

# 6. Licht, Farbe und Bilder

- 6.1 Licht und Farbe: Physikalische und physiologische Aspekte
- 6.2 Farbmodelle
- 6.3 Raster-Bilddatenformate 
  - Grundbegriffe für Bildspeicherung und -Bearbeitung
  - Verlustfrei komprimierende Formate 
- 6.4 Verlustbehaftete Kompression bei Bildern
- 6.5 Weiterentwicklungen bei der Bildkompression

Literatur:

John Miano: Compressed Image File Formats, Addison-Wesley 1999

# Beispiel Bitmap-Format: Tagged Image File Format TIFF

- Entwickelt ca. 1980 von Aldus  
(Firma Aldus inzwischen von Adobe übernommen)
  - Portabilität, Hardwareunabhängigkeit, Flexibilität
- Unterstützt ca. 80 verschiedene Varianten zur Datenspeicherung und deren Kombination
  - z.B. schwach aufgelöstes „Preview“-Bild und hochaufgelöstes Bild
  - Farbmodell explizit angegeben
- Kann Metainformation (z.B. über Ursprungshardware) speichern
- Kompression möglich, aber nicht vorgeschrieben
- Grundstruktur:
  - Header
  - Liste von *Image File Directories*
    - » Image File Directory: Liste von Tags  
(jeweils pro Tag: Typ, Datentyp, Länge, Zeiger auf Daten)
  - Datenbereich

# Windows BMP-Format

- Standardformat aus Microsoft DOS und Windows
- Rasterformat mit zulässigen Farbtiefen 1, 4, 8 und 24 bit
- Verwendet eine Farbpalette (*color table*) (bei niedrigeren Farbtiefen als 24 bit)
- Besteht aus:
  - Kopfinformation
  - Farbtabelle
  - Daten
- Datenablage zeilenweise
- 4- und 8-bit-Variante unterstützen Lauflängen-Kompression:
  - RLE4 und RLE8
  - Zwei Bytes (RLE8) bzw. Halbbytes (RLE4) als Einheit:
    - » Erstes Byte: Anzahl der beschriebenen Pixel
    - » Zweites Byte: Index in Farbtabelle für diese Pixel
- Spezielle Variante mit Alphakanal: „BMP4“

# Beispiel zu RLE in Windows BMP

- RLE8: Zwei Bytes
  - Erstes Byte: Wiederholungszähler
  - Zweites Byte: Zu wiederholender (Pixel-)Wert
- "Fluchtsymbol" (*escape code*): Wiederholungszähler mit Wert 0
  - Gefolgt von 0: Zeilenvorschub
  - Gefolgt von 1: Bildende

- Beispiel:

04 15 00 00 02 11 02 03 00 01 (Hexadezimal)

bedeutet:

15 15 15 15  
11 11 03 03  
Bildende

# GIF-Format: Allgemeines

- GIF = Graphics Interchange Format
  - eingeführt von CompuServe 1987 („GIF87a“)
  - Heute verwendete Version von 1989 („GIF89a“) mit kleinen Modifikationen
- Verlustfreie Kompression (mit LZW)
- Kleiner Farbumfang (max. 256 Farben in einem Bild)
- Flexible Anzeigeeoptionen (z.B. *interlaced* und Animation)
- Optimal für kleinere Grafiken und Gestaltungselemente
- Wenig geeignet für hoch auflösende Bilder (z.B. Fotos)
- Patent-Streit:
  - Unisys hat Patent auf den verwendeten LZW-Algorithmus
  - 1999: Ankündigung von Lizenzforderungen für GIF-Grafiken
  - Initiativen zum Ersatz von GIF (z.B. durch PNG)
- Im folgenden: Beispielhafte Konzepte aus GIF

# Color Table in GIF

- Eine GIF-Datei kann mehrere Bilder enthalten.
- Farbtabelle (Paletten)
  - entweder global für alle enthaltenen Bilder (Global Color Table)
  - oder lokal je Bild
- Lokale Farbtabelle hat Vorrang vor globaler Tabelle
- Hintergrundfarbe für Gesamtdarstellung möglich, wenn globale Farbtabelle existiert
- Sortierung der globalen Farbtabelle:
  - Reihenfolge der Farben in globaler Farbtabelle nach Häufigkeit sortiert



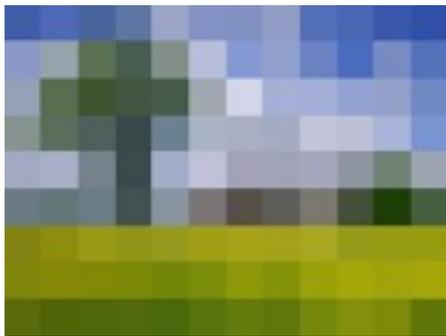
# Transparenzfarbe in GIF

- In GIF (89) kann eine Farbe der Tabelle als „transparent“ gekennzeichnet werden.
  - Pixel dieser Farbe werden nicht angezeigt, statt dessen Hintergrund
  - Das ist keine echte Transparenz im Sinne eines Alphakanals!



# Interlacing in GIF

- Ziel: Kürzere empfundene Ladezeit für Betrachter, z.B. bei Web-Grafik
- Bild wird schrittweise in Zeilen aufgebaut
  - 1. Durchlauf: Jede 8. Zeile beginnend in Zeile 0
  - 2. Durchlauf: Jede 8. Zeile beginnend in Zeile 4
  - 3. Durchlauf: Jede 4. Zeile beginnend in Zeile 2
  - 4. Durchlauf: Jede 2. Zeile beginnend in Zeile 1



# LZW-Algorithmus beim GIF-Format

- In den Datenbereich eingetragen werden:
  - Indizes in die aktuelle Farbtabelle (Länge meist 8 bit) als Repräsentation von Einzel-Pixeln
  - Weitere Indizes (Länge zwischen Pixel-Indizes+1 und 12 bit) als Repräsentation von Pixelfolgen (zeilenweise)
- Startbelegung der LZW-Code-Tabelle
  - ist implizit mit der Farbtabelle gegeben
- Rücksetzen der LZW-Codierung
  - Spezieller Reset-Code (*clear code*) erlaubt völligen Neustart der Codierung
  - Im Prinzip an jeder Stelle möglich, v.a. am Beginn eines neuen Bildes
- Packen von Bitcodes in Bytes
  - Codes werden in Bytes (8-bit-Worte) gepackt
  - Platzersparnis

# Animated GIF

- GIF-Datei mit mehreren Bildern als einfacher „Film“
  - Bilder enthalten verschiedene Stadien der Animation
  - Anzeigeprogramm zeigt zyklisch die verschiedenen Bilder an, mit definierter Wartezeit dazwischen
- Praktische Bedeutung:
  - Eine der einfachsten Formen, Besucher von Web-Seiten vom eigentlichen Inhalt abzulenken ...
  - Heutzutage sehr schwach im Vergleich zu Animationstechniken wie Macromedia Flash (sh. später)
  - Dennoch: Einfach handzuhaben und plattformübergreifend stabil implementiert



# Portable Network Graphics PNG („Ping“)

- Geschichte:
  - Ausgelöst durch Lizenzforderungen für GIF-Format (1994)
  - Arbeitsgruppe beim W3C für PNG, standardisiert 1996, offen und lizenzfrei
- Ziel:
  - Besserer Ersatz für GIF, teilweise auch Ersatz für JPEG
  - Langsam zunehmende praktische Verbreitung, z.B. durch Wikipedia
- Farbtiefen:
  - 24 oder 48 bit „TrueColor“, 8 oder 16 bit Graustufen, Paletten bis 256 Farben (optional)
- Hauptvorteile:
  - Völlig verlustfrei ("Deflate"-Algorithmus: Lempel-Ziv- + Huffman-Kompression)
  - Echter Alpha-Kanal
  - Gamma-Korrektur (Gamma-Wert der Quellplattform speicherbar)
  - Verbessertes Interlacing (7-Pass-Algorithmus „Adam7“)
  - Bessere Kompression (Kompressionsfilter)
  - Integritätstest für Dateien (*magic signature*, CRC-32)

