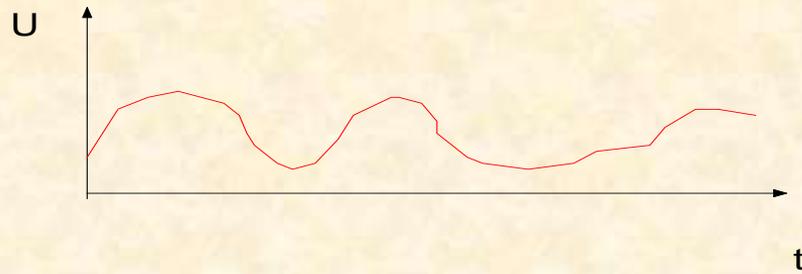
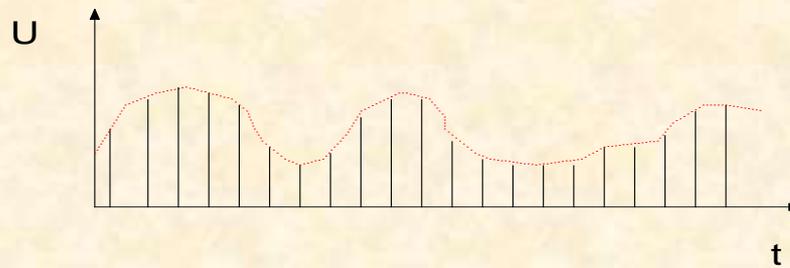


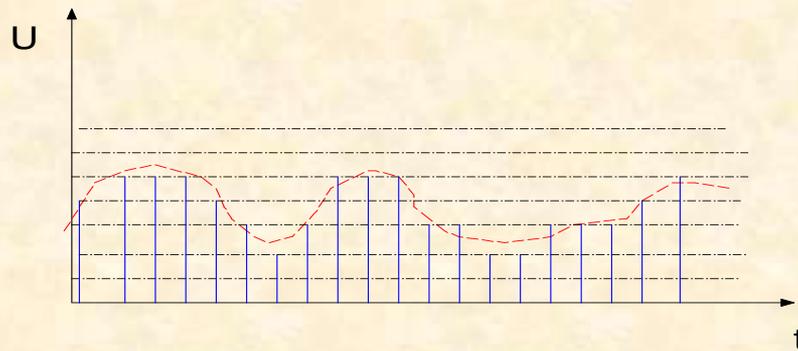
# Digitalisierung



analoges Signal



PAM



Quantisierung

# Datenreduktion

- **Redundanzreduktion (verlustfrei):**  
mehrfach vorhandene Informationen werden nur einmal übertragen, das Signal ist ohne Verluste rekonstruierbar – max. 2:1 Reduktion
- **Irrelevanzreduktion (verlustbehaftet):**  
Unwichtiges wird nicht übertragen, die entstehenden Fehler sind nicht bzw. fast nicht wahrnehmbar – max. 10:1 Reduktion
- **Relevanzreduktion (verlustbehaftet):**  
wahrnehmbare Fehler werden aus ökonomischen Gründen in Kauf genommen – max. 100 :1 Reduktion

# Redundanzreduktion RLE

## RLE (RLC) = Run Length Encoding

- sogenannte Lauflängencodierung
- mehrere gleiche Werte, die in Folge auftreten, erhalten ein eigenes Symbol und werden nicht mehr einzeln übertragen:
- $X < 128$ : übernimm die folgenden  $x+1$  Zeichen  
 $X \geq 128$ : das nächste Zeichen kommt  $257-x$  mal
- Max. 128 Zeichen übernehmbar, max. 128 Kopien eines Zeichens

Beispiel:	<b>1 . 0 0 0 . 0 0 0 , - E u r o</b>	
1.000.000,-Euro	49 46 48 48 48 46 48 48 48 44 45 69 117 114 111	(ASCII)
	1 49 46 254 48 0 46 254 48 5 44 45 117 114 111	(RLE)
1000000,-Euro	_____	(ASCII)
	_____	(RLE)

