



Lehrstuhl für Informatik II
Prof. Dr. Jürgen Albert
Dipl.Math. Stefan Frank

Seminar Bilddatenkompression

MPEG-2 Kompression

für

Videosequenzen

Marius Heuler

Sommersemester 1995

Inhaltsverzeichnis

1	Einführung	3
1.1	Gründe für die Entwicklung	3
2	Der MPEG-2 Standard	3
2.1	Anforderungen	3
2.2	Einsatzgebiete von MPEG-1	4
2.3	Erweiterung bei MPEG-2	4
2.4	Einsatzgebiete von MPEG-2	4
3	Aufbau des MPEG-Standards	5
3.1	Grundgerüst	5
3.2	Funktionsweise	5
3.3	Technischer Aufbau	9
3.4	Bildkodierung	10
3.5	Funktion eines Kodierers	15
3.6	Funktion eines Dekoders	15
3.7	Bitstromsyntax	16
4	Einschränkungen/Grenzen	18

1 Einführung

1.1 Gründe für die Entwicklung

Im Zeitalter des Multimedia-Computers wurde bald auch nach einer Möglichkeit zur Darstellung von Filmen/Videos auf Computern gesucht. Falls man aber die Datenraten betrachtet, die für unkomprimierte digitale Videodarstellungen nötig sind (z.B. Fernsehen: 290 MBit/s – 370 MBit/s), wird schnell klar, daß eine Komprimierung nötig ist. Die mögliche Datenrate wird dabei durch die Möglichkeiten der vorhanden Hardware bestimmt (z.B. CD-ROM ≥ 1.5 MBit/s, Netze 2 – 10 MBit/s, Festplatten ≥ 10 MBit/s). Eine verlustfreie Komprimierung, mit der etwa ein Kompressionsfaktor von 2 bis 4 erreicht werden kann (z.B. ZIP - Format), reicht daher nicht mehr aus. Eine stärkere Komprimierung kann aber nur mit einer verlustbehafteten Komprimierung erreicht werden. Folglich mußte ein Verfahren gefunden werden, das bei möglichst guter Qualität eine vorgegebene Kompressionsrate erreicht. Der bekannteste Standard, der dafür entwickelt wurde, ist der sog. MPEG (Moving Pictures Expert Group) Standard. Der nun in der Version 2 vorliegende MPEG-Standard beschreibt Verfahren zur digitalen Kompression und Übertragung von Video und Audio über verschiedene Medien. In diesem Seminar über Bilddatenkompression wird auf die Audioübertragung nicht weiter eingegangen.

2 Der MPEG-2 Standard

2.1 Anforderungen

Der zu entwickelnde Standard sollte dabei folgende Kriterien erfüllen:

- Bildqualität wie PAL/NTSC Fernsehen
- Optional auch HDTV Fernsehen
- Eignung auch für langsame Peripherie (z.B. CD-ROM)
- Codiermodus mit geringer Verzögerung $\Delta t \leq 150ms$ (z.B. Videokonferenz)
- direkter Zugriff auf einzelne Bilder
- Eignung auch für „one-way“ Datenwege
- Robustheit gegenüber Übertragungsfehlern
- Kompatibilität
- hohe Flexibilität in Bezug auf
 - Qualität
 - Datenrate

- Bildgröße

Natürlich können hierbei nicht alle Ziele gleich gut erreicht werden, da sie sich teilweise entgegenwirken.

2.2 Einsatzgebiete von MPEG-1

Der schon weit verbreitete MPEG-1 Standard mit einer Datenrate von meist 0.5 bis 1.5 MBit/s wird im Moment schon für folgende Aufgaben eingesetzt.

- CD-I (Computer Interaktiv) Spielekonsole von Philips
- Video-CD
- Computerspiele
- Elektronische Datenbanken
- Aus-/Fortbildung

2.3 Erweiterung bei MPEG-2

Wie man sieht, wird der MPEG-1 Standard hauptsächlich im Heim- und Spielbereich verwendet. Um auch weitere Möglichkeiten im professionellen Bereich und bei zukünftigen Anwendungen zu erschließen, wurde der MPEG-1 Standard überarbeitet und erweitert, um nun als MPEG-2 von der ISO/IEC (International Organisation for Standardisation) verabschiedet zu werden. Die wichtigsten Neuerungen beim MPEG-2 Standard sind:

- Codierung auch von Halbbildfolgen möglich (Interlace)
- Skalierbarkeit (verschiedene Qualitätsstufen in einem Videostrom)
- verschiedene Abtastraten der Farbkomponenten (4:2:0, 4:2:2 und 4:4:4)

2.4 Einsatzgebiete von MPEG-2

- digitale Fernsehübertragung
 - über Satellit
 - über terrestrische Strecken
 - über Breitbandkabel („one way“)
 - über Netze (ATM)
 - Studio zu Studio Überspielung in Profiqualität
- „Video on demand“
- interaktives Fernsehen
- Videokonferenzen

3 Aufbau des MPEG-Standards

3.1 Grundgerüst

Obwohl im MPEG2-Standard der Datenstrom und die Dekodierung sehr genau beschrieben sind, bietet er für die Erzeugung des Videostroms nahezu volle Freiheit in Bezug auf Bildgröße, Bildrate, Kompression usw.. Damit nicht jeder Dekoder alle Möglichkeiten des für einen sehr weiten Einsatzbereich definierten Standards beherrschen muß, wurden Untermengen des Standards als sogenannte **Profiles** definiert. In einem Profile darf nur eine genau definierte Untermenge der Datenstrom-Syntax verwendet werden, was die Implementierung für bestimmte Zwecke sehr erleichtert - man vergleiche z.B. die Anforderungen von HDTV-Übertragung über ein ATM-Netz mit Videospiele auf CD-ROM. Da auch in einem Profile noch sehr große Freiheiten z.B. bei der Bildgröße vorhanden sind, wurden noch **Levels** eingeführt, die bestimmte Grenzwerte bei den Bildparametern festlegen.

