



# Inhalt

- Geschichte JPG
- Allgemeine Kompression
- Kompressionsarten
- Kompressionsvorgang
- JPG 2000



### Geschichte

- Anfänge in den späten 70ern durch Bedürfnisse des Videotextes
- 1988 – gab es kein Bildformat mit guter Kompression
- Joint Photographic Experts Group (JPEG) zusammengeschlossen, um erstmals einen Kompressionsstandard für Standbilder zu etablieren
- Ziele
  - akzeptable Komplexität
  - Unabhängigkeit von der Bildbeschaffenheit
  - Beeinflussbare Bildqualität/Kompression
  - Keinen universellen Algorithmus, sondern angepasst (verlustfrei...)
- 1992 – erste Entwurf veröffentlicht / publiziert
- 1993 – als ISO 10918-1 veröffentlicht
- für alle verfügbar, große Beliebtheit
- heute Weitverbreitetstes Bildformat, z.B. bei Digitalkameras und Internet

### Allgemeines zur Komprimierung

- Internetstandard
- Anforderungen
  - Hohe Kompressionsrate
  - Hohe Geschwindigkeit beim Kodieren und Dekodieren
  - Geringer Qualitätsverlust für das menschliche Auge

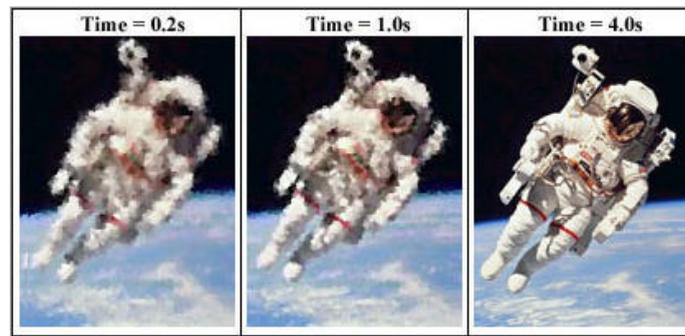
### Kompressionsarten – Modi

- Hierarchischer Modus
  - Abspeichern unterschiedlicher Auflösungen
- Verlustfreier Modus
  - Zwar ohne Qualitätsverlust, aber kaum Komprimierung
- Sequentieller Modus
  - Ein Durchgang (oben links bis unten rechts)
  - Beste Kompressionsrate
  - Leichte Implementierung
- Progressiver Modus
  - Kodieren und Dekodieren in mehreren Durchgängen
  - Pro Durchgang Verbesserung der Bildqualität

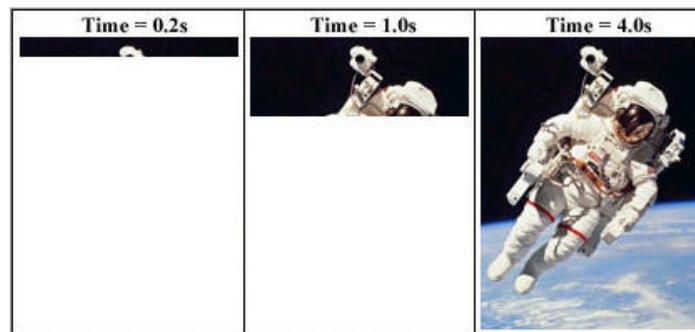


# JPEG

## - Vortrag Medientechnik



*progressive encoding*



*standard encoding (sequential)*

## Der Komprimierungsvorgang

- Farbraumveränderung
- Rasterung
- Diskrete Kosinustransformation
- Quantisierung
- Entropiecodierung

## Farben – das Farbmodell

Schemata werden zur mathematischen Beschreibung von Farben benötigt.  
Grund der Transformation in andere Modelle:

Menschliches Auge nimmt Helligkeitsunterschiede leichter wahr als Farbunterschiede

Vergleich: Mathe (Zahl als Bruch/ Dezimal etc.)

