Inhalt

- Multimedia-Daten
  - Audio ✓
  - Images
    - Codierung (Farbtiefen, Alpha-Kanal)
    - verlustlose/verlustbehaftete Bildkompression
    - wichtige Formate
  - Video
    - analoges Video-Signal
    - Standards (PAL, NTSC, HDTV, CCIR)
    - digiales Video
    - Standards (DVB)
    - Codierung (YUV, 4:2:2, ...)
    - Kompressionsverfahren (M-JPEG, MPEG, Quicktime)

Images

- Kurze Wiederholung
  - Arten von Bilddaten
    - Bit-Maps
      - bestehend aus Bildpunkten (Pixel), Zeilen/Spalten von Pixeln
      - jedes Pixel hat zugewiesene Bit-Anzahl (1,2,4,8,16,24,32 üblich)
      - jedes Pixel hat Farbe/Grauwert
      - Kompression verlustlos/verlustbehaftet (Auge sieht nicht alles)
    - Vektorbilder
      - Knoten
      - Verbindungen zwischen Knoten (Line Segements)
      - Angaben über Farbe, Linienstärken, Füllungen von Polygonen,...
      - Kompression über entsprechende Kodierung

Images

- Bit-Maps - Pixel
  - Anzahl der Bits/Pixel bestimmt Anzahl der darstellbaren Farben
  - Schwarz/Weiss-Bilder
    - 1 Bit/Pixel
  - Farbbilder
    - jedes Pixel wird durch RGB-Triplet „eingefärbt“
    - RGB-Werte können direkt im Pixel codiert sein
      - 24 Bit/Pixel: je 8 Bit für R/G/B Wert
      - 32 Bit/Pixel: zusätzliche Transparenzangabe (Alpha-Channel)
    - RGB-Werte über „Palette“ (color LookUp Table, LUT)
      - Pixel enthält Index für Tabelleneintrag
      - Tabelle selbst enthält RGB-Werte
      - 2 Bit/Pixel: 4 Farbeinträge möglich (CGA)
      - 4 Bit/Pixel: 16 Farbeinträge möglich (VGA, EGA)
      - 8 Bit/Pixel: 256 Farbeinträge möglich (GIF, VGA)
      - Problem: welche Farben in Tabelle? (Dithering)

Images

- Bit-Maps - Farbmodelle
  - RGB Modell heute sehr verbreitet, additives System
  - CMY(K) Modell (Cyan, Magenta, Yellow, blacK), subtraktives System, Pigemente subtrahieren Farbe von Weiß (Papier), in Druckbereich verwendet
  - HSV (Hue=Farbe, Saturation=Weißanteil, Value=Leuchtkraft), in Grafikformaten selten verwendet
  - YUV (Y=Helligkeit, U=Rot-Cyan-Balance, V=Gelb-Blau Balance), auch YCrCb (Chrominanz Rot/Blau), in Videosystemen im Einsatz, auch in MPEG/JPEG/...

$$\begin{bmatrix} Y \\ C_r \\ C_b \end{bmatrix} = \begin{bmatrix} .299 & .587 & .114 \\ -.169 & -.331 & .500 \\ .500 & -.419 & -.081 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0 & 1.404 \\ 1.000 & -.3434 & -.712 \\ 1.000 & 1.773 & 0 \end{bmatrix} * \begin{bmatrix} Y \\ C_r \\ C_b \end{bmatrix}$$

Images

- Ansätze zur Datenreduktion bei Bilddaten
  - verlustlose Verfahren
    - Entropiecodierung
      - über Histogramm und anschließende Codierung
    - Lauflängencodierung (RLE)
      - Folgen gleichen Inhalts werden durch Tupel (Inhalt, Zählerkennung, Zähler) codiert (z.B. PCX/TGA Format)
    - LZW-Codierung (Lempel, Ziv, Welch)
      - Zuordnung von Codeworten zu sich wiederholenden Zeichenfolgen (z.B. GIF/TIFF-Format) durch Sukzessive Codevergabe
  - verlustbehaftete Verfahren
    - Ausnutzung der Eigenschaften des visuellen Wahrnehmungssystemes (Auge/Gehirn) und der Aufnahme/Wiedergabegeräte
    - Reduktion der Farben/Auflösung/Details

Images

- Ansätze zur Datenreduktion bei Bilddaten
  - Und wieder ein bisserl Physiologie:
    - Auge kann gleichzeitig nur ca. 10.000 Farben sehen, maximal können ca. 60.000 Farben unterschieden werden
    - Auge ist auf Grün am empfindlichsten, dann Rot, Blau
    - Auge kann absolute Farben nicht wahrnehmen, feine Farbunterschiede werden nur bei benachbarten Flächen erkannt
    - Auge kann Farbe von kleinen Objekten nicht wahrnehmen, sieht nur Summenfarbe (-> Farb-Triplets auf Bildschirmen!)
    - Auflösung des Auges ist beschränkt (ca. 600 Linien im Sehzentrum)
    - Auge sieht besser Hell/Dunkel-Unterschiede als Farbunterschiede (Details liegen in Konturen, Farben füllen Flächen -> Malbuch für Kinder)
  - Ansätze zur Datenreduktion bei Bilddaten
    - Reduktion der Farbauflösung
    - Reduktion der Ortsauflösung (Pixelzahl)
    - Quantisierung (Details weg, die nicht gesehen werden)

Images

- Reduktion der Farbtiefe
  - Auge kann ca. 60.000 Farben unterscheiden
  - Reduktion 24 Bit/Pixel (True Color) auf 16 Bit/Pixel (High Color)
    - in der Regel nicht wahrnehmbar
    - wie teilt man Bits auf?
      - 5 Bit für Rot, Grün, Blau -> 1 Bit überzählig
      - Lösung: Auge für Grün am empfindlichsten: 6 Bit für Grün
  - Reduktion von 24 Bit/Pixel auf 8 Bit/Pixel
    - stark sichtbar
    - nur über Palettendarstellung möglich
    - in Palette meist häufigsten Bildfarben (durch Historgramm)
    - Annäherung nicht mehr vorhandener Farben durch Dithering
    - Problematisch: sanfte Farbübergänge, wechselnde Farben bei Bildfolgen durch unterschiedliche Paletten, vorgegebene Systempaletten (VGA)
    - weitere Reduktion: nur in Spezialfällen, Extrem: S/W Bild

