

## Grundlagen der Bildkompression (Teil 3) und Einführung in JPEG

Dr. Detlev Marpe  
Fraunhofer Institut  
für Nachrichtentechnik HHI



Fachhochschule für Technik  
und Wirtschaft Berlin  
University of Applied Sciences

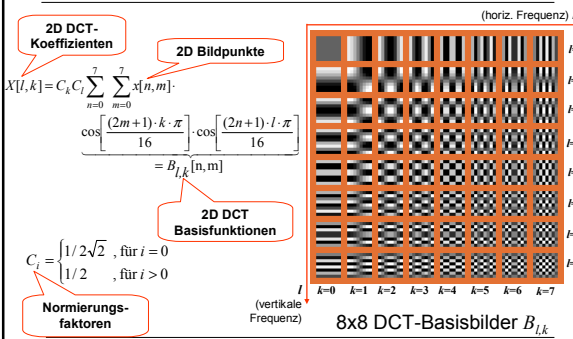


Fraunhofer Institut  
für Nachrichtentechnik  
Heinrich-Hertz-Institut

### 8x8 DCT – Berechnungsvorschrift

$$X[l, k] = C_l C_k \sum_{n=0}^7 \sum_{m=0}^7 x[n, m] \cdot \cos\left[\frac{(2m+1) \cdot k \cdot \pi}{16}\right] \cdot \cos\left[\frac{(2n+1) \cdot l \cdot \pi}{16}\right]$$

$$= B_{l,k}[n, m]$$

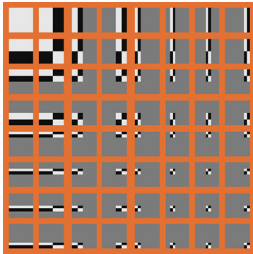
$$C_i = \begin{cases} 1/\sqrt{2}, & \text{für } i = 0 \\ 1/2, & \text{für } i > 0 \end{cases}$$


8x8 DCT-Basisbilder  $B_{l,k}$

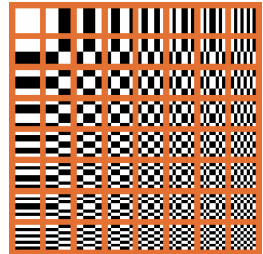
### 8x8 DCT – Berechnungsaufwand

- Direkte 1D 8-DCT Implementierung als Matrixmultiplikation erfordert 64 Multiplikation und 56 Additionen
- **Schnelle 1D 8-DCT** Implementierung möglich durch geschickte Ausnutzung von Redundanzen und Symmetrien in der Definition der DCT-Basisfunktionen: **22 Multiplikationen** und **28 Additionen**
- [Arai, Agui, and Nakajima; 1988 (AAN88)]: Schnell(st)e 1D 8-DCT durch Faktorisierung in dünnbesetzte Matrizen: **5 Multiplikationen, 8 Skalierungen und 29 Additionen**
- Ausführung der 2D 8x8-DCT nach AAN88 als separable Hintereinanderausführung von acht 1D 8-DCTs der Spalten und acht 1D 8-DCTs der Zeilen (der Spaltenkoeffizienten)
- ⇒ **Schnelle 2D 8x8-DCT: 16x5 = 80 Multiplikationen + 16x29 = 464 Additionen**

### Weitere Transformationen / Basisbilder



Haar-Basisbilder



Walsh-Hadamard-Basisbilder

Enthalten nur die Werte {+1,0,-1}; geringer Berechnungsaufwand (keine Multiplikationen)

### Wie findet man die „optimale“ Transformation?

- **Anschaulich:** Drehung in ein Koordinatensystem entlang der **Hauptachsen der Punktwolke** der Verbundverteilungsdichte
- **Mathematisch:** Neue Basisvektoren sind **Eigenvektoren der Kovarianzmatrix** des Eingangssignals
- ⇒ **Karhunen-Loève Transformation (KLT)**
- **Vorteile:**
  - Optimale Dekorrelation
  - Optimale Energiekonzentration
- **Nachteile:**
  - Signalabhängig
  - Keine schnellen Algorithmen
- **Strategie:** Annäherung an KLT durch schnelle Transformation