Die volle Syntax kann in zwei Hauptteile aufgeteilt werden. Und zwar:

- **nicht-skalierbare Syntax**

In der nicht skalierbaren Syntax wird der Videostrom über die drei Frametypen I, P und B kodiert, wobei der Frametyp frei nach den Anforderungen ausgewählt werden kann. Eine Erweiterung ist für Zeilensprung-Videos (PAL, NTSC Fernsehen) vorgesehen, wobei freigestellt ist, ob die beiden Halbbilder getrennt oder als ein Vollbild kodiert werden.

- **skalierbare Syntax**

Mit der in MPEG-2 neu eingeführten Skalierbarkeit werden vor allem Anwendungen bei Fernsehübertragungen erweitert. Der Videostrom besteht dann nicht nur aus einem Layer, sondern aus mehreren, die z.B. verschiedene Qualitätsstufen oder Formate enthalten. Der Dekoder kann sich dann je nach Fähigkeit/Anforderung des Darstellungssystems die nötigen Daten aus dem Videostrom holen. Der große Vorteil der skalierbaren Syntax gegenüber Übertragung von mehreren unabhängigen Videostreamen ist die Datenreduktion, da die Daten in den einzelnen Layern auf den Grundlayer aufbauen.

3.2 Funktionsweise

Um die geforderte hohe Kompression bei trotzdem guter Bildqualität zu erreichen, benutzt der Standard zwei Verfahren:

- Das Komprimieren von kompletten Einzelbildern mit einem blockbasierendem DCT-Verfahren (Diskrete Cosinus Transformation), welches etwa dem JPEG-Standard für Einzelbilder entspricht.
- Das Ausnutzen von Abhängigkeiten (Korrelation) zwischen aufeinanderfolgenden Einzelbildern, d.h. der Bildinhalt ändert sich meist nur wenig von Bild zu Bild. (vgl. auch Abbildung (1))

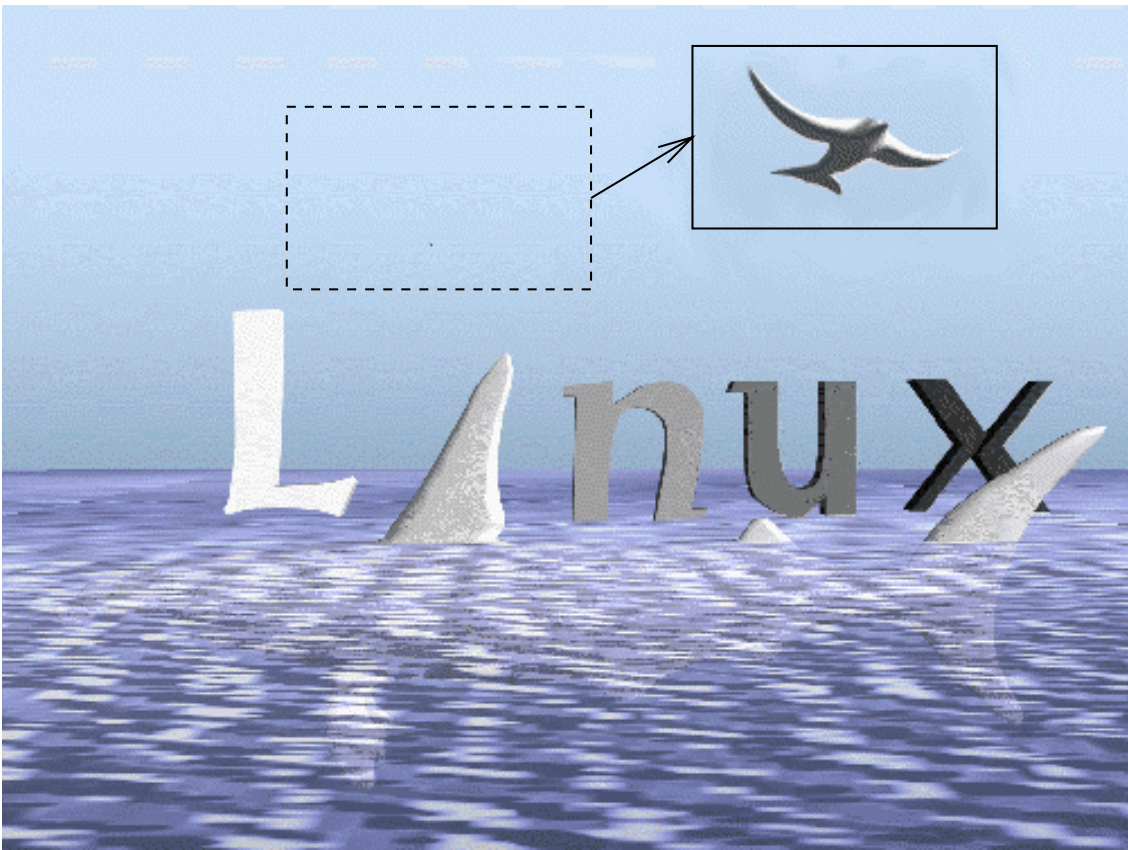
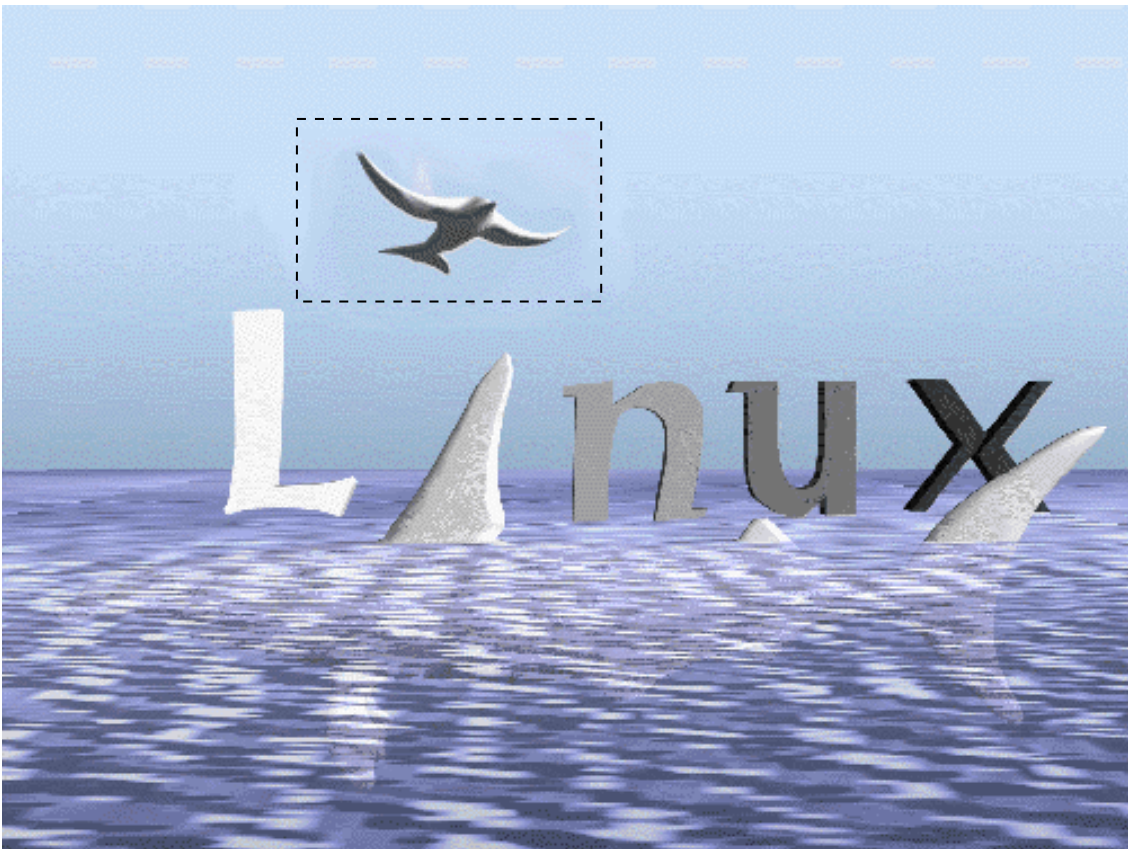
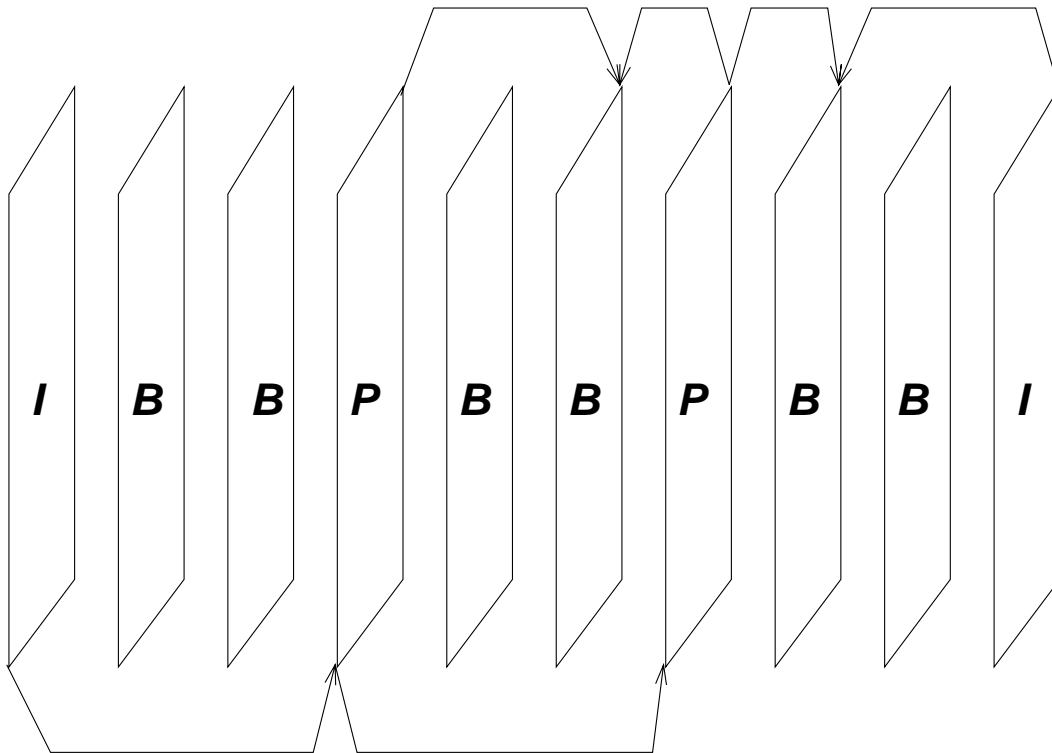


