


## 5. Licht, Farbe und Bilder

- 5.1 Licht und Farbe: Physikalische und physiologische Aspekte
- 5.2 Farbmodelle
- 5.3 Raster-Bilddatenformate
- 5.4 Verlustbehaftete Kompression bei Bildern   
JPEG

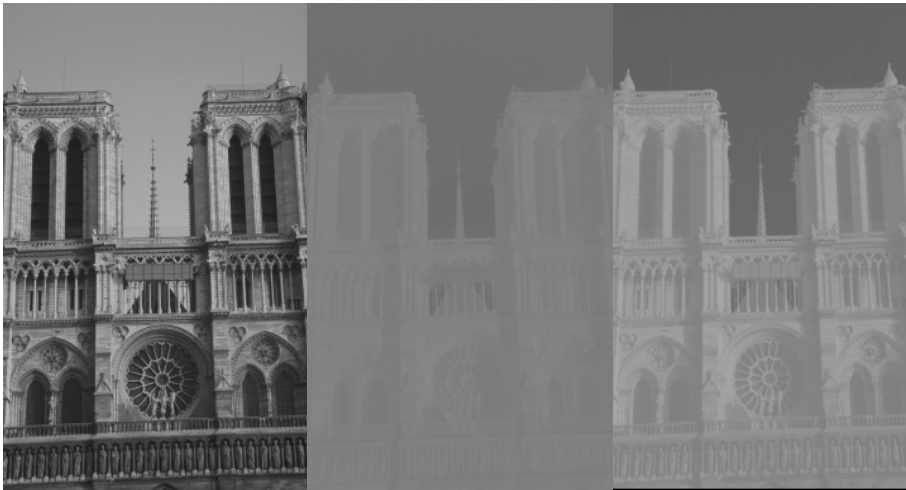
Weiterführende Literatur:

John Miano: Compressed Image File Formats - JPEG, PNG, GIF, XBM, BMP,  
Addison-Wesley 1999

## Warum und wann verlustbehaftet komprimieren?

- Durch Aufnahme aus der realen Welt erzeugte Bilder (v.a. Fotos) sind sehr groß
  - z.B. 4 Mio. Pixel mit je 24 bit = 12 MByte
- Das menschliche Auge wertet nicht alle Informationen des Bildes gleich gut aus
  - z.B. Helligkeit vs. Farbigkeit
  - z.B. Feinabstufungen von Verläufen
- Mit verlustbehafteten Kompressionsverfahren wird
  - ein oft sehr hoher Gewinn an Speicherplatz erzielt
  - der subjektive Eindruck des Bildes kaum verändert
- Bekanntestes Verfahren: JPEG
- Achtung: Für Archivierung von hochwertigen Bild-Originalen eignet sich JPEG nur bedingt (bei Einstellung von geringen Kompressionsgraden)
  - Alternativen z.B.: TIFF, PNG

## Luma- und Chromainformation: Vergleich

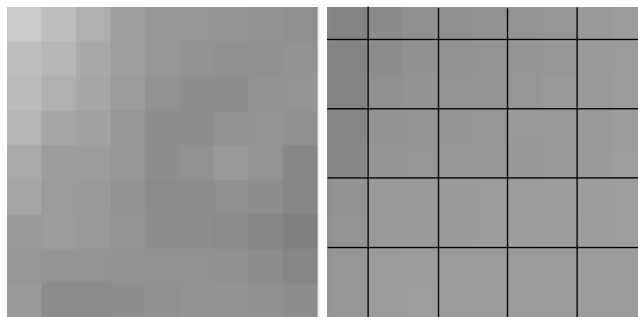


Helligkeit (L-Kanal)

Rot/Grün-Differenz  
(a-Kanal)

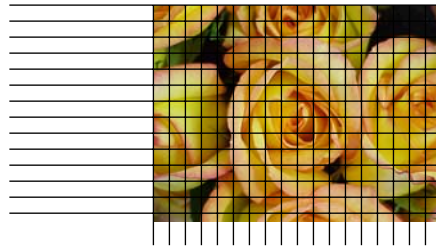
Blau/Gelb-Differenz  
(b-Kanal)

## Chroma-Subsampling



- In vielen Fällen genügt eine geringere Auflösung für die Farbinformation (Chroma) als für die Helligkeit (Luma).
  - Passende Farbmodelle: YUV, YIQ, Lab
  - Teilweise aber abhängig vom Darstellungsinhalt
- Chroma-Subsampling = niedrigere Abtastrate für Farbinformation
  - Speicherplatzersparnis im Beispiel 50% (bei gleichem Subsampling für b)