# Redundanzreduktion VLE

- VLE (VLC) = Variable Length Encodierung
- die Codeworte haben unterschiedliche Länge
- häufig auftretende Werte erhalten ein kurzes Codewort
- selten auftretende Werte erhalten ein langes Codewort
- Beispiel Morsealphabet: e : Punkt  
f : Punkt Punkt Strich Punkt
- verbreitete Methoden für VLE sind:
  - die Shannon/Fano-Komprimierung
  - die (adaptive) Huffman Komprimierung
  - die arithmetische Komprimierung
  - die Komprimierungen nach Lempel-Ziv-Welch
  - (Fax-)Komprimierungen nach CCITT 3/4

# VLE nach Shannon-Fano

Bilde Wurzel (top-down) mit Liste aller Symbole, gewichtet mit Häufigkeiten

While Blätter nicht-elementig do

    Sei L die Liste eines mehrelementigen Blattes

    Bilde zwei etwa gleichgewichtige Listen L0, L1

    und hänge sie als Söhne an Liste L

    (Kanten erhalten 0/1)

end;

**Beispiel:** Die Zeichen a, b, c, d, e sind in einem Text wie folgt verteilt:

*Symbol Häufigkeit*

**a**        **15**

**b**        **7**

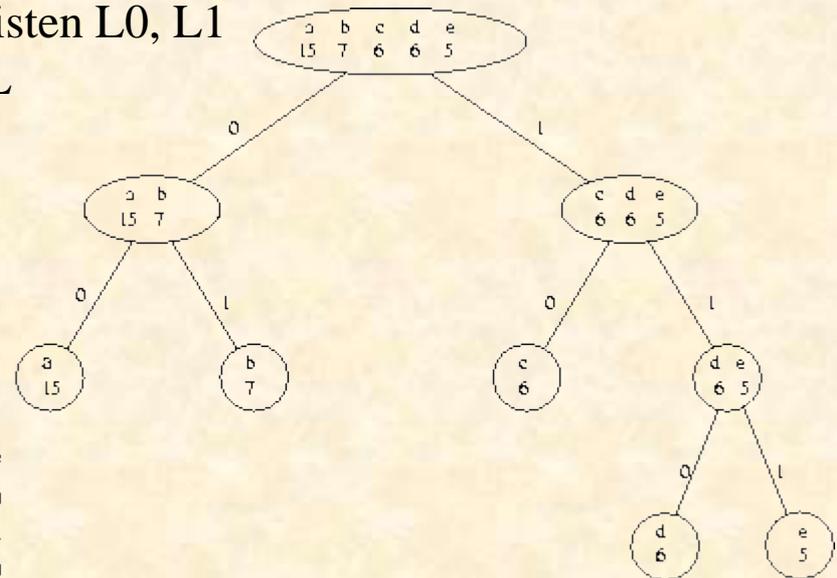
**c**        **6**

**d**        **6**

**e**        **5**

**39 (\*3 Bit = 117 Bit)**

Symbol	Code
a	00
b	01
c	10
d	110
e	111



Nach Anwendung des Algorithmus entsteht obiger Codebaum, Nachrichtenlänge **89 Bit**:

# VLE nach Huffman

Bilde Wald mit Blättern mit Symbolen, gewichtet mit Haeufigkeiten

While noch\_kein\_Baum do

Seien L0, L1 die beiden Wurzeln mit kleinstem Gewicht

Bilde Vater L mit Soehnen L0, L1

(Kanten erhalten 0/1,

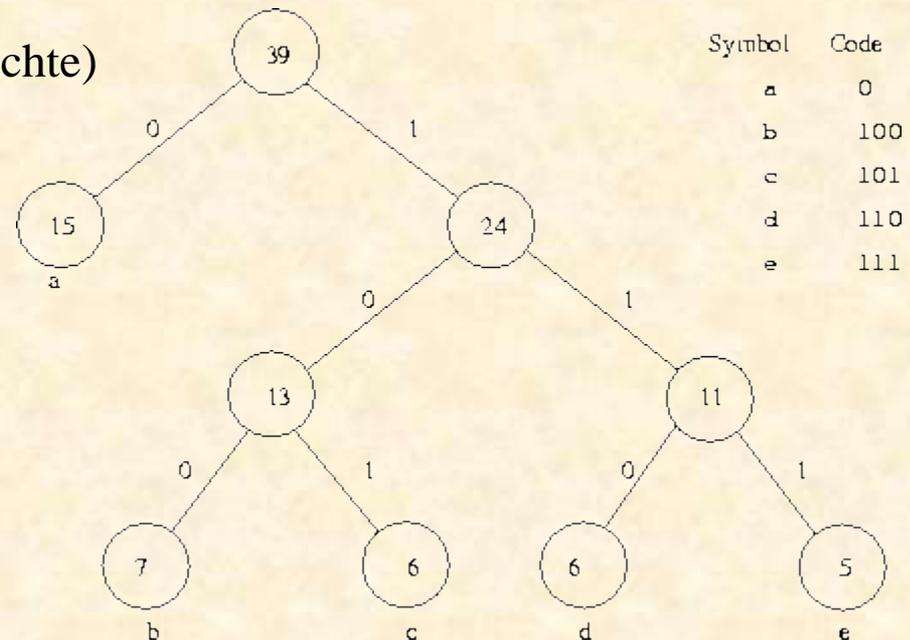
Vater erhaelt Summe der Sohngewichte)

