

Diana-Seminar

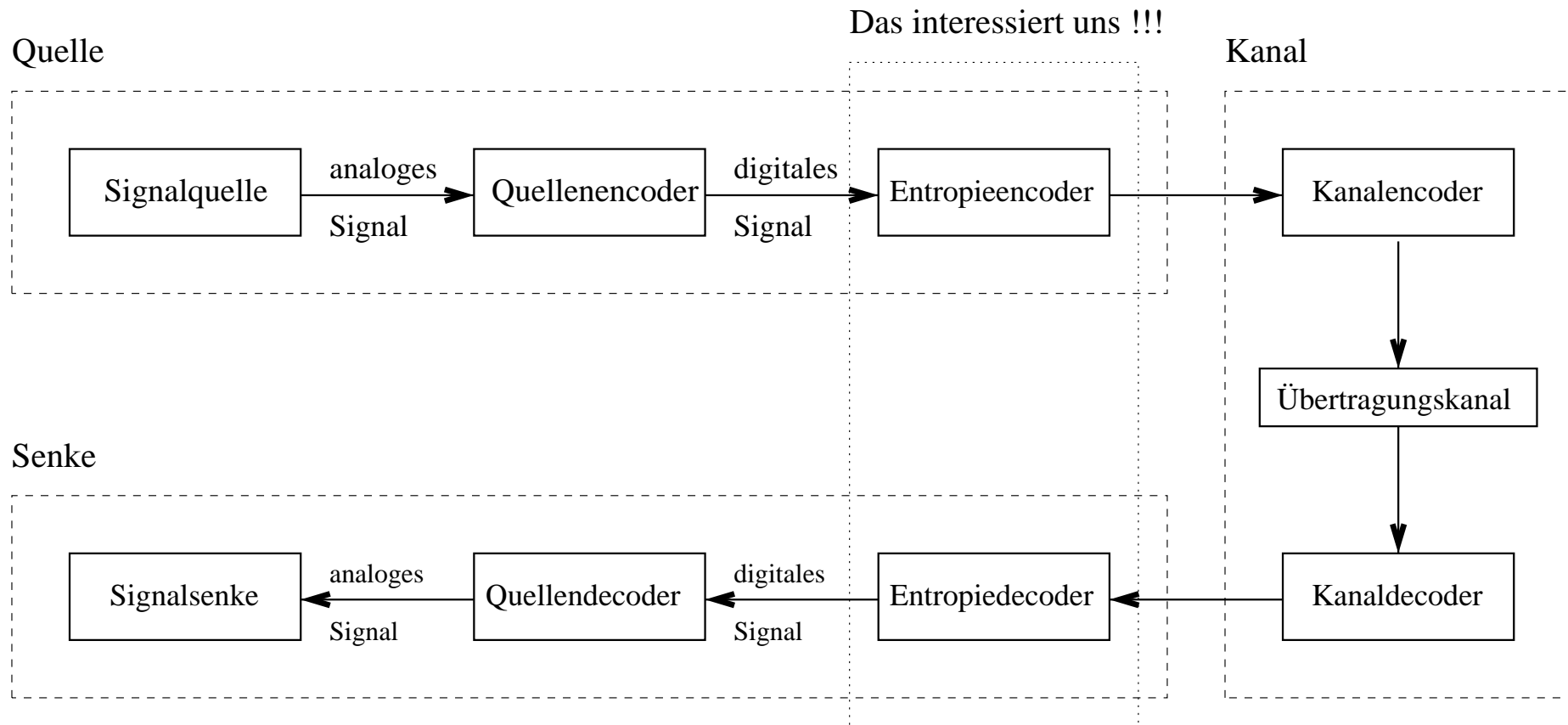
## **Videübertragung**

Ralf Staudemeyer

## Kodierungsverfahren

- Quellenkodierung  
A/D Wandlung
- Entropiekodierung  
Verringerung der Redundanz  
→ das interessiert uns !!!
- Kanalkodierung  
vom Netzwerk vorgegeben  
→ siehe ATM-Vortrag

# Das Übertragungsmodell



## Digitalisierung

- Audio  
Umwandlung von Wellen in Bitwerte
- Video  
Umwandlung von Bildpunkten in Pixel

→ *Multimedia (Audio + Video = 2 Medien)*

Problem: Datenmenge muß reduziert werden

## A/D Wandlung

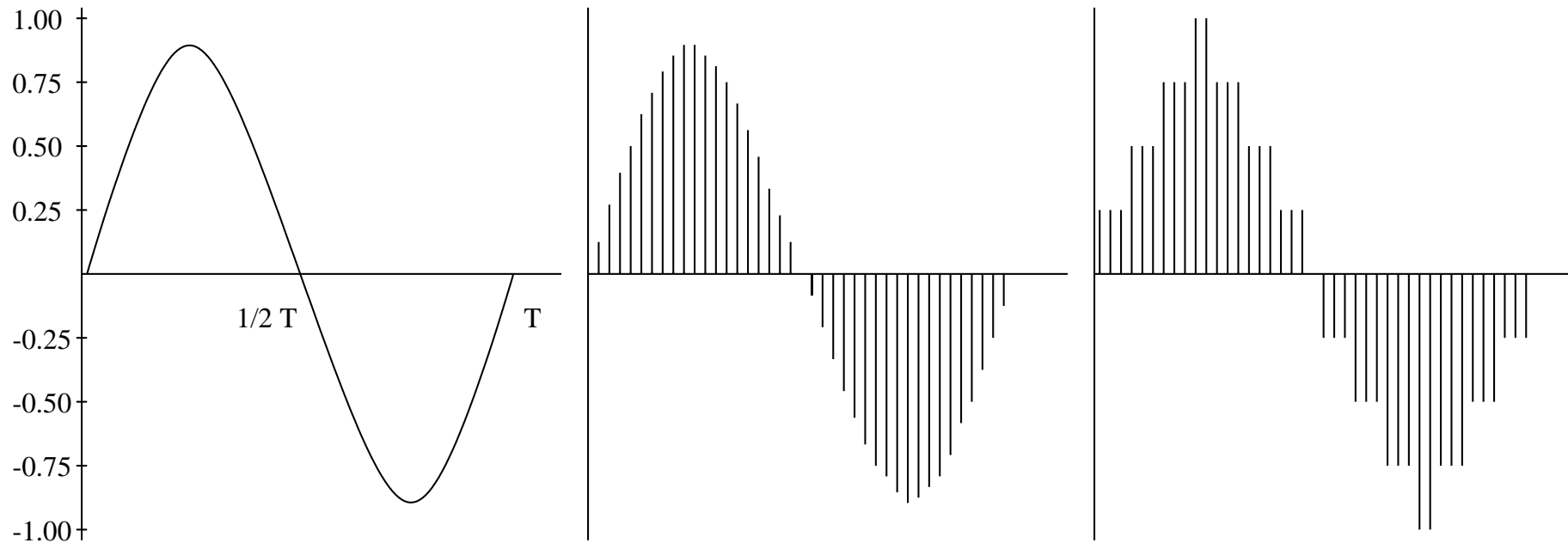
- elektrische Spannung  $\rightarrow$  Binärzahl
- alle  $\Delta T$  Sekunden abtasten  
Nyquist-Theorem
- Quantisierung durch Interpolation

## Nyquist-Theorem

max. Datenrate eines rauschfreien Kanals.

Ein beliebiges Signal durch einen Tiefpassfilter der Bandbreite  $H$  geführt, kann nur durch Abtastwerte von  $2H/s$  wieder hergestellt werden.

### A/D-Wandlung



Sinuswelle

Abtasten der Sinuswelle

Quantisieren auf 3 Bit

## Beispiel: Telefon / Audio CD

- PCM: 8000 8-Bit Samples pro Sekunde (in Europa)  
Datenrate 64000 bps  
Frequenzen über 4kHz gehen verloren
- Audio CD: 44100 16-Bit Samples Abtastrate  
1,411 Mbps bei Stereo  
erst ab 22kHz Verluste

## Kompressionsverfahren

- komprimierte Bilder
  - JPEG
- bewegte komprimierte Bilder
  - MJPEG
- komprimierte Filme
  - MPEG
- the tricky stuff from the motion experts
  - MPEG2/4/7/?