Abbildung 1: Beispiel für Prädiktion

Prädiktion von B-Frames



Prädiktion von P-Frames

Abbildung 2: Beispiel für die verschiedenen Frametypen

Im MPEG-Standard werden zu diesem Zweck drei Bildertypen (Frames) definiert, und zwar:

- **I-Frames**
Diese Bilder werden ohne Ausnutzung der Korrelation aufeinanderfolgender Bilder gespeichert, d.h. sie hängen nicht von folgenden oder vorherigen Bildern ab. Dies bedeutet auch, daß nur an diesem Bildtyp die Darstellung des Films, z.B. beim direkten Zugriff bzw. Vor-/Zurückspulen oder nach Fehlern, begonnen werden kann. Diese Bilder werden auch als „intra-codierte“ Bilder bezeichnet.
- **P-Frames**
Diese werden zusätzlich auch aus einem vorangegangenen P- oder I-Frame vorhergesagt und heißen daher prädizierte Bilder (P-Frames).
- **B-Frames**
Die B-Frames (bidirektional interpoliert/prädiziert) sind der universellste Bildertyp. Es können, im Unterschied zu den P-Frames, zusätzlich auch Verweise auf ein nachfolgendes P- oder I-Frame vorhanden sein. Natürlich

muß das nachfolgenden Bild, auf das verwiesen wird, zur (De)kodierung vorgezogen werden, was eine Anzahl von Bildspeichern erfordert und die Gesamtverzögerungszeit des (De)kodiervorgangs erhöht. Aber dafür werden sie auch am effizientesten komprimiert.

Die Abbildung (2) stellt ein Beispiel zu den verschiedenen Frametypen dar. Die unterschiedliche Größe der einzelnen Frametypen kann man leicht in Abbildung (3) erkennen.