end;

Nachrichtenlänge nun **87 Bit**.

**Satz:**

Huffman nie länger als *Shannon-Fano*!

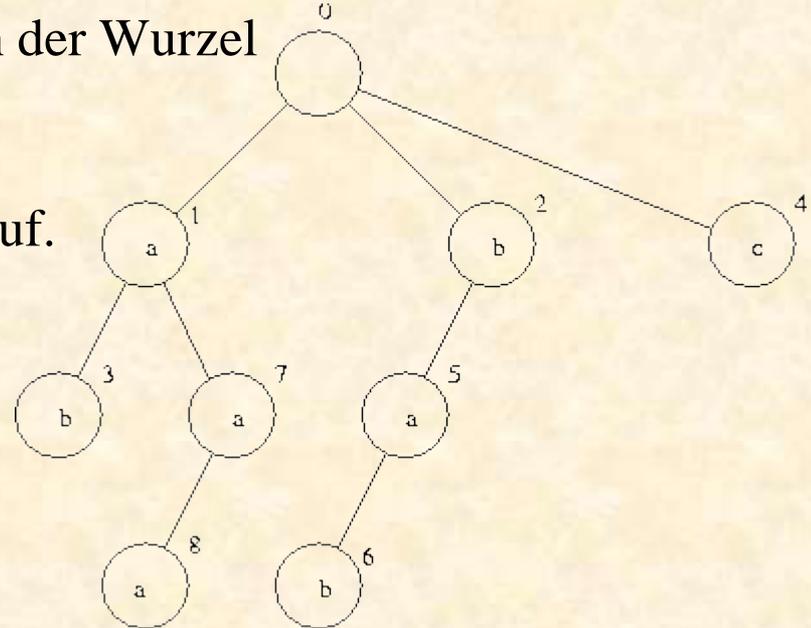


# LZ-78-Komprimierung

Der Kompressionsalgorithmus baut einen Baum auf und erzeugt als Ausgabe eine Folge von Token. Jedes Token besteht aus einer Adresse im Baum (d.h. Verweis auf einen Knoten) und einem Zeichen. Das Token repräsentiert den String, der dem Weg von der Wurzel bis zum referierten Knoten entspricht, verlängert um das betreffende Zeichen. Sender und Empfänger bauen den Baum auf.

Beispiel für LZ 78:

Der String ababcbababaaaaaa  
erzeugt den folgenden Baum:

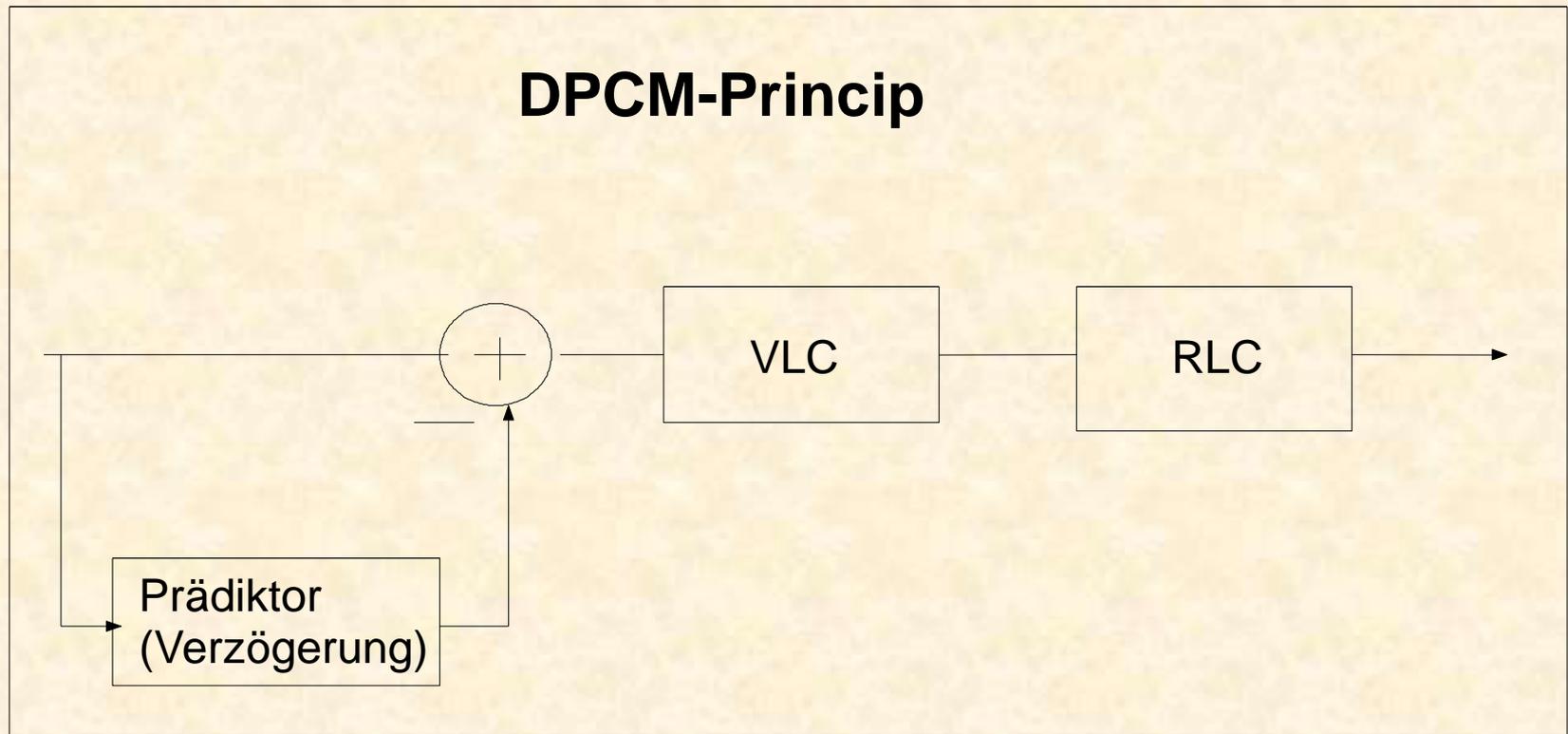


Input	a	b	ab	c	ba	bab	aa	aaa	aa
Output	0a	0b	1b	0c	2a	5b	1a	7a	7EOF

# DPCM - Differential Puls Code Modulation

- Verfahren, um eine günstige Verteilung der Auftrittswahrscheinlichkeit der Bilddaten zu erreichen (für eine effiziente VLE)
- Prinzip:  
Ausnutzung der Regelmäßigkeiten eines Bildes und Übertragung nur der Differenzen benachbarter Bildpunkte. Die Differenzbildung kann so betrachtet werden, dass aus den übertragenen Bildpunkten eine Vorhersage (Prädiktion) für die nächsten gebildet wird. Vom eigentlichen Signal wird diese Prädiktion subtrahiert und nur noch die Abweichung übertragen.

