ohne oder nur mit wenig Qualitätsverlust auf bis zu 10% ihrer Originalgröße zur komprimieren.

Die JPEG-Dekodierung läuft in den folgenden Schritten ab:

Abb. 6: *JPEG-Dekodierung*

DCT-Transformation

Wie in den vorangegangenen Abschnitten schon erwähnt, ist die Aufgabe der DCT die Dekorrelation der Signalwerte. Zu diesem Zweck wandelt die DCT eine Reihe von Punkten im 3D-Raum in Frequenzwerte um. Die Inverse Diskrete Kosinus Transformation (IDCT = *Inverse Discrete Cosine Transform*) kehrt diese Umwandlung wieder um.

DCT:

$$DCT(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \left[\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} Pixel(x,y) \times \cos\frac{(2x+1)i\pi}{2N} \cos\frac{(2y+1)j\pi}{2N} \right]$$

IDCT:

$$Pixel(x,y) = \frac{1}{\sqrt{2N}} \left[\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(i)C(j)DCT(i,j) \times \cos\frac{(2x+1)i\pi}{2N} \cos\frac{(2y+1)j\pi}{2N} \right]$$

$$\text{mit } C(u), C(v) = \frac{1}{\sqrt{2}} \text{ für } u, v = 0 \text{ und } C(u), C(v) = 1 \text{ sonst}$$

Die DCT arbeitet mit einer $N \times N$-Matrix, die aus Pixelwerten besteht. Das Ergebnis ist eine $N \times N$-Matrix aus Frequenzkoeffizienten. Für den JPEG-Algorithmus wurde $N = 8$ gewählt. Für eine Grafik auf dem Bildschirm werden dabei die Koordinaten x, y und der Farbwert $z = f(x, y)$ der Pixel genommen. Bei Farbbildern werden die einzelnen Farbebenen getrennt nacheinander kodiert und dekodiert. Für die Kodierung wird das Eingabebild in 8×8 Pixel-Blöcke aufgeteilt und die vorzeichenlosen Integerwerte im Bereich von $[0, 2^p - 1]$ werden so geschoben, daß sie im Bereich von $[-2^{p-1}, 2^{p-1} - 1]$ liegen. Die DCT liefert für jeden 8×8 Pixel-Block eine 8×8 Matrix mit DCT-Koeffizienten:

Orginalbild

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

Farbwerte des Orginalbildes

235,6	-1,0	-12,1	-5,2	2,1	-1,7	-2,7	1,3
-22,6	-17,5	-6,2	-3,2	-2,9	-0,1	0,4	-1,2
-10,9	-9,3	-1,6	1,5	0,2	-0,9	-0,6	-0,1
-7,1	-1,9	0,2	1,5	0,9	-0,1	0,0	0,3
-0,6	-0,8	1,5	1,6	-0,1	-0,7	0,6	1,3
1,8	-0,2	1,6	-0,3	-0,8	1,5	1,0	-1,0
-1,3	-0,4	-0,3	-1,5	-0,5	1,7	1,1	-0,8
-2,6	1,6	-3,8	-1,8	1,9	1,2	-0,6	-0,4

DCT-Koeffizienten

Abb. 7: *Anwendung der DCT auf einen 8x8 Pixelblock*

Die Behandlung der Ausgangsdaten mit der DCT führt sie in eine äquivalente Repräsentation in einem Frequenzraum über. Die Interpretation der Matrix ist, daß die links oben angeordneten Werte gröbere “langwellige” Bilddetails beschreiben als die rechts unten angeordneten Koeffizienten. Entfernt man sich weiter von dieser linken oberen

Ecke der Matrix besitzen die Koeffizienten nicht nur niedrigere Werte, sie werden für die Beschreibung des Bildes auch weniger wichtig. Die DCT trennt also Wichtiges von weniger Wichtigem, ohne Informationen zu zerstören.

Quantisierung

Die DCT ist eine verlustlose Transformation, die den “verlustbehafteten” Schritt der Quantisierung vorbereitet. Die verschiedenen Ansätze und Methoden zur Quantisierung, sowie deren Einsatz im Rahmen von Kompressionssystemen für Bilder, werden in einem der folgenden Abschnitte (Siehe *Quantisierung* auf Seite 11) diskutiert. Das Ziel der Quantisierung in diesem speziellen Kontext ist die Darstellung der DCT-Koeffizienten mit der Genauigkeit, die für die gewünschte Bildqualität notwendig ist. Der JPEG-Algorithmus realisiert die Quantisierung mit einer Quantisierungsmatrix, die für jeden DCT-Koeffizienten einen Quantum-Wert $Q(u, v)$ enthält. Die Quantisierungsformel lautet:

$$F^Q(u, v) = \left\lfloor \frac{F(u, v)}{Q(u, v)} \right\rfloor$$

Damit erhält man:

235,6	-1,0	-12,1	-5,2	2,1	-1,7	-2,7	1,3	
-22,6	-17,5	-6,2	-3,2	-2,9	-0,1	0,4	-1,2	
-10,9	-9,3	-1,6	1,5	0,2	-0,9	-0,6	-0,1	
-7,1	-1,9	0,2	1,5	0,9	-0,1	0,0	0,3	DCT-Koeffizienten
-0,6	-0,8	1,5	1,6	-0,1	-0,7	0,6	1,3	
1,8	-0,2	1,6	-0,3	-0,8	1,5	1,0	-1,0	
-1,3	-0,4	-0,3	-1,5	-0,5	1,7	1,1	-0,8	
-2,6	1,6	-3,8	-1,8	1,9	1,2	-0,6	-0,4	

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

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

quantisierte DCT-Koeffizienten

Abb. 8: *Quantisierung*

Die Quantisierungsmatrix kann während der Komprimierung definiert werden. Die ISO hat allerdings auch aufgrund von psychovisuellen Tests eine Quantisierungsmatrix für Implementierungen des JPEG Algorithmus vordefiniert. Für die Bestimmung der Quantum-Schrittgröße gibt der Anwender einen Qualitätsfaktor vor, so daß sich die Quantum-Werte folgendermaßen berechnen:

$$Q(u, v) \;=\; Q(u, v) \times \text{Qualität}$$

Mit Hilfe der Rekonstruktionsformel wird der Prozeß bei der Dekomprimierung wieder umgekehrt:

$$F^{Q'}(u, v) \;=\; F^{Q}(u, v) \times Q(u, v)$$

Damit erhält man:

240	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

rekonstruierte DCT-Koeffizienten

Abb. 9: *Rekonstruktion*

Durch Anwendung der IDCT läßt sich aus den rekonstruierten DCT-Koeffizienten das Originalbild wieder rekonstruieren, allerdings mit den durch die Quantisierung entstandenen Verlusten.