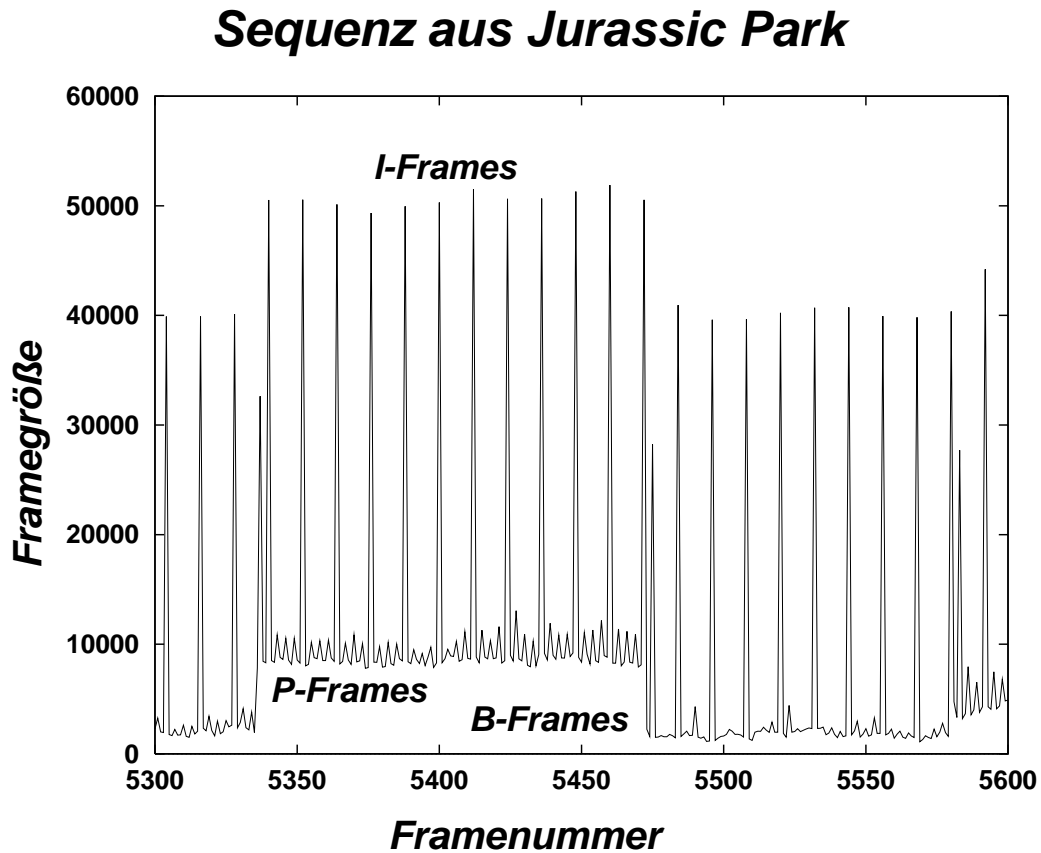


Abbildung 3: Beispiel für die Größe der Frametypen

3.3 Technischer Aufbau

Slice

Ein Slice ist eine Folge einer beliebigen Anzahl von Makroblöcken. Die Slices können von Bild zu Bild unterschiedlich groß sein, dabei muß nur der ganze Frame von Slices bedeckt sein (siehe Abb. 4). Der Grund, warum Makroblöcke überhaupt zu Slices zusammengefaßt werden, liegt in der Fehlertoleranz. Der Dekoder kann nämlich bei einem Fehler nur den nächsten Slice-Startcode finden und dort weiterdekodieren.

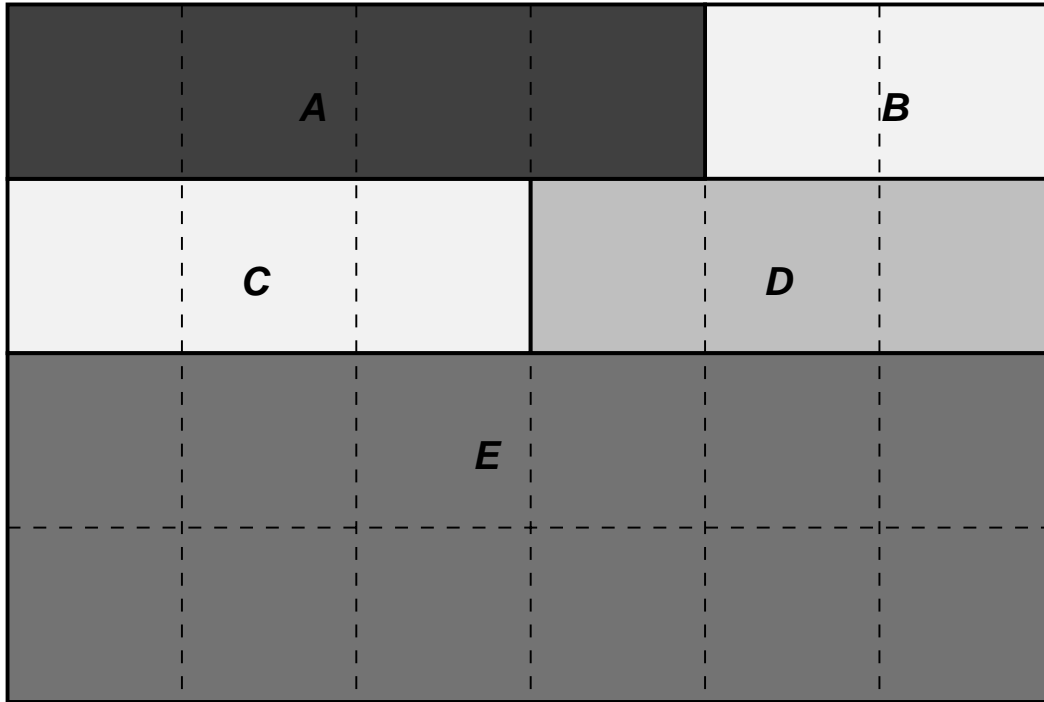
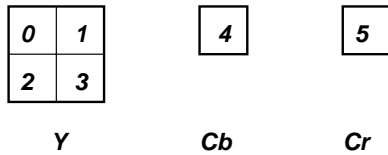