# PNG: Beispiel



# Echter Alpha-Kanal in PNG

- Alpha-Werte pro Pixel gespeichert
  - 4 Bytes pro Pixel: „RGBA“-Farbmodell
  - Ermöglicht elegante Schatten und Übergänge zwischen Grafik und Hintergrund
- Vermeidet Wechselwirkungen zwischen Anti-Aliasing und Transparenzfarbe
  - Bei „binärer Transparenz“ wie in GIF oft „weißer Rand“ um transparente Grafiken aufgrund von Anti-Aliasing (erzeugt nicht-transparente Farben)



# Kompressionsverbesserung durch Filter in PNG

- Beispiel:
  - Wertfolge 1, 2, 3, 4, 5, 6, 7, 8, 9, ...
  - Komprimiert extrem schlecht mit LZ-artigen Algorithmen
- Filter (Prädiktion):
  - Ersetze alle Zahlen (außer der ersten) durch die Differenz zur vorhergehenden
  - Wertfolge: 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
  - Komprimiert exzellent! (viele Wiederholungen)
- Filter in PNG:
  - Sub: Differenz zum linksstehenden Byte
  - Up: Differenz zum darüberstehenden Byte
  - Average: Differenz zum Durchschnitt der Sub- und Up-Bytes
  - Paeth: Differenz zum *Paeth-Prediktor* (siehe nächste Folie)
    - » Benutzt linksstehendes, darüberstehendes und "links oben" stehendes Byte
  - Heuristiken zur Wahl des passenden Filters

# Paeth-Prediktor

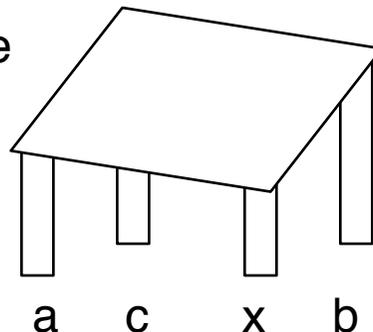
	$c$	$b$	
	$a$	$x$	

$$Px = a + b - c$$

Erfinder: Alan W. Paeth

- Den Prädiktor  $a+b-c$  kann man sich am einfachsten algebraisch erklären:
- Seien  $Ra = f(x_1, y_1)$ ,  $Rb = f(x_2, y_2)$ ,  $Rc = f(x_1, y_2)$ ,  $Rx = f(x_2, y_1)$ .
- Sei  $f$  linear in  $x$  und  $y$ , d.h.  $f(x, y) = Ax + By$ .
- $Ra + Rb - Rc = Ax_1 + By_1 + Ax_2 + By_2 - Ax_1 - By_2 = Ax_2 + By_1 = Rx$

Geometrische  
Erklärung:



# Welches Format wofür?

- Für Web-Grafiken (klein, geringe Farbanzahl)
  - GIF oder PNG
- Für Bilderzeugung mit Scanner oder Austausch über diverse Geräte hinweg:
  - TIFF
- Für hochauflösende Bilder mit vielen Farben (Fotos)
  - JPEG (wegen wesentlich besserer Kompression)
  - Bei grossen einheitlichen Farbflächen evtl. auch PNG (beste Qualität)

# 6. Licht, Farbe und Bilder

- 6.1 Licht und Farbe: Physikalische und physiologische Aspekte
- 6.2 Farbmodelle
- 6.3 Raster-Bilddatenformate
  - Grundbegriffe für Bildspeicherung und -Bearbeitung
  - Bitmap-Formate
  - Verlustfrei komprimierende Formate
- 6.4 Verlustbehaftete Kompression bei Bildern 
- 6.5 Weiterentwicklungen bei der Bildkompression

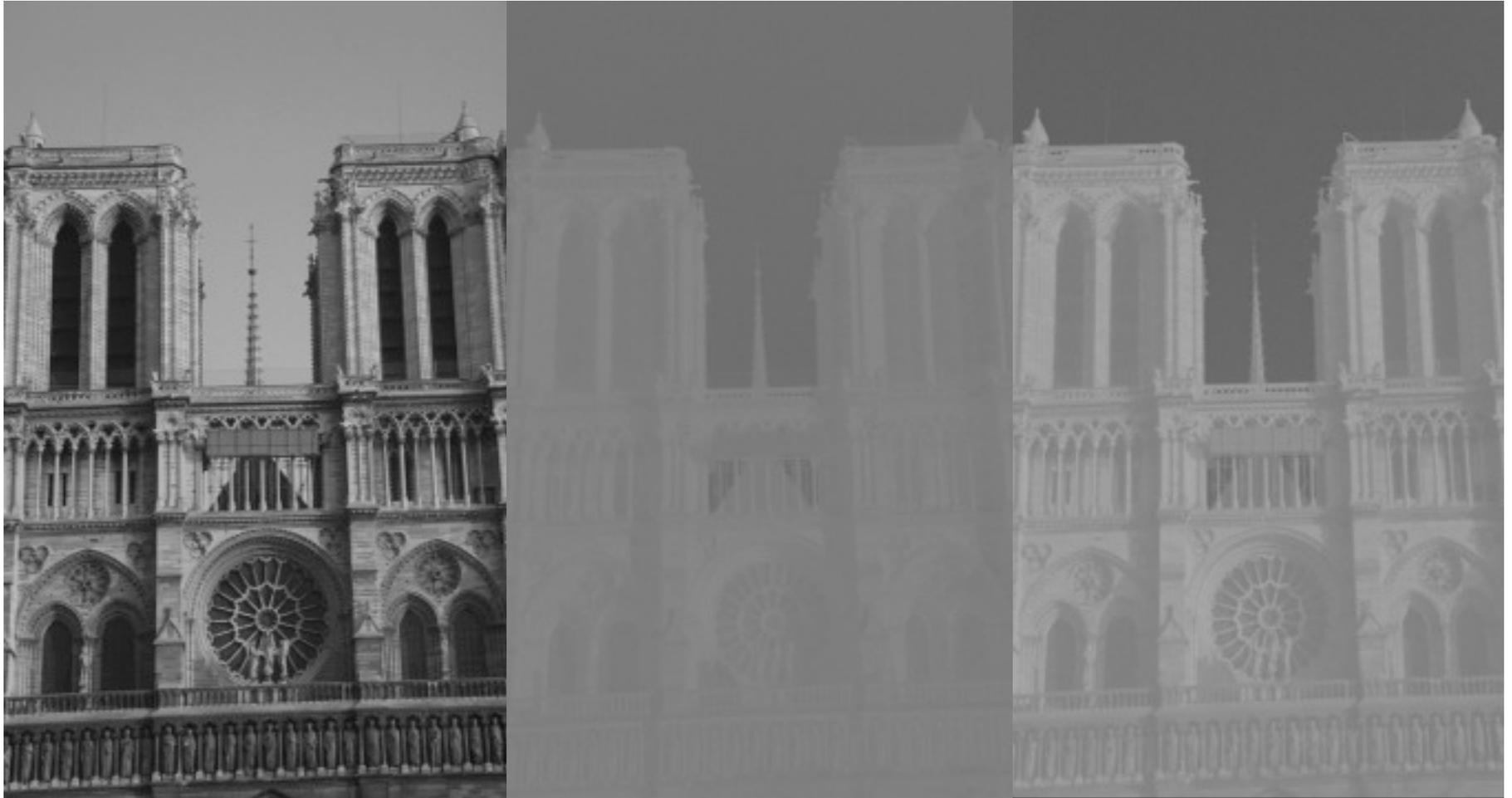
Weiterführende Literatur:

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. 12 Mio. Pixel mit je 24 bit = 36 MByte)
- Das 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.: RAW, TIFF, PNG