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

Farbwerte des Originalbildes

144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	156
163	164	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158

Farbwerte des rekonstruierten Orginalbildes

Abb. 10: *Vergleich der originalen und der rekonstruierten Pixelwerte*

Ein Vergleich zwischen Qualität und Dateigröße:

Abb. 11: *JPEG: Vergleich zwischen Qualität und Dateigröße*

Verlustfreie Kompression

Im letzten Schritt der JPEG-Kompression wird eine Lauflängen-Kodierung der Nullwerte und eine Entropie-Kodierung auf der Basis von Huffmann-Codes oder arithmetischer Kodierung durchgeführt. Zur Unterstützung dieser Techniken werden die quantifizierten DCT-Koeffizienten mit Hilfe eines “Zick-Zack Kurses” angeordnet: [ISO94b]

Abb. 12: *Erhöhung der Lauflänge für die Lauflängenkodierung*

Dieser “Zick-Zack” Kurs erhöht die Lauflänge für die Lauflängenkodierung, da er die höchsten Elementwerte zuerst und die niederen Werte zuletzt abarbeitet.

Pyramiden Schemata (*pyramidal schemes*)

Pyramiden Schemata können als ein Spezialfall einer hierarchischen Kodierung angesehen werden, da sie aus einer Menge von Bildern mit verschiedenen abgestuften Auflösungen desselben Ausgangsbildes bestehen. Die wichtigste Variante eines Pyramiden Schemas ist die Laplace Pyramide [Burt83]. Der Grundgedanke dieses Schemas ist, daß ein Großteil der visuellen Bildinformationen auch noch in einer auflösungsreduzierten Version des Ausgangsbildes vorhanden sind. Folgendes Schema verdeutlicht die zugrundeliegende Idee:

Abb. 13: *Laplace Pyramide: Kodierung*

Die Abbildung zeigt den Aufbau einer zweistufigen Pyramide. Ausgehend von einem Bild I_0 mit einer vertikalen Auflösung R_{V_0} und einer horizontalen Auflösung R_{H_0} erhält man durch Downsampling ein Bild I_1 mit einer Auflösung $R_{V_1} < R_{V_0}$ und $R_{H_1} < R_{H_0}$. Der Begriff “Downsampling” steht hier für eine Kombination aus Subsampling und einem Low-Pass Filter. Das Bild I_1 wird anschließend komprimiert und stellt die erste Stufe mit der niedrigen Auflösung der zweistufigen Pyramide dar. Dies ist das Bild I_1^c mit einer Auflösung R_{V_1} und R_{H_1}. An dieser Stelle sei extra darauf hingewiesen, daß die Art des Kompressions-Algorithmus nicht spezifiziert wird. Die Formulierung der Laplace’schen Pyramide wird hier unabhängig von einem speziellen Kompressions-Algorithmus formuliert und läßt sich somit in dieser Form mit einem beliebigen Algorithmus zur Kompression von Bildern kombinieren und anwenden.

Für die Berechnung der zweiten Pyramidenstufe wird das Bild I_1^c dekomprimiert. Anschließend werden ein Upsampling und eine Interpolation auf Pixelbasis durchgeführt. Damit erhält man ein Bild I_0' mit derselben Auflösung wie das Ausgangsbild I_0. Aus diesen beiden Bildern berechnet man auf Pixelbasis das Differenzbild

$$I_0^D = I_0 - I_0'$$

und komprimiert dieses, so daß man als zweite Pyramidenstufe das Bild I_0^c erhält.

Die Dekodierung startet dann mit der ersten Pyramidenstufe I_1^c für die niedrige Auflösung und verwendet I_0^c zur Berechnung der hohen Auflösung:

Abb. 14: *Laplace Pyramide: Dekodierung*

Dieser Prozeß kann zur Produktion von Pyramiden mit beliebig vielen Stufen verwendet werden. Es besteht somit die Möglichkeit, Bilder in beliebig vielen Auflösungen so zu kodieren, daß höhere sich auf niedrigere Auflösungen beziehen. Höhere Auflösungen setzen sich somit additiv aus niedrigeren Auflösungen zusammen.

Frequenzband-Analyse (*subband analysis*)

Das Ziel einer Subband- bzw. Wavelet-Analyse ist die Zerlegung eines Signals bzw. Bildes in einzelne “elementare” Signalbereiche bzw. Bildteile, die zusammen wiederum das gesamte Signalspektrum beschreiben [Croch83] [Smith1990] [Vetterli84]. Jeder dieser Signalbereiche enthält Informationen, die einem Teil des Gesamtspektrums entsprechen. Erreicht wird dies durch das Filtern des Signals mit Hilfe einer Filterbank, in der jeder Filter ausschließlich einen bestimmten Frequenzbereich abdeckt. Nach dem Filtern werden im allgemeinen die einzelnen Ergebnisse der Filter gemäß ihrem Anteil am Gesamtspektrum in der Auflösung reduziert:

Abb. 15: *Analyse Filterbank*

In der Art der iterierten Zerlegung des Gesamtsignals existieren von diesem Schema die verschiedensten Variationen. Auf der Dekoderseite wird dieser Prozeß umgekehrt, indem für die einzelnen Signalbereiche ein Upsampling, eine Filterung und eine anschließende Addition zum Gesamtsignals durchgeführt wird:

Abb. 16: *Synthese Filterbank*

Grundlegende Forderungen an eine Subband-Filterbank sind die folgenden Fähigkeiten und Eigenschaften:

– Auf der Empfänger- bzw. Dekoderseite muß ein von Alias-Effekten freies Signal ankommen.
– Das Gesamtsystem darf nicht zu einer linearen Verzerrung führen (*perfect reconstruction filter banks*).
– In der Bildverarbeitung ist außerdem eine Phasen-Linearität wünschenswert.
– Um möglichst unkorrelierte Frequenzbänder (*subbands*) zu erhalten, sollten sich die Filterbereiche der Analysefilter nicht überlappen.

Für die ersten drei Forderungen wurden Filterbänke mit den entsprechenden Eigenschaften entwickelt, während die vierte Eigenschaft nur approximiert werden kann. Die Anwendung einer Subband-Analyse auf digitale Bilder, also endliche Signale, bedeutet, daß diese Signale konzeptuell in unendliche Signale umgewandelt werden müssen, was z.B. durch eine periodische Wiederholung des endlichen Signals erreicht werden kann.

Der Zusammenhang zwischen einer Subband-Analyse und der Transformation von Pixelblöcken besteht darin, daß die Transformation eines Pixelblockes als Anwendung einer aus FIR-Filtern bestehenden Filterbank interpretiert werden kann. Es bestehen auch enge Verbindungen zu den schon erwähnten Pyramiden Schemata, wie anhand der Abbildungen *Laplace Pyramide: Kodierung*, *Laplace Pyramide: Dekodierung*, *Analyse Filterbank* und *Synthese Filterbank* für den Fall $N = 2$ leicht zu sehen ist [Rioul91]. Während die oberen Zweige der Schemata gleich bleiben, stellen die unteren Zweige ein Signal mit jeweils der halben Samplingrate dar. In jeder Stufe wird das niederfrequente Signal wieder in zwei Signalteile zerlegt:

Abb. 17: *Subband-Analyse — Pyramiden Schema*

Die Anwendung einer Subband-Analyse auf mehrdimensionale Signale läßt sich durch die Erweiterung des Schemas auf die mehrfache Anwendung einer eindimensionalen Subband-Analyse in jede der Dimension (*separable subband decomposition*) vornehmen. Desweiteren sind auch spezielle nicht separable Filter- und Subsampling-Strukturen einsetzbar [Vetterli84].