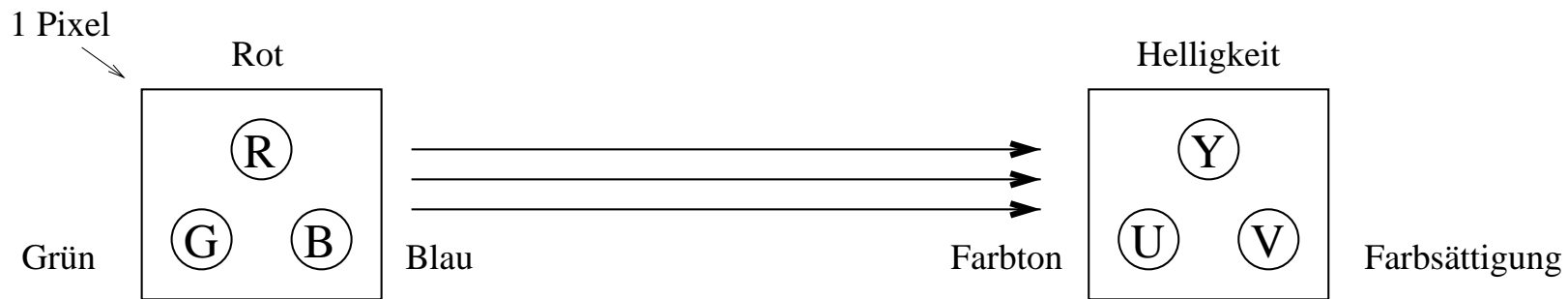
## JPEG-Norm

Eingabe →

1. Änderung des Farbmodells (RGB → YUV)
2. Unterabtastung (Subsampling)
3. Discrete Cosinus Transformation (DCT)  
für 8x8 Blöcke
4. Quantisierung
5. Lauflängenkodierung (RLC)
6. statische Kodierung (Huffman/VLC)

→ Ausgabe

## Umwandlung von RGB nach YUV



(R)

Luminanz

(Y)

$(0.299R - 0.587G + 0.114B)$

(G)

Chrominanz  $C_r$   
(Blaudifferenz)

(U)

$(B - Y)$

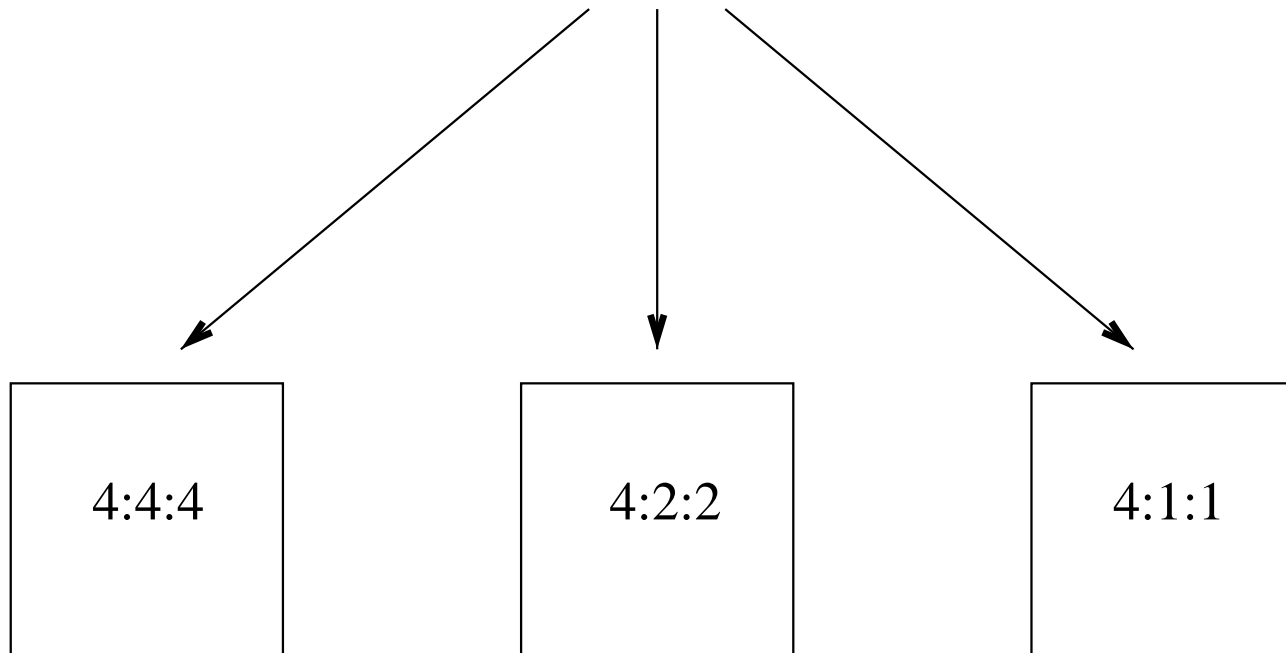
(B)

Chrominanz  $C_b$   
(Rotdifferenz)

(V)