### Leistungsvergleich von 8x8 Transformationen

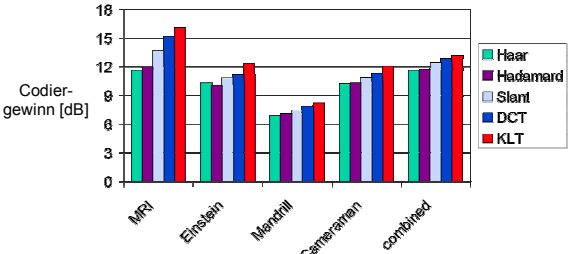


Image	Haar	Hadamard	Slant	DCT	KLT
MRI	12	12	14	15	16
Einstein	10	11	11	12	12
Mandrill	7	7	8	8	8
Cameraman	9	10	11	12	12
combined	11	12	12	13	13

aus: B. Girod, Stanford Univ., Handouts

### Basisfunktionen von DCT und KLT im Vergleich

8x8 DCT Basisblöcke

bildunabhängig!

8x8 KLT Basisblöcke

Für gegebenes Bild optimiert!

Original-bild

7 © Detlev Marpe, Kai Uwe Barthel

### Sukzessive Approximation mit bis zu 8 von 64 Basisfktn

Original

Rekonstruktion mit

1      2      4      8 Koeffizienten (höchster Energie)

8x8 KLT

8x8 DCT

Probleme durch visuelle Artefakte an den Blockgrenzen !

8 © Detlev Marpe, Kai Uwe Barthel

### Einfluss der Blockgrößen auf die 2D DCT

Image	2x2	4x4	8x8	16x16	32x32
MRI	9	13	15	16	17
Einstein	6	10	12	13	14
Mandrill	5	8	9	10	11
Cameraman	7	10	11	12	13
combined	8	11	12	13	14

9 © Detlev Marpe, Kai Uwe Barthel

### „WWW“ der Bild-/Videocodierstandardisierung

- **Warum wird standardisiert?**
  - Standards ermöglichen **Kompatibilität** und **Interoperabilität**
  - Standards bündeln Expertenwissen (**State-of-the-Art Technologie**)
  - Produkte können **vorgefertigte HW-/SW-Implementierungen** der Standards verwenden
- **Was wird standardisiert?**
  - Nur **Syntax** (Form und Struktur) des Bitstroms und **Decoder**
  - Erlaubt implementationspezifische Optimierung im Encoder, z. B. bezüglich Komplexität, Qualität und Bitrate
- **Wer standardisiert?**
  - **ITU-T Telecommunication Standardization**
    - Study Group 16: **Multimedia Services, Systems, and Terminals**
  - **ISO/IEC JTC1 SC29 Coding of Audio, Picture, Multimedia and Hypermedia Information**
    - **WG 1: Digital Compression and Coding of Still Pictures (JBIG, JPEG)**
    - **WG11: Generic Coding of Moving Pictures and Assoc. Audio Information (MPEG)**

10 © Detlev Marpe, Kai Uwe Barthel

### Der JPEG-Standard

#### Encoder (Baseline Modus)

```

    graph LR
        subgraph Codierschicht
            B[Bild] --> BK[Blockteilung der Helligkeits- und Farbkomponenten]
            BK --> ZD[2D-DCT]
            ZD --> S((+))
            S --> LQ[Lineare Quantisierung]
            LQ --> EC[Entropie-codierung]
            EC --> M[Multiplexieren]
            M --> C[Code]
        end
        subgraph Kontrollschicht
            S --> PG[Prädiktion des Gleichanteils]
            PG --> LQ
            PG --> EC
            EC --> QM[Quantisierungsmatrix]
            EC --> VT[VLC-Tabellen]
            EC --> AC[Adaptive Komponenten, Erzeugung der Headerinformation]
            AC --> HI[Headerinformation und Nebeninformation]
            HI --> M
        end
    
```