Wavelet Analyse (*wavelet analysis*)

Spätestens die Verbindung der Subband-Analyse zur Wavelet-Theorie hat diesen Ansatz zu einem wichtigen Ansatz für die Signal-Analyse und Verarbeitung werden lassen. Der bei der Signal-Analyse mit Hilfe von Wavelets verwendete Ansatz ist ein gebräuchlicher Ansatz in der Analyse von Signalen, bei dem ein Signal $f(x)$ als gewichtete Summe von Basis-Funktionen dargestellt wird:

$$f(x) = \sum_i c_i \Psi_i(x)$$

Die $\Psi_i(x)$ sind die Basisfunktionen und die c_i sind die Koeffizienten oder Gewichte. Zur Vermeidung von Redundanzen in der Repräsentation sollten die Ψ_i orthogonal sein. Die Signaleigenschaften sind in den Koeffizienten kodiert, da die Basisfunktionen Ψ_i fest sind. Im Vergleich zu anderen Ansätzen, die z.B. mit Hilfe der DCT arbeiten, können im Falle der Wavelets die Ψ_i allerdings in Abhängigkeit von den Bildeigenschaften gewählt werden. Eine enge Verbindung zwischen einer Fourier- und Wavelet-Transformation ist allein schon aufgrund ihrer Definitionen erkennbar:

– Fourier Transformation:

$$\tilde{f} := \int_{\mathbb{R}} f(t) e^{-2\pi i t} dt \qquad \text{für } f \in L^1$$

– Wavelet Transformation:

$$W_\Psi f(a, b) := |a|^{-1/2} \int_{\mathbb{R}} f(t) \overline{\Psi}((t-b)/a) dt \qquad \text{für } a, b \in \mathbb{R},\ a \neq 0 \text{ und } f, \Psi \in L^2$$

Bezogen auf die Kompression von Bildern liegen die Hauptvorteile einer Wavelet-Transformation in deren Eigenschaften im Frequenzraum. Verschieden breite Definitionsbereiche erlauben eine Beeinflussung des Verhältnisses von Zeit- und Frequenzauflösung:

– Eine Basis-Funktion mit einem breiten Definitionsbereich erlaubt die Transformation eines großen Signalbereiches und gibt die niederfrequenten Bildinformationen genauer wieder.
– Eine Basis-Funktion mit einem schmalen Definitionsbereich transformiert dagegen nur einen kleinen Signalbereich, wobei in diesem Fall das zeitliche Verhalten des Signals besser wiedergegeben wird:

Abb. 18: *Zeit- und Frequenzauflösung — FT, FWT, WT*

Verwendet man diesen Ansatz in der Bildkompression und entwirft ein Kompressionsschema auf der Basis einer Wavelet-Transformation, so ähnelt das resultierende Schema sehr stark Schemata, die auf der DCT basieren ($\Rightarrow$ JPEG).

– Wavelet-Kodierung:

Abb. 19: *Wavelet-Kodierung*

– Wavelet-Dekodierung:

Abb. 20: *Wavelet-Dekodierung*

Die Wavelet-Transformation und ihre Inverse lassen sich durch Anwendung eines Paares angepaßter QM-Filter (QMF = *quadrature mirror filter*) mit Operationen effizient implementieren. Die Implementierung einer Wavelet-Transformation als Anwendung eines speziellen Filters verdeutlicht auch, warum eine Wavelet-Kodierung als eine Art von Subband-Kodierung betrachtet werden kann.

Jedes QMF Paar besteht aus einem Tiefpaß-Filter und einem Hochpaß-Filter , welche die Bandbreite des Signals in zwei Hälften teilen. Die folgende Abbildung illustriert die Anwendung einer zweidimensionalen Wavelet-Transformation eines Bildes

durch die Anwendung zweier separabler eindimensionaler Wavelet-Transformationen:

Abb. 21: *Zweidimensionale Wavelet-Transformation*

Zuerst wird das Bild $f(x, y)$ entlang der X-Achse mit Hilfe der QM-Filter gefiltert, so daß ein Tiefpaß-Bild $f_L(x, y)$ und ein Hochpaß-Bild $f_H(x, y)$ entsteht. Da die Bandbreite der beiden gefilterten Bilder f_L und f_H entlang der X-Achse die halbe Bandbreite von f ist, kann entlang der X-Achse auch ein Downsampling um den Faktor 2 durchgeführt werden, ohne daß ein Informationsverlust auftritt. Im zweiten Schritt werden f_L und f_H dann entlang der Y-Achse gefiltert und in der Auflösung reduziert.

Die inverse Wavelet-Transformation stellt sich folgendermaßen dar, wobei beim Upsampling Nullen zwischen die Bildwerte entlang der jeweiligen Achse eingefügt werden:

Abb. 22: *Zweidimensionale Inverse Wavelet-Transformation*

Die Wavelet-Transformation dekorreliert die Pixelwerte des Originalbildes und konzentriert die Bildinformationen in einer relativ geringen Anzahl von Koeffizienten. Dies läßt sich z.B. anhand des folgenden Histogramms zeigen, welches einerseits die Verteilung der Pixelwerte des Originalbildes und andererseits die des transformierten Bildes zeigt:

Abb. 23: *Wavelet-Transformation: Verteilung der Pixelwerte*

Durch die Anwendung der Wavelet-Transformation gehen keine Bildinformationen verloren, sondern es wird nur eine andere Darstellung gewählt. Diese Darstellung der Bildinformationen benötigt nicht mehr Speicher wie die ursprüngliche Darstellung, besitzt allerdings verschiedene, für die Datenkompression vorteilhafte Eigenschaften. Jede Änderung der Koeffizientenwerte führt zu proportionalen Änderungen bei den rekonstruierten Pixelwerten. Das Löschen von kleinen Koeffizientenwerten mit einer Schwellenwert-Funktion $T(t, x)$ mit $T(t, x) = 0$ für $|x| < t$ und $T(t, x) = x$ sonst führt zu keinen signifikanten Verzerrungen im rekonstruierten Bild. Mit dem Schwellenwert t läßt sich somit der Kompressionswert steuern. Für höhere Kompressionsraten wird man vor einer Kodierung allerdings noch eine Quantisierung für jene Koeffizienten durchführen, die durch $T(t, x)$ nicht null geworden sind.

In neueren Arbeiten werden insbesondere bei hohen Kompressionsraten Vorteile von Verfahren auf der Basis einer Subband-Zerlegung mit Wavelets im Vergleich zu Verfahren auf der Basis von Transformationen von Pixelblöcken aufgezeigt. Mit einer Wavelet Subband-Zerlegung läßt sich im allgemeinen bis zur einer Kompressionsrate von 0,1bpp noch eine gute Bildqualität erreichen [Hilton94]. Diese subjektiv gute Bildqualität bei hohen Kompressionsraten beruht zum einen im wesentlichen auf der Eigenschaft von Wavelets, feine Bilddetails gut wiedergeben zu können und zum anderen darauf, daß Quantisierungsfehler aufgrund der Überlappung der Basisfunktionen nicht so sichtbar durch das HVS sind:

Abb. 24: *Beispielbilder*

6. Räumlich-zeitliche Transformationen

Die Verallgemeinerung bzw. Erweiterung der bisher besprochenen Techniken zur zweidimensionalen Bildanalyse im Bezug auf eine Analyse von Videosignalen entspricht der Erweiterung der Techniken um eine dritte zeitliche Dimension. Die Behandlung der dritten zeitlichen Dimension läßt dabei eine grundsätzliche Unterscheidung der Ansätze in zwei verschiedenen Klassen zu.