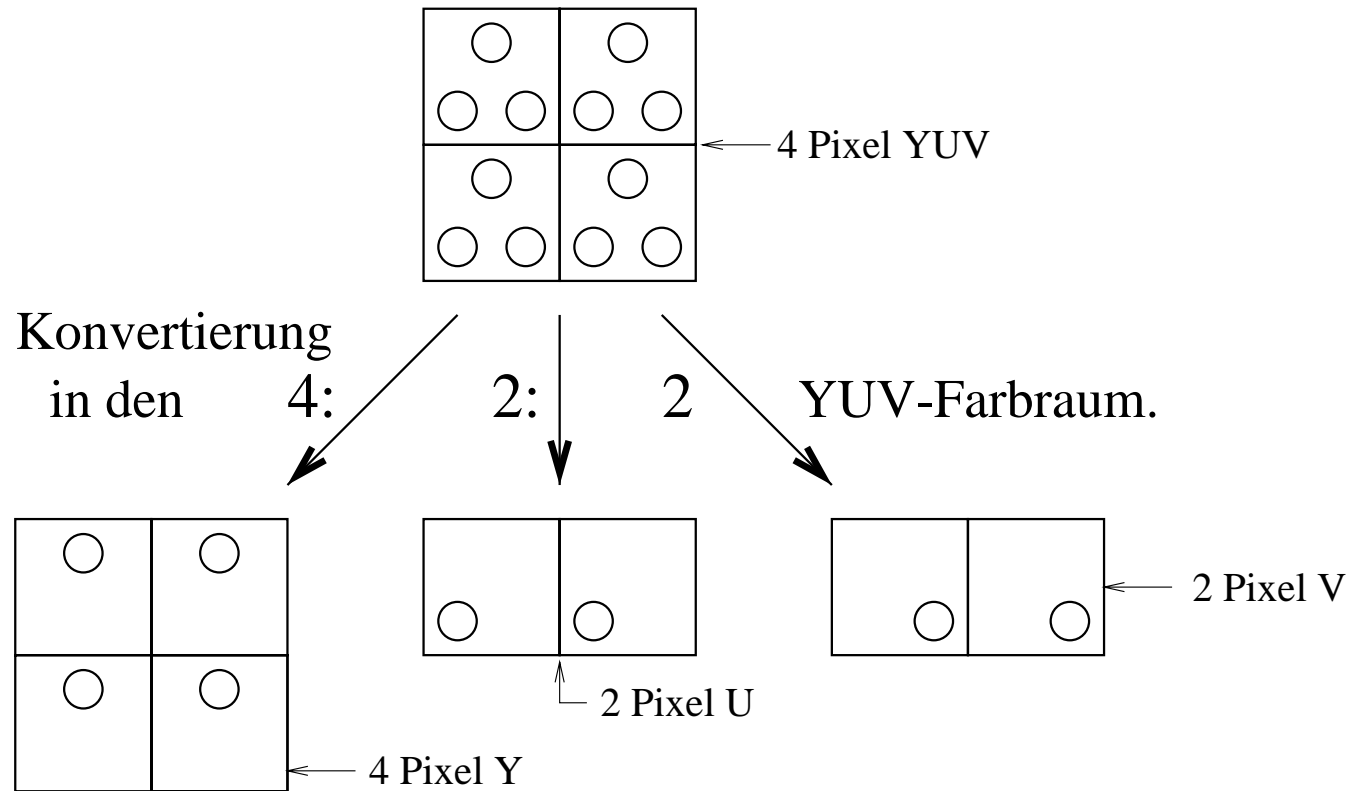
$(R - Y)$

# Subsampling



Verhältnis Y:U:V

# Subsampling



## Diskrete Cosinus Transformation

- 8x8 Pixel Block
- für jeden Pixel des Blockes ist  $Y$  (Helligkeit) bekannt
- enthaltene Frequenzen müssen ermittelt werden
  - erfolgt durch zeilen- und spaltenweise Behandlung der 8x8 Blöcke mit den DCT-Koeffizienten
  - handelt sich um Matrizenmultiplikation der Helligkeitsverteilung mit der DCT-Matrix
- Umrechnung ist sehr aufwendig
  - deswegen Stückeln in Blöcke
    - Bestimmung der Frequenzanteile
    - Reduzierung auf das Wesentliche

## Blockbehandlung in der Praxis

- Skalierung der Helligkeitswerte auf -128 bis 128
- Multiplikation mit transponierter DCT-Matrix
- Zwischenergebnis wird mit DCT-Matrix multipliziert
- Ergebnis repräsentiert die Frequenzverteilung im 8x8 Pixelblock
  - links oben wichtigste Information
  - rechts unten unwichtige Information

## Quantisierung

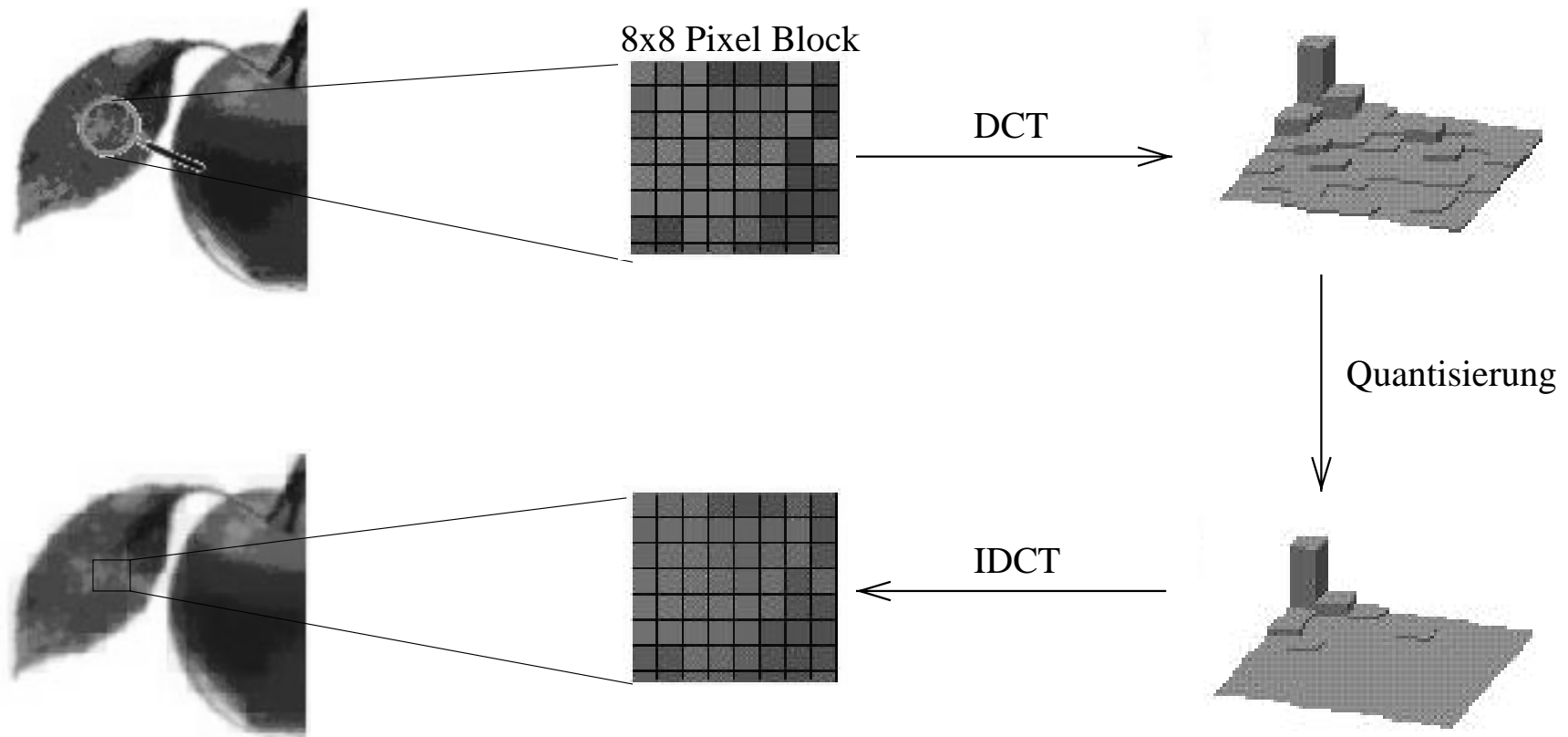
... ist die Aussonderung der weniger wichtigen DCTs

- Unterteilung des Ergebnisses in diskrete Stufen
- Kompressionsfaktor hängt maßgeblich von Feinheit der Unterteilung ab

## Quantisierung in der Praxis

- Je nach Kompressionsgrad Eliminierung der weniger wichtigen Informationen
- Trick der Quantisierung
  - Wert muß einen Mindestwert überschreiten, um signifikant zur Gesamtinformation des Bildausschnittes beizutragen
  - Grenze bleibt uns überlassen
    - \* kleiner Wert - geringe Kompression - gute Qualität
    - \* großer Wert - hohe Kompression - schlechte Qualität

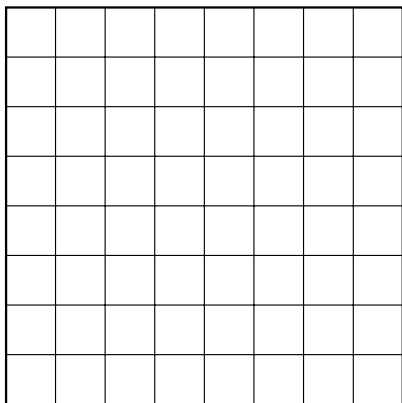
# Beispiel für DCT/Quantisierung



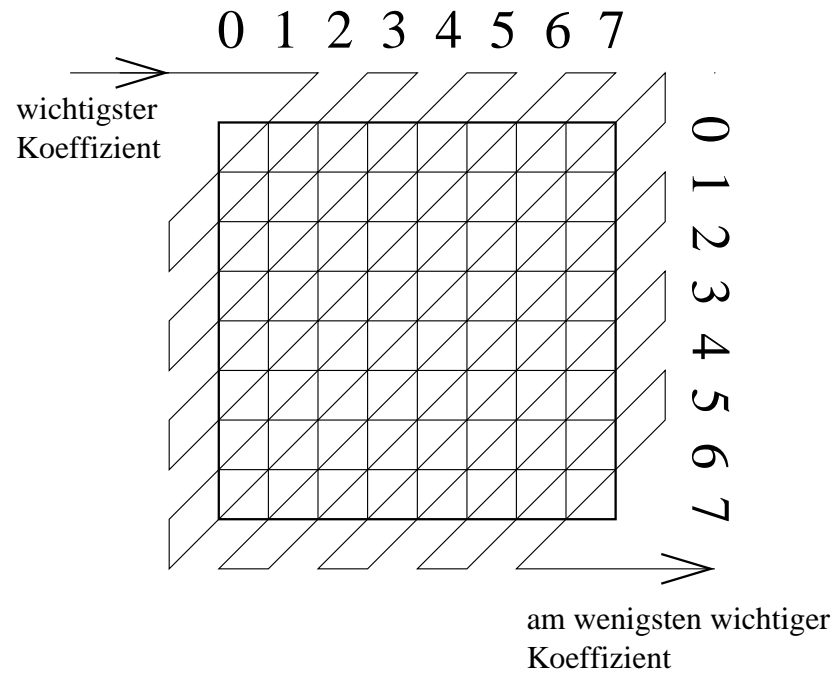
# nach der Quantisierung

Ergebnis der DCT

(8x8 Block mit 64 Koeffizienten  
zwischen 0 und 255)