### **Das RGB Modell – additive Farbmischung**

- 3 Grundfarben Rot Grün Blau
- jede andere Farbe lässt sich mischen
- alle zu gleichen Teilen = weiß

### **CMYK – substraktive Farbmischung**

- Cyan, Magenta und Gelb
- Zusammen theoretisch schwarz, praktisch aber nur dunkelbraun  
→ Zugabe von schwarz

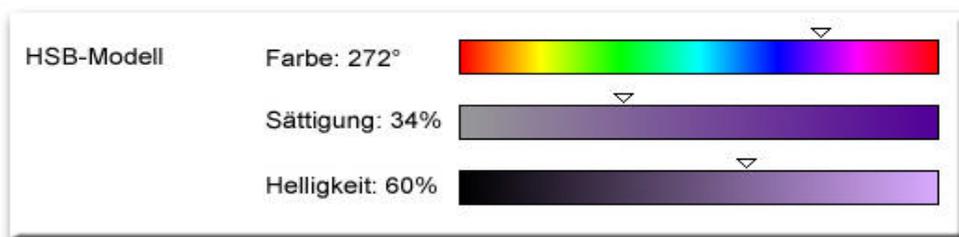


### Das HSB Modell

- Farbton (Hue)
  - Reine Farbinformation
  - Skala bis 360°
- Sättigung (Saturation)
  - Verhältnis von reiner Farbstärke zu unbunten Anteilen
  - 0% meint immer Grauton, 100% kräftigste reine Farbe
- Helligkeit(Brightness)
  - 0% schwarz, 100% weiß



Ausgangsfarbtone, "blasses Dunkel-Lila"



gleiche Farbe dargestellt in unterschiedlichen Farbmodellen

### Beispiel Farbraumveränderung:

- RGB zu YUV
- YUV entspricht in etwa dem HSB Modell
- Y = Luminanz (Helligkeit)
- U = Farbton
- V = Chrominanz (Farbsättigung)

### Rechnung

$$Y = 0,299R + 0,587G + 0,114B$$

$$U = 0,493(B - Y)$$

$$V = 0,877(R - Y)$$

### Subsampling / Downsampling

- Grauinformationen (Luminanz) bleibt erhalten
- 2 oder 4 Pixel jeder Farbkomponente (Ton und Sättigung) werden zu einem Pixel gemittelt
- deutliche Reduktion der Dateigröße



## Komprimierung – Schema

Originalbild



RGB >> YUV

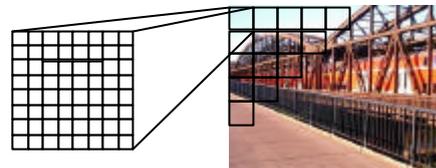
$$Y = 0,299R + 0,587G + 0,114B$$

...

Subsampling

YUV	YUV	YUV	Y
YUV	YUV	Y	Y

Rasterung



Indexverschiebung

$$F_{Index}(u, v) = F(u, v) - 128$$

DCT Transformation

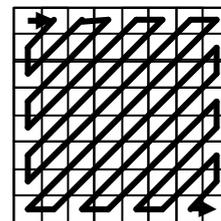
$$F(u, v) = \frac{1}{4} C(u) C(v) \left[ \sum_{x=0}^7 \sum_{y=0}^7 \cos\left(\frac{(2x+1)u\mathbf{p}}{16}\right) * \cos\left(\frac{(2y+1)v\mathbf{p}}{16}\right) \right]$$

$$C(u), C(v) = \begin{cases} 1 & u, v = 0 \\ \frac{1}{\sqrt{2}} & \text{sonst} \end{cases}$$

Quantisierung

$$F_{Quant}(u, v) = \frac{F(u, v)}{Q(u, v)}$$

Entropiecodierung



JPEG Bild





### Erklärungen zur Kompression

#### **Rasterung**

- Einteilung des Bildes in Felder mit 8x8 Pixeln
- auch größere Felder möglich
  - aber Zeitverlust, da 8x8 Pixel optimal
- Anwendung der Komprimierung auf jedes Feld einzeln
  - So genanntes „Blocking Verfahren“

#### **Index Shifting**

- Helligkeitswerte (Y) liegen im Wertebereich 0..255
- Algorithmus benötigt eine Gleichverteilung um 0
  - Aufgrund der Kosinus Funktion
- Verschiebung um 128 Werte nach unten
- neuer Wertebereich: -128..127

#### **DCT Transformation**

- Transformierung der Pixelwerte in den Frequenzbereich
- Vorbereiten der Helligkeitswerte auf Komprimierung
- keine Komprimierung der Daten

#### DC Koeffizient

- direct-current term
- liegt in der DCT Tabelle links oben
- beschreibt durchschnittlichen Grauwert des Bildes