Die erste Klasse von dreidimensionalen Bildanalyse-Techniken stellt die direkte Erweiterung der zweidimensionalen Analyse-Techniken dar. Die zweidimensionalen Analyse-Techniken werden um eine dritte Dimension erweitert, was dazu führt, daß die Techniken zu Dekorrelation der Daten in allen drei Dimensionen in gleicher Art und Weise angewendet werden. Eine Unterscheidung beim Einsatz der Techniken aufgrund von Eigenschaften, die sich aus der räumlichen oder zeitlichen Struktur der Daten ergeben, entfällt vollkommen. Diese fehlende Berücksichtigung der unterschiedlichen Struktur der Daten im Bezug auf die zwei räumlichen und die eine zeitliche Struktur führt im wesentlichen auch zu dem Ergebnis, daß die Anwendung eines solchen Ansatzes nur moderaten Erfolg bei der Kompression von Videodaten erbringt.

Der Einsatz dieser Klasse von Analyse-Techniken ist vor allem bei dreidimensionalen Daten mit drei räumlichen Dimensionen, sogenannten Volumendaten, sinnvoll. Diese kommen vorallem in der Medizintechnik vor, wo nach Einsatz eines Bildgebungsverfahren, wie z.B. ein CT oder ein MRT, ein Objekt als eine Menge von Pixeln

mit drei räumlichen Dimensionen vorliegt. Gerade in einem medizinischen Umfeld muß man allerdings beim Einsatz von Kompressionsverfahren sehr umsichtig sein, vor allem deshalb, da es sich um verlustbehaftete Verfahren handelt.

Die zweite Klasse von dreidimensionalen Bildanalyse-Techniken versucht die verschiedenen Strukturen der Dimensionen für eine möglichst optimale Kodierung auszunutzen, was zum Einsatz von verschiedenen Analyse-Techniken für räumliche und zeitliche Dimensionen führt. Für prädiktive Verfahren ergeben sich aus dieser Überlegung heraus nun zwei Möglichkeiten zur Vorhersage von Pixeln eines Videosignals. Entsprechend dem zweidimensionalen Fall ist natürlich eine Vorhersage für einen Pixelwert aufgrund von Pixelwerten aus demselben Bild möglich, eine sogenannte räumliche Vorhersage (*spatial prediction*). Eine Vorhersage, die sich auf Pixelwerte aus vorangegangenen, schon gesendeten Bilder bezieht, wird als zeitliche Vorhersage (*temporal prediction*) bezeichnet [Netra80].

Auch für die Verfahren, die auf der Transformation von Pixelblöcken beruhen, wurde die Benutzung von dreidimensionalen Pixelblöcken sehr früh wieder verworfen [Netra80] [Roese77]. Die Gründe dafür liegen in der Summe der Nachteile einer solchen Verwendung:

- Durch die Einbeziehung der zeitlichen Dimension in die Transformation ergibt sich, in Abhängigkeit von der Ausdehnung der Pixelblöcke auf der zeitlichen Achse, eine zeitliche Verzögerung beim Senden und Empfangen des Videosignals.
- In Abhängigkeit von der zeitlichen Ausdehnung der Pixelblöcke ergeben sich auch zusätzliche Anforderungen an einen Pufferspeicher beim Sender und Empfänger der Daten.
- Entsprechend den Artefakten in den beiden räumlichen Dimensionen treten auch zeitliche Artefakte auf, die allerdings weitaus sichtbarer sind als die räumlichen.
- Die Performance ist relativ gering.

Auch für die anderen besprochenen zweidimensionalen Verfahren wurden dreidimensionale Erweiterungen vorgestellt, wie z.B. die dreidimensionale hierarchische [Croch83], Subband- [Karlsson88] und Wavelet- [Lewis90] Kodierung. Die Effizienz all dieser Ansätze ist allerdings durch die geringe Korrelation zwischen gleichen Bildbereichen bei aufeinanderfolgenden Bildern und dem Vorhandensein von Bewegung begrenzt. Dieses Problem versucht man mit der Bewegungs-Kompensation (*motion compensation*) zu begegnen.

Bewegungs-Kompensation (*motion compensation*)

Bewegungs-Kompensation, kurz MC für *motion compensation*, setzt Bildbereiche aufeinanderfolgender Bilder einer Videosequenz in Beziehung zueinander und versucht dann aufgrund dieser Beziehung eine möglichst effektive Kodierung zu erzielen. Die Hauptanwendung von MC ist die Vorhersage eines gegebenen Bildes aufgrund eines vorangegangenen Referenzbildes. Die heute gebräuchlichsten Verfahren zur MC gehen von einem sehr einfachen Bewegungsmodell aus und versuchen, für dieses Modell jene Parameter zu finden, welche die beste Vorhersage aus dem Referenzbild ermöglichen [Jain81b]. Fast alle heute verwendeten Bewegungsmodelle definieren eine Bewegung als eine reine Translation eines Bildbereiches. Die Konsequenz aus diesem Modell ist,

daß die Vorhersage eines Pixelblockes in einem Bild als Verschiebung eines gleichgroßen Pixelblockes aus dem Referenzbild kodiert wird. Die Beziehung zwischen den beiden Pixelblöcken wird durch die Verschiebung hergestellt und mit einem Bewegungsvektor (*motion vector*) kodiert. Anstelle des verschobenen Pixelblockes wird dann der Bewegungsvektor gespeichert bzw. gesendet:

Abb. 25: *Pixelblock basierende MC*

Bei *very low bit-rate* (VLBR)-Anwendungen stellt sich die Kodierung der Bewegungsvektoren allerdings als Problem heraus, da mit deren Übertragung auch die Übertragung weiterer zusätzlicher Daten notwendig ist, was bei VLBR-Anwendungen nicht immer gewährleistet werden kann.

Innerhalb dieses Bewegungsmodells ist die Hauptaufgabe die Berechnung der Bewegungsvektoren. Fast alle heutzutage angewendeten Methoden beruhen dabei auf dem Prinzip des *block matching*. Ausgehend von einem zu kodierenden Pixelblock im aktuellen Bild einer Videosequenz wird beim *block matching* in vorangegangenen Bildern nach einem gleichgroßen Pixelblock gesucht, der dem zu kodierenden Pixelblock möglichst ähnlich ist. Als Maß für die Ähnlichkeit von zwei Pixelblöcken dienen meist Maße wie z.B. der mittlere quadratische Fehler zwischen den beiden Pixelblöcken.

Abb. 26: *block matching*

Durch den Einsatz von Hardware ist mittlerweile sogar eine erschöpfende Suche im Referenzbild möglich, wobei eine Kodierung mit Hilfe von MC allerdings noch immer sehr stark von äußeren Parametern wie z.B. maximale Bit-Rate, Real-Zeit oder erforderliche Bildqualität geprägt wird.