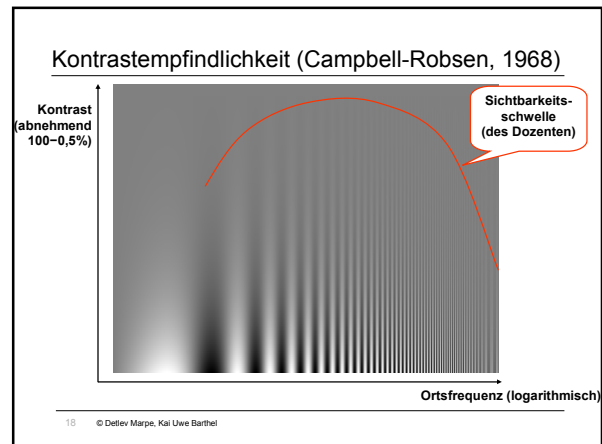
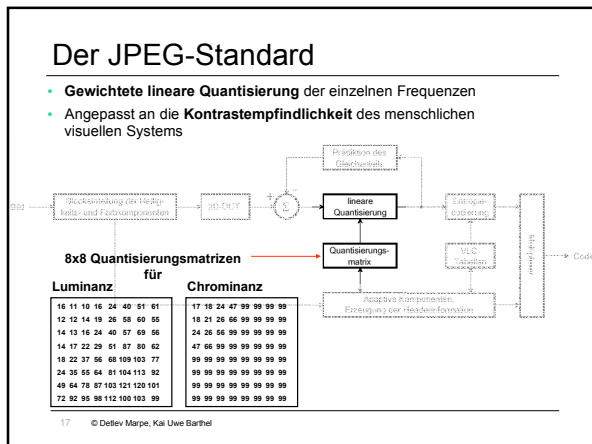
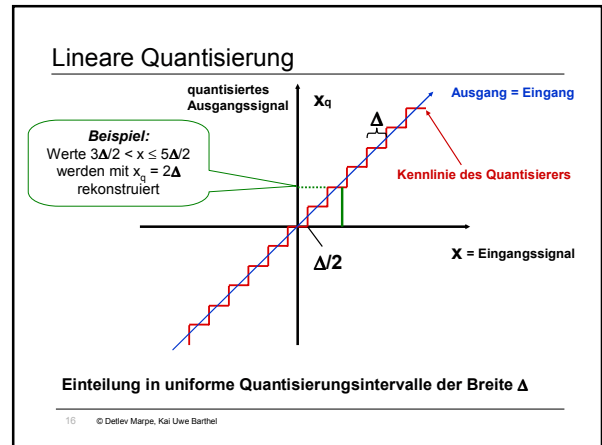
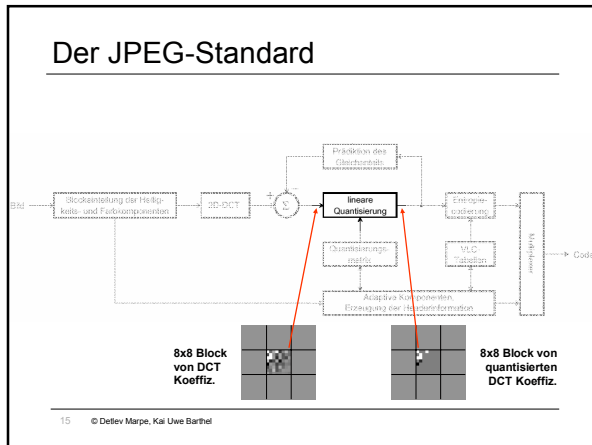
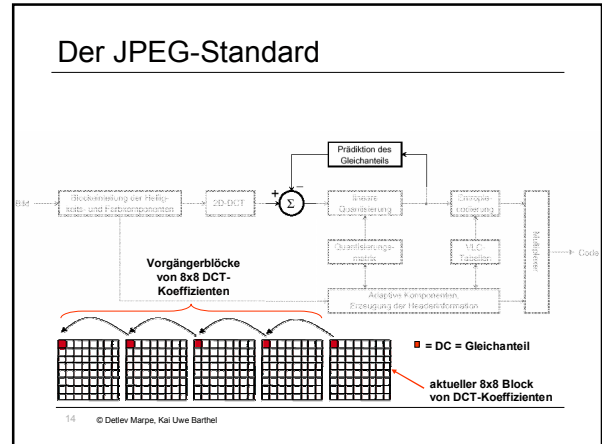
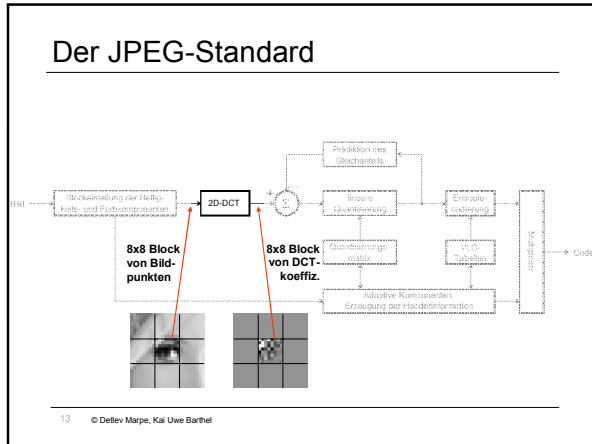
11 © Detlev Marpe, Kai Uwe Barthel