#### AC Koeffizienten

- alternating-current term
- alle anderen Felder in der DCT Tabelle

#### **Quantisierung**

- Voraussetzung zur Reduzierung der Datenmenge
- menschliche Auge reagiert empfindlicher auf niedrige Frequenzen
- entspricht den DCT Werten links oben
- Quantisierung durch Berechnung mit Quantisierungstabelle
  - Q-Tabelle besteht wie DCT-Tabelle aus 8x8 Werten
  - umgekehrte Verteilung der Werte
  - hohe Werte: rechts unten
  - niedrige Werte: links oben
- $DCT\_quant(u,v) = DCT(u,v) / Q(u,v)$
- nach Berechnung
  - Reduzierung des Wertebereiches links oben
  - Zusammenfassung ähnlicher Werte rechts unten
  - viele Werte werden 0
- Q-Tabelle kann frei bestimmt werden



# JPEG

## - Vortrag Medientechnik

---

- Einfluss auf Qualität und Komprimierung der Daten
- muss mit Bild mitgeliefert werden, da sie frei wählbar ist

### **Entropiecodierung**

- Durchführung für Redunanzreduktion
- Verwendung von Algorithmen mit variablen Längen
- getrennte Komprimierung von DC und AC Werten
- aufgrund der starken Abhängigkeit aufeinander folgender Blöcke wird der DC Koeffizient als Differenz zum vorherigen Block codiert

### DC-Werte

- Codierung mit Hilfe der VLC Tabellen
  - Codierung mit Hilfe von 2 Symbolen
    - Symbol 1: | Anzahl Bits für Symbol 2 |
    - Symbol 2: | DC Wert oder Differenz |
- Beispiel  
DC = 13  
VLC = 110 (für die Länge 4 Bits)  
DC\_bin = 1101  
komplette Bitfolge: | 110 | 1101 | = 1101101

### AC-Werte

- Auslesen der Werte im Zick-Zack Verfahren
- Aneinanderreihung von gleichen Werten (speziell 0) sehr wahrscheinlich
- Codierung von einer Nullfolge gefolgt von einem AC Wert ungleich 0
- Codierung im RUN-LEVEL Verfahren
  - RUN = Anzahl der Nullen
  - LEVEL = AC Wert (ungleich 0)
- für häufig auftretende Werte gibt es eine Standard Tabelle
  - modifizierter Huffman Code
- Werte außerhalb der Tabelle können definiert werden
  - Zusammenstellung der Bitfolgen durch den Escape Code gefolgt von Bitfolgen für die RUN und LEVEL Werte
- Übertragung ohne Trennbits
  - Ende einer Bitfolge kann kein Anfang einer Bitfolge sein



# JPEG

## - Vortrag Medientechnik

Standard Huffmann Tabelle

RUN	LEVEL	Bitfolge
0	1	1, erster AC
0	1	11, sonst
0	2	0100
0	3	00101
0	4	0000110
0	5	00100110
...	...	...
1	1	011
1	2	000110
1	3	00100101
1	4	0000001100
1	5	000000011011
...	...	...
EOB		10
ESCAPE		000001

RUN Codierung

RUN	Bitfolge
0	000000
1	000001
2	000010
...	...
62	111110
63	111111

LEVEL Codierung

LEVEL	Bitfolge
-255	10000000 00000001
...	...
-128	10000000 10000000
-127	10000001
...	...
-1	11111111
1	00000001
...	...
127	01111111
128	00000000 10000000
...	...
255	00000000 11111111

VLC Wertebereichstabelle

Bits	Wertebereich	
0	0	
1	-1	1
2	-3,-2	2,3
3	-7,-4	4..7
4	-15..8	8..15
5	-31..16	16..31
6	-63..32	32..63
7	-127..-64	64..127
8	-255..-128	128..255

VLC Bit Tabelle

VLC	angehängte Bits
100	0
00	1
01	2
101	3
110	4
1110	5
11110	6
111110	7
1111110	8



### JPEG 2000

- JPG hat 2 Nachteile: keine zusätzlichen Informationen (Metatags) & bei hoher Kompression Artefaktbildung (Rechtecke)
- -> 1996 Arbeitsbeginn an JPG 2000
- 1999 erste Entwurf fertig
- 2000/2001 ISO Standard
- wird noch weiter entwickelt, bei modernen Programmen (z.B. PhotoImpact 7) schon integriert, auch Plug-ins für IE, Netscape oder Adobe Photoshop erhältlich

