EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Teilnahme auf ALMA registrieren

Bildkomprimierung

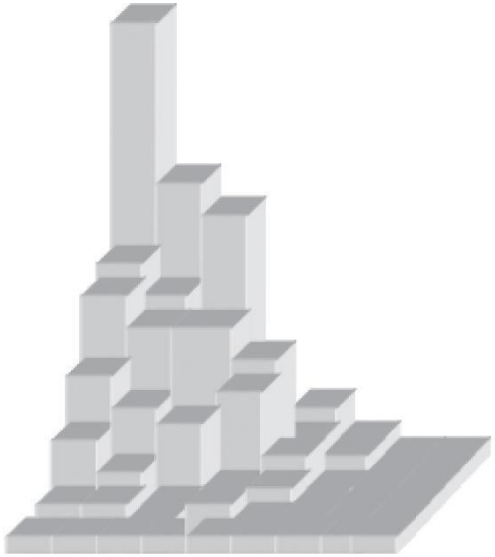
JPEG - Kompressionsschritte

1. **Farbraumumrechnung** vom RBG- zum YCbCr-Farbraum (verlustfrei)
2. **Chroma-Subsampling** (verlustbehaftet)
3. **DCT** – Diskrete Cosinus Transformation (verlustbehaftet durch Rundungsfehler und endlich viele Frequenzen)
4. **Quantisierung** → kleinere Zahlen für Frequenzanteile = effizientere Codierung (verlustbehaftet)
5. **Umsortierung durch Zick-Zack-Schema** (verlustfrei)
6. **Entropiekodierung:** z.B. RLE mit Huffman-Kodierung (verlustfrei)

DCT – Diskrete Cosinus Transformation

1. Unterteilung des Bilds in 8x8 Pixelblöcke → viele Pixel werden ähnliche Farben haben
   → kaum hohe Frequenzen, die Kanten repräsentieren
2. Berechnung der Cosinus-Frequenzanteile

