

JPEG Kompression — technische Realisierung

Christian Gollwitzer

Experimentalphysik V

20. Januar 2005

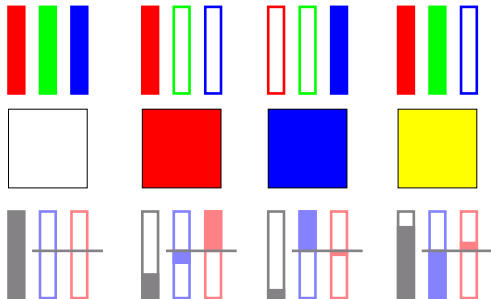
Schema der JPEG Kompression

- ▶ Farbraumkonvertierung RGB \rightarrow YCbCr
- ▶ Subsampling der Farbkomponenten Cb, Cr
- ▶ Zerlegung in Blöcke 8×8
- ▶ 2D Kosinustransformation (DCT)
- ▶ Quantisierung der DCT-Koeffizienten (Verlust!)
- ▶ Verlustfreie Kompression der quantisierten Koeffizienten (RLE, Huffman)

Farbraum YCbCr (CCIR Recommendation 601)

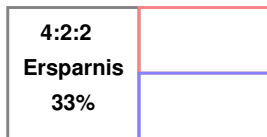
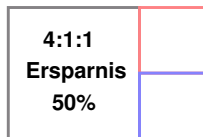
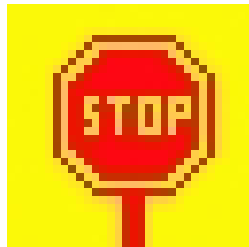
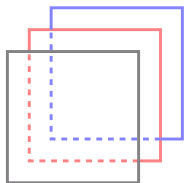
Das Auge ist auf Helligkeit sehr empfindlich. → Repräsentation der Farbe durch Helligkeit (Y) und Farbkomponenten (Cr, Cb)

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.29900 & 0.58700 & 0.11400 \\ -0.16874 & -0.33126 & 0.50000 \\ 0.50000 & -0.41869 & -0.08131 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$



Subsampling der Chroma-Komponenten (optional)

Chroma-Komponenten werden nicht mit voller Auflösung gespeichert → Platzersparnis bis 50%



2D-Kosinustransformation von Blöcken 8×8

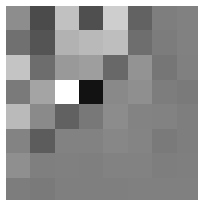
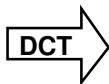
$$F_{uv} = \frac{c_{uv}}{4} \sum_{x=0}^7 \sum_{y=0}^7 f_{xy} \cdot \cos\left(\frac{\pi}{16}(2x+1) \cdot u\right) \cdot \cos\left(\frac{\pi}{16}(2y+1) \cdot v\right)$$

mit

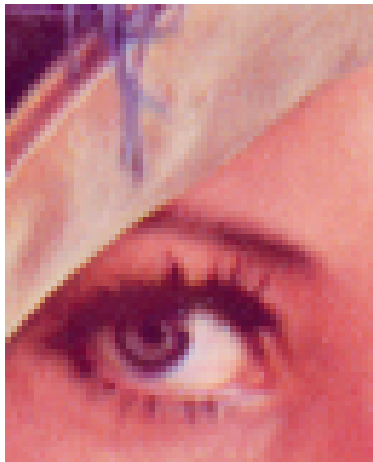
$$c_{uv} = \begin{cases} \frac{1}{2}, & \text{wenn } u = v = 0 \\ \frac{1}{\sqrt{2}}, & \text{wenn entweder } u = 0 \text{ oder } v = 0 \\ 1 & \text{sonst} \end{cases}$$