# Datenreduktion DPCM



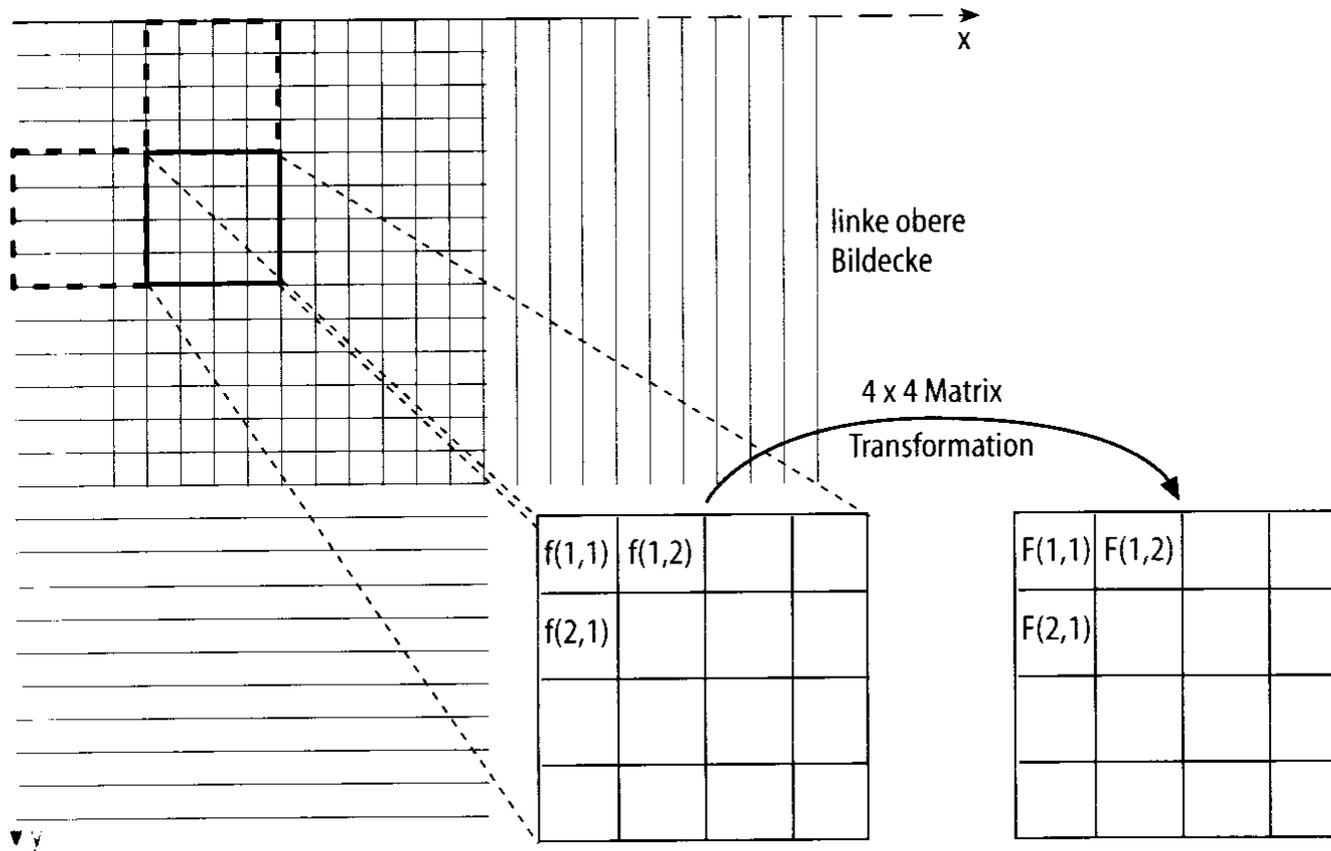
# Bilddatenreduktion

- **Bildcodierung** - Zerlegung des Bildes in Pixel und Beschreibung der Grau- bzw. Farbwerte jedes Pixels durch Zahlen
- um eine effiziente Datenreduktion zu erhalten, erfolgt danach eine Transformation der Daten und eine Codewortzuweisung (wird auf der Empfängerseite wieder rückgängig gemacht).
- Das entsprechende Codierungs- /Decodierungspaar wird als Codec bezeichnet.