### Der JPEG-Standard

```

    graph LR
        B[Bild] --> BK[Blockteilung der Helligkeits- und Farbkomponenten]
        BK --> ZD[2D-DCT]
        ZD --> S((+))
        S --> LQ[Lineare Quantisierung]
        LQ --> EC[Entropie-codierung]
        EC --> M[Multiplexieren]
        M --> C[Code]
        
        S --> PG[Prädiktion des Gleichanteils]
        PG --> LQ
        PG --> EC
        
        EC --> QM[Quantisierungsmatrix]
        EC --> VT[VLC-Tabellen]
        EC --> AC[Adaptive Komponenten, Erzeugung der Headerinformation]
        AC --> HI[Headerinformation und Nebeninformation]
        HI --> M
    
```

12 © Detlev Marpe, Kai Uwe Barthel



### Der JPEG-Standard

- Zick-Zack Auslesen der quantisierten AC Koeffizienten (2D → 1D)
- Lauflängencodierung der Nullkoeffizienten

The diagram illustrates the JPEG standard process. It starts with 'quantisierte Koeffizienten' (quantized coefficients) in a 2D grid. A 'Zick-Zack Auslesen' (zig-zag reading) path is shown, starting from the top-left and moving in a zig-zag pattern through the grid. Red dots indicate non-zero coefficients, while black dots indicate zero coefficients. The process then moves to 'Entropiecodierung' (entropy coding) using 'VLD-Tabellen' (VLC tables) to generate a 'Bitstrom' (bitstream).

19 © Detlev Marpe, Kai Uwe Barthel

### Entropiecodierung von JPEG (Spielzeugbeispiel)

This example shows the entropy coding process for a 4x4 block. It starts with a 4x4 block of pixel data (110, 112, 109, 111; 106, 105, 107, 107; 98, 97, 97, 95; 92, 90, 94, 91). This is transformed into DCT coefficients (405, 0.3, -0.3, 1.3; 29.7, -0.4, 0.7, -3.2; -0.8, -0.5, -0.3, -0.6; -2.3, 2.2, -0.5, -1.6). These are then quantized using a quantization matrix  $\Delta_i$  (1, 1, 1, 1; 1, 1, 2, 2; 2, 2, 2, 2; 1, 1, 2, 4) to produce a 1D data stream (405, 0, 30, 0, 0, 0, 1, 0, 0, -2, 2, 0, -2, 0, 0, 0). Finally, run-length coding is applied, resulting in a sequence of (Run, Level) pairs: (1,30), (3,1), (2,-2), (0,2), (1,-2), and an End Of Block (EOB) marker.