Zig-Zag-RLC



## Run Length Coding

Datenstrom:

00000111100000000111

Vereinfachung:

(5x0);(4x1);(8x0);(3x1)

Codierung:

(6,4);(18,3)

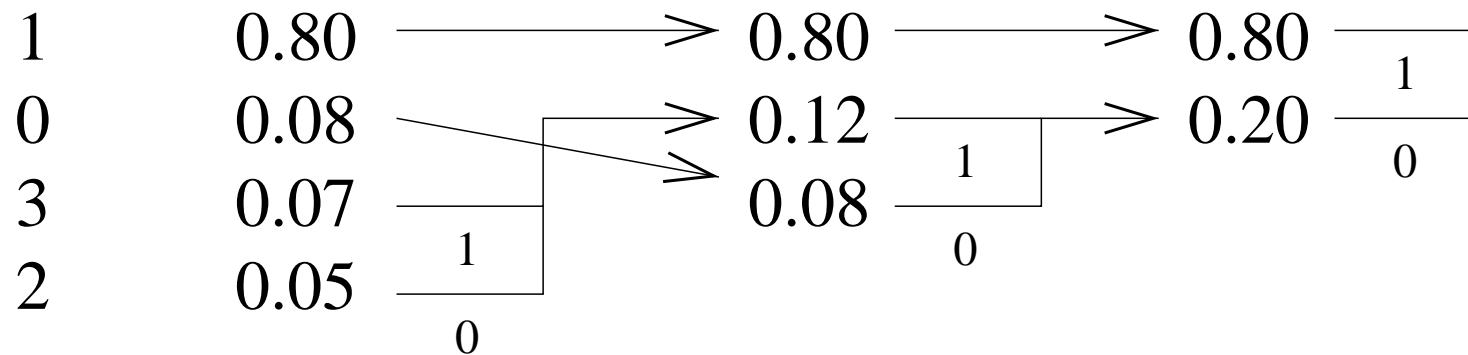
(Position, Anzahl der Einsen)

## Huffman-Kodierung

- **Ansatz:**  
verschiedene Häufigkeiten von gleichlangen Bitfolgen
- **Ziel:**  
je größer die Häufigkeit, desto kürzer die codierte Bitfolge
- **Wie ?**

Zeichen	bin-Code	rel. Häufigkeit	Huffmann-Code
0	00	0.08	00
1	11	0.80	1
2	10	0.05	010
3	01	0.07	011
mittl. Codewortl.	2Bit		<u>1.32Bit</u>

Zusammenfügen und Sortieren



## Qualität der Bilder

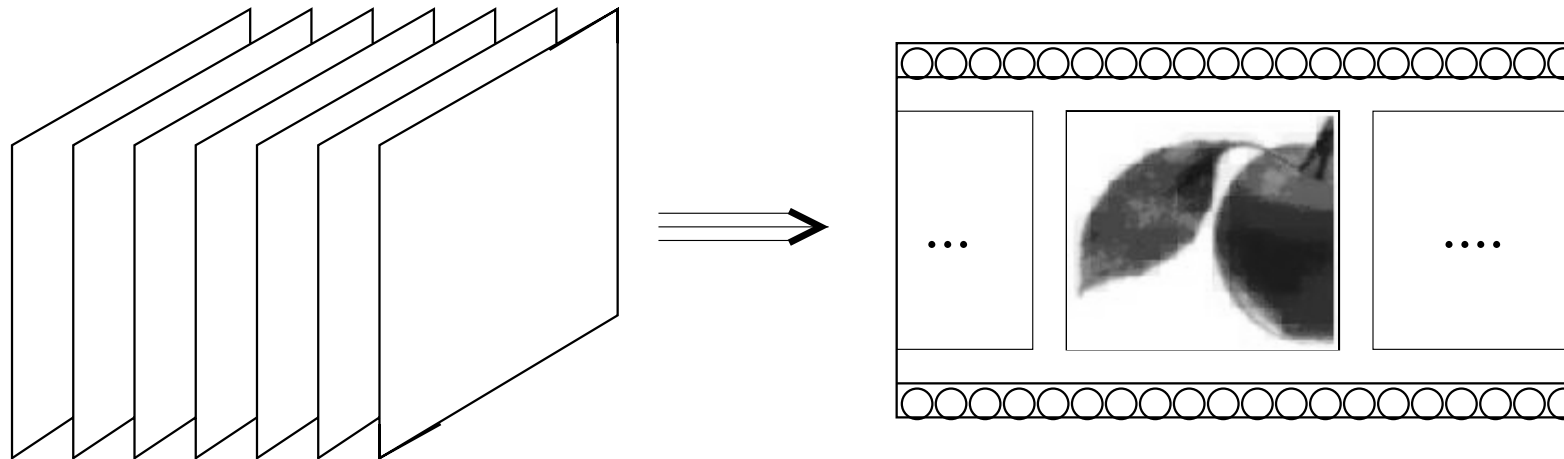
Qualität ist abhängig von

Subsampling und Quantisierung

beide Verfahren sind verlustbehaftet

hohe Kompression → schlechte Qualität

# Motion-JPEG



einzelne JPEG-Bilder

MJPEG-Videostrom

## Motion-JPEG kontra MPEG

MJPEG	MPEG
kein Standard	Bewegungskompensation
Intraframekomprimierung	Interframekomprimierung
Zugriff auf Einzelbild möglich	höhere Kompressionsrate
	sehr hoher Rechenaufwand

## MPEG

Motivation:

- 1s Video nach CCIR 601  
768 x 576 x 3Byte x 25Frames  
→ 31.64MByte/s

Ziel:

- 1.2MBit/s  
entspricht Single Speed CD-Rom Laufwerk
- max. 74 min Video auf eine CD-ROM  
→  $\frac{650MByte}{150kByte/s * 60s} = \underline{73,9min}$

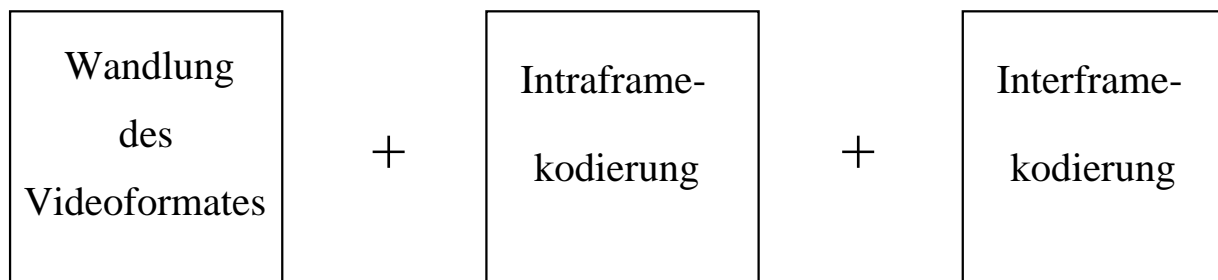
## Was ist MPEG ?

- MPEG - Motion Pictures Experts Group
- MPEG ist

Standbildkompression  
und  
Bewegungskompensation.