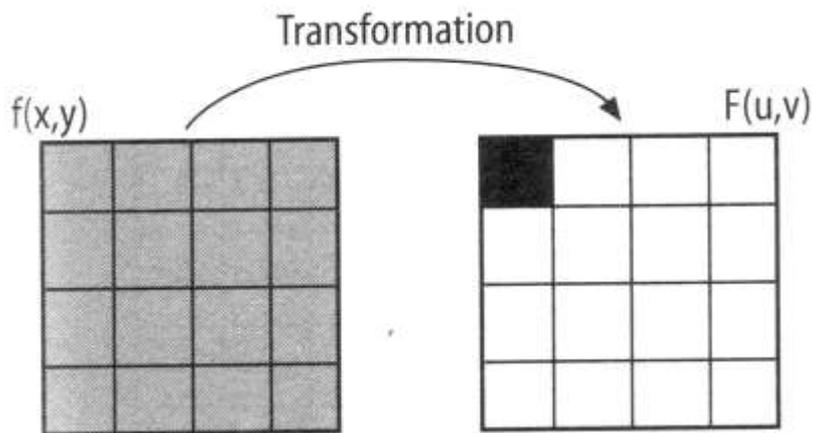
# Bilddatenreduktion

- DCT - Diskrete Cosinustransformation
- Transformation nicht auf Grund der Pixel, sondern auf Grund der sogenannten Ortsfrequenzen
- das Bild wird i.A. in quadratische Pixelblöcke (8x8) aufgeteilt und die Frequenzverteilung codiert. Im Ergebnis relativ kleine Zahlenwerte, die zu übertragen sind und nach Rundung und Einbeziehung von Gewichtungstabellen ist eine sehr effiziente VLE möglich
- bei einer reinen DCT handelt es sich um eine Intraframe - Reduzierung

# Bilddatenreduktion

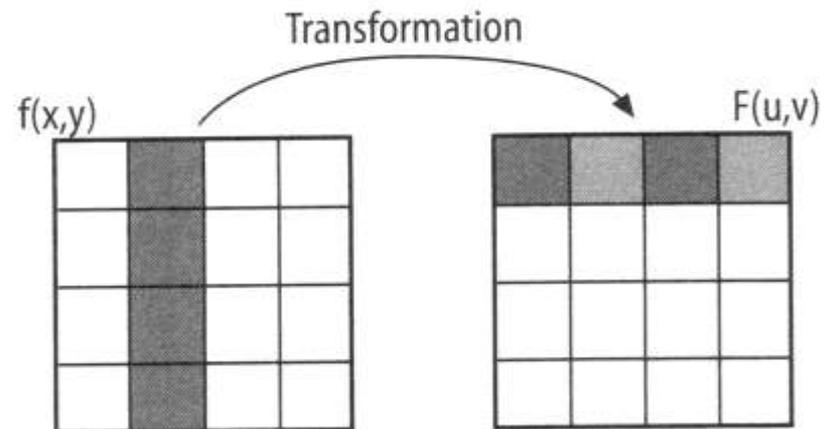


# Bilddatenreduktion



Gleichmäßige Fläche, geringe Frequenz –  
Konzentration auf die DC-Spektral-Komponente

Abb. 3.47. DCT-Beispiele



Vertikaler Puls, höhere Frequenz in x-Richtung –  
Konzentration auf die x-Spektral-Komponenten