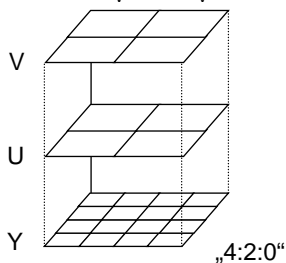
# Abtastraten für Bilder



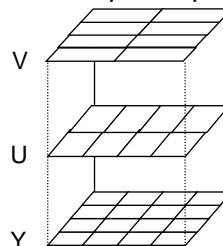
- Abtastrate: Wieviele Pixel pro Längeneinheit des Bildes?
- Mehrdimensionalität:
  - Horizontale Abtastrate ( $H$ )
  - Vertikale Abtastrate ( $V$ )
- Bei Sub-Sampling:
  - Verschiedene Abtastraten für verschiedene Komponenten des Bildes (Farben, evtl. Alphakanal)

## Subsampling

Y:  $H_Y = 4, V_Y = 4$   
U:  $H_U = 2, V_U = 2$   
V:  $H_V = 2, V_V = 2$



Y:  $H_Y = 4, V_Y = 4$   
U:  $H_U = 4, V_U = 2$   
V:  $H_V = 2, V_V = 2$



- $H$  und  $V$ : Zahl der berücksichtigten Pixel je 4x4-Block (*subsampling rate*)
  - horizontal und vertikal
- Subsampling wird bei verschiedenen digitalen Bildverarbeitungstechniken benutzt
  - in JPEG (optional)
  - auch in diversen digitalen Video-Aufzeichnungs-Standards



## Notation für Subsampling

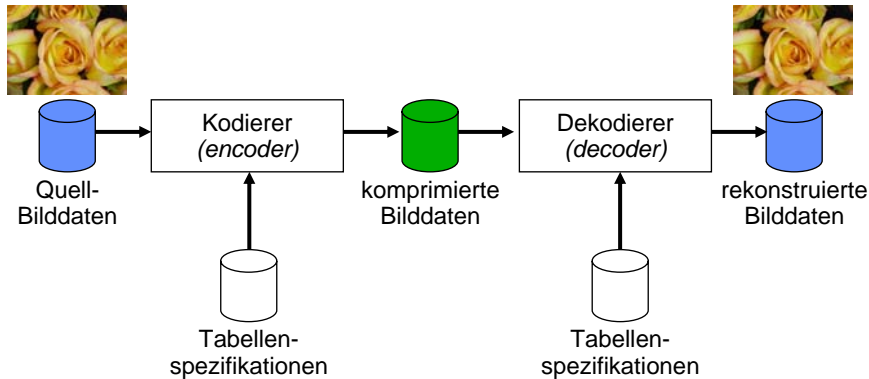
- Übliche Notation für Subsampling von Farben:  
x:y:z
- Ursprüngliche Bedeutung: Horizontales Frequenzverhältnis für Luma (x) zu den beiden Chroma-Kanälen (y, z)
  - Vertikales Subsampling?
  - Beide Chroma-Kanäle haben in der Praxis immer die gleiche Abtastrate !
- Heutige Bedeutung:
  - x: Immer „4“, historische Reminiszenz
  - y: Faktor für das Subsampling von Chroma gegenüber Luma
  - z: Falls z=y: kein vertikales Subsampling  
Falls z=0: vertikales Subsampling von 2:1 für beide Chroma-Kanäle
- Beispiele :
  - 4:2:0  $H_V=4, V_V=4, H_U=2, V_U=2, H_V=2, V_V=2$   
(bei JPEG weit verbreitet)
  - 4:2:2  $H_V=4, V_V=4, H_U=2, V_U=4, H_V=2, V_V=4$



## JPEG: Hintergrund

- JPEG = „Joint Photographics Expert Group“
  - „Joint“ wegen Zusammenarbeit von Arbeitsgruppen zweier Organisationen (ISO und CCITT/ITU)
  - Arbeit seit 1982, Vergleichsverfahren 1987, Auswahl einer „adaptiven Transformationskodierung basierend auf Diskreter Cosinus-Transformation (DCT)“
  - 1992: ITU-T Recommendation T.81 + Internationaler Standard ISO 10918-1
- Wichtige Eigenschaften/Anforderungen:
  - Unabhängigkeit von Bildgröße, Seitenverhältnis, Farbraum, Farbvielfalt
  - Anwendbar auf jedes digitale Standbild mit Farben oder Grautönen
  - Sehr hohe Kompressionsrate
  - Parametrisierbar in Qualität/Kompression
  - Realisierbar durch Software und Spezial-Hardware: gute Komplexität
  - Sequentielle und progressive Dekodierung
  - Unterstützung von verlustfreier Kompression und hierarchischer Verfeinerung der Bildqualität