Abbildung 4: Aufbau eines Frames aus Slices

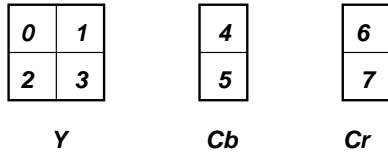
Makroblock

Ein Makroblock besteht aus drei rechteckigen Matrizen, einer Luminanz- (Y) und zwei Chrominanzmatrizen (Cb und Cr), in denen die Ergebnisse der DCT über die vier Blöcke, die den Makroblock bilden, als ganze Zahlen gespeichert sind. Ein Makroblock beschreibt jeweils 16x16 Bildpunkte und ist die kleinste Einheit im MPEG-Standard, auf die auch die Bewegungskodierung aufbaut. Das Chrominanzformat bestimmt dabei die Größenverhältnisse der einzelnen Matrizen:

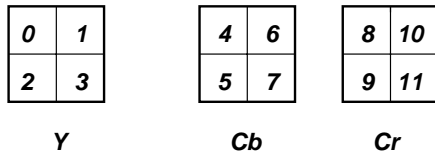
- **4:2:0 Makroblock**



- 4:2:2 Makroblock (nur MPEG2)



- 4:4:4 Makroblock (nur MPEG2)



Block

Ein Block ist 8 mal 8 Bildpunkte groß und wird als Eingabe für die diskrete Cosinustransformation (DCT) benutzt. Sie stellt eine Art spektrale Analyse des Bildinhaltes dar. Vier Blöcke sind jeweils zu einem Makroblock zusammengefaßt, und das Ergebnis der DCT dieser Blöcke bildet die Matrizen des Makroblocks. In Abbildung (5) wird der Zusammenhang noch einmal erläutert.

3.4 Bildkodierung

Wie weiter oben beschrieben, verwendet der MPEG-Standard grundsätzlich zwei Verfahren zur Bildkodierung:

- Einzelbildkodierung
- Bewegtbildkodierung

Die Bewegtbildkodierung findet dabei nur in den P- oder B-Frames statt und wird dort ergänzend zur normalen Codierung des Makroblocks angewandt. Zur Bildkodierung werden die folgenden Makroblocktypen definiert:

- **intra-type**

Dieser Makroblock wird vollkommen autonom abgespeichert. Dazu wird der Makroblock in 4 Blöcke von jeweils 8x8 Bildpunkten aufgeteilt. Diese werden dann blockweise einer diskreten Cosinustransformation unterworfen. Das Ergebnis der DCT wird dann quantisiert, d.h. Werte innerhalb eines Intervalls werden zu einem Repräsentativwert zusammengefaßt, und mit variabler Codelänge gespeichert. Auf dieses Verfahren will ich hier nicht

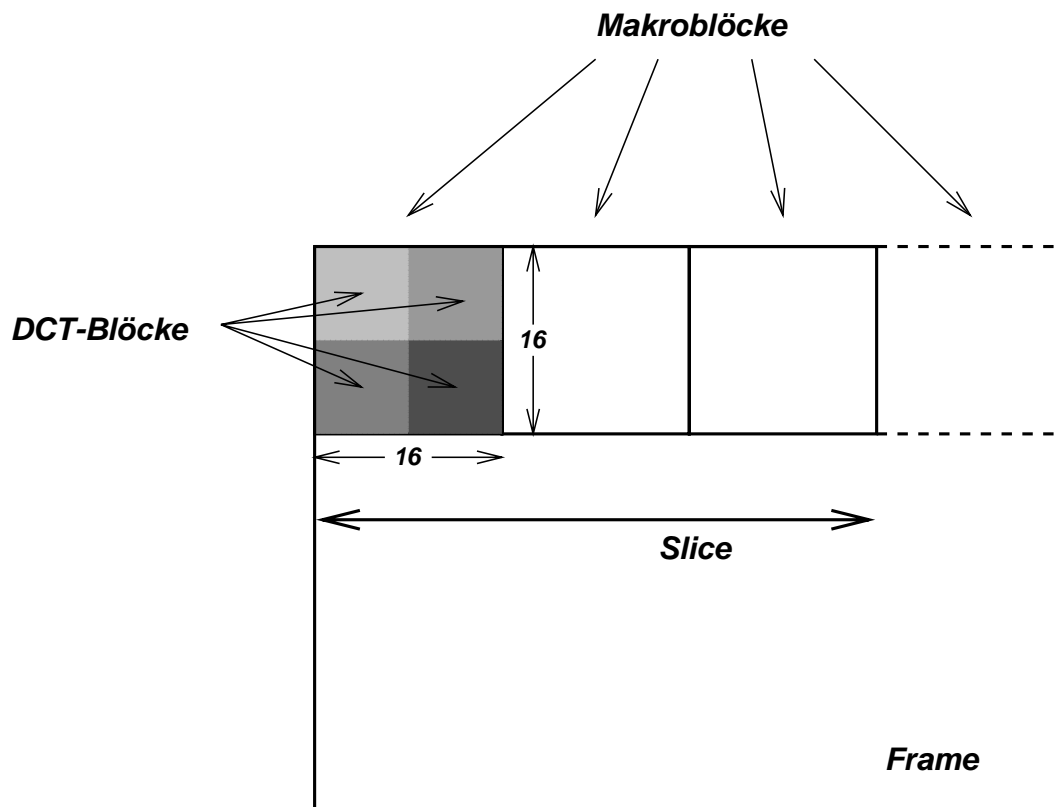


Abbildung 5: Zusammenhang zwischen Frame/Slice/Makroblock und Block

weiter eingehen, da es genauso im JPEG-Standard Verwendung findet (siehe JPEG-Seminar).

(in I-, P- und B-Frames)

- **backward-predicted-type**

Bei diesem Makroblocktyp wird ausgenutzt, daß sich bestimmte Bildteile in den zeitlich folgenden Bildern kaum verändern bzw. nur eine andere Position im Bild einnehmen. Dazu wird dieser Makroblock aus dem Bildinhalt des letzten vorausgegangenen I- oder P-Frame vorhergesagt („predicted“). Der Bewegungsvektor, der mit dem Makroblock gespeichert wird, bestimmt dabei die Position auf dem Referenzbild. Die noch vorhandene Abweichung zwischen dem zu kodierenden Makroblock und der Referenz, auf die der Bewegungsvektor zeigt, wird wie bei **intra-type** kodiert. Der Bildunterschied ist i.a. gering, so daß dessen Kodierung wesentlich weniger Platz braucht als bei kompletter Einzelbildkodierung. Die Kodierung des Restunterschieds kann dabei auch entfallen.