# Luma- und Chromainformation: Vergleich

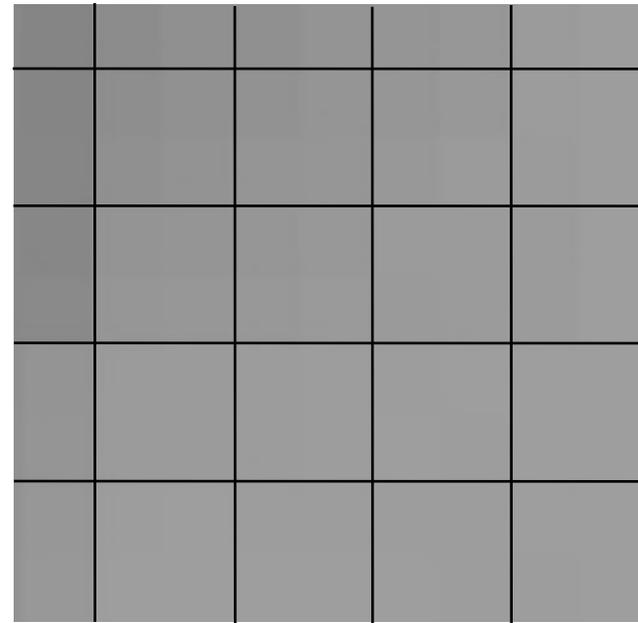
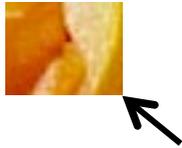


Helligkeit (L-Kanal)

Rot/Grün-Differenz  
(a-Kanal)

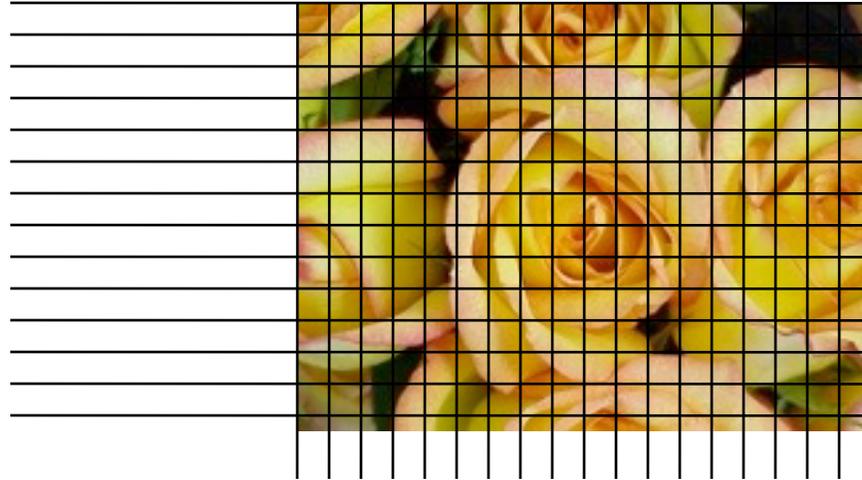
Blau/Gelb-Differenz  
(b-Kanal)

# Chroma-Subsampling



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

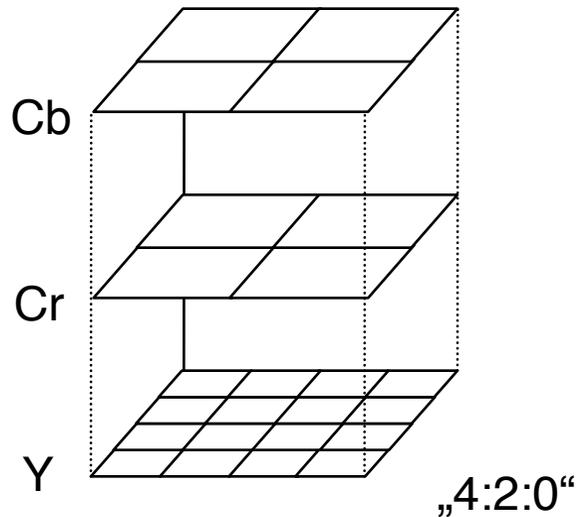
# Abtastraten für Bilder



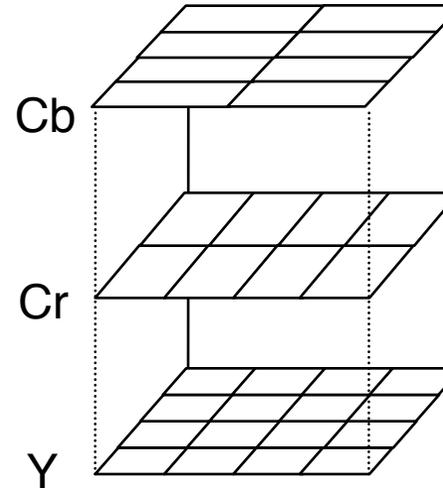
- Abtastrate: Wieviele Pixel pro Längeneinheit des Bildes?
- Mehrdimensionalität:
  - Horizontale Abtastrate ( $H$ )
  - Vertikale Abtastrate ( $V$ )
- Bei Sub-Sampling:
  - Verschiedene Abtastraten für verschiedene Komponenten des Bildes (Farben, evtl. Alphakanal)

# Subsampling

$$\begin{aligned} Y: H_Y &= 4, V_Y = 4 \\ Cr: H_{Cr} &= 2, V_{Cr} = 2 \\ Cb: H_{Cb} &= 2, V_{Cb} = 2 \end{aligned}$$



$$\begin{aligned} Y: H_Y &= 4, V_Y = 4 \\ Cr: H_{Cr} &= 4, V_{Cr} = 2 \\ Cb: H_{Cb} &= 2, V_{Cb} = 4 \end{aligned}$$