Images

- Bsp: Farbreduktion

Reduktion 24 Bit/Pixel auf 8 Bit/Pixel und 3 Bit/Pixel-Palettendarstellung

Images

- Reduktion der Ortsauflösung
  - abhängig vom Wiedergabesystem (oft durch Übertragungsstandard vorgegeben)
  - Bsp: LCD-Display mit 320x200 Pixel, Originalbild vor Übertragung schon reduzieren
  - bei Übertragung Verkleinerung durch
    - Pixel Dropping (Weglassen von Pixel, z.B. jedes 2.)
    - Pruning: Verkleinerung durch Interpolation (2-dimensionale Filter)
  - bei Wiedergabe Vergrößerung durch
    - Pixel Replikation: Problem der Blockbildung (und Treppenbildung - Aliasing)
    - Resizing: Filterfunktion zur Bildvergrößerung samt Anti-Aliasing-Funktion (Schaffung von „weichen“ Übergängen)

Images

- Bsp: Reduktion Ortsauflösung (Bild 480x658 -> 120x165 -> 480x658)

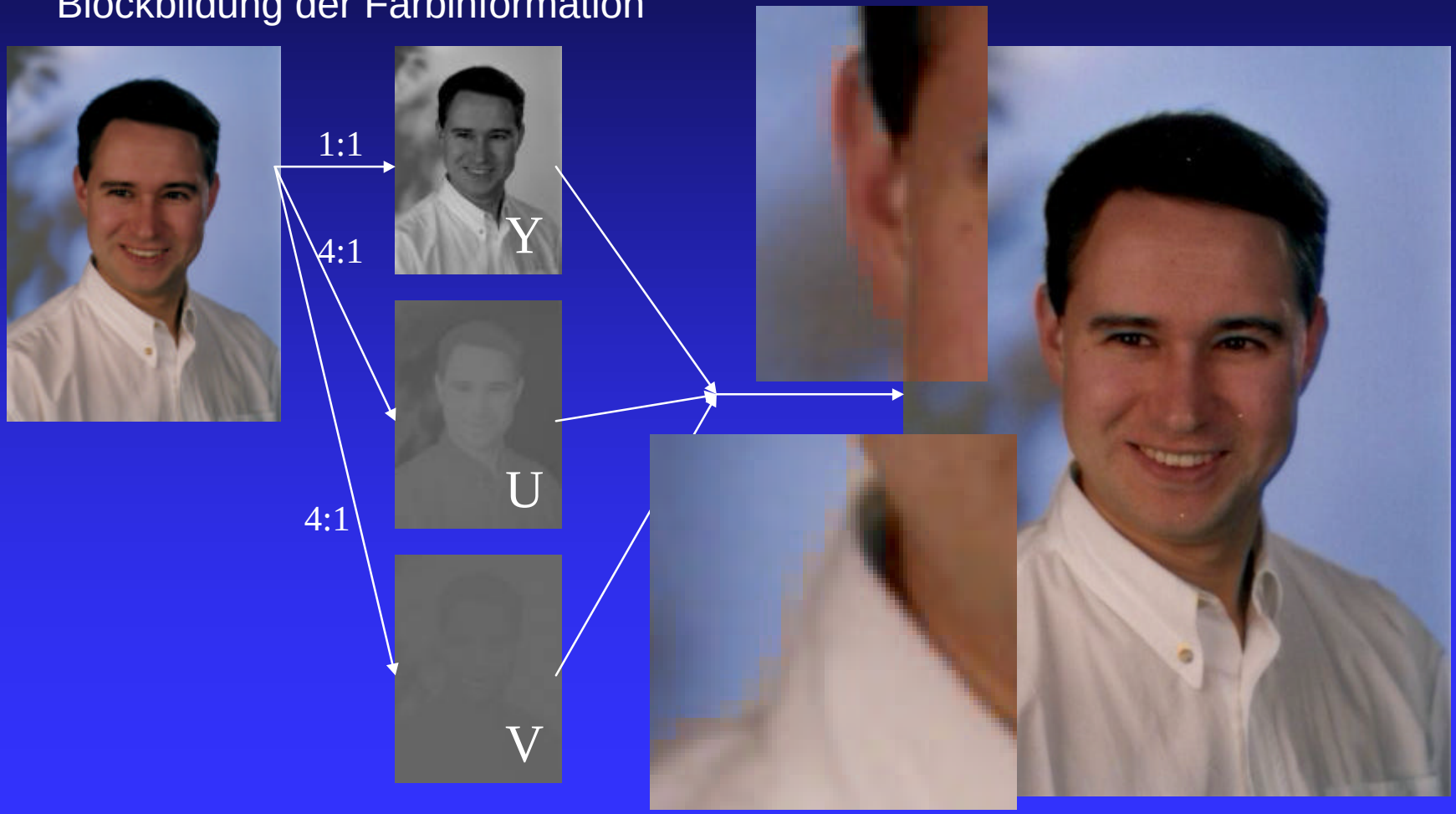
Images

- Reduktion der Farbauflösung (Color-Subsampling)
  - Auge hat schlechtere Farbauflösung wie Helligkeitswahrnehmung
  - Schärfeinformation primär aus Helligkeitsanteil gewonnen
  - YUV-Modell (z.B. MPEG, JPEG)
    - Pixeldarstellung mit jeweils 8 Bit / YUV-Wert liefert volle Auflösung
    - Reduktion der Farbinformation
      - 1/2 Auflösung horizontal „4:2:2“ -> 8 Bit Y pro Pixel, 8 Bit U/V für jedes 2. Pixel -> 30% Reduktion
      - 1/2 Auflösung horizontal/vertikal „4:2:0“ -> Farbinfo nur mehr für jedes 2. Pixel und jede 2. Zeile -> 50% Reduktion
  - andere
    - Block Tuncation Coding (BTC)
      - Regionen von NxN Pixel durch Mittelwert + Abweichung der einzelnen Pixel vom Mittelwert dargestellt (1 Bit Diskretisierung)
    - Color Cell Compression
      - wie BTC, jedoch zusätzliche Farb-Info für 0/1 Bit-Darstellung