# JPEG-Architekturmodell



## JPEG-Modi

- Charakteristika:
  - Verlustbehaftet oder verlustfrei
  - sequentiell, progressiv oder hierarchisch
  - Abtasttiefe (für bis zu 4 Komponenten)
  - (Entropie-)Kompressionsverfahren: Huffman- oder arithmetische Kodierung
- **Basismodus (*baseline process*):**
  - Verlustbehaftet (DCT), 8 bit Tiefe, sequentiell, Huffman-Kodierung
- **Erweiterter Modus (*extended process*):**
  - Verlustbehaftet (DCT), 8 oder 12 bit Tiefe, sequentiell oder progressiv, Huffman-Kodierung oder arithmetische Kodierung, mehr Tabellen
- **Verlustfreier Modus (*lossless process*):**
  - Verlustfrei (kein DCT), 2 – 16 bit Tiefe, sequentiell, Huffman-Kodierung oder arithmetische Kodierung
- **Hierarchischer Modus (*hierarchical process*):**
  - Baut auf erweitertem oder verlustfreiem Modus auf, Mehrfach-Frames

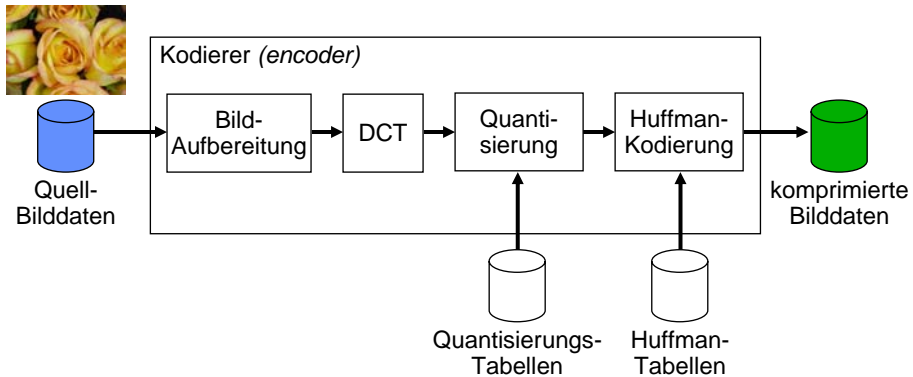
meist verwendet

selten verwendet

ungebräuchlich

## Schritte der JPEG-Kodierung

- Hier nur die gebräuchlichste Variante:  
verlustbehaftet, sequentiell, 8-bit-Daten, Huffman-Kodierung



DCT = *Discrete Cosinus Transformation*



## JPEG-Kodierung: Bildaufbereitung (1)

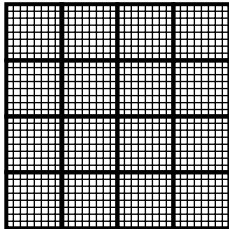
- Bild wird generell in 8 x 8-Pixel-Blöcke (*data units*) eingeteilt
  - Am Rand wird „aufgefüllt“
- Bild kann theoretisch aus bis zu 255 Komponenten (*components*) bestehen
  - Verbreitet: 3 oder 4, nach Farbmodell
- Verzahnte (*interleaved*) oder nicht-verzahnte Reihenfolge:
  - Ablage der Komponenten nacheinander nicht ideal:
    - » Z.B. könnten 3 Farbkomponenten *nacheinander* erscheinen
    - » *Pipelining* in der Verarbeitung erfordert vollständige Information über einen Bildanteil
  - Verzahnte Ablage: Einheiten, die je mindestens eine *data unit* jeder Komponente enthalten: *Minimum Coded Units (MCU)*
  - Maximal vier Komponenten können verzahnt werden

## JPEG-Kodierung: Bildaufbereitung (2)

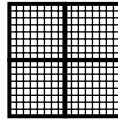
Subsampling

- Interleaving bei gleichzeitigem Chroma-Subsampling:
  - Jede Komponente eingeteilt in *Regionen* aus  $H_c \times V_c$  Data Units ( $H_c$  und  $V_c$  Subsampling-Raten der Komponente  $c$ )
  - Jede Komponente von links oben nach rechts unten zeilenweise gespeichert
  - MCUs enthalten Data Units aus allen Komponenten anteilig