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

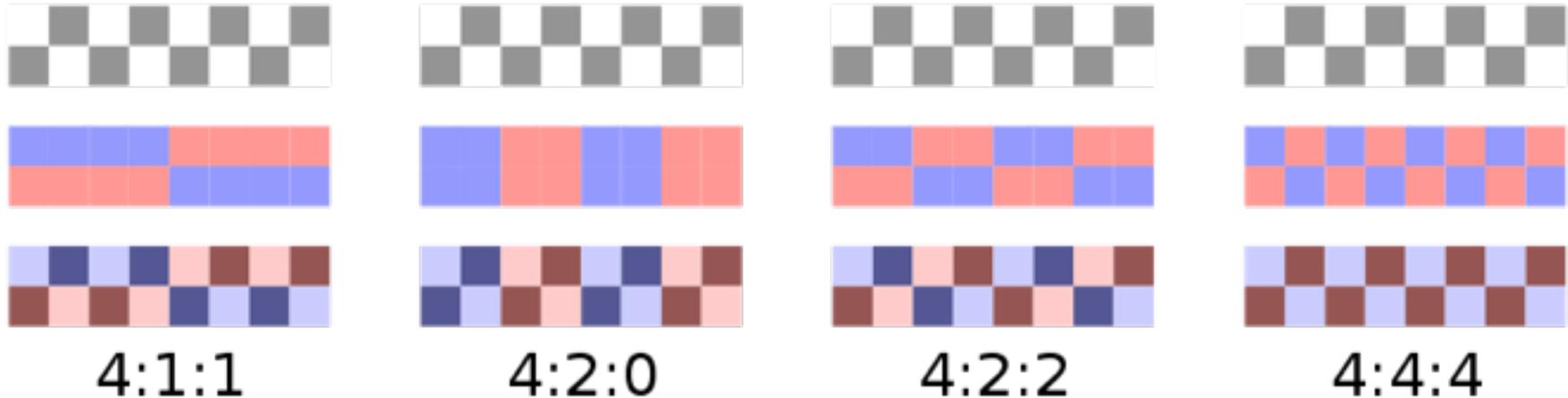


# Notation für Subsampling

- Übliche Notation für Subsampling von Farben:  
x:y:z
  - Vertikales Subsampling oft nicht genutzt
  - Ursprüngliche Bedeutung:  
Horizontales Frequenzverhältnis für Luma (x) zu den Chroma-Kanälen (y, z)
- Heutige Bedeutung:
  - Beide Chroma-Kanäle immer gleich abgetastet
  - x: Anzahl der Luma-Samples; in der Regel „4“
  - y: Anzahl der Cr/Cb-Chroma-Samples, horizontal, in der ersten Zeile
  - z: Anzahl der zusätzlichen Cr/Cb-Chroma-Samples in der zweiten Zeile  
(z=y: kein vertikales Subsampling, z=0: vertikales Subsampling 2:1)
- Beispiele :
  - 4:2:2  $H_Y=4, V_Y=4, H_{Cr}=2, V_{Cr}=4, H_{Cb}=2, V_{Cb}=4$
  - 4:1:1  $H_Y=4, V_Y=4, H_{Cr}=1, V_{Cr}=4, H_{Cb}=1, V_{Cb}=4$
  - 4:2:0 entspricht  $H_Y=4, V_Y=4, H_{Cr}=2, V_{Cr}=2, H_{Cb}=2, V_{Cb}=2$   
(bei JPEG weit verbreitet)
- Bandbreitenformel: Summe der drei Zahlen geteilt durch 12



# Beispiele zur Notation für Subsampling



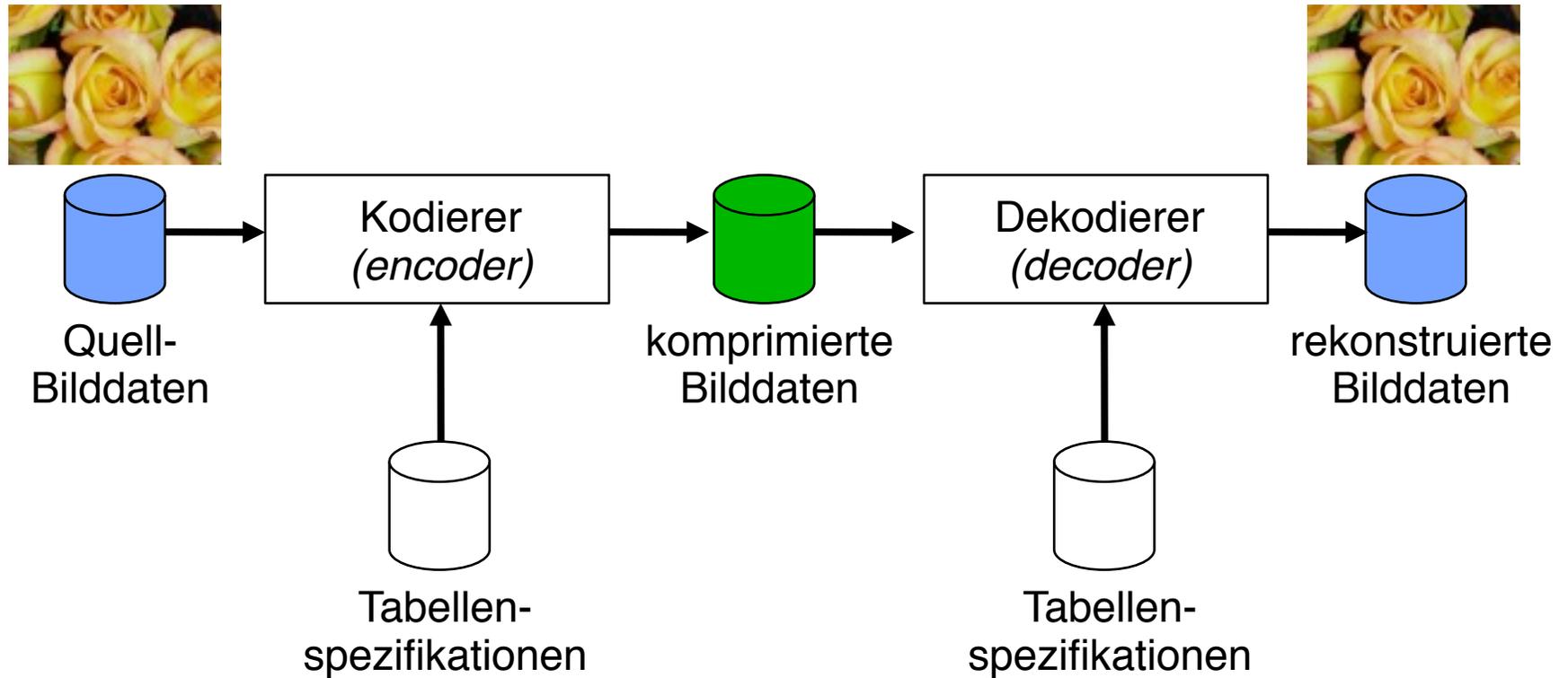
- x: Anzahl der Luma-Samples; in der Regel „4“
- y: Anzahl der Cr/Cb-Chroma-Samples, horizontal, in der ersten Zeile
- z: Anzahl der zusätzlichen Cr/Cb-Chroma-Samples in der zweiten Zeile (z=y: kein vertikales Subsampling, z=0: vertikales Subsampling 2:1)

Bild: Wikipedia

# JPEG: Hintergrundinformation

- 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



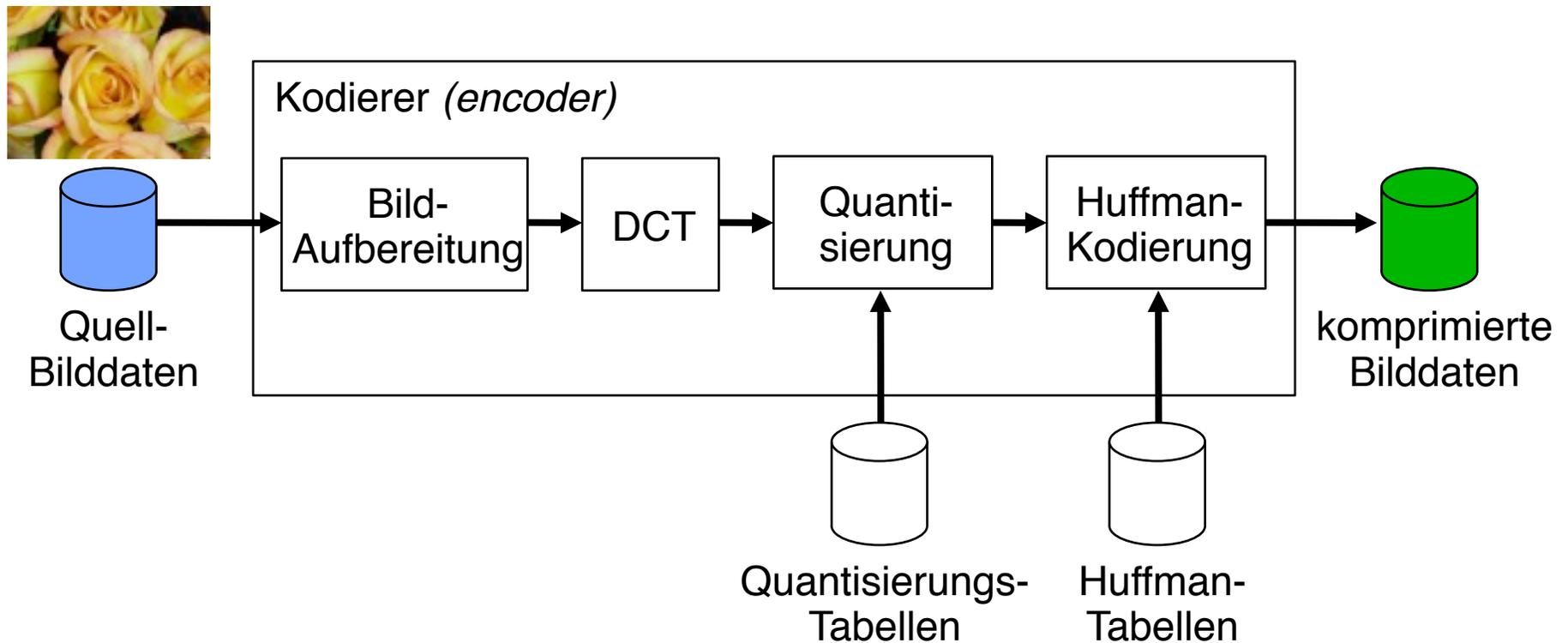
# JPEG-Modi

- 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)