(in P- und B-Frames)

- **forward-predicted-type**

Dieser Typ entspricht im Prinzip dem **backward-predicted-type** Makroblock. Nur wird hier nicht als Referenz ein vorausgegangenes Frame, son-

dern das zeitlich als nächstes folgende I- oder P-Frame benutzt. Da dazu auf zeitlich nachfolgende Frames zugegriffen wird, müssen die nachfolgenden Bilder schon dekodiert vorliegen. Dies erzwingt die Nutzung von Bildpuffern und bewirkt eine zeitliche Verzögerung, d.h. der Dekoder hinkt mit der Darstellung der Frames der Dekodierung um den Abstand bis zum nächsten I- oder P-Frame hinterher. In bestimmten Bereichen, wie z.B. Videokonferenzen, führt das zu Problemen.

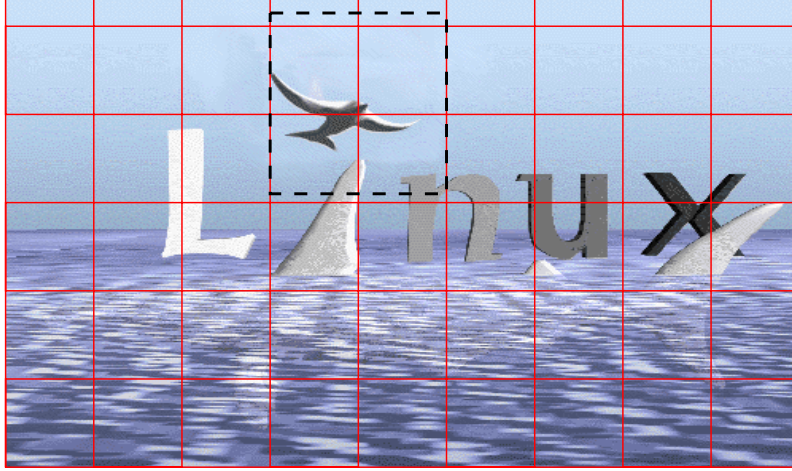
(in B-Frames)

- **average-type**

Bei diesem Typ werden zwei Bewegungsvektoren, einer zeigt entsprechend dem **backward-predicted-type** auf ein vorausgegangenes Frame, der andere, wie beim **forward-predicted-type** auf ein nachfolgendes Frame. Als Bildinhalt wird das arithmetische Mittel der beiden Referenzmakroblöcke verwendet. Der Vorteil von diesem Typ liegt vor allem darin, daß bei der Mittelwertbildung Rauschen und Bildfehler verringert werden.

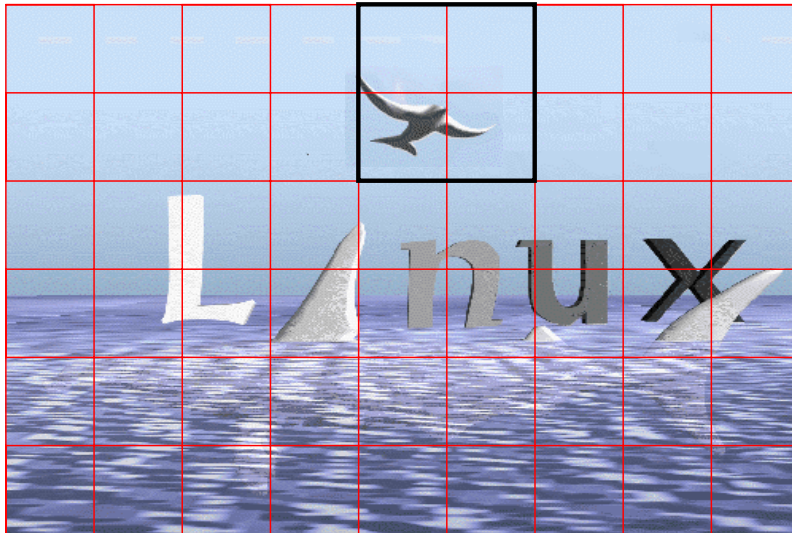
(in B-Frames)

Besonders deutlich werden die verschiedenen Makroblocktypen in den Abbildungen (6, 7). Welche Frametypen (I, P oder B) verwendet werden, bestimmt der Anwender beim Kodieren. Welche Makroblöcke der Kodierer in den einzelnen Frames verwendet, entscheidet der Kodierer anhand der Vorgaben der Anwendung selbst.



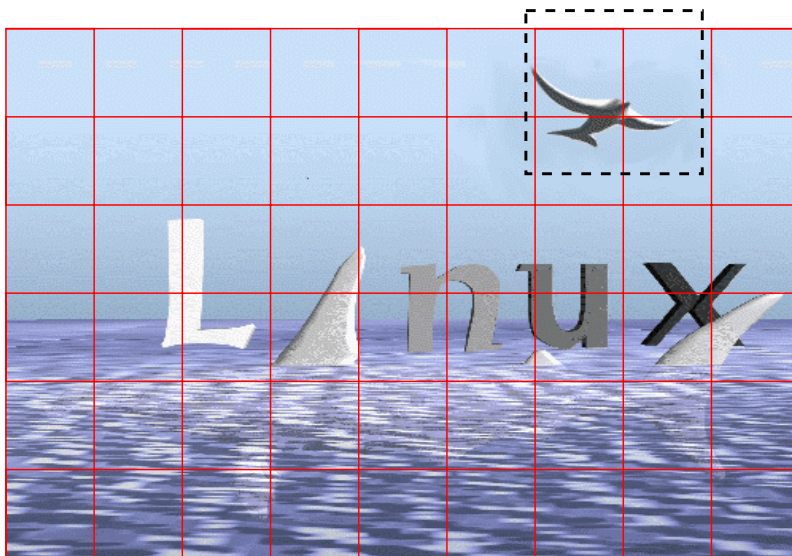
I-Frame

Nummer: t



B-Frame

Nummer: $t+k$



I-Frame

Nummer: $t+i$

Abbildung 6: Beispielbildfolge für Abbildung (7)