$C_{0,0}$= DC-Koeffizient = Grundfarbe des Pixelblocks

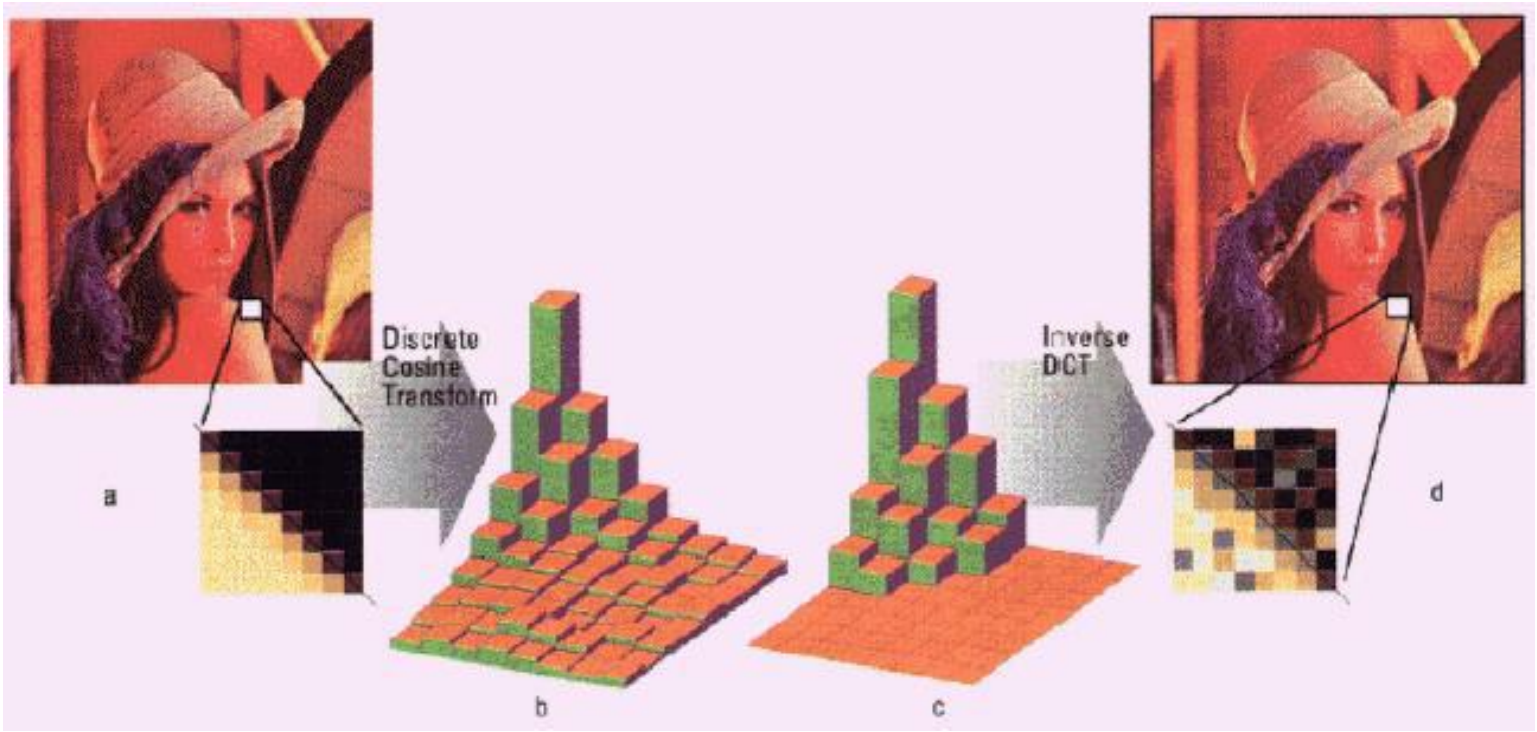


Vorteile:

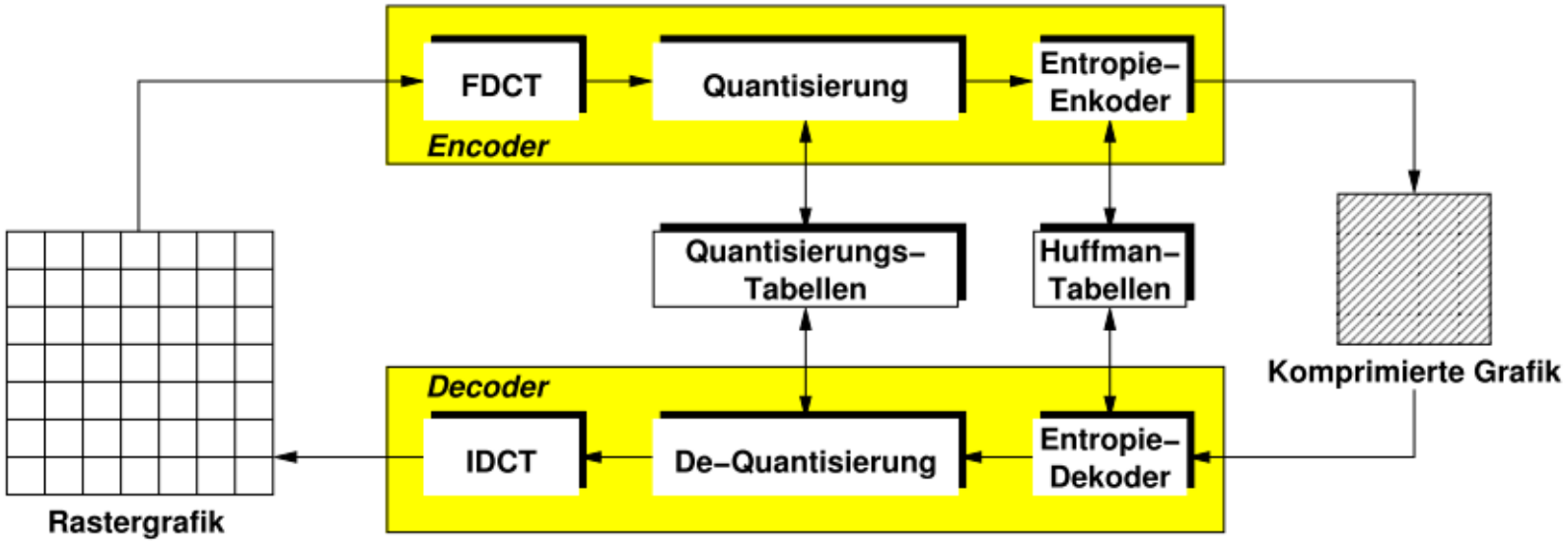
- Große (wichtige) Frequenzanteile liegen gesammelt in der linken oberen Ecke
- Im Gegensatz zur DFT reelle Gewichtungen