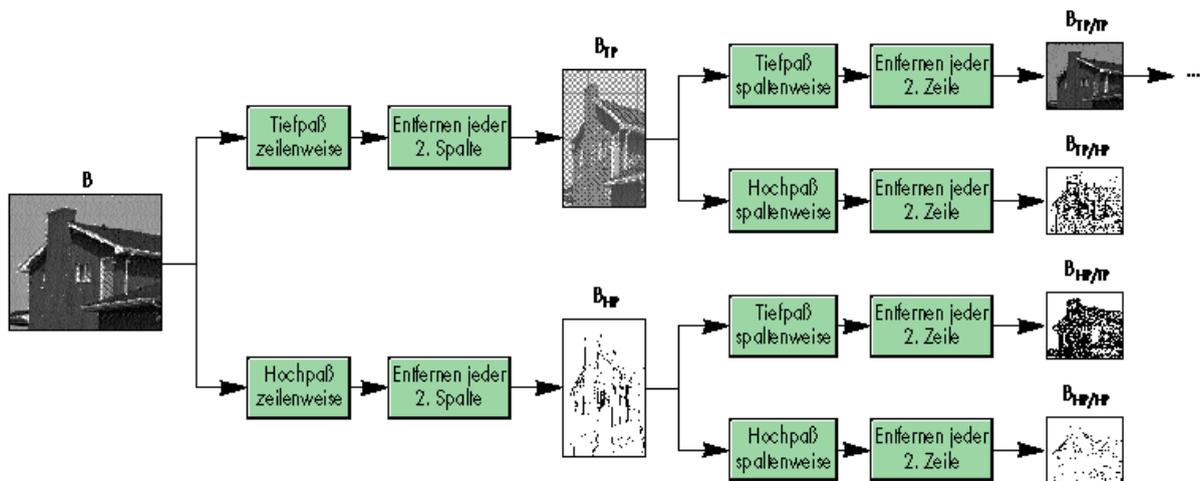
### **Funktionsweise**

- diskrete Kosinus Transformation durch Wavelettransformation (verschiedene Wellen) ersetzt
- nicht mit 8x8 Blöcken sondern fortlaufend kodiert, guckt wie schnell sich Bildgegebenheiten ändern
- Bild wird in Waveletkoeffizienten zerlegt, so dass Schritt für Schritt immer größere Bildstrukturen herausgefiltert werden
- Koeffizienten bestehen aus Hoch- und Tiefpass gefilterten Versionen des Bildes & sind im hohen Maße redundant (zerlegt das Bild in zwei Frequenzbereiche)
- Jeder Filterdurchlauf erzeugt ein geglättetes Bild mit niedrigen Frequenzanteilen sowie ein komplementäres mit hohen Frequenzen
- Beide Hälften sind so beschaffen, dass sich bei beiden Bildern jeweils jede zweite Pixelspalte entfernen lässt
- Nochmals angewendet jedoch entlang der Vertikalen -> 4 Bilder (1 Bild entspricht dem Durchschnittssignal, die anderen drei sind Detailsignale)
- Das Durchschnittssignal-Bild lässt sich genauso weiter transformieren und reduzieren - theoretisch so lange, bis es nur noch ein Pixel groß ist, aber nur bei Bildern bei denen die Kantenlänge eine Zweierpotenz ist
- Nach dieser Wavelettransformation in Vorwärtsrichtung liegen die Teilbilder in Form einer Matrix von Koeffizienten vor
- Die eigentliche Kompression erfolgt dann wie bei JPEG in der Quantisierungsschritt



# JPEG

## - Vortrag Medientechnik



### Vorteile

- spart min. 20% Speicherplatz gegenüber JPG, Größe bis zu 0.1 Bit pro Pixel und darunter
- Metatags integrierbar (Autor, Quelle ...)
- Bild lässt sich in verschiedene Größen gleichzeitig speichern & übertragen
- Einzelne Bildbereiche können mit unterschiedlicher Qualität gespeichert werden
- Keine Artefakte und Geisterschatten mehr
- Passwort, Wasserzeichen und Verschlüsselung möglich

### Nachteile

- Bei hoher Kompression allgemeine Unschärfe, bei Fotos aber zu vernachlässigen
- JPEG 2000 nicht zu JPG kompatibel
- Zwei bis dreimal rechenintensiver als JPG