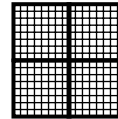
Subsampling für Y:  $H_Y = 4, V_Y = 4$ , für U:  $H_U = 2, V_U = 2$ , für V:  $H_V = 2, V_V = 2$



Y



U

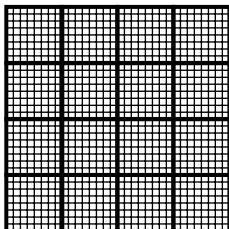


V

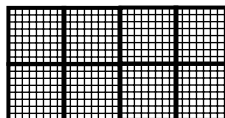
## JPEG-Kodierung: Bildaufbereitung (3)

Subsampling

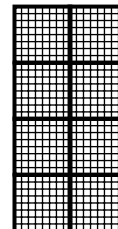
Subsampling für Y:  $H_Y = 4, V_Y = 4$ , für U:  $H_U = 4, V_U = 2$ , für V:  $H_V = 2, V_V = 4$



Y



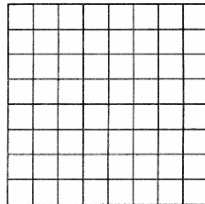
U



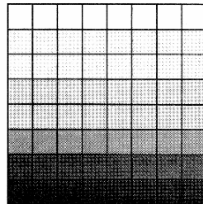
V

# Ortsfrequenz

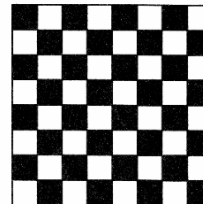
- Ortsfrequenz (oder: räumliche Frequenz, *spatial frequency*)
  - Häufigkeit der Wiederholung einer im Bild erkennbaren Eigenschaft über die räumliche Ausdehnung
  - Maßeinheit: 1/Längeneinheit
  - z.B. Dichte von Linien auf Papier: Anzahl Striche pro cm
- Meist: Anzahl von Helligkeitsschwankungen pro Längeneinheit
- 2-dimensionale Frequenz (horizontal und vertikal)



Ortsfrequenz 0



Ortsfrequenz  
0 horizontal,  
niedrig vertikal



Ortsfrequenz  
hoch  
horizontal und vertikal

# Diskrete Cosinus-Transformation (DCT)

- Grundmotivation: JPEG-Schritte
  - Menschliche Sehvernehmung sehr empfindlich für niedrige und mittlere Frequenzen (Flächen, deutliche Kanten), wenig empfindlich für hohe Frequenzen (z.B. feine Detaillinien)
  - Deshalb Zerlegung der Bildinformation in Frequenzanteile (ähnlich zu Fourier-Transformation)
- Prinzip von DCT:
  - (in einer oder zwei Dimensionen...)



Datenpunkte und Koeffizienten sind bei JPEG jeweils 8 x 8 - Integer - Blöcke



## (Forward) DCT: Mathematische Definition

$$F(u,v) = \frac{1}{4} c_u c_v \sum_{x=0}^7 \sum_{y=0}^7 f(x,y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

wobei

$x, y$  Koordinaten für die Datenpunkte einer Quell-Dateneinheit  
( $x, y = 0, \dots, 7$ )

$u, v$  Koordinaten für die Ziel-Koeffizienten ( $u, v = 0, \dots, 7$ )

$f(x,y)$  Datenwert (Sample)

$F(u,v)$  Koeffizientenwert

$$c_u, c_v = \frac{1}{\sqrt{2}} \quad \text{falls } u, v = 0$$

$$c_u, c_v = 1 \quad \text{sonst}$$

- Die Berechnung der Formel läßt sich auf eine einfache Matrixmultiplikation mit konstanten Matrixeinträgen reduzieren.
- Aus technischen Gründen Sample-Wertebereich zuerst in  $(-128, +127)$  verschoben

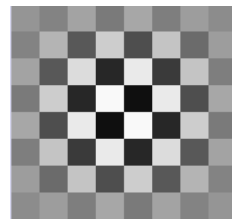
## Beispiele für DCT-Transformation



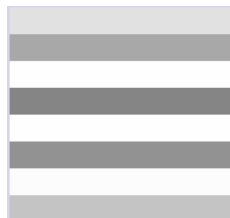
$F(0,1) = 500$ ,  
alle anderen  $F(u, v) = 0$