## Das MPEG-Verfahren

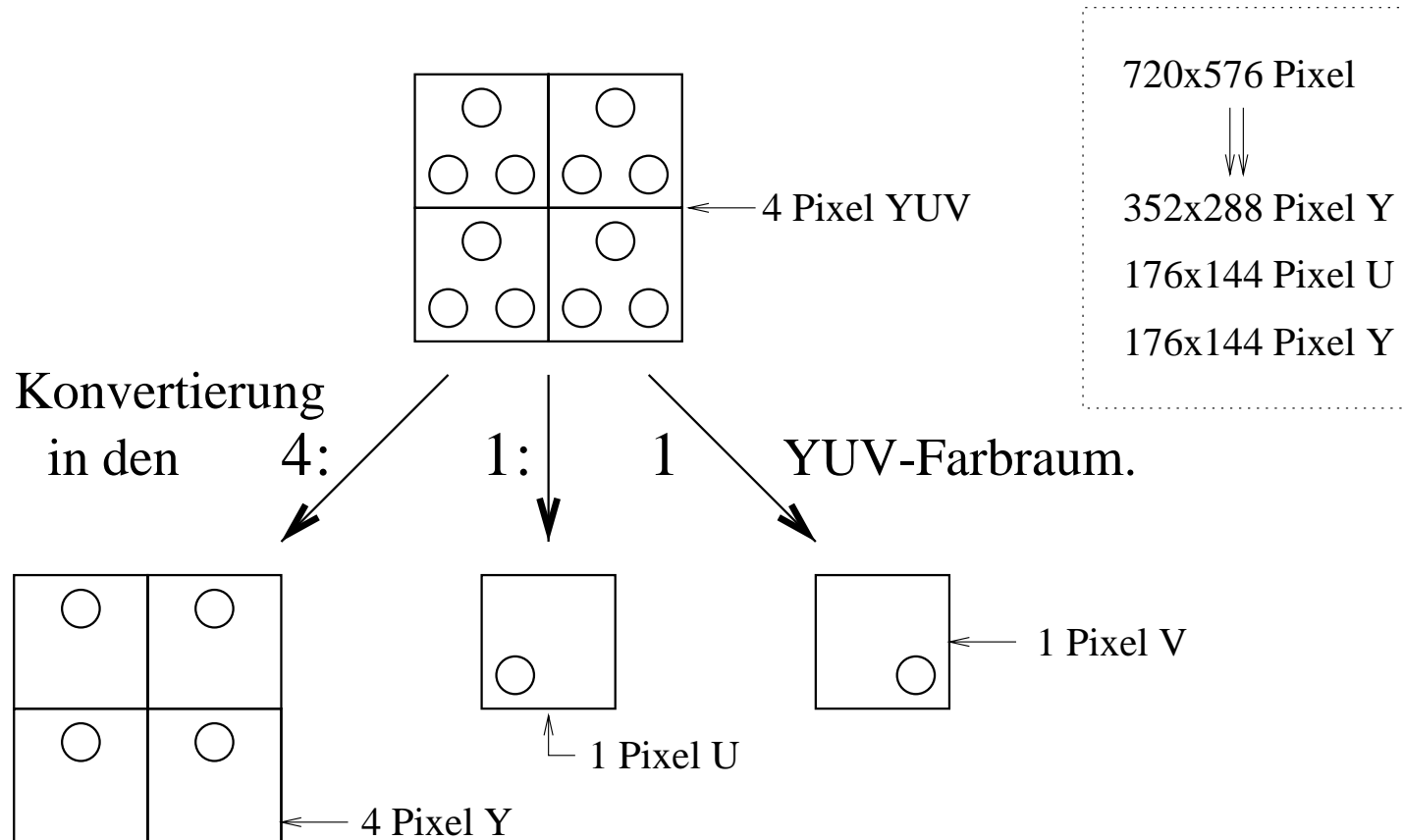
Grundbestandteile:



## Dezimierung der Auflösung

- Halbierung der Auflösung  
720x576 Pixel → 360x288 Pixel
  - Weglassen der Randpixel  
auf jeder Seite 4  
wegen Makroblockbildung  
→ 352x288 Pixel
- ... dann Konvertierung nach YUV