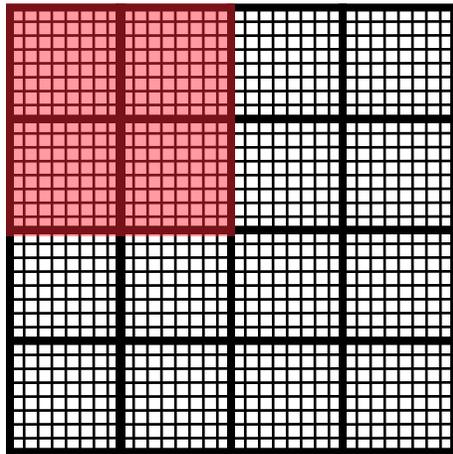
- Bild wird generell in 8 x 8-Pixel-Blöcke (*data units*) eingeteilt
  - Am Rand wird „aufgefüllt“
- Bild kann theoretisch aus bis zu 255 Komponenten (*components*) bestehen
  - Verbreitet: 3 oder 4, nach Farbmodell
- Verzahnte (*interleaved*) oder nicht-verzahnte Reihenfolge:
  - Ablage der Komponenten 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)

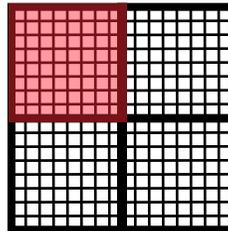
Subsampling

- Interleaving bei gleichzeitigem Chroma-Subsampling:
  - Jede Komponente  $c$  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

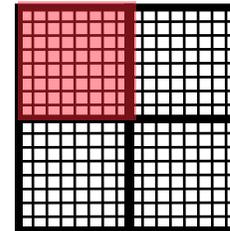
Beispiel: MCU bei 4:2:0-Subsampling  
( $H_Y = 4, V_Y = 4, H_{Cr} = 2, V_{Cr} = 2, H_{Cb} = 2, V_{Cb} = 2$ )



Y



Cr

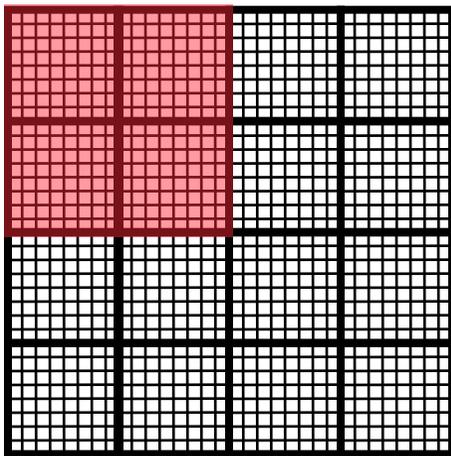


Cb

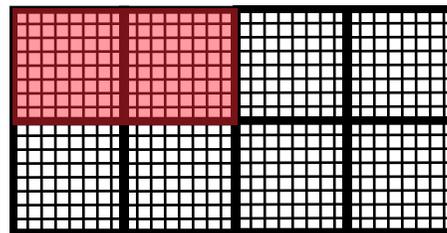
# JPEG-Kodierung: Bildaufbereitung (3)

Subsampling

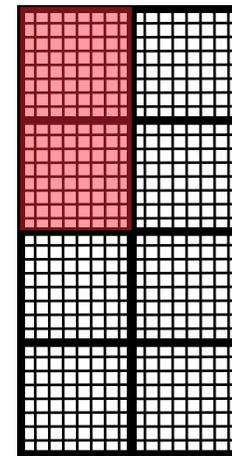
Subsampling für Y:  $H_Y = 4$ ,  $V_Y = 4$ , für Cr:  $H_{Cr} = 4$ ,  $V_{Cr} = 2$ , für Cb:  $H_{Cb} = 2$ ,  $V_{Cb} = 4$



Y



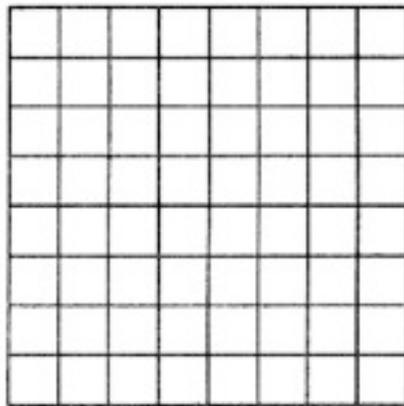
Cr



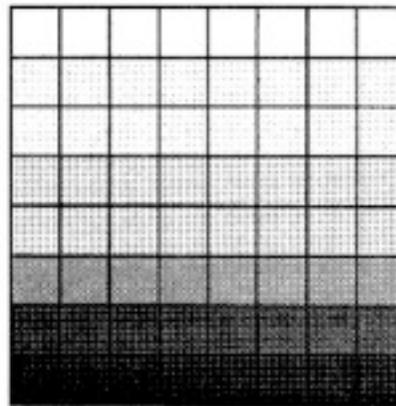
Cb

# Ortsfrequenz

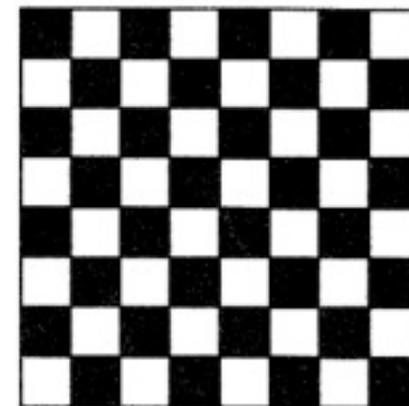
- Ortsfrequenz (oder: räumliche Frequenz, *spatial frequency*)
  - Häufigkeit der Wiederholung einer im Bild erkennbaren Eigenschaft über die räumliche Ausdehnung
  - Maßeinheit: 1/Längeneinheit
  - z.B. Dichte von Linien auf Papier: Anzahl Striche pro cm
- Meist: Anzahl von Helligkeitsschwankungen pro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