Hybride Kodierung

Durch den Einsatz von prädiktiven Verfahren in der zeitlichen Dimension und Transformationsverfahren in den beiden räumlichen Dimensionen erhält man eine leistungsfähige Kodierungsmethode, wie sie heute in fast allen Kompressionssystemen für Videosequenzen eingesetzt wird. Eine Ausnahme bilden da nur jene Verfahren, die sich auf eine reine Intraframe-Kodierung beschränken, wie z.B. die momentan im PC Umfeld sehr populären MJPEG-Systeme.

Kodierungen, die prädiktive Verfahren für die zeitliche und Dekorrelationsverfahren für die räumlichen Dimensionen verwenden, werden deshalb auch hybride Kodierungen (*hybrid coding*) genannt. Diese Namensgebung ist im Umfeld der Kompression von Bildinformationen nicht eindeutig und steht z.B. im Kontext reiner Intraframe-Kodierung für die Kombination einer Entropie- und einer Transformations-Kodierung.

Die Integration von MC-Verfahren und räumlicher Verfahren zur Dekorrelation ist leicht, wenn beide Methoden mit sich nicht überschneidenden Pixelblöcken arbeiten. In diesem Falle muß nur entschieden werden, ob der zu kodierende Pixelblock nicht prädiktiv (intraframe) kodiert werden soll, weil keine passende Prädiktion gefunden wurde, oder ob er prädiktiv (interframe) kodiert werden kann. Im letzteren Fall wird der Vorhersagefehler vor der Kodierung und dem Senden mit dem Bewegungsvektor noch räumlich dekorreliert, auch wenn dies nach [Ronda90] nicht immer effektiv ist:

Abb. 27: *Intra- und Interframe-Kodierung am Beispiel MPEG*

Die Transformation der Pixelblöcke wird bei der Interframe-Kodierung in derselben Art und Weise wie bei der Intraframe-Kodierung angewendet, so daß von den dort vorgenommenen Optimierungen, wie z.B. bei der Quantisierung, auch hier profitiert werden kann.

Prinzipiell arbeiten hybride Kodierungsschemata dabei mit drei Abstraktionsebenen für Daten aus einer Videosequenz:

Pixelblock: (pixel block)
Ein Pixelblock ist eine Menge von Pixeln, auf die die Transformation, im allgemeinen die DCT, angewendet wird. Die einzelnen Pixelwerte bestehen aus den Farbwerten des Bildes bezüglich einer bestimmten Farbdarstellung. Meist ist dies die YC_bC_r-Darstellung, da diese Darstellung eine leichtere an das HVS angepaßte Bearbeitung zuläßt, wie z.B. die Darstellung im RGB-System.

Makroblock: (macro block)
Zur besseren Handhabung und Bearbeitung werden die Pixelblöcke im allgemeinen in sogenannte Makroblöcke gruppiert. Ein Makroblock stellt einen zusammenhängenden Bildausschnitt des zu kodierenden Bildes dar, wobei die Größe und Ausdehnung von dem speziellen Kompressionssystem abhängen.

Bild: (picture)
Die größte Informationseinheit ist ein einzelnes Bild, auch *field* oder *frame* genannt.

Für jede dieser verschiedenen Abstraktionsebenen existieren spezifische Verfahren zur Ver- und Bearbeitung.

Bild-Kodierung

Unter einer Kodierung von Bildern aus einer Videosequenz versteht man in diesem Kontext die Ver- und Bearbeitung der Videodaten auf dem Niveau ganzer Bilder:

Abb. 28: *Bild-Kodierung*

Die erste Verarbeitungseinheit in diesem Schema, die “Bildumordnung”, ist für jene Bilder notwendig, deren Referenzbilder in der Zukunft liegen. Bei diesen Bildern stimmt die Verarbeitungsreihenfolge nicht mehr mit der zeitlichen Reihenfolge überein, da für die Kodierung und Dekodierung des aktuellen Bildes auf zukünftige Bilder zugegriffen werden muß. Dies hat dann natürlich zur Folge, daß ein solches System einen

Bildpuffer besitzen muß, in dem all jene Bilder gespeichert sind, welche als Referenzbild für das aktuell zu kodierende bzw. dekodierende Bild dienen können. Dieser Puffer ist das *prediction memory*, in welchem je nach Komplexität des Gesamtsystems ein oder mehrere mögliche Referenzbilder gespeichert sind. Die Verarbeitung eines Bildes besteht nun aus der Abarbeitung aller Verarbeitungsschritte für alle enthaltenen Makroblöcke und daraus, daß eine dekodierte Version des aktuellen Bildes im Puffer gespeichert wird, so daß es für folgende Bilder als Referenzbild dienen kann.

Makroblock-Kodierung

Die Verarbeitung eines Bildes auf der Basis von Makroblöcken ist die Schicht, auf der die Verfahren zur Prädiktion angewendet werden:

Abb. 29: *Makroblock-Kodierung*

Zuerst werden für den zu kodierenden Makroblock eine Menge von möglichen Prädiktionen aus dem *prediction memory* ausgesucht. Jede dieser Prädiktionen besteht aus einem Makroblock, der entweder aus einem einzelnen Bild stammt oder eine Linearkombination von zwei Makroblöcken aus zwei verschiedenen Bildern darstellt. Die Pixelwerte der Makroblöcke für die Prädiktion werden direkt aus den Bildern extrahiert oder durch eine lineare Interpolation gewonnen, so daß die Suche durchaus eine Auflösung von weniger als einem Pixel haben kann.

Nachdem die beste Prädiktion gefunden wurde, wird diese vom zu kodierenden Makroblock abgezogen, so daß man einen Makroblock mit dem Prädiktionsfehler erhält. Dieser Makroblock wird anschließend kodiert und gesendet. Der Makroblock mit dem Prädiktionsfehler kann für den Fall, daß keine Prädiktion gefunden wurde, auch aus dem Ausgangs-Makroblock selbst bestehen (Intraframe-Modus). Zusätzlich zum Makroblock werden noch der Prädiktionsmodus und der Bewegungsvektor kodiert und gesendet. Abschließend wird der Makroblock vom Kodierer in derselben Weise dekodiert, wie vom Dekodierer beim Empfangen ausgeführt wird. Dieses Ergebnis wird dann im *prediction memory* für nachfolgende Bilder zur Prädiktion gespeichert.

Pixelblock-Kodierung

Die Verarbeitung auf Pixelblock-Niveau startet mit einer zweidimensionalen DCT des Makroblockes mit dem Prädiktionsfehler:

Abb. 30: *Pixelblock-Kodierung*

Nach der Quantisierung werden die Koeffizienten noch Entropie-kodiert, um sie möglichst kompakt zu speichern bzw. zu versenden. Für die Speicherung des Makroblockes im *prediction memory* muß der quantisierte Pixelblock wieder rekonstruiert und invers transformiert werden.