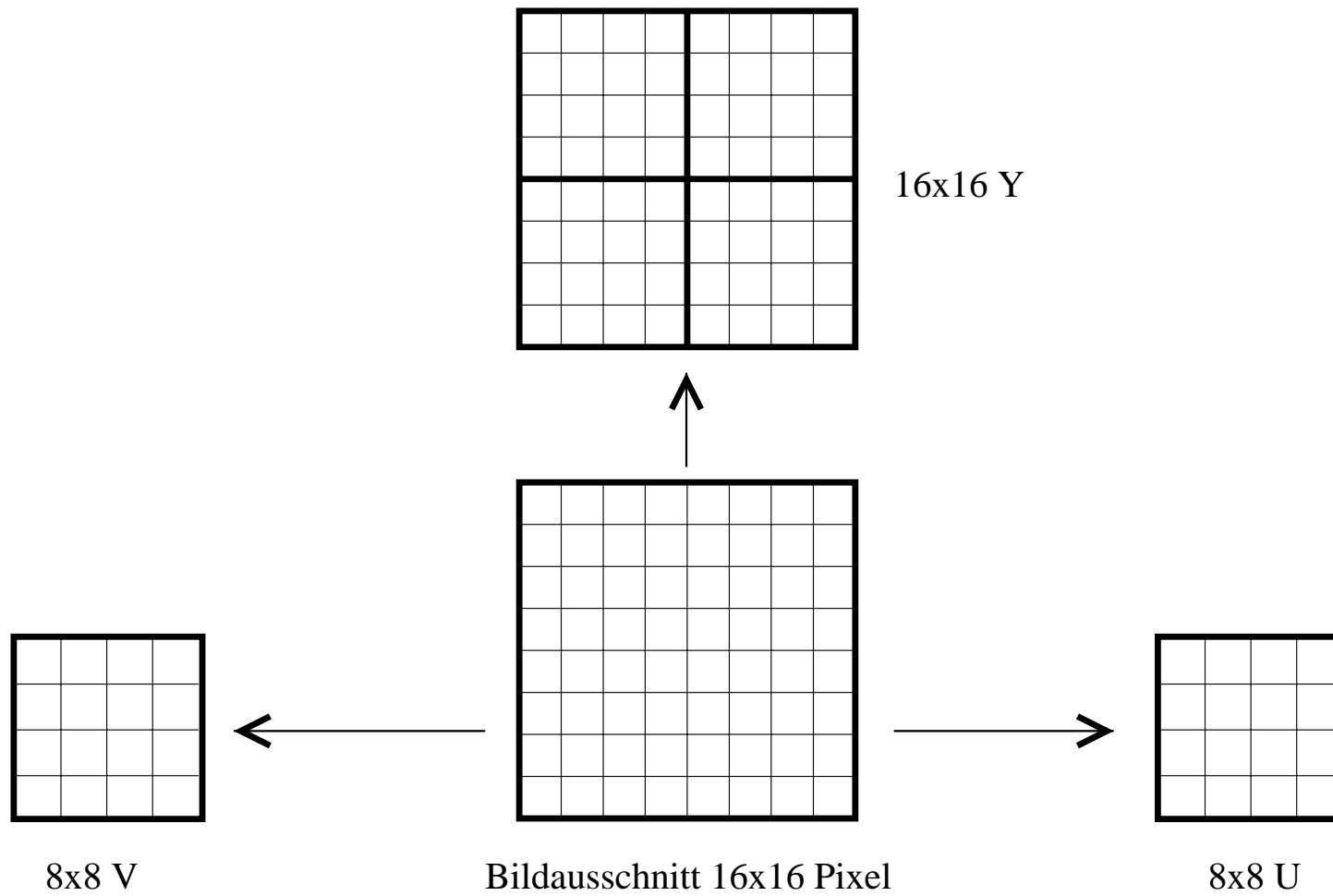
## Subsampling bei MPEG



## Makroblockbildung

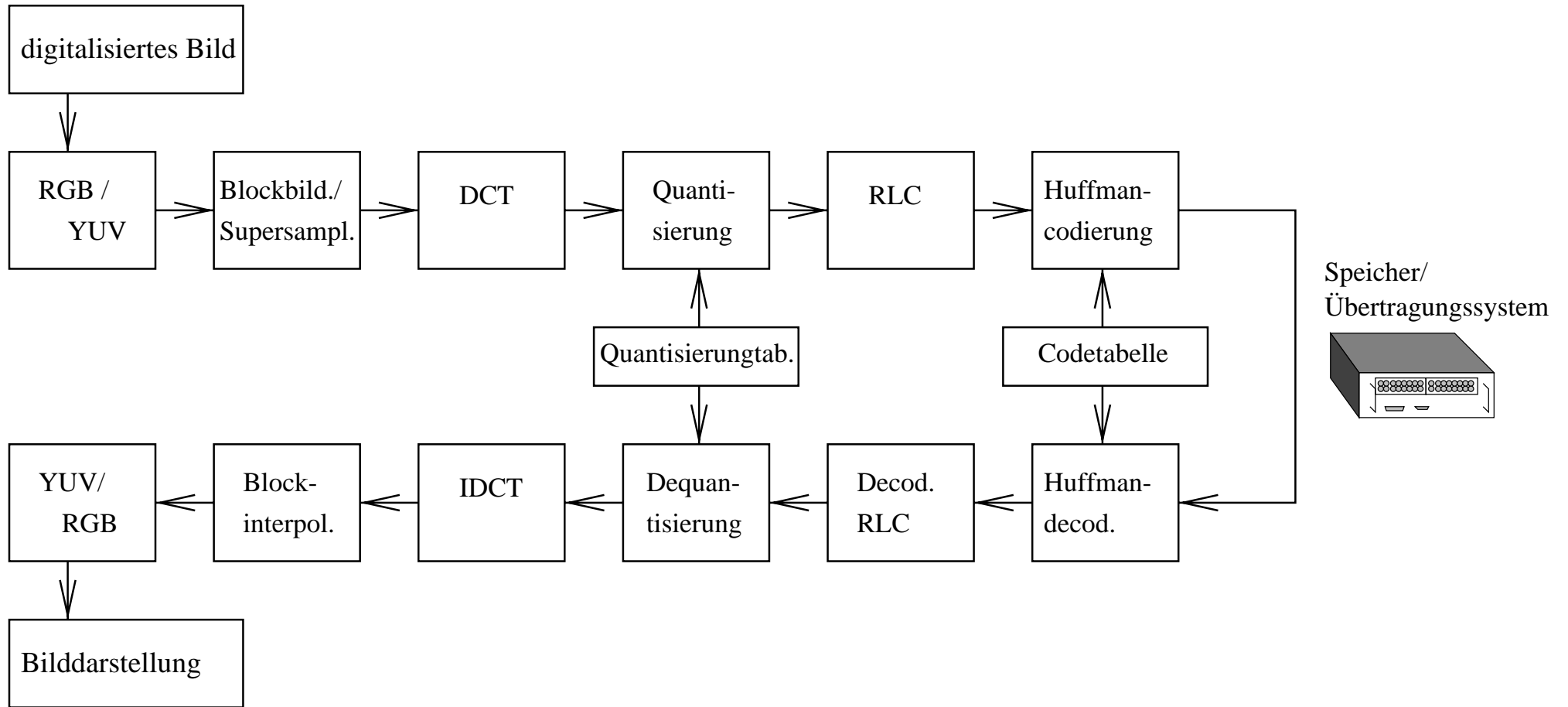
- Blöcke von 16x16 Pixel werden gebildet
- für Makroblock werden 6 Blöcke (8x8) kodiert
  - 4x Luminanz
  - 1x Chrominanz  $C_b$
  - 1x Chrominanz  $C_r$

# Makroblockbildung



## Intraframekodierung

- ist nicht festgeschrieben  
meist JPEG
  
- I-Bild:
  - vollständiger Inhalt des zugeordneten Videobildes
  - dient als Referenzbild
  - geringste Kompressionsrate (hohe Qualität)



Bei JPEG bzw. MJPEG sind Codierung und Decodierung gleich aufwendig.

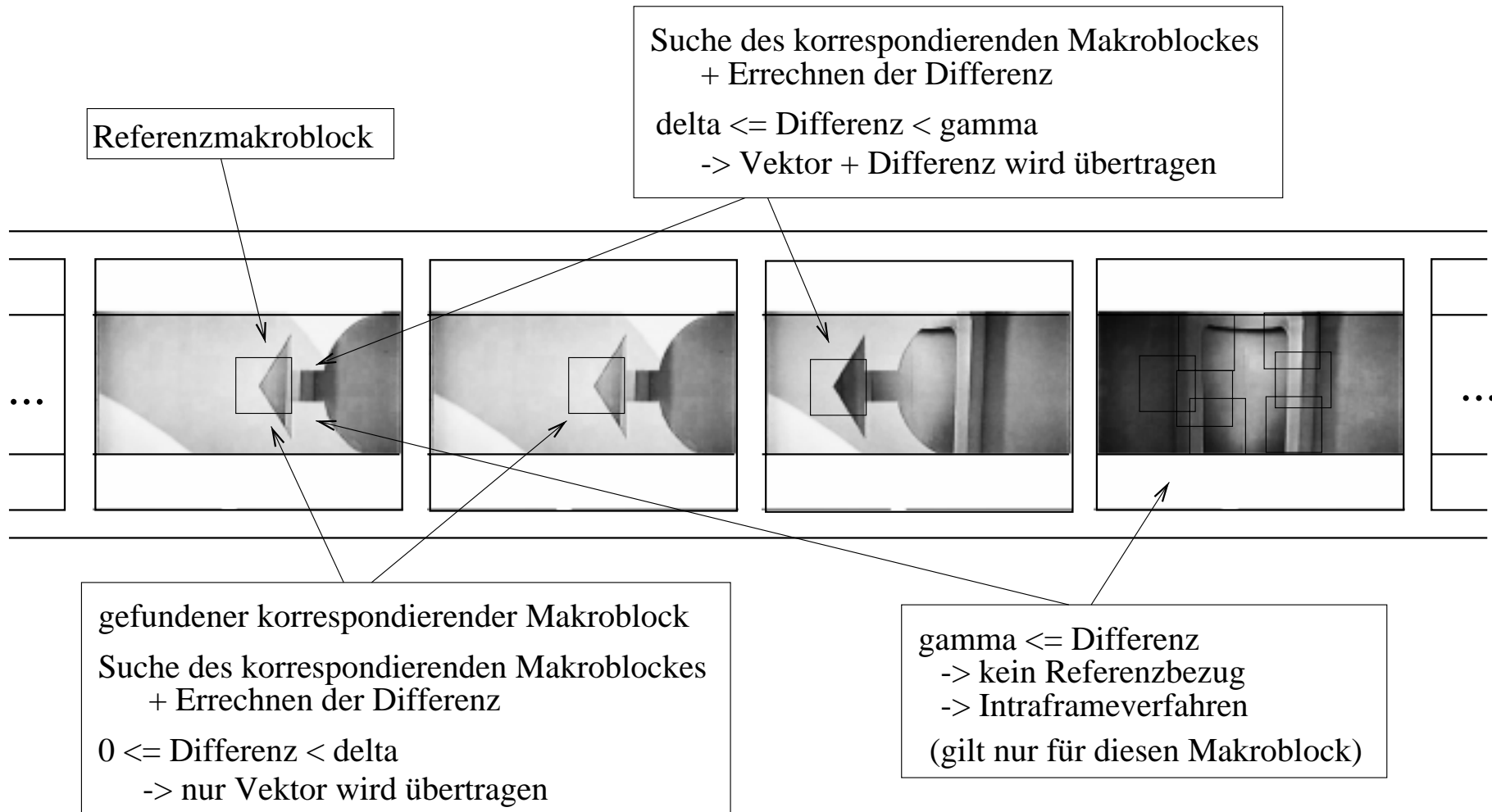
## Bewegungskompensation

- ermöglicht hohe Kompressionsraten
- Überlegung:

Bei aufeinanderfolgenden Bildern werden nur die zeitlich veränderten Blöcke kodiert und übertragen.

... und wie funktioniert nun dies?

- Suche eines Makroblockes (16x16 Pixel)  
in einem Suchbereich (48x48 Pixel) nach Ähnlichkeit  
48x48 = 2304 Möglichkeiten den Block auszuwerten
- an jeder Position 256 Luminanz- und 128 Chrominanz- werte.