Ortsfrequenz 0



Ortsfrequenz  
0 horizontal,  
niedrig vertikal

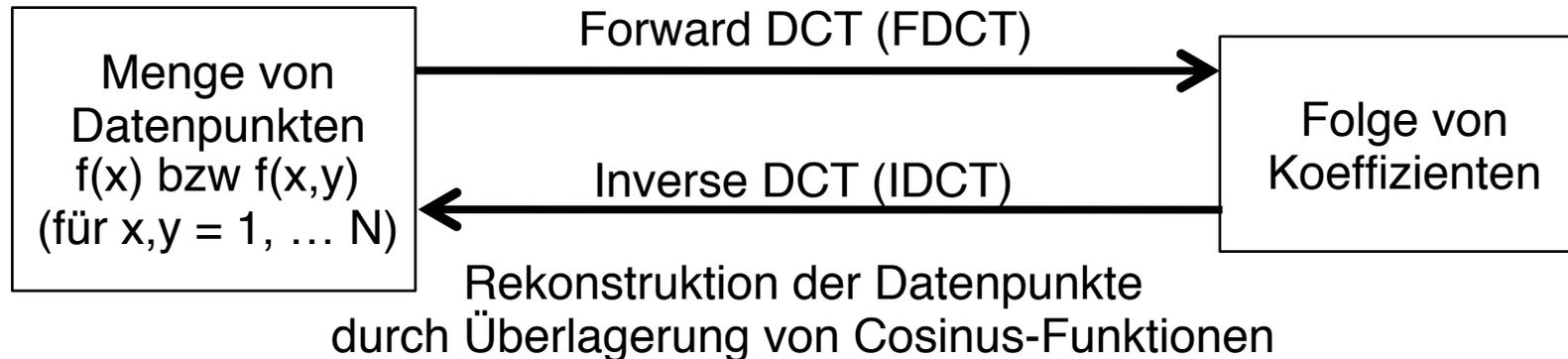


Ortsfrequenz  
hoch  
horizontal und vertikal

# Diskrete Cosinus-Transformation (DCT)

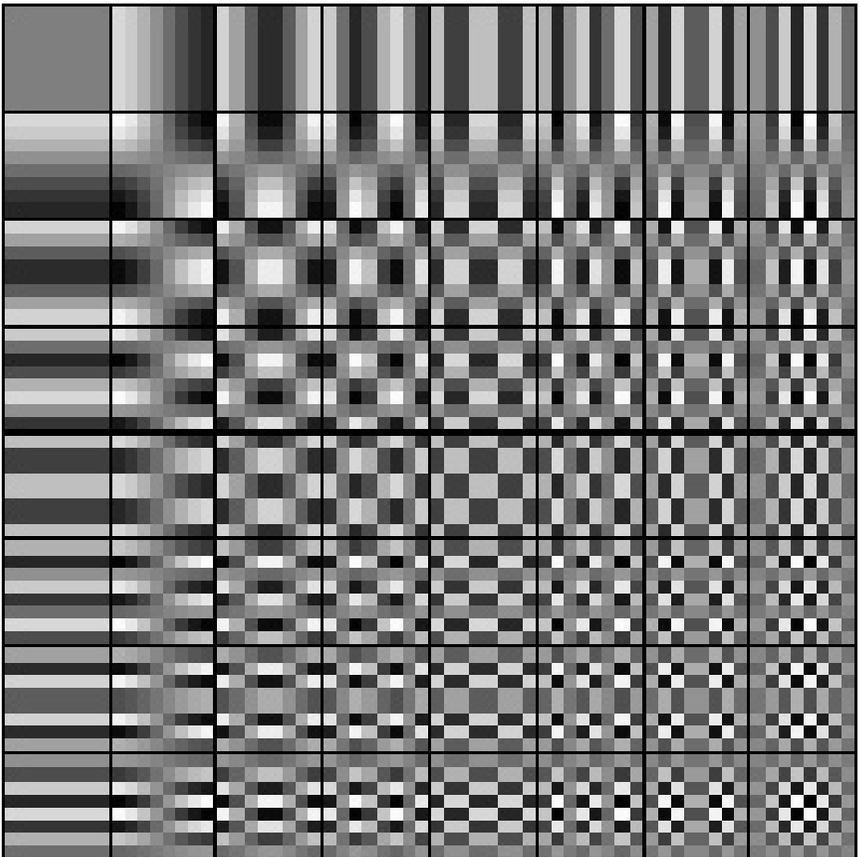
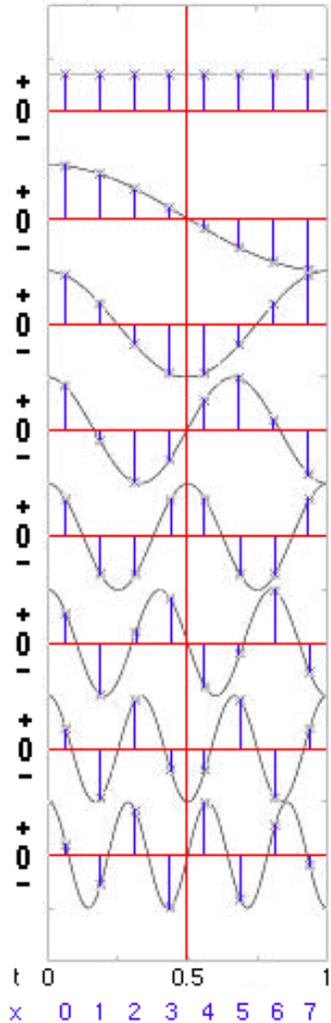
JPEG-Schritte

- Grundmotivation:
  - Menschliche Sehwahrnehmung 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

# Basisfunktionen der DCT in 1D und 2D



# (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

# Matrixdarstellung zur Durchführung einer DCT

$$\begin{bmatrix}
 \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{2} \\
 \frac{1}{2}\cos\left(\frac{1}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{3}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{5}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{7}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{9}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{11}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{13}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{15}{16}\pi\right) \\
 \frac{1}{2}\cos\left(\frac{1}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{3}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{5}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{7}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{9}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{11}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{13}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{15}{8}\pi\right) \\
 \frac{1}{2}\cos\left(\frac{3}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{9}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{15}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{21}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{27}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{33}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{39}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{45}{16}\pi\right) \\
 \frac{1}{2}\cos\left(\frac{1}{4}\pi\right) & \frac{1}{2}\cos\left(\frac{3}{4}\pi\right) & \frac{1}{2}\cos\left(\frac{5}{4}\pi\right) & \frac{1}{2}\cos\left(\frac{7}{4}\pi\right) & \frac{1}{2}\cos\left(\frac{9}{4}\pi\right) & \frac{1}{2}\cos\left(\frac{11}{4}\pi\right) & \frac{1}{2}\cos\left(\frac{13}{4}\pi\right) & \frac{1}{2}\cos\left(\frac{15}{4}\pi\right) \\
 \frac{1}{2}\cos\left(\frac{5}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{15}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{25}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{35}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{45}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{55}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{65}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{75}{16}\pi\right) \\
 \frac{1}{2}\cos\left(\frac{3}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{9}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{15}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{21}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{27}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{33}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{39}{8}\pi\right) & \frac{1}{2}\cos\left(\frac{45}{8}\pi\right) \\
 \frac{1}{2}\cos\left(\frac{7}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{21}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{35}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{49}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{63}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{77}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{91}{16}\pi\right) & \frac{1}{2}\cos\left(\frac{105}{16}\pi\right)
 \end{bmatrix}$$

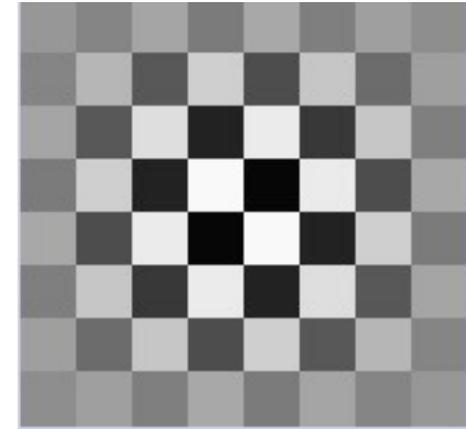
# Beispiele für DCT-Transformation



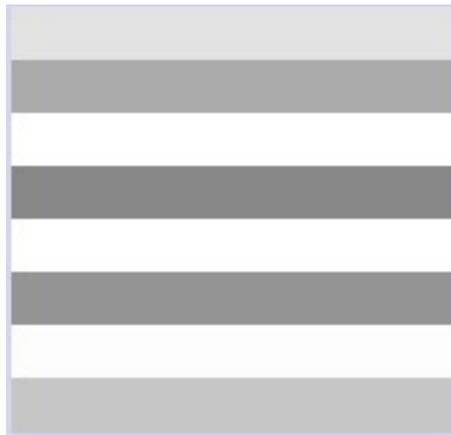
$F(0,1) = 500$ ,  
alle anderen  $F(u, v) = 0$



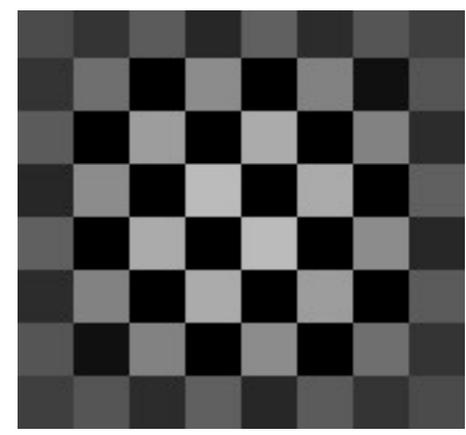
$F(7,0) = 500$ ,  
alle anderen  $F(u, v) = 0$