Kodierung des markierten Makroblocks des Frames (t+k)


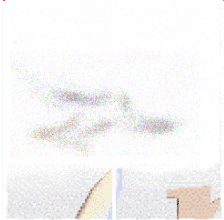
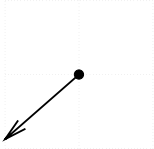
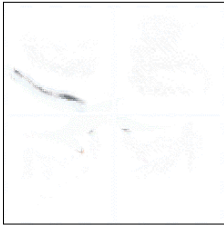
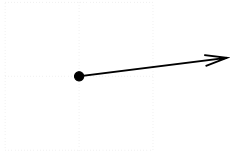
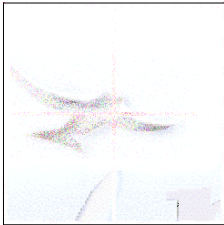
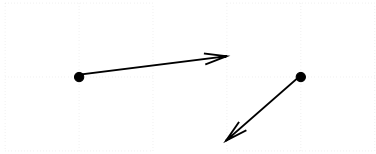
<i>Makroblocktyp</i>	<i>Bildkodierung</i>	<i>Bewegungsvektor</i>
<i>intra</i> <i>(in allen Frametypen)</i>		
<i>backward-predicted</i> <i>(in P- oder B-Frames)</i>		
<i>forward-predicted</i> <i>(in B-Frames)</i>		
<i>average</i> <i>(in B-Frames)</i>		

Abbildung 7: Kodierung des mittleren Frames aus Abbildung (6)

3.5 Funktion eines Kodierers

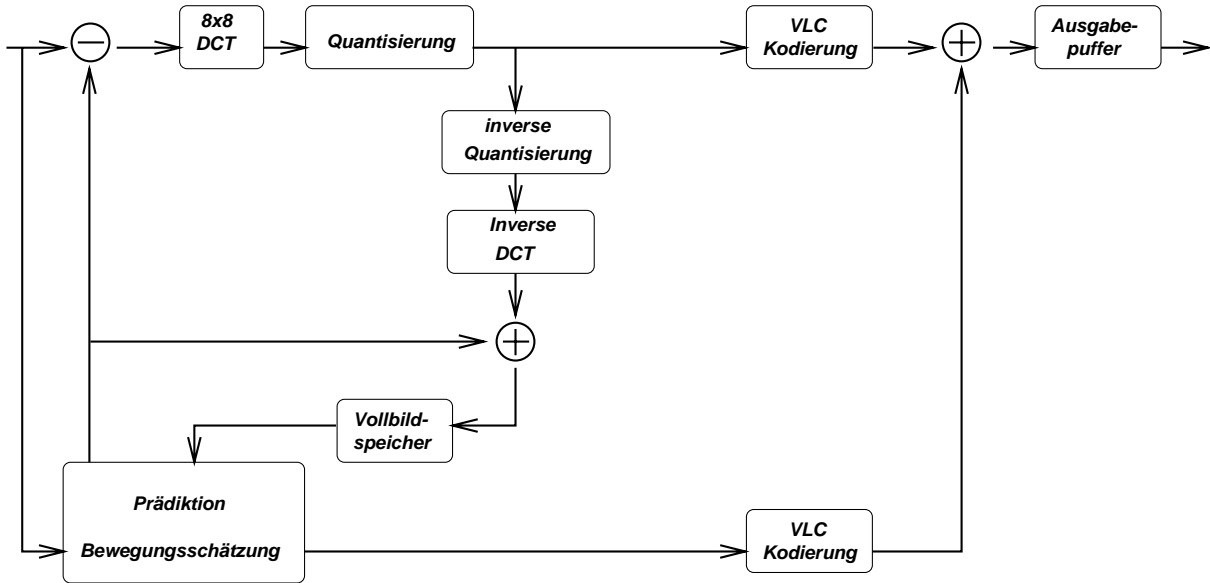


Abbildung 8: Blockschaltbild eines Kodierers

3.6 Funktion eines Dekoders

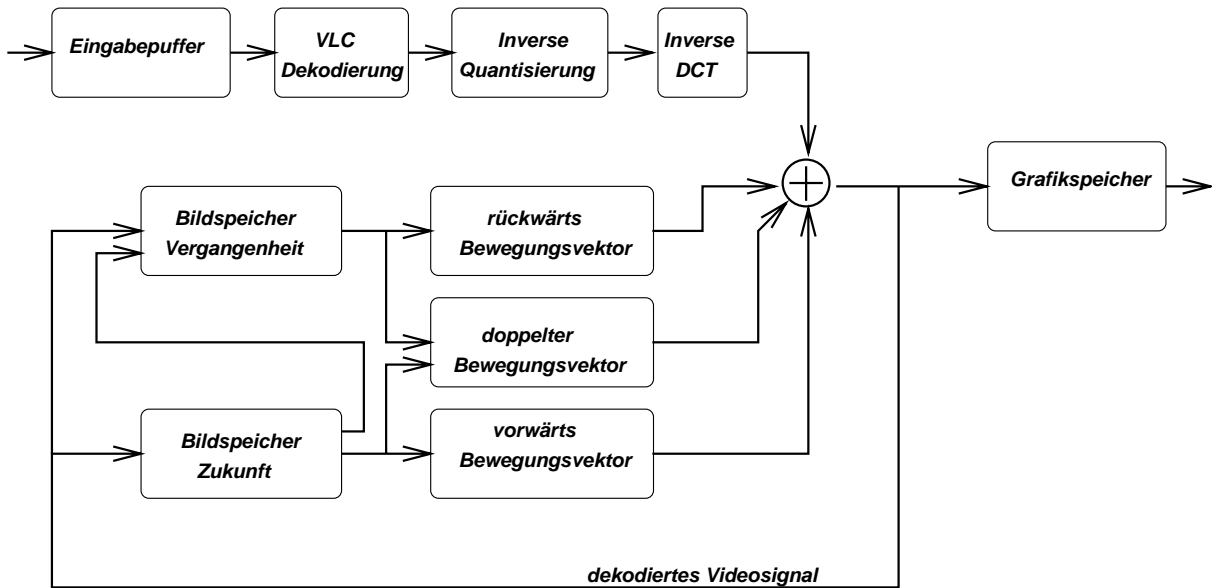


Abbildung 9: Blockschaltbild eines Dekodierers

3.7 Bitstromsyntax

Der Bitstrom ist in 6 Ebenen („layers“) strukturiert:

Sequenzebene (Sequence layer)
Bildgruppenebene (Group of picture layer)
Bildebene (Picture layer)
Sliceebene (Slice layer)
Makroblockebene (Macroblock layer)
Blockebene (Block layer)

Jede Ebene hat einen eindeutigen Startcode, der sonst nicht im Datenstrom vorkommt. Dies ist nötig, da der Bitstrom mit variabler Codelänge gespeichert ist.

- Die Sequenzebene ist einer längeren Folge von Bildern zugeordnet. Hier werden die grundsätzlichen Daten des MPEG-Stroms definiert:
 - horizontale Bildgröße
 - vertikale Bildgröße
 - Seitenverhältnis (0.6 bis 1.2)
 - Bildrate (z.B. 30 Bilder/s)
 - Bitrate (z.B. 1.5 MBit/s)
 - Quantisierungsparameter
 - minimale Puffergröße
 - Skalierdaten
 - Chrominanzformat (z.B. 4:2:0)
 - Benutzerdaten (frei verfügbar)
- Alle Bilder in einer Bildgruppen können unabhängig von anderen Bildgruppen dekodiert werden, d.h. die Bildgruppen sind in sich abgeschlossen. Ein Beispiel für eine Bildgruppe findet man in Abbildung (2).
- In der Bildebene wird jedes Bild mit einem Bildvorspann begonnen. Darin werden folgende Daten übergeben:
 - Nummer des Bildes (zeitlich)
 - Bildtyp (I, B oder P)
 - Bildstruktur (Vollbild, 1. Teilbild oder 2. Teilbild)
 - Parameter für DCT
- In der Sliceebene wird nur die vertikale Position des Slice und die Anzahl der enthaltenen Makroblöcke mitgeteilt.

- Hier werden folgende Daten abgelegt:
 - Makroblockadresse (es müssen nicht alle kodiert werden)
 - Makroblocktyp (siehe Abschnitt 3.4)
 - Bewegungsvektor (bei **average-type** zwei Vektoren)
- In der Blockebene werden die Koeffizienten der DCT gespeichert.

4 Einschränkungen/Grenzen

Leider hat auch jeder noch so gute Standard seine Einschränkungen und Grenzen, auf die im folgenden kurz eingegangen werden soll.

- **Codierungsaufwand**

Der Codierungsaufwand bei MPEG ist sehr hoch, was bei den Anforderungen an den Kodierer auch klar ist. Bei Bewegtbildkodierung muß z.B. für jeden einzelnen Makroblock eines Bilds das Referenzbild untersucht werden. Außerdem lassen sich die DCT und die variable Codelänge auf normalen Prozessoren schlecht implementieren. Daher braucht man bei Softwarekompression ein Vielfaches der Laufzeit zum Kodieren. Echtzeitkompression ist im Moment nicht bei höchster Qualität und ansonsten nur mit Parallelverarbeitung und Signalprozessoren möglich.

- **Dekodierungsaufwand**

Softwaredekodierung ist nur mit schnellem Prozessor (z.B. Pentium, Workstation) und bei geringer Bildgröße („Briefmarkengröße“) bzw. Qualität möglich. Dafür ist mit relativ preisgünstiger Hardware Vollbilddarstellung möglich. Die Kosten für Dekoderkarten werden etwa auf Niveau von CD-ROM Laufwerken sinken.

Natürlich bleibt die starke Kompression der Videodaten nicht ohne Folgen für die Qualität. Diese Auswirkungen sind aufgrund des intelligenten Kodierungsverfahrens meist nicht bzw. kaum sichtbar.

- **Codierung von Rauschen**

Bei stark verrauschten Bildern ist keine Korrelation mehr zu den benachbarten Bildern vorhanden, was nur noch Kodierung im Intra-Modus zuläßt. Bei starker Kompression wird das Rauschen gefiltert.

- **Schnelle Bewegung**

Wenn der Bewegungsverfolger eine Bewegung nicht mehr erfassen kann, werden die Makroblöcke überwiegend im Intra-Modus kodiert, was eine starke Verschlechterung der Bildqualität zur Folge hat.

- **Scharfe Kanten**

Falls im Bild gleichmäßig gefärbte Flächen durch scharfe Kanten getrennt werden, so kann das bei vielen Kanten durch die Quantisierung zu Einbußen der Bildqualität führen (z.B. große Schrift auf einfarbigem Grund). Bei stark strukturierten Gebieten fallen diese Fehler weniger stark auf. Diese Probleme der Komprimierung sind die gleichen wie bei dem JPEG Standard für Standbilder.

Literatur

- [1] O. Rose, „Statistical Properties of MPEG Video Traffic and their Impact Modeling in ATM Systems“, Institute of Computer Science, University of Würzburg, COST 242 TD(95)01, Jan. 1995
- [2] P. Pancha and M. Zarki, „MPEG Coding for Variable Bit Rate Video Transmission“, IEEE Communication Magazine, p. 54 – 66, May 1994
- [3] A. Knoll, „Kurzbeschreibung des MPEG-2 Standards“, Deutsche Bundespost Telekom, FTZ, Darmstadt, April 1993
- [4] D. Le Gall, „MPEG: a Video Compression Standard for Multimedia Applications“, Communications of the ACM, Vol. 34, No.4, 46 – 58, April 1991
- [5] ISO/IEC 13818-2, „Information Technology - Generic Coding of Moving Pictures and Associated Audio“, Recommendation H.262, MPEG-2 Draft, March 1994
- [6] ISO/IEC 11172-2, „Coding of Moving Pictures and Associated Audio for Digital Storage Media up to 1.5 MBit/s“, MPEG-1 Draft, 1993