Images

- Bsp: Reduktion der Farbauflösung (Color-Subsampling)

  Bild in YUV-Kanäle zerteilt, U/V subgesampelt, wieder zusammengefügt -> Blockbildung der Farbinformation



Images

- Reduktion durch Quantisierung
  - Auge hat unterschiedliche Empfindlichkeiten für Ortsfrequenz (Bilddetails) -> Vorgehensweise bei Bildern, wie bei verlustbehafteten Audiocodecs -> Reduktion der Frequenzanteile
  - verlustbehaftete Bildkompression
    - Dekorrelation: Transformation in eine Darstellung, wo weniger Abhängigkeiten der Elemente untereinander entstehend
    - Anschliessende Reduktion der Daten durch Quantisierung
    - nachher eventuell noch Entropiecodierung
  - Methoden
    - Diskrete Cosinus Transformation (DCT)
      - Darstellung des Bildinhaltes/-ausschnittes durch Überlagerung 2-dimensionaler Cosinus-Schwingungen
      - Problem: Kanten im Bild -> Spektrum unendlich
    - Wavelet-Transformation
      - anstelle Sinus/Cosinus-Schwingungen, andere Basisfunktionen

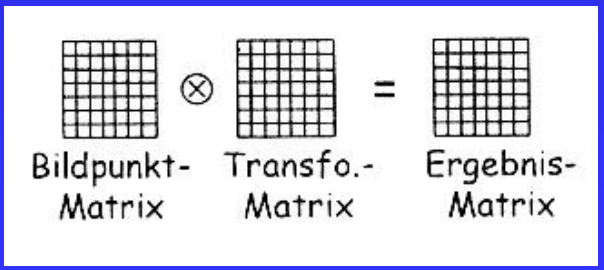
Images

- DCT (Diskrete Cosinus Transformation) - 1
  - von FFT abgeleitet, jedoch ohne Imaginärteil
  - Transformiert 2D-Abbildung in Frequenzbereich
  - Berechnung
    - Bild in quadratische Blöcke NxN Pixel aufgeteilt
    - Pixelwerte f(u,v) mit 0<= u,v < N

$$F(u,v) = \frac{2}{N} C(u)\ C(v) \left[\sum_{k=0}^{N} \sum_{l=0}^{N} f(k,l) \cos \frac{u\pi(2k+1)}{2N} \cos \frac{v\pi(2l+1)}{2N} \right]$$

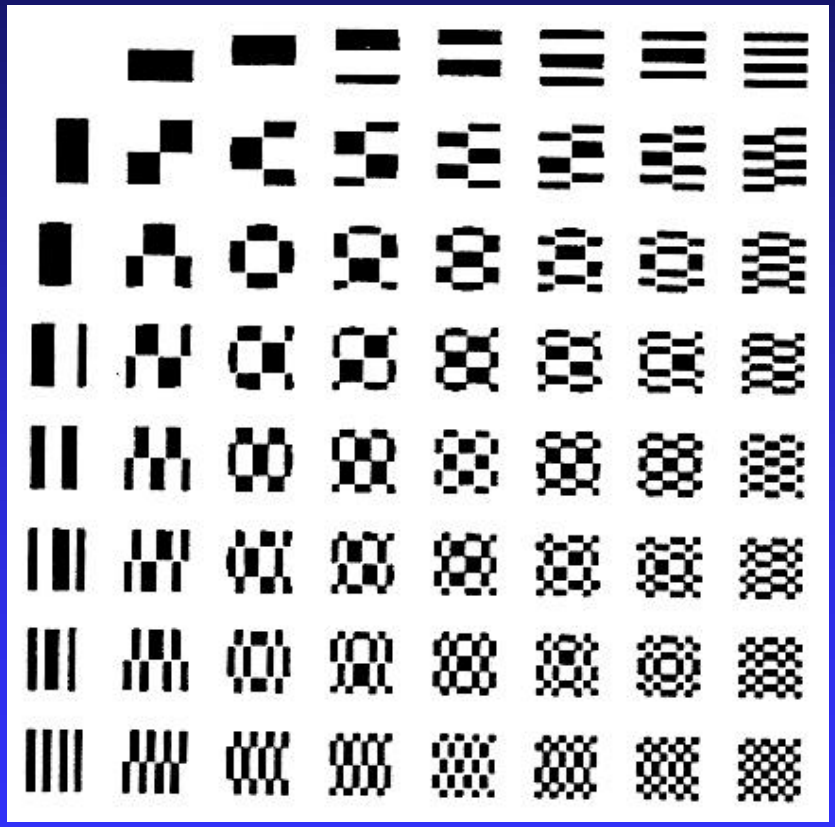
$$C(u)\ bzw.\ C(v) = \frac{1}{\sqrt{2}}\ für\ u,v = 0\ sonst\ 1$$