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



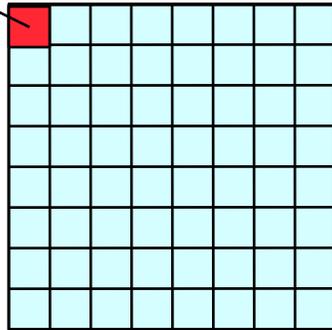
$F(7,0) = 500$ ,  $F(0,0) = 600$   
alle anderen  $F(u, v) = 0$



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

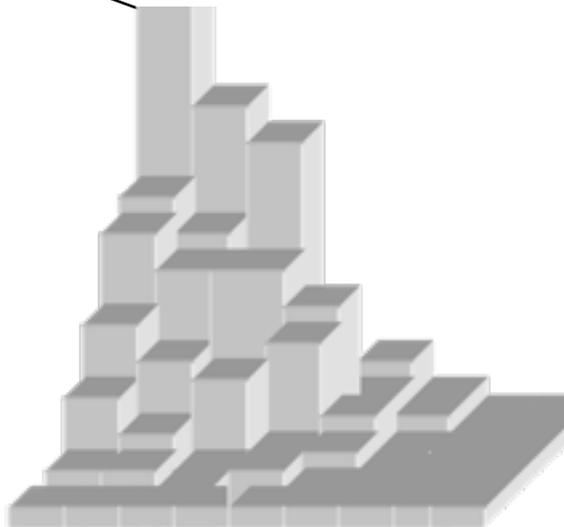
# Interpretation der DCT-Koeffizienten

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

# 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

JPEG-Schritte

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

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

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

# Rechenbeispiel: Quantisierung

$$\begin{pmatrix} 93 & 2 & -8 & -7 & 3 & 1 & 1 & -2 \\ -38 & -58 & 11 & 17 & -3 & 5 & 5 & -3 \\ -84 & 63 & -1 & -17 & 2 & 7 & -4 & 0 \\ -51 & -37 & -10 & 13 & -10 & 5 & -1 & -4 \\ -85 & -42 & 50 & -8 & 18 & -5 & -1 & 1 \\ -63 & 66 & -13 & -1 & 2 & -6 & -2 & -2 \\ -16 & 14 & -37 & 18 & -12 & 4 & 3 & -3 \\ -53 & 31 & -7 & -10 & 23 & 0 & 2 & 2 \end{pmatrix}$$

DCT-Koeffizienten

$$\begin{pmatrix} 31 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -7 & -8 & 1 & 1 & 0 & 0 & 0 & 0 \\ -12 & 7 & 0 & -1 & 0 & 0 & 0 & 0 \\ -5 & -3 & 0 & 0 & 0 & 0 & 0 & 0 \\ -7 & -3 & 3 & 0 & 0 & 0 & 0 & 0 \\ -4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

quantisierte DCT-Koeffizienten

$$\begin{pmatrix} 3 & 5 & 7 & 9 & 11 & 13 & 15 & 17 \\ 5 & 7 & 9 & 11 & 13 & 15 & 17 & 19 \\ 7 & 9 & 11 & 13 & 15 & 17 & 19 & 21 \\ 9 & 11 & 13 & 15 & 17 & 19 & 21 & 23 \\ 11 & 13 & 15 & 17 & 19 & 21 & 23 & 25 \\ 13 & 15 & 17 & 19 & 21 & 23 & 25 & 27 \\ 15 & 17 & 19 & 21 & 23 & 25 & 27 & 29 \\ 17 & 19 & 21 & 23 & 25 & 27 & 29 & 31 \end{pmatrix}$$

Quantisierungsmatrix

<http://www.mathematik.de/>

# Informationsverlust durch Quantisierung



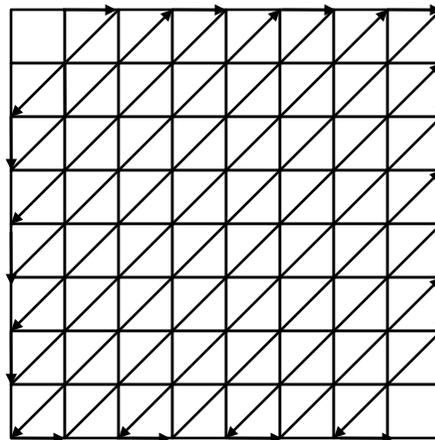
Bei JPEG-Kompressions-Algorithmen ist der Grad der Quantisierung wählbar: „Trade-Off“ zwischen Speicherplatzersparnis und Bildverfälschung (Artefakten)



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“)



# JPEG-Kodierung: Entropie-Kompression

JPEG-Schritte

- 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 Beispielveilteilungen im JPEG-Standard beschrieben
  - Alternative: Durch zusätzlichen Durchlauf über die Daten errechnen

# JFIF Dateiformat

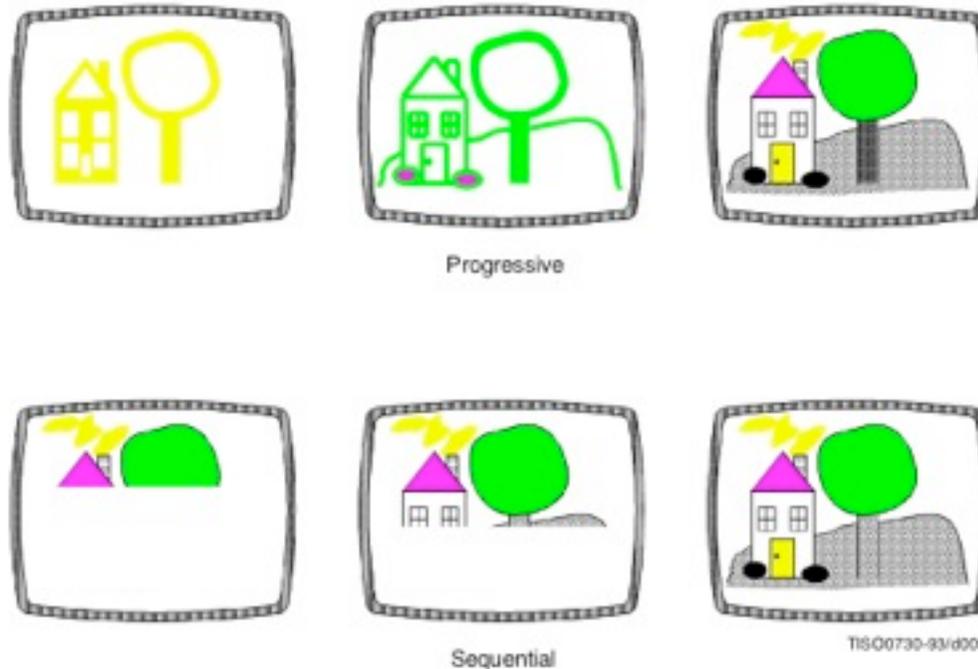
- Der JPEG-Standard definiert das Dateiformat nicht im Detail.
- De-Facto-Standard: JFIF (JPEG File Interchange Format)
  - inoffiziell (David Hamilton 1992)
- Neuer offizieller Standard: SPIFF (Still Picture Interchange File Format)
  - von der JPEG
  - spät eingeführt, kompatibel mit JFIF, aber wesentlich flexibler
- JFIF definiert:
  - „Signatur“ zur Identifikation von JPEG-Dateien („JFXX“)
  - Farbraum
  - Pixeldichte
  - Vorschaubilder („Thumbnails“)
  - Zusammenhang Pixel – Abtastfrequenz

# 6. Licht, Farbe und Bilder

- 6.1 Licht und Farbe: Physikalische und physiologische Aspekte
- 6.2 Farbmodelle
- 6.3 Raster-Bilddatenformate
- 6.4 Verlustbehaftete Kompression bei Bildern
- 6.5 Weiterentwicklungen bei der Bildkompression
  - Progressives und hierarchisches JPEG 
  - JPEG XR und WebP
  - Wavelet-basierte Verfahren (verlustbehaftet), insb. JPEG 2000
  - Prädiktionsbasierte Verfahren (verlustfrei)

# Progressives JPEG

- Ein Durchlauf (*scan*) durch die JPEG-Daten kann Verschiedenes bewirken:
  - Ausgabe einer Komponente des Bildes
  - Ausgabe einer unscharfen Vorversion des Bildes
- *Progressive Coding* verbessert die Bildqualität in aufeinander folgenden *scans*.