# Bilddatenreduktion

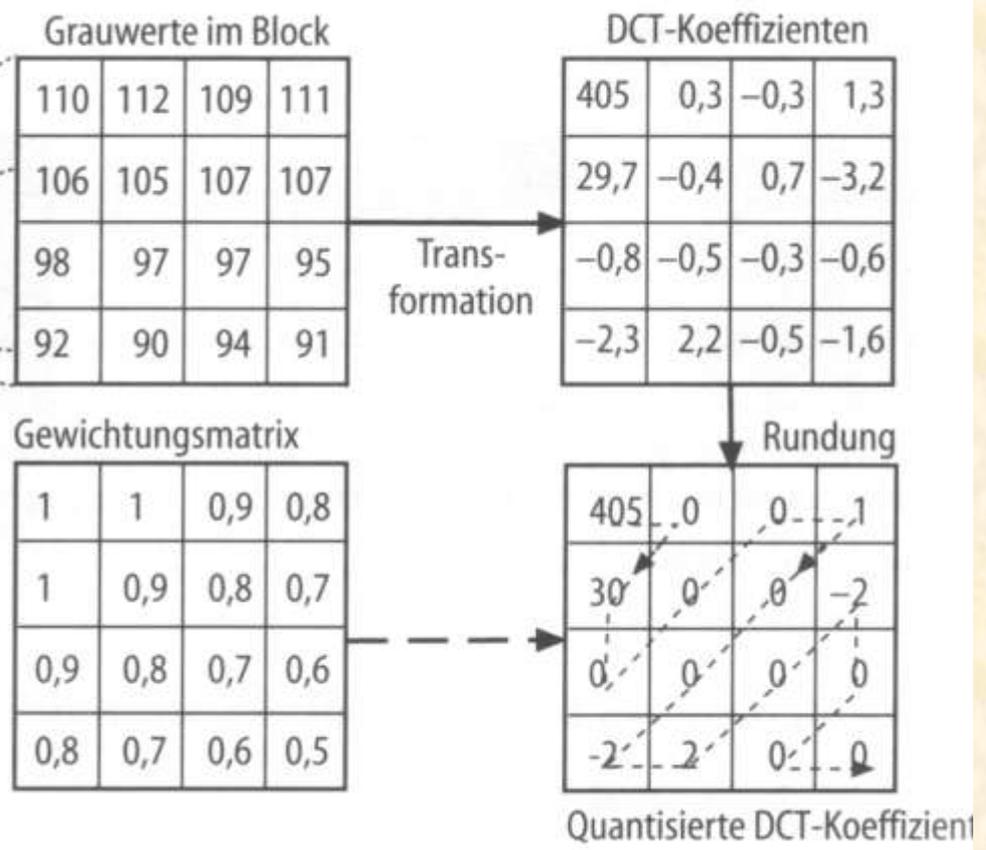
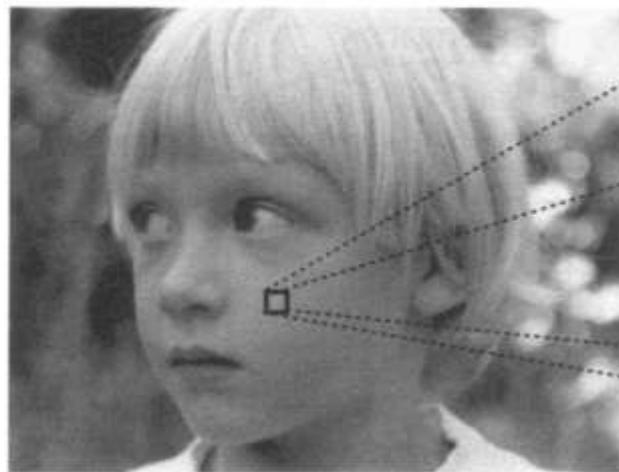


Abb. 3.48. DCT mit Gewichtung und Rundung

# Bilddatenreduktion

- Hybride DCT
- Kombination von DPCM und DCT
- mit Hilfe der DPCM incl. Prädiktion wird zunächst die Differenz von aufeinanderfolgenden Bildern ermittelt und anschließend mit der DCT transformiert.
- Das daraus resultierende Signal (Zahlen) wird im i.A. noch mit VLE weitercodiert.

# Bilddatenreduktion

- Eine Gesamtkompression auf 1:100 bei akzeptabler Qualität ist problemlos zu erreichen. Kompressionen bis zu 1:500 bei verminderter Qualität(Auflösung) und damit Übertragungsraten von 64 Kbit/s sind erzielbar.
- *Hybride DCT ist die Grundlage für DVB und MPEG*

# Videodatenreduktion

Tabelle 3.4. Datenreduktion durch Minderung der Bildpunktanzahl

Signal	Abtasttakt	Werte/Zeile	Zeilenzahl	Datenraten	Gesamtrate	Format
R	13,5 MHz	864	625	108 Mbit/s	324 Mbit/s	4:4:4
G	13,5 MHz	864	625	108 Mbit/s		
B	13,5 MHz	864	625	108 Mbit/s		
Y	13,5 MHz	864	625	108 Mbit/s	216 Mbit/s	4:2:2
C <sub>R</sub>	6,75 MHz	432	625	54 Mbit/s		
C <sub>B</sub>	6,75 MHz	432	625	54 Mbit/s		
Y	13,5 MHz	720	576	83 Mbit/s	166 Mbit/s	4:2:2 nur aktives Bild
C <sub>R</sub>	6,75 MHz	360	576	41,5 Mbit/s		
C <sub>B</sub>	6,75 MHz	360	576	41,5 Mbit/s		
Y	13,5 MHz	720	576	83 Mbit/s	125 Mbit/s	4:2:0 nur aktives Bild
C <sub>R</sub> C <sub>B</sub>	6,75 MHz	360	576	41,5 Mbit/s		
Y	6,75 MHz	360	288	21 Mbit/s	31 Mbit/s	4:2:0 nur aktives Bild
C <sub>R</sub> C <sub>B</sub>	6,75 MHz	180	288	10,4 Mbit/s		