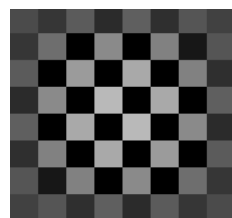
$F(7,0) = 500$ ,  
alle anderen  $F(u, v) = 0$



$F(7,7) = 500$ ,  
alle anderen  $F(u, v) = 0$



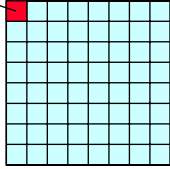
$F(7,0) = 500$ ,  $F(0,0) = 600$   
alle anderen  $F(u, v) = 0$



$F(7,7) = 500$ ,  $F(0,0) = -600$   
alle anderen  $F(u, v) = 0$

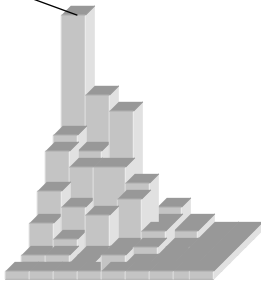
## Interpretation der DCT-Koeffizienten

DC-Koeffizient



Alle anderen:  
AC-  
Koeffizienten

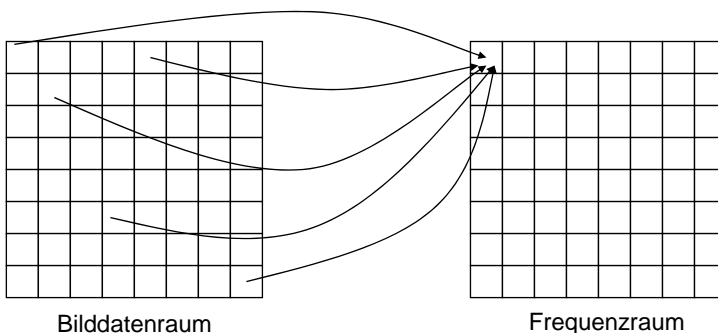
DC-Koeffizient  $F(0,0)$



- Der DC-Koeffizient gibt den Grundton des beschriebenen Bereichs (8x8) im Bild an (in der aktuellen Komponente)
- Die AC-Koeffizienten geben mit aufsteigenden Indizes den Anteil „höherer Frequenzen“ an, d.h. die Zahl der (vertikalen bzw. horizontalen) Streifen
- Z.B.:
  - $F(7,0)$  gibt an, zu welchem Anteil extrem dichte waagrechte Streifen vorkommen;
  - $F(0,7)$  gibt an, zu welchem Anteil extrem dichte senkrechte Streifen vorkommen

DC = Gleichstrom  
AC = Wechselstrom

## DCT: Zusammenhang Datenraum - Frequenzraum



- Ein Punkt im Frequenzraum fasst die Informationen aus dem aktuell betrachteten Bilddatenraum (8x8 Pixel) zusammen.
- Kanten erscheinen als Anteile hoher Frequenzen; bei Flächen sind die hohen Frequenzen fast Null
  - Gute Voraussetzung für spätere Kompression der Null-nahen Werte durch Entropiekodierung

## Inverse DCT: Mathematische Definition

$$f(x, y) = \frac{1}{4} \sum_{x=0}^7 \sum_{y=0}^7 c_u c_v F(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

wobei

$x, y$  Koordinaten für die Datenpunkte einer Quell-Dateneinheit  
( $x, y = 0, \dots, 7$ )

$u, v$  Koordinaten für die Ziel-Koeffizienten ( $u, v = 0, \dots, 7$ )

$f(x, y)$  Datenwert (Sample)

$F(x, y)$  Koeffizientenwert

$$c_u, c_v = \frac{1}{\sqrt{2}} \quad \text{falls } u, v = 0$$

$$c_u, c_v = 1 \quad \text{sonst}$$

- Die Berechnung ist fast identisch mit der Vorwärts-Transformation.
- Mathematisch gesehen, ist der Prozeß verlustfrei!
  - Verluste entstehen aber durch Rundungsfehler

## JPEG-Kodierung: Quantisierung

- Entscheidender Schritt zum **Informationsverlust** und damit zur starken Kompression!