    - damit reduziert sich DCT auf blockweise Matrixmultiplikation



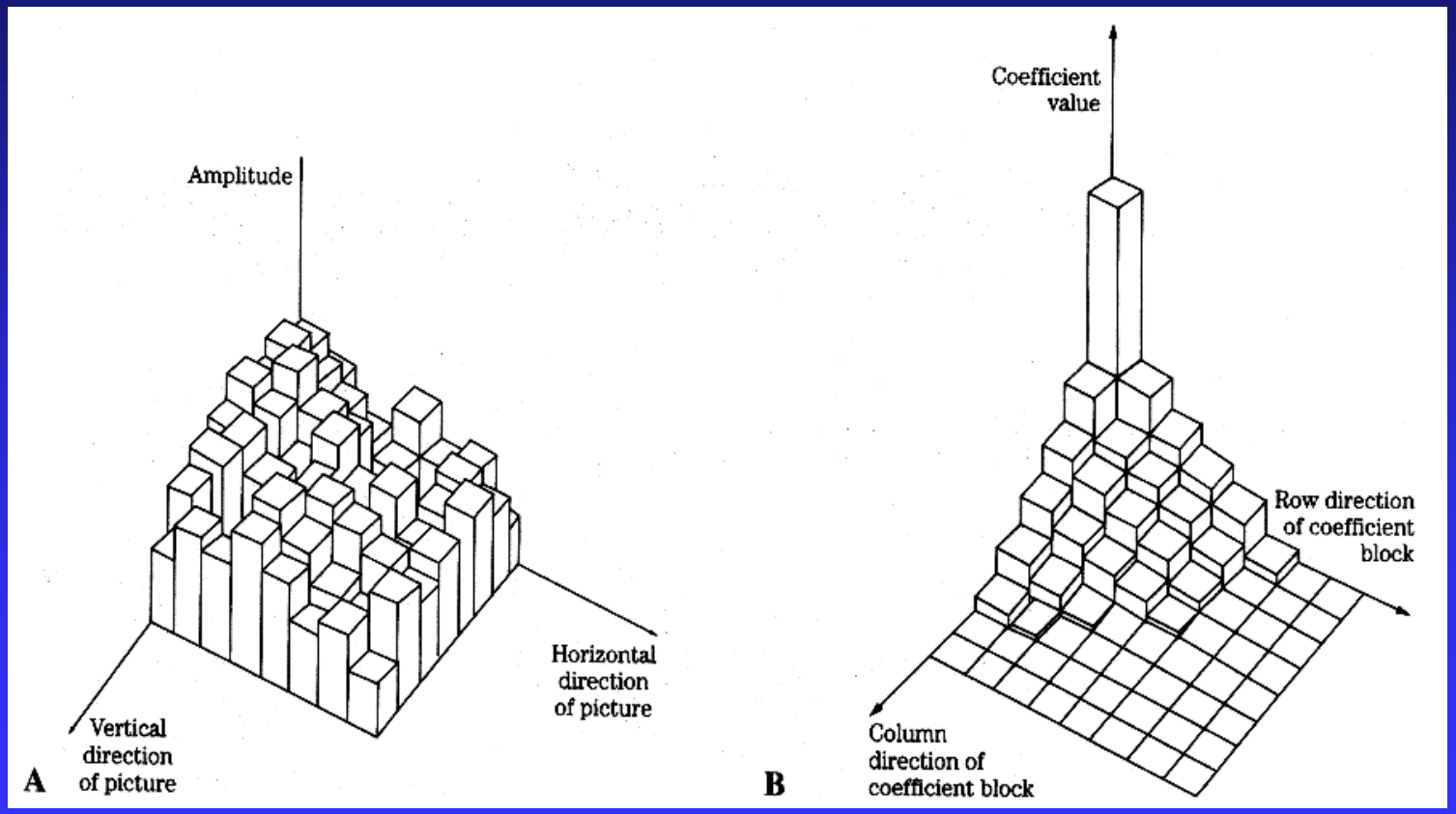
Images

- DCT - 2
  - DCT liefert Matrix der Frequenzanteile des Bildblockes
  - links oben: Gleichanteil (DC-Koeffizient), „Grundhelligkeit“
  - rechts unten: höchstfrequente Anteile, Details des Blockes