Prädiktionsschemata (*prediction schemes*)

Während die ersten Prädiktionsschemata auf zeitlich stationären Schemata (*time stationary prediction schemes*) beruhten und z.B. immer nur den direkten Vorgänger eines Bildes für dessen Prädiktion verwendeten, sind die heutigen Systeme in Bezug auf eine zeitliche Prädiktion variabler geworden. Sie erlauben eine Prädiktion für ein Bild aus einer Menge von Bildern, die mit einem bestimmten zeitlichen Abstand in der Vergangenheit oder auch der Zukunft zum kodierenden Bild liegen. Für eine effektivere Bearbeitung einer Videosequenz und die Anwendung eines Algorithmus zur Interframekompression wird eine Videosequenz somit in Bildgruppen aufgeteilt:

Abb. 31: *Aufteilung einer Videosequenz in Bildgruppen*

Diese Bildgruppen haben im allgemeinen alle die gleiche systembedingte Anzahl von Bildern. Die Gruppen werden zusammenhängend komprimiert, so daß der direkte Zugriff auf einzelne Bilder einer Gruppe aus den komprimierten Daten nicht mehr ohne weiteres möglich ist. Ein direkter Zugriff besteht nur noch auf das erste Bild einer Bildergruppe. Dabei hat es sich als sinnvoll erwiesen, etwa drei Zugriffsmöglichkeiten pro Sekunde zu ermöglichen, was (je nach Videonorm) zu acht bis zehn Bildern pro Gruppe führt.

Bildtypen (frame types)

Innerhalb der Bildgruppen werden die einzelnen Bilder je nach Art ihrer Kodierung und damit auch nach ihren Eigenschaften unterschieden. Das Anfangsbild einer jeden Bildgruppe wird nur einer Intraframe-Kompression unterworfen und heißt deshalb Intra-Frame (*I-Frame*). Es wird also keiner zeitlichen Prädiktion unterworfen und nur als Standbild komprimiert und dekomprimiert.

Die Kompression aller anderen Bilder der Gruppe erfolgt mit dem I-Frame als Referenzbild. Aufgrund der Art der zeitlichen Prädiktion unterscheidet man bei den auf das I-Frame folgenden Bildern einer Bildgruppe zwei Typen, die sogenannten P- und die B-Frames.

Abb. 32: *I-Frames*

Die P-Frames (*predicted picture*) benutzen das vorangegangene I-Frame oder ein vorangegangenes P-Frame als Referenzbild, eine Prädiktion findet also nur mit Bezug auf die Vergangenheit statt.

Abb. 33: *P-Frames*

Zwischen I- und P-Frames oder zwischen P-Frames können noch die sogenannten B-Frames (*bidirectional picture*) liegen. Bei der Kodierung der B-Frames können vorangegangene und nachfolgende Bilder als Referenzbilder dienen, so daß sich eine zeitliche Prädiktion sowohl auf die Vergangenheit als auch auf die Zukunft beziehen kann.

Abb. 34: *B-Frames*

Die Einteilung der einzelnen Bilder in I-, P- und B-Frames bedeutet nicht, daß alle Makroblöcke eines Bildes mit einem bestimmten Prädiktionsschema kodiert sind. Die Eigenschaft eines Bildes, ein P- oder B-Frame zu sein, heißt nur, daß mindestens ein Makroblock mit dem entsprechenden Prädiktionsschema kodiert ist:

Abb. 35: *Referenzkodierung*

Die erreichbare Kompressionsrate und allerdings auch der Aufwand zur Kodierung ist bei den B-Frames am größten, gefolgt von den P- und I-Frames. Die Referenzkodierung ist im allgemeinen im Hinblick auf die Kompressionsrate sehr effektiv, benötigt aber umfangreiche Such- und Vergleichsoperationen, um die Referenzbilderfelder minimaler Differenz zu finden. Die Kompression unter Nutzung der Interframekompression ist damit wesentlich rechenintensiver als die Dekompression, so daß im allgemeinen auf diesem Verfahren basierende Systeme im Bezug auf ihr Zeitverhalten asymmetrisch sind.

Die Anwendung der Interframe-Kompression hat, wie oben schon kurz erwähnt, auch zur Folge, daß die ursprüngliche Bildreihenfolge umgestellt werden muß. Bei der Dekodierung benötigt man zuerst das I-Frame, dann das nächste P-Frame, und mit diesen beiden Frames können die dazwischen liegenden B-Frames dekodiert werden. Ein direkter Zugriff ist nur auf die I-Frames möglich, denn nur sie können für sich genommen dekodiert werden. Sie bilden deshalb die Zugriffspunkte (*access points*) in der komprimierten Bildfolge.

Abb. 36: *Umstellung der Bildreihenfolge bei der Interframekompression*

7. Quantisierung

Die Quantisierung stellt den eigentlichen Verarbeitungsschritt zur Reduktion der Datenmenge und damit zur verlustbehafteten Kompression dar. Man unterscheidet bei der Quantisierung zwischen der skalaren und der Vektor-Quantisierung. Bei der skalaren Quantisierung wird jedes Sample einzeln quantisiert, während bei der Vektor-Quantisierung mehrere Samples zu einem Vektor zusammengefaßt und dann durch einen Näherungsvektor ersetzt werden.

Skalare Quantisierung

Der Algorithmus zum Entwurf eines, im Sinne des mittleren quadratischen Fehlers, optimalen skalaren Quantisierers wurde von Lloyd und Max unabhängig voneinander entwickelt [Lloyd82] [Max60]. Die Optimalität in Bezug auf die Verzerrung des Signals durch den Quantisierer ist für den Lloyd-Max-Quantisierer für den Fall gegeben, daß der quantisierte Wert anschließend mit einer festen Kodewortlänge kodiert wird. Für den Fall, daß die sich anschließende Kodierung eine Entropie-Kodierung ist, kann gezeigt werden, daß der im obigen Sinne optimale Quantisierer der Uniforme Quantisierer ist [Jayant84]. Diese Tatsache und die einfache Implementierbarkeit sind die Hauptgründe dafür, daß in fast allen heutigen Standards zur Bildkompression uniforme Quantisierer verwendet werden. Ein weiterer Grund für den Einsatz eines uniformen Quantisierers kann die Notwendigkeit einer Bitraten-Kontrolle sein, welche den Einsatz eines sehr variablen Quantisierers verlangt.

Vektor-Quantisierung

Die Quantisierung von ganzen Blöcken von Samples anstelle der Quantisierung einzelner Samples wird Vektor-Quantisierung (VQ) genannt [Gray84]. Diese Art der Quantisierung besitzt die Eigenschaft, räumliche Redundanz von Bildinformationen für eine kompakte Kodierung ausnutzen zu können. Setzt man die VQ zur Kodierung von Videosignalen ein, so kann dies sowohl für eine Intra- als auch für eine Interframe-Kodierung geschehen.