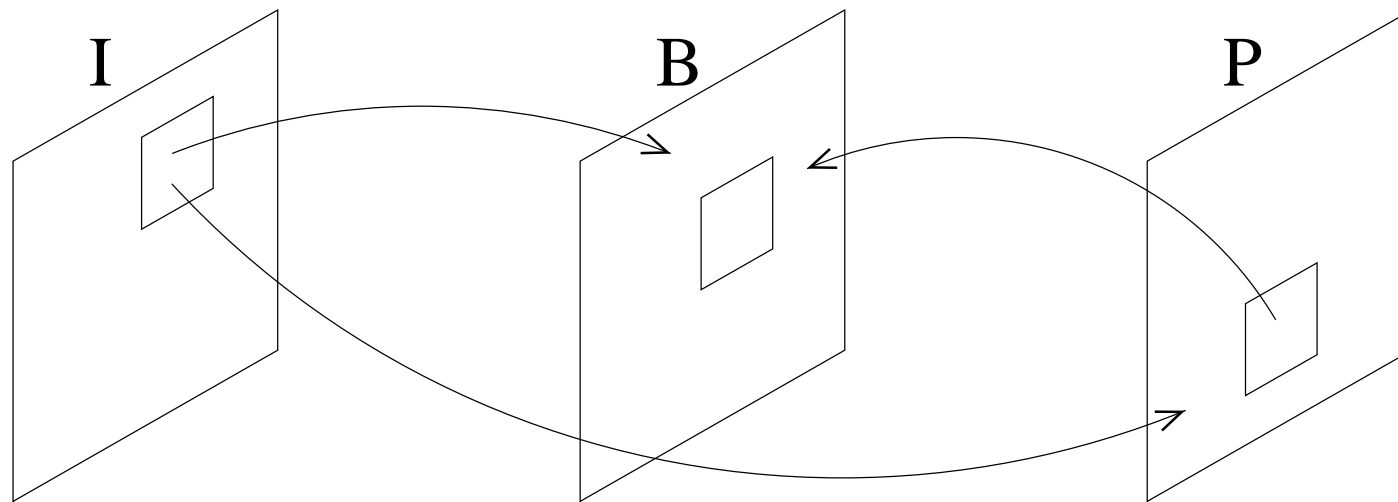


## Interframekodierung

neue Bildtypen

- P- und B- Bilder
- mittels Bewegungskompensation
- “P” = vorhergesagtes Bild  
höhere Kompressionsrate als I-Bilder
- “B” = bidirektionales Bild  
höchste Kompressionsrate
- außerdem “D”-Bilder  
zum schnellen Vor-/Rückspulen (optional)

- Kodierfolge eigentlich I-P-B
- zur Veranschaulichung I-B-P

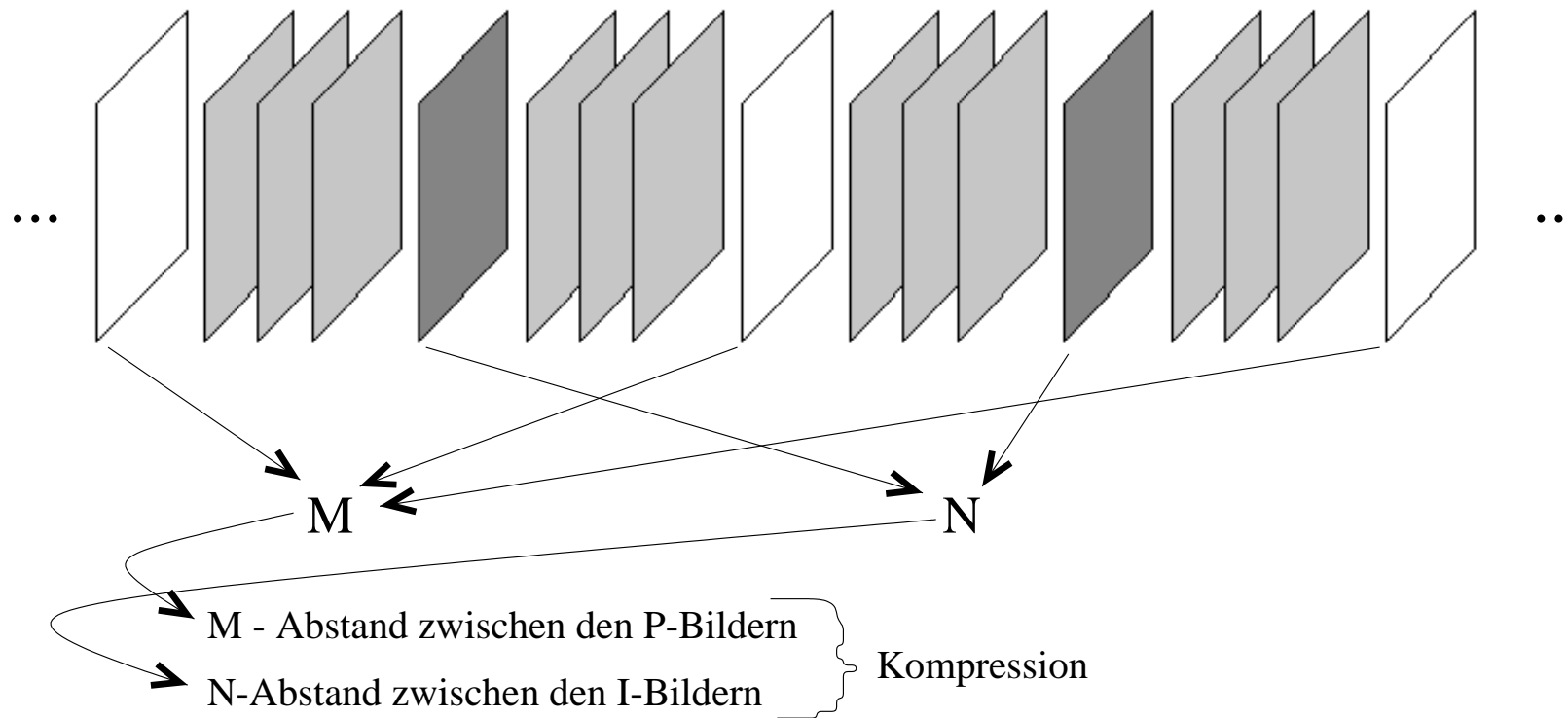


Ermittlung der Differenz zu  
a) vorhergehenden  
b) nachfolgenden  
c) beiden  
Bild(ern).