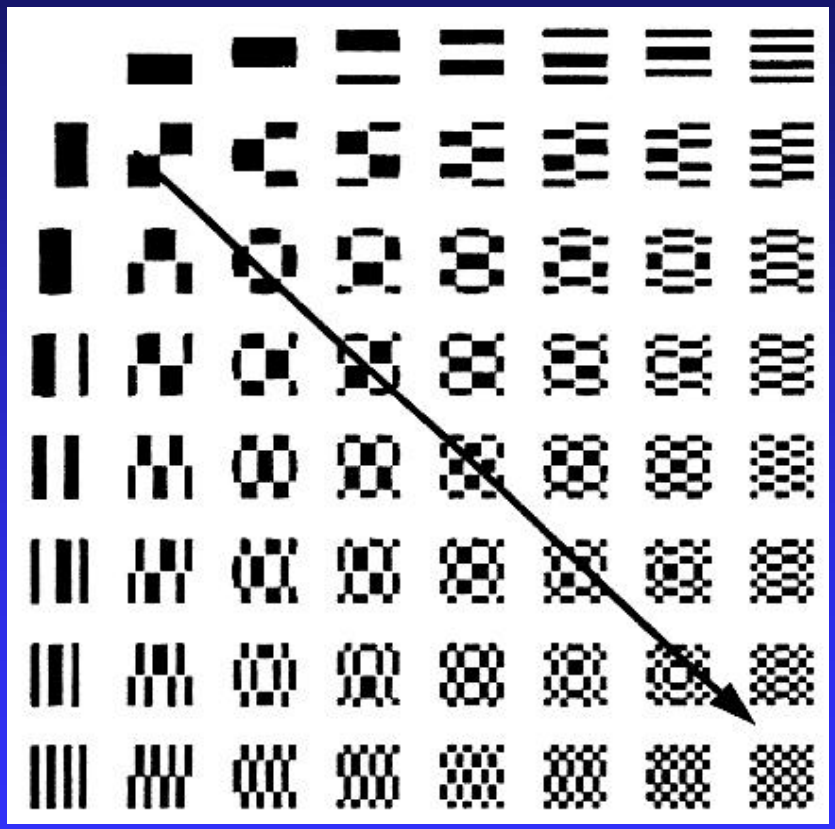
Images

- DCT - 3
  - Transformation vom Bild (Pixel) in Spektrum



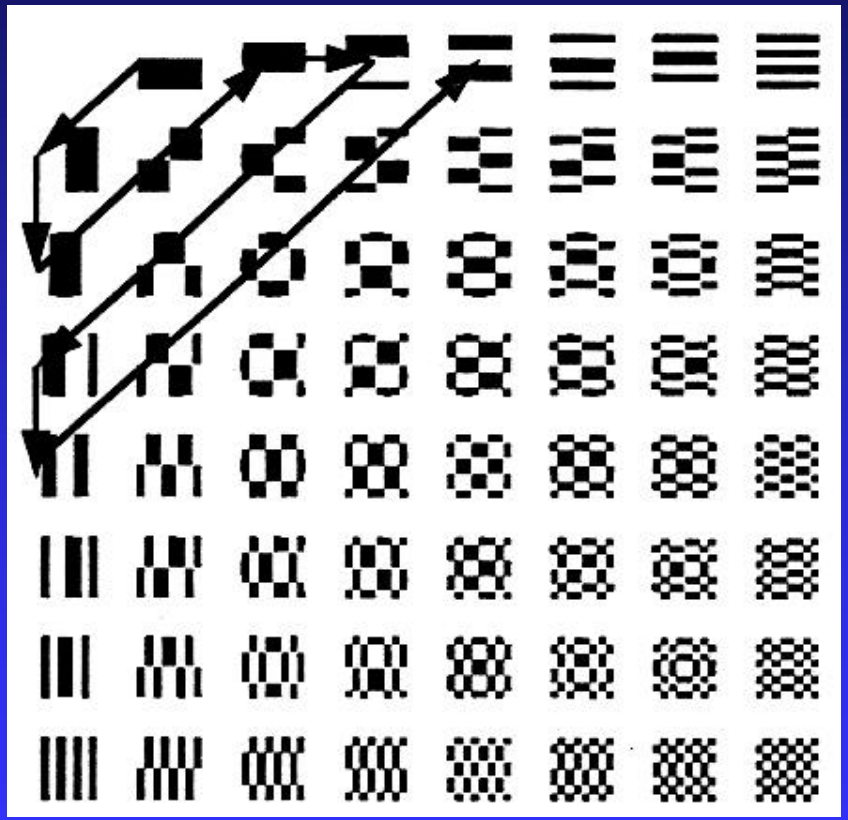
Images

- DCT - 4
  - Ansatz zur Datenreduktion
    - abnehmende Wahrscheinlichkeiten von l.o. nach r.u.
    - abnehmende Relevanz (Details)



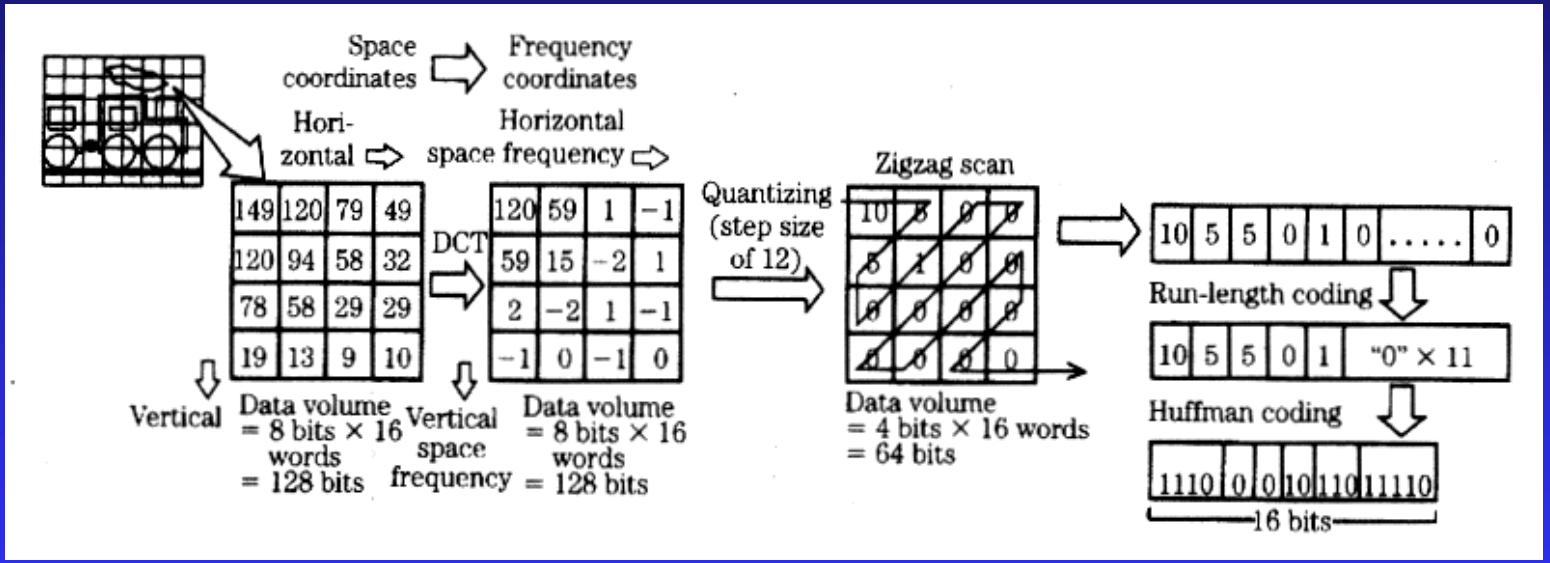
Images

- DCT - 5
  - Ansatz zur Datenreduktion
    - Auslesen der Koeffizienten im Zickzack-Scan
    - lange Nullketten zu erwarten
    - anschließende Quantisierung - Reduktion
    - anschließend Entropiecodierung (z.B. Huffman) - Kompression