JPEG-Schritte

- Runden der Koeffizienten erzeugt viele Null-Werte und ähnliche Werte
- Damit besser mit nachfolgenden verlustfreien Verfahren komprimierbar

- Quantisierungstabelle:

- Enthält 64 vorgegebene und konstante Bewertungs-Koeffizienten  $Q(u, v)$
- Bedeutung: Bewertung der einzelnen Frequenzanteile des Bildes
- Größere Tabelleneinträge bedeuten stärkere Vergrößerung
- Konkrete Tabellen nicht Bestandteil des Standards (nur zwei Beispiele)
  - » Typisch: Verschiedene Bewertung für hohe und niedrige Frequenzen
- Benutzte Quantisierungstabellen werden als Bestandteil der komprimierten Daten abgelegt und bei Dekompression benutzt

- Berechnung:

- Division Frequenz-Koeffizient / Bewertungskoeffizient und Rundung

$$F'(u, v) = \text{Round} \frac{F(u, v)}{Q(u, v)}$$

Typische  
Tabelle

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

## Informationsverlust durch Quantisierung



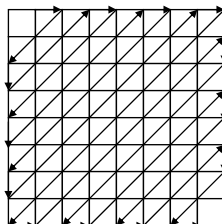
Bei JPEG-Kompressions-Algorithmen ist der Grad der Quantisierung wählbar: „Trade-Off“ zwischen Speicherplatzersparnis und Bildverfälschung (Artefakten)



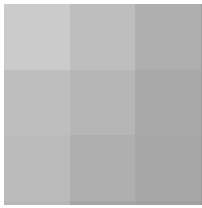
Artefakte treten bei Kanten und Details auf, kaum bei Flächen

## Vorbereitung zur Weiterverarbeitung

- Quantisierte Frequenzwerte:
  - werden in linearer Reihenfolge ausgegeben
  - unterschiedliche Behandlung DC- und AC-Koeffizienten
- DC-Koeffizienten:
  - Benachbarte Dateneinheiten haben oft ähnlichen Grundton
  - Deshalb separat extrahiert (alle DC-Koeffizienten des Bildes in ein „Grob bild“)
- AC-Koeffizienten:
  - Ausgabe nach absteigender Frequenz („Zick-Zack“)



## Prädiktive Codierung: Grundidee



45	48	60
50	55	65
52	60	70

-10	-7	+5
-5	55	+10
-3	+5	+15

- Feste Werte nur für einige Basispixel definieren
- Für andere Pixel nur Differenz speichern
- Potentielle Vorteile:
  - an sehr vielen Stellen kleinere Bitbreite für Differenzwerte möglich
  - Wiederholung gleicher Differenzwerte ermöglicht effektive Kompression mit verlustfreien Verfahren
- Verlustfrei, in JPEG kombiniert mit weiteren (verlustbehafteten) Techniken

## JPEG-Kodierung: Entropie-Kompression

- Vorletzter Schritt: „Statistische Modellierung“
  - » DC-Koeffizienten: Prädiktive Codierung (*Differenzen*)
  - » AC-Koeffizienten: im Wesentlichen Lauflängen-Codierung
- Letzter Schritt: Entropie-Kodierung
  - Wahl zwischen Huffman-Algorithmus und arithmetischer Kompression
  - Getrennt für DC- und AC-Koeffizienten
- Woher kommen die Häufigkeitsverteilungen?
  - Zwei Beispielverteilungen im JPEG-Standard beschrieben
  - Alternative: Durch zusätzlichen Durchlauf über die Daten errechnen

JPEG-Schritte

## JPEG Datenströme

- Ausgabe der JPEG-Kompression
  - Besteht aus Komponenten
  - Getrennt durch *marker* (2 Bytes, erstes Byte xFF)
- Beispiele für Marker:
  - Start of image (SOI)
  - End of image (EOI)
  - Start of frame, baseline (SOF<sub>0</sub>)
  - Start of frame, extended sequential (SOF<sub>1</sub>)
  - Start of frame, progressive (SOF<sub>2</sub>)
  - Start of frame, lossless (SOF<sub>3</sub>)
  - Define Huffman table (DHT)
  - Define quantization tables (DQT)
  - Define restart interval (DRI)
  - Application specific (APP<sub>0</sub> – APP<sub>15</sub>)
  - ...

## JFIF Dateiformat

- Der JPEG-Standard definiert das Dateiformat nicht im Detail.
- De-Facto-Standard: JFIF (JPEG File Interchange Format)
  - inoffiziell (David Hamilton)!
- Neuer offizieller Standard: SPIFF (Still Picture Interchange File Format)
  - von der JPEG
  - spät eingeführt, kompatibel mit JFIF, aber wesentlich flexibler
- JFIF definiert:
  - „Signatur“ zur Identifikation von JPEG-Dateien („JFXX“)
  - Farbraum
  - Pixeldichte
  - Vorschaubilder („Thumbnails“)
  - Zusammenhang Pixel – Abtastfrequenz