Quantisierung der DCT

- DCT-Koeffizienten werden durch Quantisierungsmatrix geteilt
  → kleine Zahlen mit kürzerer Kodierung
- **Verlustbehaftet**, da kleine Gewichte zu Nullen werden



Vom Bild zur JPEG-Datei und zurück



BESPRECHUNG ÜBUNG 8

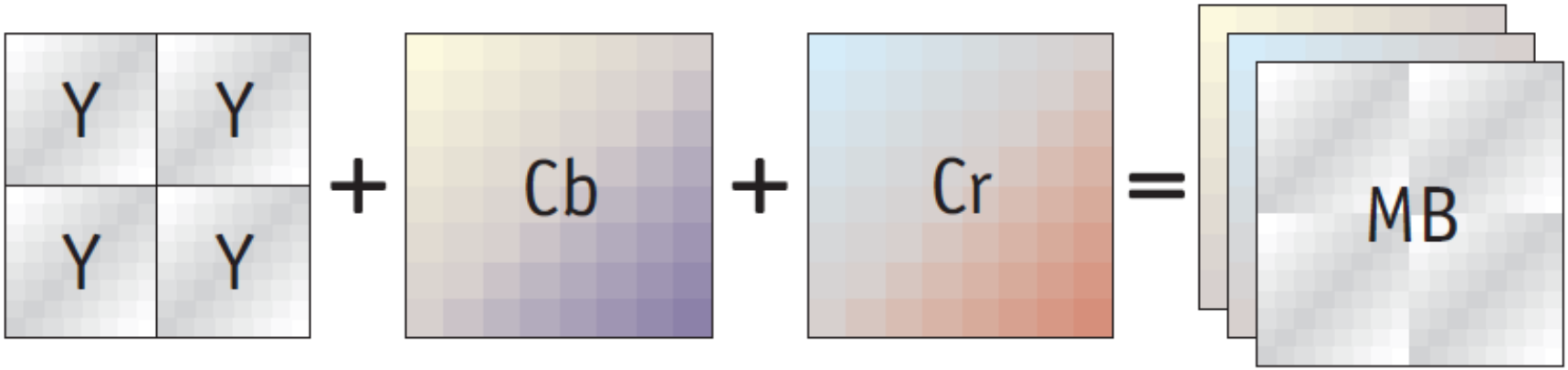
Videokomprimierung

Kompressionsansätze

- **Intra-Bild-Kodierung (räumlich):**
  „Redundanzen“ in einzelnen Bildern ausnutzen, um Datenvolumen einzusparen
  - DCT
  - Vektorquantisierung
  - Konturbasierte Kodierung
- **Inter-Bild-Kodierung (zeitlich):**
  „Redundanzen“ in Bildabfolgen ausnutzen, um nur Teile von späteren Bildern speichern zu müssen
  - Differenzkodierung
  - Bewegungskompensation

Makroblöcke

- Pro 16x16 Bildausschnitt speichern wir einen Makro-Block:



- Wegen dem Chroma-Subsampling besteht dieser aus:
  - 4 Luminanz-Blöcken (je 8x8)
  - 2 Chrominanz-Blöcke (je 16x16, aber nur jeder zweite Pixel wird abgetastet → 4:2:0)
- Größe der Blöcke ist vorteilhaft für die DCT

Intra-Bild-Kodierung (räumlich)

- **Vektorquantisierung**
  1. Bild in kleine Pixelblöcke aufteilen
  2. Ähnliche Blöcke als Gruppen sortieren
  3. Durchschnittsblock der Gruppe ersetzt originale Blöcke
  4. Position und neue Bildblöcke werden kodiert