Images

- DCT - 6
  - Beispiel Bildkompression



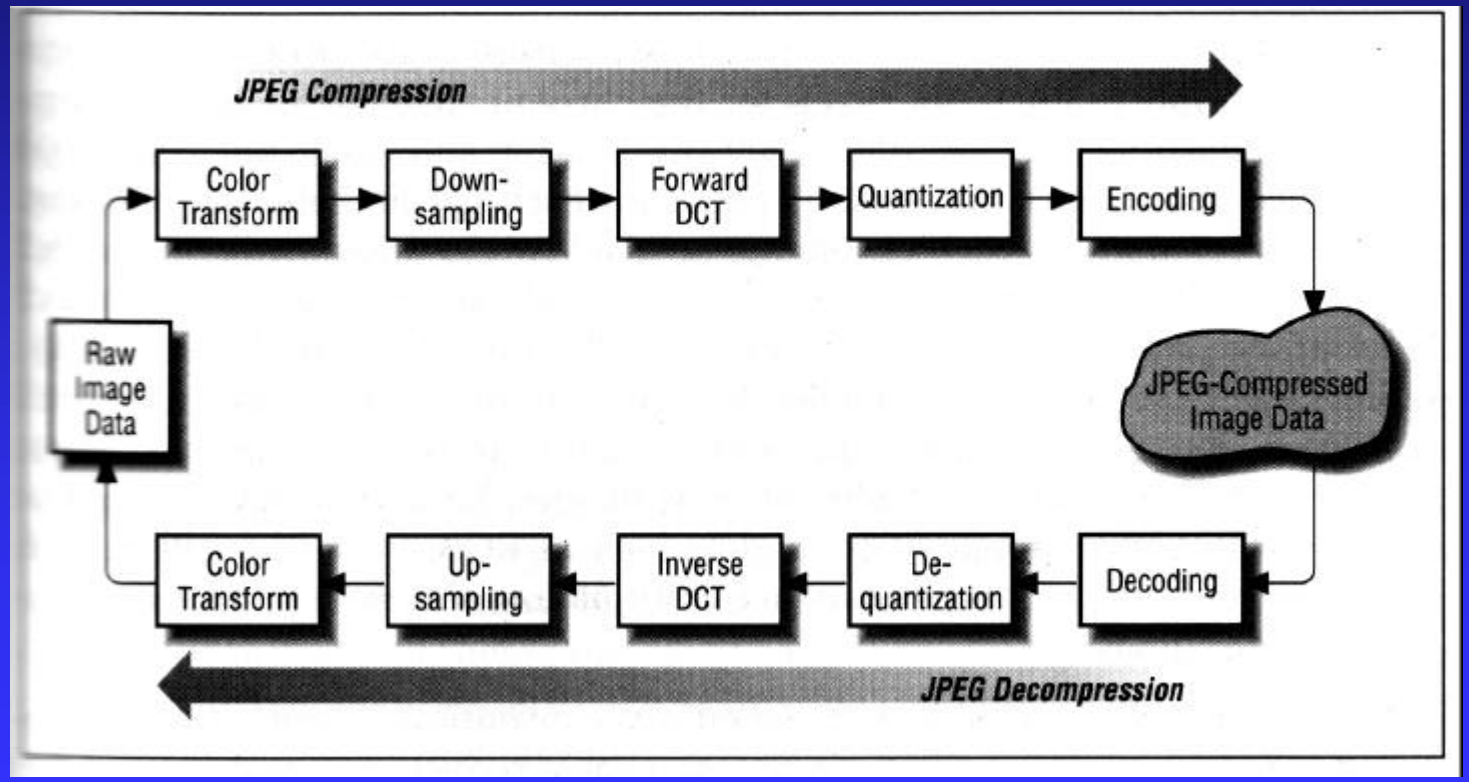
Images

- JPEG - Bildformat
  - JPEG (Joint Photographics Expert Group), 1992 ISO-Norm
  - Bildformat zur Speicherung von 24 Bit Truecolor-Bildern
  - Einsatz von Color Subsampling, DCT und Entropiecodierung
  - verlustbehaftete Kompressionsfaktoren von 20:1 und mehr erreichbar

  - Baseline System
    - sequentielle Codierung des Bildinhaltes
  - Extended System (optional)
    - progressive Codierung des Bildinhaltes
    - arithmetic encoding (10-15% kürzere Codes wie Huffman)
    - lossless compression

Images

- JPEG - Bildformat - 2
  - Baseline System
    - Encoder/Decoder hat 5 Verarbeitungsschritte



Images

- JPEG - Bildformat - 3
  - Baseline System
    1. Color Transform
       - Bild wird in Farbkomponenten unterteilt
       - Farbbilder meist in YCrCb-Format konvertiert
    2. Downsampling
       - Subsampling der Farbinformationen im 4:2:2 oder 4:2:0 Format
       - Helligkeitsinformationen bleiben 1:1 erhalten
    3. DCT
       - auf jede Komponenten wird DCT in 8x8 Pixel-Blöcken angewendet
    4. Quantisierung
       - jede DCT-Koeffizientenmatrix wird durch Quantisierungsfaktoren aus vordefinierten Tabellen dividiert und in Integers umgewandelt
       - Quantisierung erfolgt für Luminanz/Chrominanz getrennt
       - Chrominanz stärker quantisiert (Farbsehen), ebenso Koeffizienten höherer Ordnung (Auflösung)
       - Tabellen in Codecs von ISO/JPEG, durch Q-Faktor (User) verändert

Images

- JPEG - Bildformat - 4
  - Baseline System

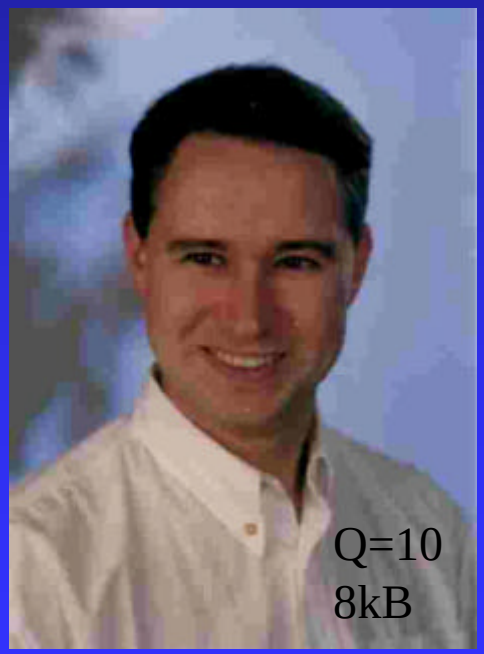
    5. Encoding
       - Zick-Zack-Scan der quantisierten Koeffizientenmatrix
       - Huffman-Codierung zur Entropiereduktion

  - Decoder
    - durchläuft Schritte umgekehrt
    - gleicher Aufwand wie bei Encoder

  - wichtig für JPEG
    - richtige Wahl des Q-Faktors für Quantisierung, sonst starke Artefaktbildung