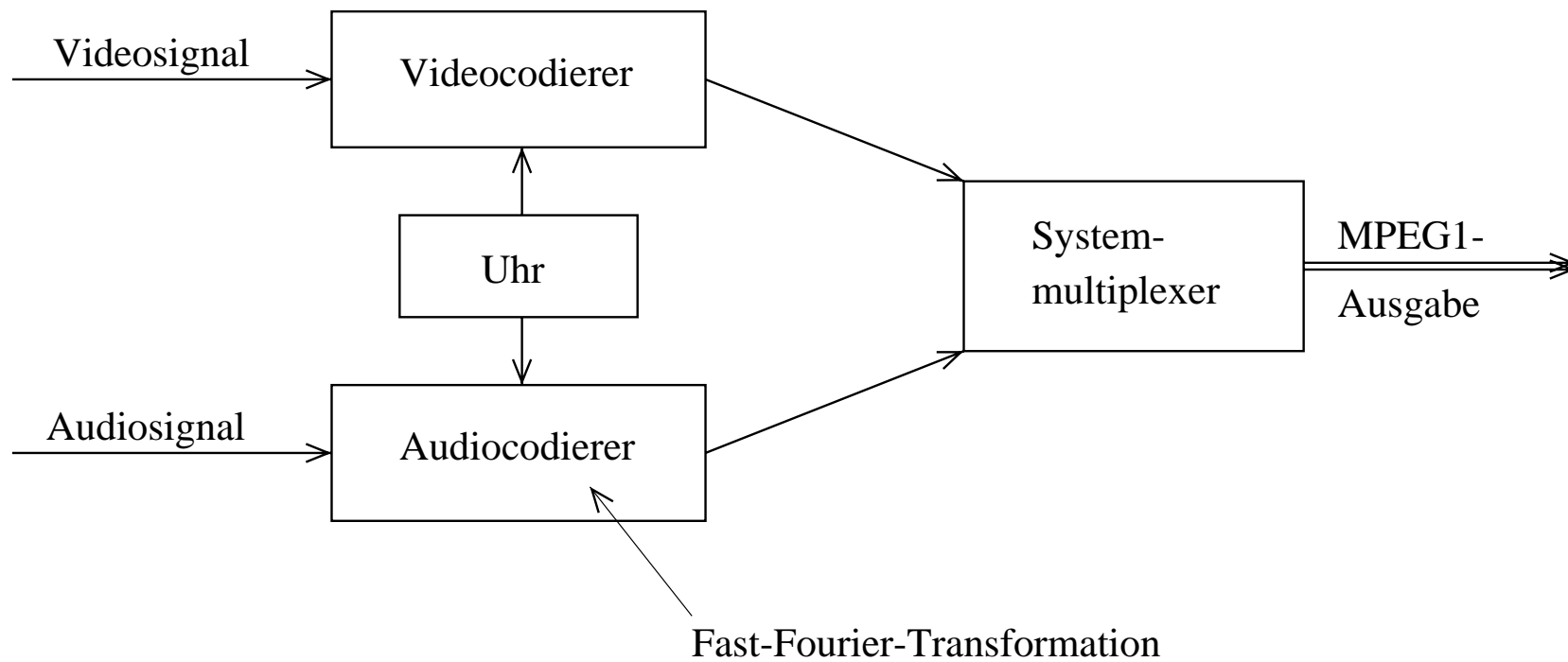
} beste Methode  
aussuchen

Kodierung des  
Bewegungsvektors

# Organisation der GOPs



## Synchronisation der Audio und Videoströme in MPEG



## MPEG2

kein Nachfolger, sondern Untermenge von MPEG  
+ zusätzliche Funktionen

- Unterstützung verschiedener Auflösungen (s.g. Level)  
(352x240), (720x480), (1440x1152), (1920x1080)
- keine "D"-Rahmen
- DCT mit 10x10 Blöcken  
50% mehr Koeffizienten → bessere Qualität
- 5 Profile mit unterschiedlichen Eigenschaften  
Software/Hardware Codierung  
HDTV

## MPEG2 - Profile

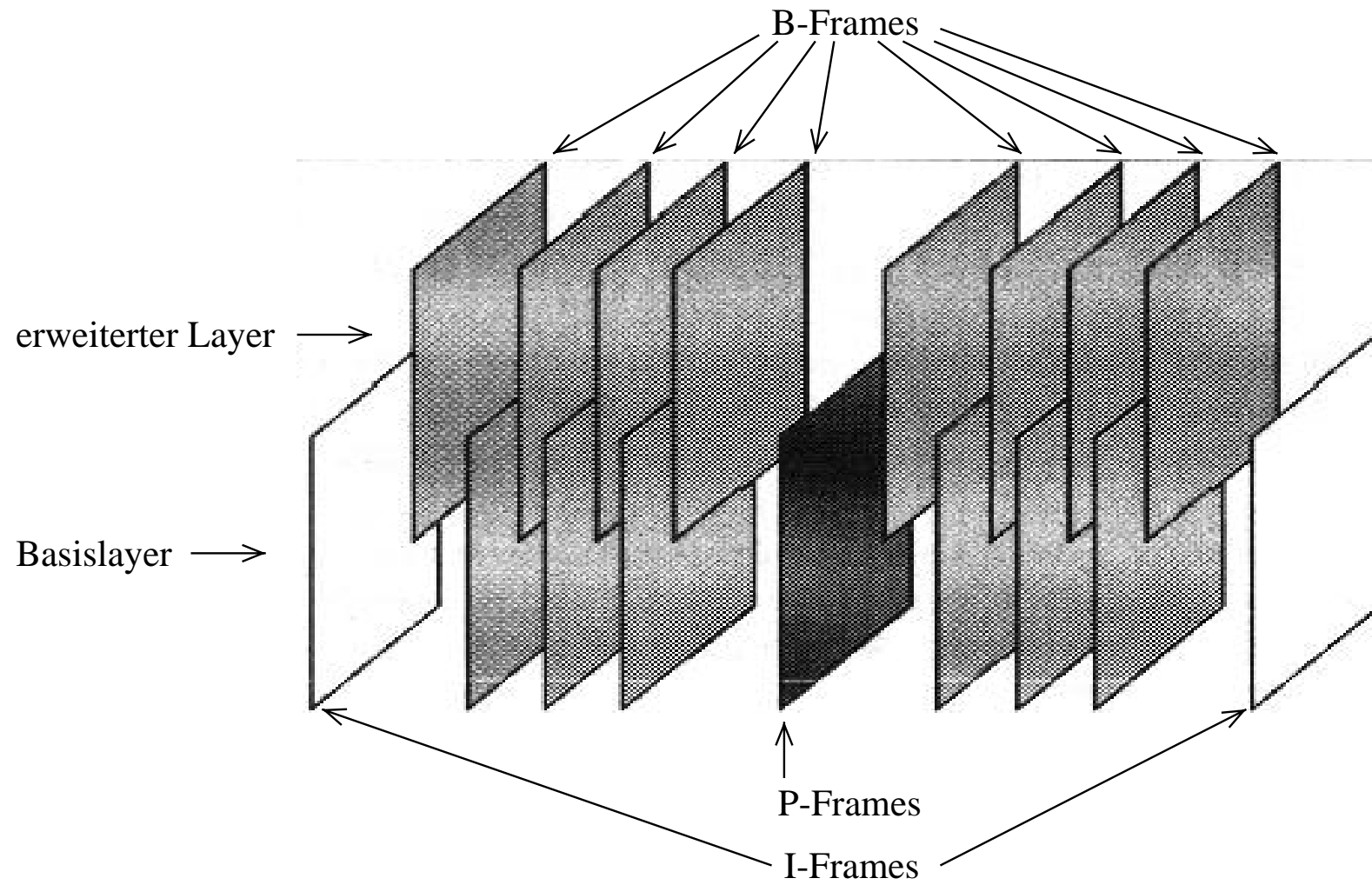
5 Profile \* 4 Level = 20 Kombinationen

Auswahl:

- Simple/Main Profile - für Echtzeitanforderungen  
im 4:2:0 Format ohne Bitratenangabe  
kostengünstige Implementierung  
+ geringe Anforderungen

## MPEG2 - Profile

- SNR Profile - 2 Videoebenen in unterschiedlicher Qualität  
SNR - signal-to-noise ratio  
Versorgung von einfachen und aufwendigen En-/Decodern
- Spatial Profile - HDTV in 2 Qualitätsstufen  
3 Ebenen im 4:2:0 Format, flex. Bitraten  
TV- und HDTV-Stream gleichzeitig möglich
- High Profile - für prof. Studioanwendungen  
wie Spatial Profile mit 4:2:2 Format  
hoher Bandbreitenbedarf



## weitere Modifikationen...

- unbegrenzte Zahl elementarer Ströme  
z.B. verschiedene Sprachen bzw. Untertitel
- PES - Pakete (Packetized Elementary Stream)  
dienen CRC, Verschlüsselungskontrolle  
Copyright, Zeitstempel etc.

→ daraus resultiert aufwendiges Multiplexing

## MPEG4

ein neuer Ansatz ... für noch kleinere Bandbreiten

- Aufteilung der Szenen in ihre Bestandteile  
→ extrem Rechenaufwendig

- Tricks

Audio: z.B. Text to Speech, MIDI

SAOL (Structured Audio Orchestra Language)

Auftrennung in Video- und Synthetische- Objekte

z.B. Shape coding, Texture coding, Sprite coding

2D-Gittermodelle, Gesichts- und Körperanimation

etc.

# Zusammenfassung

in

Computernetzwerke

Andrew. S. Tanenbaum, Prentice Hall (1998)

S.761-781