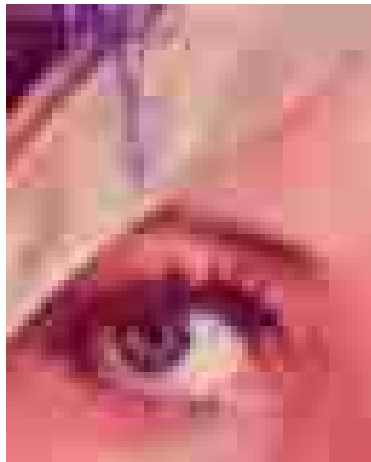
level shift



Beispiel Lena : Blockartefakte



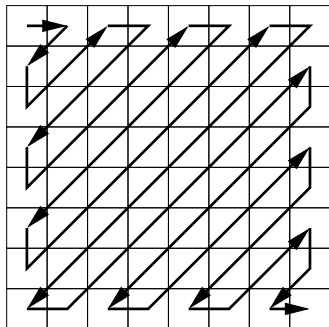
Ausschnitt aus Lena, 64×80



Ausschnitt aus Lena, JPEG
 $q=20$



Quantisierung der Koeffizienten

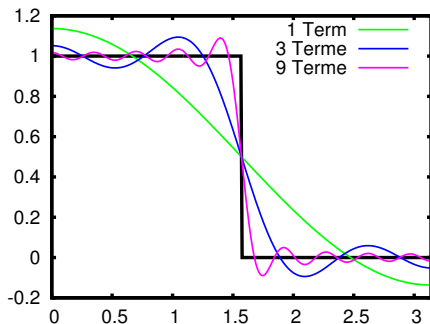


Lineare Quantisierung =
Integer-Division

$$Sq_{uv} = \text{round} \left(\frac{F_{uv}}{q_{uv}} \right)$$

Durch Runden entsteht
Verlust

Folgen der Quantisierung — 1D Beispiel



- ▶ Durch das Quantisieren werden „schwache“ Frequenzanteile unterdrückt.
- ▶ Wellen werden geglättet
- ▶ Sprünge **erzeugen** Wellen

Verlustfreie Kompression der Koeffizienten

- ▶ 0. Fourierkomponente (DC-Komponente, Mittelwert) differenzkodiert:

$$D_0 D_1 D_2 \dots \rightarrow D_0 (D_1 - D_0) (D_2 - D_1) \dots$$

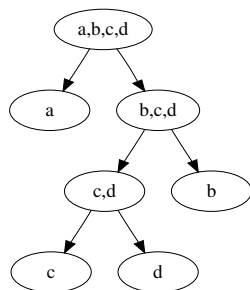
- ▶ „Zero Run-length Encoding“ (RLE) für übrige Komponenten:

$$(0, 0, 0, 3, 0, 20, 10, 0, 0, 50, \dots) \rightarrow (3, 3)(1, 20)(0, 10)(2, 50)(0, 0)$$

- ▶ Spezieller Marker (0, 0) zeigt das Ende der Koeffizienten an
- ▶ Anzahl aufeinanderfolgender Nullen immer ≤ 15
- ▶ Differenzkodierung und RLE-Output wird nach Huffman kodiert (oder arithmetische Kodierung)

Huffman-Verfahren

- ▶ Anzahl Nullen und Bitlänge des Koeffizienten zu 1 Byte (Code) kombiniert
- ▶ Der Code wird Huffman kodiert, die Bits des Koeffizienten werden angehängt










- ▶ Verfolgen der Bits von der Wurzel bis zum Blatt im Huffman-Baum liefert den Code.
- ▶ Häufige Zeichen bekommen kurzen Code, seltene Zeichen langen Code
- ▶ Aufstellen des Huffman-Baums erfolgt durch Zusammenfassen seltener Zeichen

Schema für einen „transform coder“

- ▶ Transformation zur Dekorrelation der Daten (DCT, Wavelet, Singular-Value-Decomposition, Karhunen-Loeve-Transformation, ...)
- ▶ Entfernung „überflüssiger“ Daten = Quantisierung der Koeffizienten (Lineare Quantisierung, logarithmische Quantisierung, Vektorquantisierung, ...)
- ▶ Verlustfreie Kompression der quantisierten Koeffizienten (RLE, Huffman, LZ77, arithmetische Kodierung, Burrows-Wheeler-Transformation, ...)

Literatur

-  ISO/IEC 10918-1:1993 (E) (auch: ITU CCITT Recommendation T.81)
-  <http://www.lenna.org>
-  Cristu Cuturicu, 1999: A note about the JPEG decoding algorithm
-  ISO/IEC 15948:2003 (E) (PNG)
-  <http://pmt.sourceforge.net/pngcrush/>
-  ISO/IEC 15444-1 (JPEG 2000),
<http://www.ece.uvic.ca/~mdadams/jasper/>
-  <http://links.uwaterloo.ca/fractals.home.html>