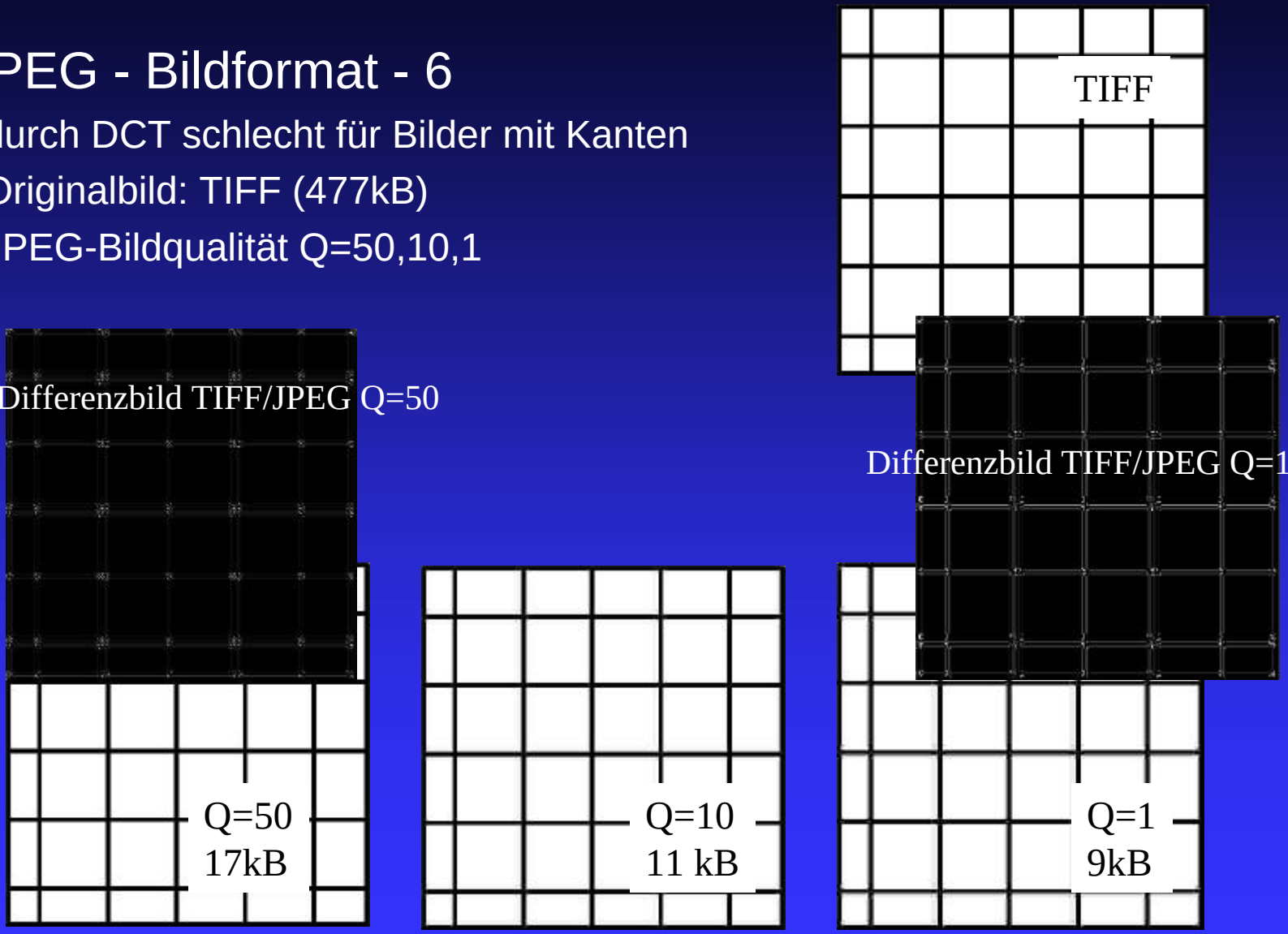
Images

- JPEG - Bildformat - 5
  - Original Bild: 156kB
  - JPEG-Bildqualität Q=75,50,10,1



Images

- JPEG - Bildformat - 6
  - durch DCT schlecht für Bilder mit Kanten
  - Originalbild: TIFF (477kB)
  - JPEG-Bildqualität Q=50,10,1

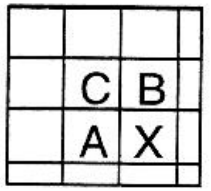


Images

- JPEG - Bildformat - 7
  - Extended System
    - Progressive Codierung
    - Bild wird in mehreren Durchgängen codiert
    - 2 Arten
      - Spectral Selection
        - zuerst nur niederfrequente Koeffizienten für grobes Bild
        - dann höher Auflösende Differenzbilder
      - Successive Approximation
        - zuerst werden nur höchstwertigen Bits der Koeffizienten codiert
        - in weiteren Schritten die niedrigwertigen Bits
    - Vorteil: Bild kann schon während Übertragung angezeigt werden (zuerst grob, dann immer feiner)
    - Nachteil: jedesmal kompletter JPEG-Decoderdurchlauf notwendig (CPU)

Images

- JPEG - Bildformat - 8
  - Extended System
    - verlustfreie Codierung
    - keine DCT, keine Blockbildung, Pixelauflösung 2-12Bit
    - verwendet 2D-DPCM Verfahren
    - setzt Pixel X über Prädiktor mit Nachbarpixel (A,B,C) in Verbindung (2x2Pixelgruppen)
      - 8 Prädiktorfunktionen definiert (3 Bit)
      - Prädiktionsfehler des Pixel + Code für Prädiktorfunktion wird übertragen