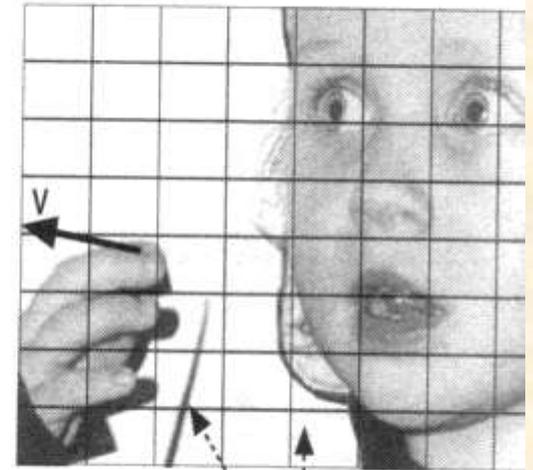
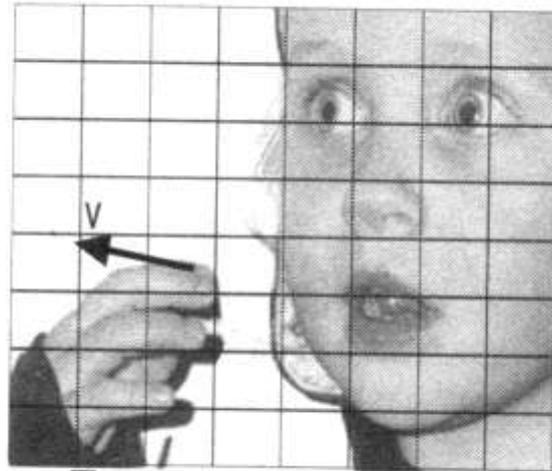
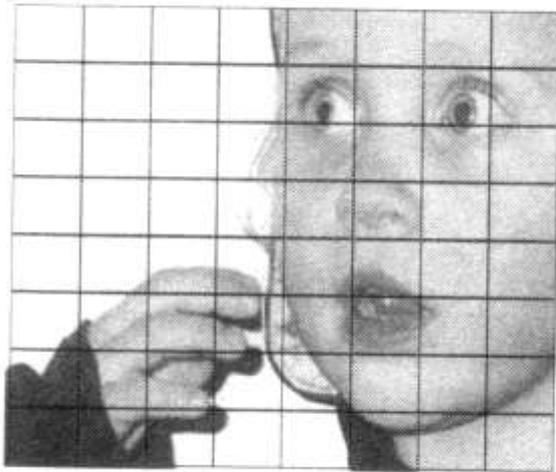
# Videodatenreduktion

- Intraframe: DPCM Ähnlichkeiten innerhalb eines Bildes
- Intraframe: DPCM Ähnlichkeiten innerhalb eines Halbbildes
- Interframe: DPCM Ähnlichkeiten zwischen benachbarten Bildern
  - wesentlich mehr Ähnlichkeiten sind zwischen benachbarten Bildern zu beobachten:  
oft wird nur ein kleiner Bereich des Bildes geändert (Bewegung)
  - wenn die Bewegung angefangen hat, also nach einigen Bildern, kann relativ einfach die weitere Richtung bestimmt werden, d.h. es muss eine Bewegungserkennung mit einer Bewegungsvorhersage erfolgen

# Videodatenreduktion

- Problem:  
der ursprünglich verdeckte Hintergrund kann nicht vorhergesagt werden
- Abhilfe:  
bidirektionale Prädiktion, d.h., es wird auf der Coderseite nicht nur das aktuelle und das vorhergehende Bild berücksichtigt, sondern auch noch das nachfolgende Bild

# Videodatenreduktion



Bestimmung des Bewegungsvektors  $v$

Bestimmung des freiwerdenden Gebiets

Bildpunktblöc

# Videodatenreduktion

- anschließend kann die Bildfolge vor der Übertragung geändert werden, dass zunächst beide Nachbarbilder am Decoder vorliegen
- hier werden jetzt die Bilder durch Integration aus den Differenzen zurückgewonnen und wieder in die richtige Reihenfolge gebracht
- Bilder, die aus unidirektionaler Prädiktion stammen, heißen **P-Frames**
- Bilder aus bidirektionaler Prädiktion heißen **B-Frames**
- Intraframe-codierte Bilder ohne Prädiktion heißen **I-Frames**