Eine Vektor-Quantisierung basiert auf der Wahl einer endlichen Menge von Vektoren als Repräsentanten für den gesamten Coderaum. Diese Vektoren, auch Codebuch (*codebook*) genannt, zerlegen den Eingaberaum in einzelne Zellen entsprechend einer Minimal-Abstands-Bedingung. Für jeden Eingabeblock wird jener Vektor aus dem Codebuch gewählt und gespeichert, der dem Eingabeblock am nächsten ist. Die Leistungsfähigkeit einer VQ hängt somit von der richtigen Wahl des Codebuches ab. Für die Bestimmung eines optimalen Codebuches muß allerdings die Verteilung der Eingabeblocks bekannt sein. Dies ist allerdings im allgemeinen nicht der Fall, so daß auf Algorithmen zur Bestimmung eines suboptimalen Codebuches zurückgegriffen werden muß, wie z.B. den LBG-Algorithmus [Gray84], welcher aus einer Menge von Trainingsdaten ein solches Codebuch berechnet.

Vektor-Quantisierung ist im allgemeinen ein sehr aufwendiger Prozeß, so daß verschiedenste Ansätze und Vorschläge für eine schnelle Bestimmung eines Codebuches existieren, wie z.B. die hierarchische VQ [Gray84]. Ist die Quantisierung zwar aufwendig, so gestaltet sich die Rekonstruktion jedoch sehr einfach, da sie im wesentlichen nur aus einem Zugriff auf das berechnete und bekannte Codebuch besteht.

Die Transformationsverfahren auf der Basis von sich nicht überschneidenden Pixelblöcken können auch als eine VQ interpretiert werden, wobei das Codebuch indirekt durch die Basisvektoren der gewählten Transformation und die anschließende skalare Quantisierung bestimmt werden. Im Vergleich der beiden Ansätze zeigen sich dann dementsprechend auch hier dieselben Kompressionsartefakte, wie der *blocking effect* und eine schlechte Kantenwiedergabe. Auch in diesem Falle treten die Artefakte vor allem bei niedrigen Bit-Raten auf. Ansätze, diese Artefakte zu vermindern, existieren

auch hier. Die Kantenwiedergabe läßt sich z.B. durch die Einführung eines Extra-Codebuches für Kanten verbessern (*classified VQ*, [Nasra88]). Die Idee der Einführung von extra Codebüchern für spezielle Bildeigenschaften ist wohl auch der Grund dafür, daß in einigen neueren Proposals zum noch zu definierenden Standard MPEG-4 die VQ zur Kodierung von Texturen vorgeschlagen wird.

Trotz einiger positiver Eigenschaften führte vor allem die hohe Komplexität bei der Berechnung dazu, daß bis heute in allen standardisierten Verfahren zur Kompression von Bildern eine skalare Quantisierung verwendet wird.

8. Kodierung

Die Ausgabe des Quantisierers kann als eine Sequenz von Symbolen mit einer festen Bitrate angesehen werden. Desweiteren wird für den Entwurf eines verlustlosen Kodierers davon ausgegangen, daß diese Symbole durch den vorgeschalteten Transformationsschritt unabhängig gemacht wurden. Damit ist prinzipiell jede Technik zur Kodierung von unabhängigen Symbolen für den Kodierer einsetzbar.

Ein einfacher und oft praktizierter Ansatz sind die *variable length codes* (VL-Codes), wie z.B. die Huffman-Kodierung, die für unabhängige Symbole mit einer Wahrscheinlichkeit von $(1/2)^n$ für $n = 0, 1, \ldots$ sogar eine optimale Kodierung ist. Für konkrete Anwendungen sind Modifikationen von VL-Codes sinnvoll, die unter Berücksichtigung einer Optimierung des Zusammenspiels zwischen Quantisierer und Kodierer entworfen werden sollten. Ein Beispiel dafür ist der Einsatz einer RL-Kodierung nach der DCT der Pixelblöcke beim JPEG Algorithmus (Siehe *Verlustfreie Kompression* auf Seite 23 und [ISO94b] [ISO94b]).

Ein weiterer, wenn auch mit höherem Rechenaufwand verbundener Ansatz ist die Arithmetische Kodierung [Witten87]. Ein Vorteil der Arithmetischen Kodierung liegt in der Eigenschaft, für nicht vollständig unabhängige Symbole, wie sie bei einer nicht ganz optimalen Transformation auftreten, die entsprechende Entropie höherer Ordnung anzunähern. Aus diesem Grunde findet sie auch immer mehr Anwendung in neueren Standards [ISO94b] [ITU95].

9. Datenfluß-Kontrolle

In einem System zur Kompression von Bilddaten ist das Modul zur Bitraten-Kontrolle verantwortlich für die Anpassung der Ausgabe des Kodierers an die vorgegebenen Parameter des Übertragungs- oder Speicherungssystems, wie z.B. die Bandbreite eines Netzwerkes. Eine Hauptaufgabe dabei ist die Kontrolle über die Sende- und Empfangspuffer, um sie vor einem Überlauf und somit vor dem Verlust von Daten zu schützen. Ein Verlust von Daten bedeutet auf der Empfängerseite den totalen Verlust von Bild- oder Tonsegmenten aus der Videosequenz und damit eine sehr starke Qualitätsminde-

rung. Der Kontrollmechanismus ist darüber hinaus für eine effiziente Zuweisung der verfügbaren Bandbreite zuständig, um eine möglichst optimale Qualität zu erhalten. Die Einführung eines Puffers führt natürlich auch zu einer Verzögerung im Sendeablauf, die proportional zur Größe des Puffers ist. Diese Verzögerungen, und damit auch die Puffer, können nicht beliebig groß werden, sondern werden von den jeweiligen Anforderungen an das Sende- und Empfangssystem in ihrer maximalen Größe vorgegeben.

Die Kontrolle der Bitrate wird im allgemeinen durch entsprechende Variationen bei der Quantisierung erreicht, obwohl sich auch andere Parameter des Gesamtsystems, wie z.B. die Bildwiederholrate oder die Bildauflösung dafür eignen [ITU93a] [ITU95]. Die Parameter zur Bitraten-Kontrolle müssen dann vom Kodierer auch zum Dekodierer übertragen werden. Dies schränkt die Art und den Umfang der Kontrollparameter natürlich je nach Kompressionssystem stark ein, da die dafür verwendete Bandbreite von der für die Videodaten zur Verfügung stehenden Bandbreite abgezogen werden muß. Fast alle eingesetzten Kontrollalgorithmen arbeiten dabei heuristisch, obwohl auch Arbeiten existieren, die einen Kontrollmechanismus aufgrund von stochastischen Eigenschaften der Daten implementieren [Ronda90].

10. Resultierende Eigenschaften und mögliche Erweiterungen Pixel-basierter Methoden

Aus den bisherigen Ausführungen über die verschiedenen Techniken und Verfahren zur Kompression von Bildinformationen lassen sich grundlegende Eigenschaften für pixelverarbeitende Systeme herausarbeiten. Während einige der nachfolgenden Eigenschaften und Einschränkungen im Pixel-basierten Ansatz selbst begründet liegen, sind andere Eigenschaften durch spezielle Implementierungen in den aktuellen Systemen bedingt. Einschränkungen und Eigenschaften, die durch spezielle Implementierungen bedingt sind, lassen sich innerhalb des Pixel-basierten Ansatzes durch die weitere Modifikation von Systemparametern oder den Einsatz von neuer Hardware bestimmt noch aufheben, so daß die Entwicklung dieses Ansatzes sicher noch nicht abgeschlossen ist. Die durch den Ansatz bedingten Einschränkungen wird man allerdings nur durch die Entwicklung und den Einsatz von neuen Methoden und Techniken aufheben können.