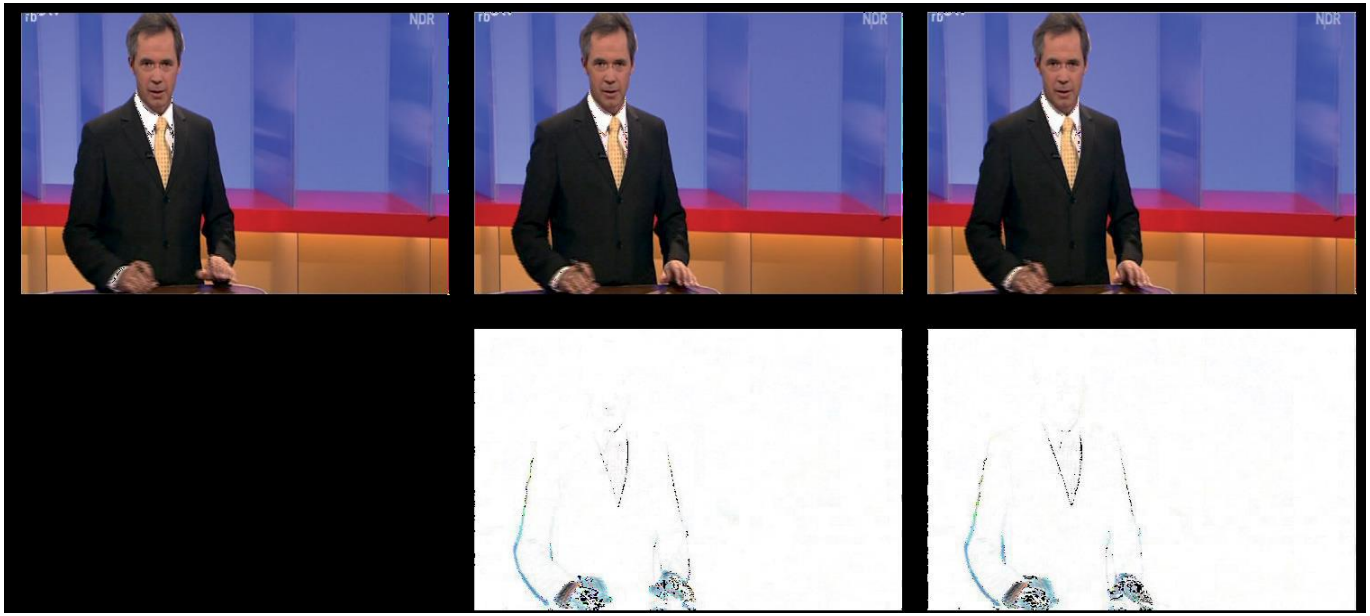
- **Konturbasierte Kodierung**
  1. Trennen Bild in Kontur und Textur
  2. Separate Speicherung dieser → effizientere Speicherung

  Problem: Finden der Konturen von Objekten

Inter-Bild-Kodierung (zeitlich)

- **Differenzkodierung**
  1. Startbild wird vollständig gespeichert
  2. Differenz zum Folgebild wird gespeichert und beim Abspielen des Videos auf das Startbild addiert

  Vorteil: Viele Pixel ändern sich nicht → viele Nullen



Inter-Bild-Kodierung (zeitlich)

- **Bewegungskompensation**
  1. Bewegung von Objekten in Bildabfolgen erkennen und abschätzen
  2. Im nächsten Bild wird der Hintergrund vom alten Bild genommen und das verschobene Objekt eingesetzt.
  3. Noch fehlende Pixel werden als Differenzbild gespeichert

  Probleme:
  Wie erkennen wir Objekte (die sich drehen)?
  Wie berechnen wir den Bewegungsvektor?

Präsenz-Übung

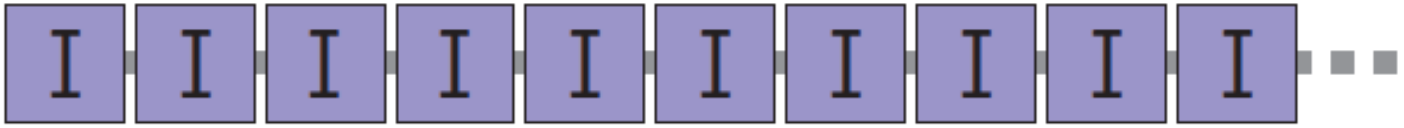
Welches der folgenden Videos lässt sich besser komprimieren? Erläutert eure Antwort an den Kodierungstypen.

- Nachrichtensendung

- Action Film

Frametypen in Videos

- **I-Frames** (intra-kodierte Frames)



- **P-Frames** (prädiktiv-kodierte Frames)

- **B-Frames** (bidirektionale prädiktiv-kodierte Frames)