20 © Detlev Marpe, Kai Uwe Barthel

### Codierung der (Run,Level)-Paare

- Repräsentierung über einen 8-bit Wert **Run-Size** RRRSSSS
- **RRRR**: 4-bit Wert zur Bestimmung der Lauflänge 0 ... 15
- **SSSS**: 4-bit Wert zur Bestimmung der Betragskategorie („Size“) des Level-Werts
- **Huffman-Codierung** von Run-Size (Baseline Modus)
- Vorzeichen und zusätzliche Bits zur Bestimmung des exakten Level-Werts werden angefügt
- Verwendung von vordefinierten Huffman-Tabellen (einstufige Encodierung; default)
- **Alternativ**: Huffman-Tabellen können auf Statistik des gegebenen Bildsignals angepasst (zweistufige Encodierung) und als Nebeninformation übertragen werden

21 © Detlev Marpe, Kai Uwe Barthel

### JPEG Qualitäten (Q: Qualitätsfaktor 1...100)

Originalbild  
768 x 512 Pixel,  
24 bit RGB

22 © Detlev Marpe, Kai Uwe Barthel

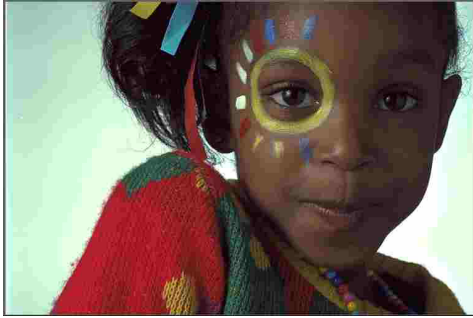
### JPEG Qualitäten (Q=50; 32202 B; CR 36,6:1)

23 © Detlev Marpe, Kai Uwe Barthel

### JPEG Qualitäten (Q=25; 19411 B; CR 60,8:1)

24 © Detlev Marpe, Kai Uwe Barthel

### JPEG Qualitäten (Q=12; 10924 B; CR 108:1)



25 © Detlev Marpe, Kai Uwe Barthel

### JPEG Qualitäten (Q=6; 6468 B; CR 182:1)



26 © Detlev Marpe, Kai Uwe Barthel

### JPEG Modi

- **Baseline**
    - Minimale Toolbox, die jeder JPEG-Decoder unterstützt
    - Sequentielle (blockweise) Verarbeitung
  - **Progressiver Modus** (statt sequentielle Verarbeitung)
    - Spektrale Selektion (von tief- zur hochfrequenten DCT-Koeffizienten)
    - Schrittweise Verfeinerung der Genauigkeit aller DCT-Koeffizienten
- Wenig unterstützt:
- **Arithmetische Codierung** (statt Huffman)
  - **Hierarchischer Modus** (progressiv bzgl. Bildauflösung)
  - **Verlustloser Modus** (DPCM-basiert, Huffman-Codierung)
- **Motion JPEG**
    - kein offizieller **Standard** (viele unterschiedliche Implementierungen)
    - Jedes Frame/Field wird unabhängig voneinander JPEG codiert
    - Verwendung zur **Videonachbearbeitung** / Digitale Kameras

27 © Detlev Marpe, Kai Uwe Barthel

### Literatur- und weitere Informationsquellen

- W. B. Pennebaker and J. Mitchell: **JPEG - Still Image Data Compression Standard**, Van Nostrand Reinhold, 1993.
  - *Die JPEG-Bibel*
- ISO/IEC International Standard 10918-1 bzw. ITU-T Recommendation T.81 (Sept. 1992)
  - <http://www.itu.int/T-REC-T.81/>
- **Hitchhiker's Guide to JPEG compression**: Java-Applet zur Demonstration von Baseline u. Hierarchischem Modus
  - <http://www.sfu.ca/~cjenning/toybox/hjpeg/>
- Offizielle JPEG/JBIG web site:
  - <http://www.jpeg.org/>
- Open-Source Projekt Independent JPEG Group:
  - <http://www.iij.org/>

28 © Detlev Marpe, Kai Uwe Barthel