Prinzipiell ist der Ansatz, auf der Basis von Pixeln zu arbeiten, auch der allgemeinste Ansatz, da er nur rein statistische Bedingungen an den Inhalt der Bilder bzw. Bildsequenzen stellt. Allerdings reichen diese Minimalanforderungen auch schon aus, um das System zu begrenzen. So ist die Qualität von prädiktiven und Transformations-Verfahren bei Bildern mit weichen Übergangen in den Bildern wesentlich besser, als bei Bildern, die scharfe Kanten und Ecken beinhalten, womit diese Verfahren für die primäre Verarbeitung dieser Art von Bildern nicht so geeignet sind.

Vom Gesichtspunkt der Kontrollierbarkeit von Systemparametern zur Qualitätskontrolle besitzen die heutigen Systeme durchaus gute Eigenschaften, zumindest solange man sie innerhalb ihrer Spezifikationen betreibt. Außerhalb dieser Bereiche zeigen

diese Systeme dann allerdings einen sehr starken und nicht mehr zu kontrollierenden Qualitätsverlust. Kontrollierbar ist die Qualität der Systeme dabei nur indirekt über die Kontrolle der Ausgabe-Bitrate des Gesamtsystems. Diese Kontrolle des Systems ist solange einfach, wie die Qualitätskurve auch für steigende Kompressionsraten keine großen Schwankungen oder Einbrüche verzeichnet.

Der Hauptparameter für die Qualitätskontrolle ist dabei die Quantisierung. Durch Modifikation der Quantisierungsstufen des uniformen Quantisierers läßt sich das Verhältnis von Kompressionsrate zur Qualität steuern. Da die Quantisierung der Hauptkontrollparameter ist, folgt auch sofort, daß beim Einsatz von Transformationsverfahren die Qualität viel leichter zu regulieren und kontrollieren ist, als beim Einsatz von prädiktiven Verfahren. Dies liegt darin begründet, daß bei den prädiktiven Verfahren die Quantisierung in die Prädiktionsschleife integriert ist, so daß eine Änderung der Quantisierungsstufen auch Auswirkungen auf die Prädiktion der folgenden Bilder bzw. Daten besitzt. Je gröber die Quantisierung wird, umso schneller und unberechenbarer fällt dann die Qualitätskurve.

Der Ansatz, mit einer disjunkten, vollständigen und gleichmäßigen Partitionierung von Bildern zu arbeiten, hat auch den Nachteil, daß alle Pixelblöcke gleich kodiert werden, unabhängig davon, ob sie z.B. Ecken und Kanten enthalten und deshalb mit mehr Bits kodiert werden müßten. Eine adaptive Partitionierung ohne feste Blockgrößen hat dagegen den Vorteil, daß größere, gleichmäßige Bildbereiche als Einheit und damit mit weniger Bits und einer besseren Kompressionsrate kodiert werden können. Eine analoge Problemstellung erhält man bei der Kodierung von Videosignalen auch für die zeitliche Dimension.

Für die zeitliche Dimension ergibt sich darüber hinaus aus der Kodierung von disjunkten, beziehungslosen Pixelblöcken ein weiteres Problem. Auch wenn der Begriff “Bewegungskompensation” es nahe legen sollte, so werden doch keine Bewegungen im Sinne von bewegten Gegenständen kodiert. Durch die Zerlegung der Bilder in Pixelblöcke werden nur die Bewegungen von einzelnen Pixelblöcken kodiert, ohne deren Zugehörigkeit zu einem bestimmten Gegenstand zu berücksichtigen. Bei der Kodierung der Bewegungsvektoren kann es dann passieren, daß Pixelblöcke, die zu einem Gegenstand gehören, mit unterschiedlichen Bewegungsvektoren kodiert werden. Bei einer anschließenden Dekodierung führt dies dazu, daß der Gegenstand nicht mehr kontinuierlich und zusammenhängend dargestellt wird und die fehlerhaft kodierten Pixelblöcke sichtbar werden.

11. Schlußfolgerungen

Auf der einen Seite sind zu den aktuellen Techniken und Verfahren zur Kompression von Bildinformationen die verschiedensten Ansätze zur Verbesserung und Erweiterung der de facto-Standards denkbar. Auf der anderen Seite lassen sich aufgrund der gemeinsamen Eigenschaften der heutigen Standards ihre Qualitätseigenschaften und damit

auch ihr Einsatzgebiet sehr genau bestimmen. Basierend auf den Ausführungen in diesem Kapitel läßt sich der folgende Zusammenhang zwischen Qualität und Kompressionsrate herstellen:

Qualität	Intraframe	Interframe
Sehr Gut	0,75 -	0,4 -
Gut	0,25 - 0,75	0,125 - 0,4
Befriedigend	0,125 - 0,25	0,05 - 0,125

Abb. 37: *Zusammenhang Qualität und Kompressionsrate (bpp)*

Für die Intraframe-Kodierung beziehen sich die angegebenen Werte auf den JPEG-Algorithmus. Bei Kompressionsraten über 0,4bpp läßt sich bei einer Interframe-Kodierung eine sehr gute Qualität in Form einer Standard TV-Auflösung mit Hilfe einer MPEG2-Kodierung erzielen. Gute Qualität für Kompressionsraten im Bereich von 0,125 bis 0,4bpp lassen sich durch eine MPEG1-Kodierung erreichen, während für Kompressionsraten von 0,05 bis 0,125bpp die H.263-Kodierung immerhin noch moderate Qualitäten für Bild-Telephon und Video-Konferenzen liefert.

Eine Leistungssteigerung dieser Systeme durch die Modifikation einzelner Bestandteile ist möglich und wahrscheinlich. Ein Hauptaugenmerk wird dabei auf die Abkehr vom Paradigma der Transformation von disjunkten Pixelblöcken gerichtet werden. Die Konsequenz daraus ist natürlich auch die Notwendigkeit der Entwicklung von neuen Verfahren zu MC, da die bisherigen Verfahren die disjunkte Zerlegung der Bildinformationen voraussetzt. So ist z.B. ein einfacher Austausch der DCT durch eine DWT in einem MPEG-System nicht möglich, sondern bedingt auch die Entwicklung und Integration neuer MC-Techniken.

Analysen von Bitraten für *very low bitrate*-Anwendungen (VLB-Anwendungen) haben gezeigt, daß die Kodierung der Bewegungsinformationen in der heutigen Form bei diesen Anwendungen einen sehr hohen Anteil an der Gesamtbitrate besitzt. Aufgrund dieser Erkenntnisse wurden mehrere Ansätze zur Verbesserung der Prädiktionsschleife vorgestellt, wie z.B. die *control grid interpolation* (CGI [Sullivan91] [Wang94]) und die *overlapped block MC* (OBMC [Orchard94]).

Die Entwicklung der aktuellen Kompressions-Standards ist auch aufgrund dieser Ansätze noch nicht abgeschlossen, so daß die obigen Angaben über die Qualität und die Kompressionsraten eher als eine empirische Momentaufnahme als prinzipielle Systemgrenzen zu verstehen sind.