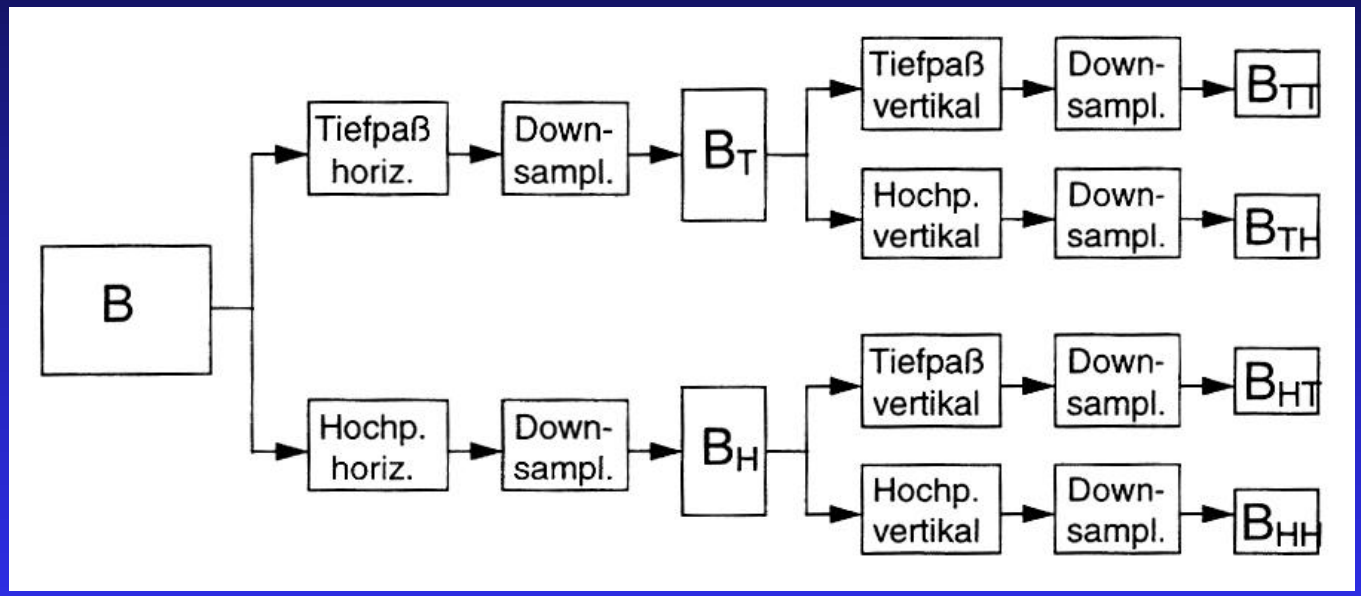
Code	Prädiktoren
0	keine Präd.
1	X=A
2	X=B
3	X=C
4	X=A+B+C
5	X=A+(B-C)/2
6	X=B+(A-C)/2
7	X=(A+B)/2

Images

- Wavelet - Transformation (informell) - 1
  - aus Mathematik, Ende der 80er
  - Problem von FFT/DCT
    - Kanten in Bildern (sehr hohe Frequenzanteile im Spektrum)
  - Idee der Wavelets
    - andere Basisfunktion anstelle Sinusschwingung zu verwenden
    - Ziel: beliebige Frequenzen mit wenig Koffezienten darzustellen
    - wie FFT/DCT: Transformation zur Dekorrelation, dann Quantisierung und Entropiecodierung
    - Transformation aber nicht blockweise sondern für ganzes Bild
    - Implementierung: Filter zerteilt Bild in horizontale/vertikale Teilbilder die jeweils auf 1/4 der Auflösung reduziert werden
    - Filter ist Digitalfilter das „Basisfunktion enthält“
  - Problem: fehlende Standardisierung, Wahl der Basisfunktion

Images

- Wavelet - Transformation - 2
  - Implementierung der Filterung



B_{TT}: geglättete Verkleinerung des Originalbildes (1/4 Auflösung)

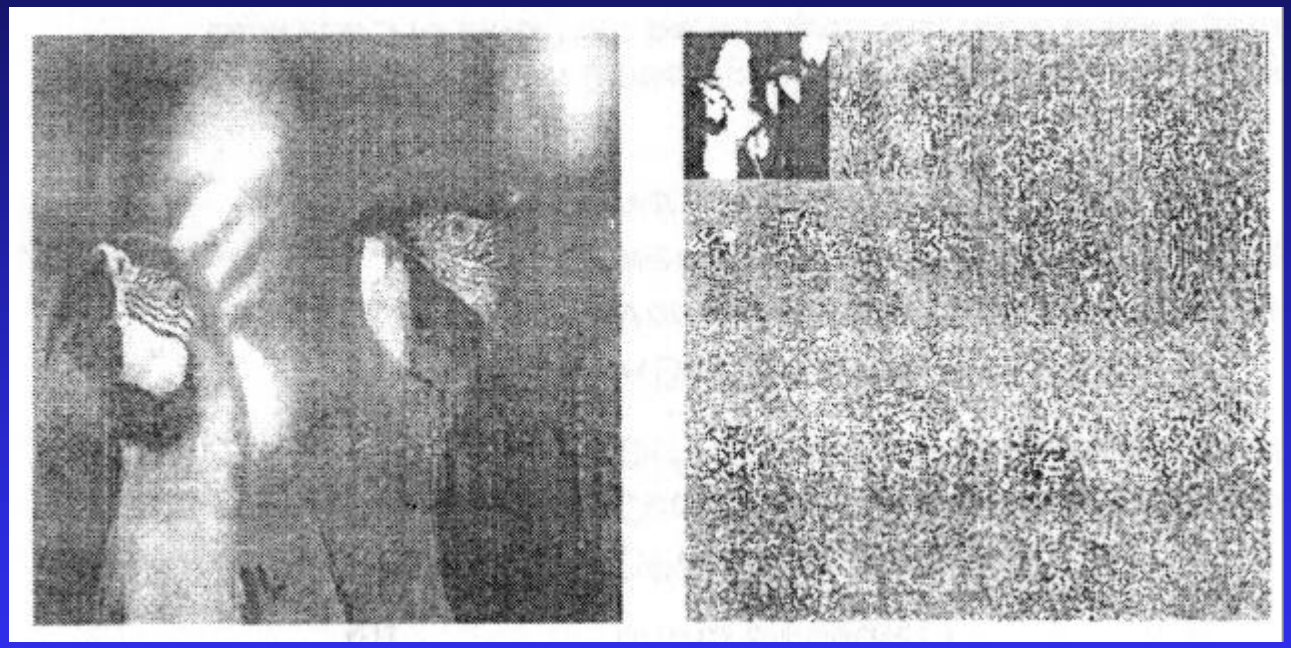
$B_{TH/HT}$: Details in der Horizontalen/Vertikalen

B_{HH}: diagonale Bildstruktur

Downsampling: Entfernen jeder 2. Zeile/Spalte (verlustfrei durch Digitalfilter)

Images

- Wavelet - Transformation - 3
  - Beispiel: Bild nach 2 Transformationsschritten



  - Decoder:
    - Einfügen von Nullzeilen/Spalten in Teilbilder
    - Rücktransformation durch Rücktransformationsfilter

Images

- Wavelet - Transformation - 4

  Beispiel: Wavelet Darstellung (Corel Photo Paint)

  JPEG (Q=50, 15 kB)

  Wavelet (3 kB)