# Progressive Kodierung durch Spektralselektion

- 8x8-Block von DCT-Koeffizienten
  - Zick-Zack-Reihenfolge geht von niedrigen Frequenzen (wenig Detail) zu hohen Frequenzen (viel Detail).
- *Band*: Teilintervall der Bildfrequenzen
  - als Intervall der DCT-Koeffizienten
- Je Band ein separater *scan*
  - Bandgrenzen im *scan header* angegeben

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Beispiel: 5 Bänder (d.h. 5 *scans*)

Band 1: DCT-Koeffizient 0 (DC)

Band 2: DCT-Koeffizienten 1 – 2

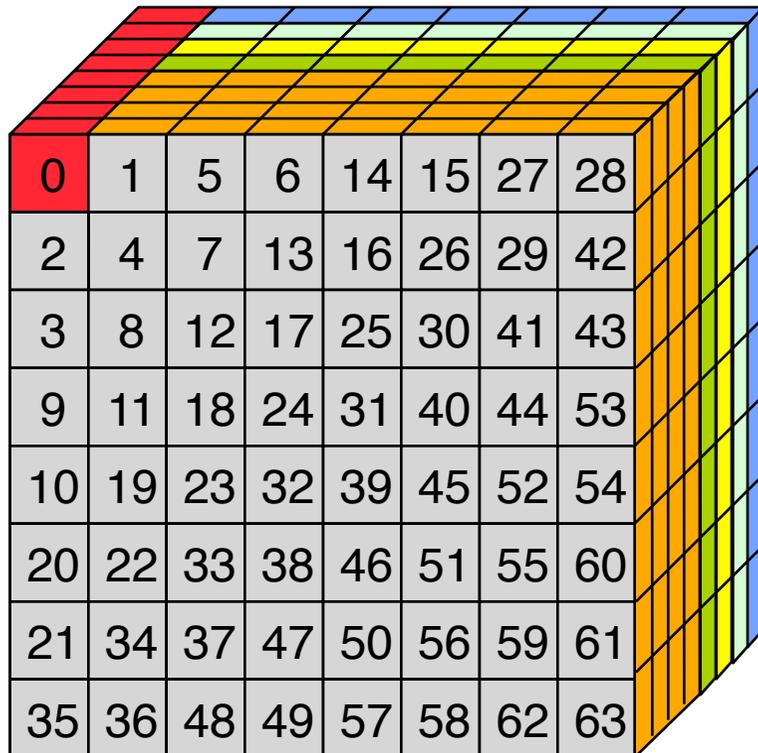
Band 3: DCT-Koeffizienten 3 – 14

Band 4: DCT-Koeffizienten 15 – 42

Band 5: DCT-Koeffizienten 43 – 63

# Progressive Kodierung durch Bit Plane Approximation

- Koeffizienten werden zunächst mit geringerer Präzision übertragen
  - Division mit Zweierpotenz bzw. Rechts-Shift (*point transform*)
  - Definition der verwendeten Transformation im *scan header*
- Fehlende Bits werden in weiteren *scans* nachgeliefert



A 3D visualization of a coefficient matrix. The matrix is shown as a grid of 8x8 cells, with the first cell (0) highlighted in red. The grid is colored in a gradient from red to blue, representing different bit planes. The numbers in the grid are: 0, 1, 5, 6, 14, 15, 27, 28; 2, 4, 7, 13, 16, 26, 29, 42; 3, 8, 12, 17, 25, 30, 41, 43; 9, 11, 18, 24, 31, 40, 44, 53; 10, 19, 23, 32, 39, 45, 52, 54; 20, 22, 33, 38, 46, 51, 55, 60; 21, 34, 37, 47, 50, 56, 59, 61; 35, 36, 48, 49, 57, 58, 62, 63.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Beispiel: 6 *scans*

Scan 1: DCT-Koeffizient 0 (DC)

Scan 2: Bits 4 – 7 der DCT-Koeffizienten  
1 – 63 (d.h. der AC-Koeffizienten)

Scan 3: Bit 3 der AC-Koeffizienten

Scan 4: Bit 2 der AC-Koeffizienten

Scan 5: Bit 1 der AC-Koeffizienten

Scan 6: Bit 0 der AC-Koeffizienten

# 6. Licht, Farbe und Bilder

- 6.1 Licht und Farbe: Physikalische und physiologische Aspekte
- 6.2 Farbmodelle
- 6.3 Raster-Bilddatenformate
- 6.4 Verlustbehaftete Kompression bei Bildern
- 6.5 Weiterentwicklungen bei der Bildkompression
  - Progressives und hierarchisches JPEG
  - JPEG XR und WebP 
  - Wavelet-basierte Verfahren (verlustbehaftet), insb. JPEG 2000
  - Prädiktionsbasierte Verfahren (verlustfrei)

# JPEG XR

- Microsoft-spezifische Fotoformate:
  - "Windows Media Photo", eingeführt mit Windows Vista
  - 2006 umbenannt in "HD Photo"
  - Seit 2009 ISO-Standard unter dem Namen "JPEG XR"
- Ähnlich zu JPEG
  - Photo Core Transformation (PCT), ähnlich zu DCT
  - Auf verschobenen 4x4-Blöcken arbeitende "Photo Overlap Transformation" vermeidet Blockartefakte
- Vorteile gegen JPEG:
  - Verlustfreie und verlustbehaftete Kompression in einem Verfahren
  - Direkter Zugriff auf Bildkacheln (Regionen)
  - Unterstützung für extrem hohe Farbtiefen (48 bit)
    - » Ziel: High Dynamic Range (HDR) Fotografie
  - Echter Alpha-Kanal

# WebP

- Verlustbehaftetes Kompressionsverfahren für Foto-Bilder
  - Angeblich ca. 40% kleiner als JPEG bei vergleichbarer Qualität
  - Von Google entwickelt (Ankauf von *on2*) und als offener Standard verbreitet
  - Basiert auf dem Video-Standard VP8, analog zum Videoformat "WebM"
  - Verwendet RIFF-Container zur Datenablage
- Basiert mehr auf *Prädiktion* von Pixelwerten im Vergleich zu JPEG
  - vgl. PNG/Paeth-Prediktor oben und nächste Vorlesung
- Quantisierung von Residual-Werten
  - d.h. Differenz zwischen prädiziertem und tatsächlichem Wert
- Noch wenig praktische Erfahrung:
  - Freigabe 30. September 2010
  - Erste Berichte über Bildqualität eher zurückhaltend

Google Wants to Kill the JPEG: Meet WebP

<http://mashable.com/2010/09/30/google-webp/>

# WebP Update 2011

- Funktionsumfang erweitert
  - jetzt auch verlustfreie Kompression (Konkurrenz zu PNG)
  - Alphakanäle
  - ICC Farbprofile
  - Metadaten
- Vorteile jetzt differenzierter beschrieben
  - 25 bis 34 % bessere verlustbehaftete Kompression
  - 28 % stärker bei verlustfreier Kompression
- Tools verfügbar: <http://code.google.com/intl/de-DE/speed/webp/>
- Durchsetzung in der Praxis bleibt weiterhin abzuwarten