



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

MIN-Fakultät
Fachbereich Informatik
Arbeitsbereich SAV/BV (KOGS)

Grundlagen der Signalverarbeitung und Robotik

Teil 1: Grundlagen der Signalverarbeitung

Vorlesung 5: Datenkompression

Benjamin Seppke

Jianwei Zhang

Datenkomprimierung

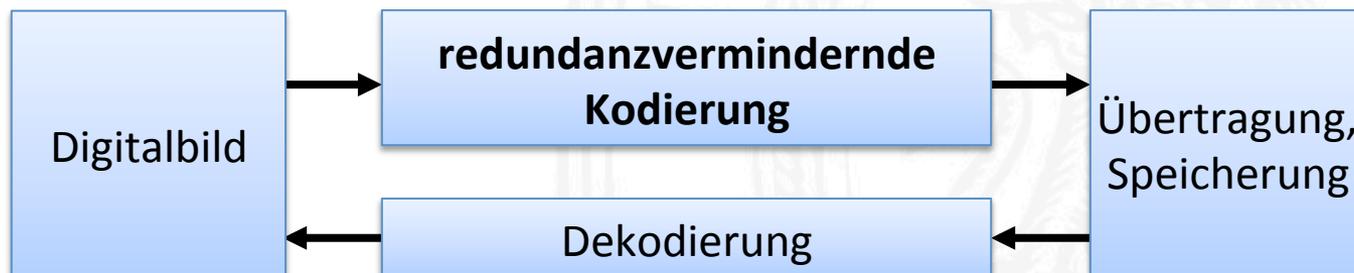
Datenkomprimierung dient der Einsparung von Speicherplatz und von Übertragungskapazität für digitale Daten. Dabei werden Daten in neuen Datenstrukturen platzsparender repräsentiert.

Wir behandeln Komprimierungstechniken am Beispiel von Bilddaten.

Bildkomprimierung ist wichtig für

- Bildarchivierung z.B. Satellitenbilder
- Bildübertragung z.B. Bilder aus dem Internet
- Multimedia-Anwendungen z.B. Bildschirmtextarbeiten

Datenkompression nutzt Redundanz zum Komprimieren aus:



Unterscheidung von Datenkompressionsarten

Verlustfreie Datenkompression:

- Originaldaten können **exakt** aus komprimierten Daten wiederhergestellt werden.
- Beispiele
 - Datei-Komprimierung, z.B. ZIP, GZ u.ä.
 - Bildformate TIF und PNG

Verlustbehaftete Datenkompression:

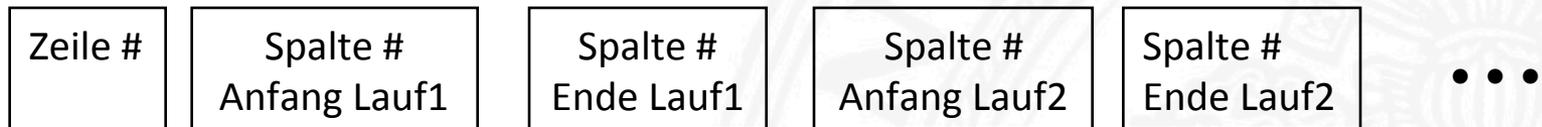
- Originaldaten können **näherungsweise** aus komprimierten Daten wiederhergestellt werden.
- Beispiele
 - JPEG Bilddateien
 - MP3 Audiodateien

Laufängenkodierung

Bilder mit wiederholten Grauwerten können durch Speichern von "Läufen" (runs) mit gleichen Grauwerten komprimiert werden.



Bei S/W Bildern (z.B. Faxdaten) wird eine spezielle Laufängenkodierung angewendet:



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0				■	■	■				■						
1		■	■	■	■	■	■	■		■						
2																
3					■		■		■		■		■	■	■	

Laufängenkodierung:

(0 3 5 9 9)
 (1 1 7 9 9)
 (3 4 4 6 6 8 8 10 10 12 14)

Probabilistische Datenkomprimierung

In einem diskreten Bild wird Information redundant kodiert, wenn

1. die Grauwerte der Pixel nicht gleichverteilt sind, und / oder
2. die Grauwerte von Pixeln korreliert sind

Die Informationstheorie beschreibt Grenzen für minimales (verlustfreies) Kodieren von Information.

Redundanz einer Kodierung von Pixeln mit G Graustufen:

$$r = b - H$$

$$b = \lceil \log_2 G \rceil = \text{Anzahl der Bits pro Pixel}$$

$$H = \sum_{g=0}^{G-1} P(g) \log_2 \frac{1}{P(g)}$$

$H =$ Entropie einer "Pixelquelle"

= mittlere Zahl von Bits, die zum Kodieren dieser Informationsquelle erforderlich sind

Die **Entropie** einer Pixelquelle mit gleichwahrscheinlichen Grauwerten ist gleich der Zahl der Bits zur Kodierung der Grauwerte.

Huffman Kodierung (1)

Ein Huffman-Code erlaubt es, Nachrichten mit minimaler mittlerer Länge, d.h. mit minimaler Redundanz, zu kodieren. "Nachrichten" (hier Grauwerte von Pixeln) werden mit unterschiedlichen Kodewortlängen kodiert.

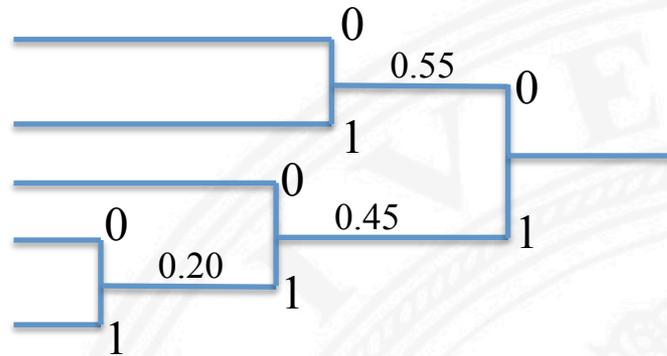
Ablauf:

1. Ordne Nachrichten nach absteigenden Wahrscheinlichkeiten.
 $g^{(1)}$ und $g^{(2)}$ seien die am wenigsten wahrscheinlichen Nachrichten.
2. Teile dem Codewort von $g^{(1)}$ eine 1 und dem Codewort von $g^{(2)}$ eine 0 zu.
3. Fasse $g^{(1)}$ und $g^{(2)}$ durch Addition der Wahrscheinlichkeiten zu einer Nachricht zusammen.
4. Wiederhole Schritte 1 bis 4, bis eine einzige Nachricht übrig bleibt.

Huffman Kodierung (2)

Beispiel:

Nachricht	Wahrscheinlichkeit
$g^{(5)}$	0.30
$g^{(4)}$	0.25
$g^{(3)}$	0.25
$g^{(2)}$	0.10
$g^{(1)}$	0.10



Resultierende Codierungen:

Nachricht	Wahrscheinlichkeit	Codierung
$g^{(5)}$	0.30	00
$g^{(4)}$	0.25	01
$g^{(3)}$	0.25	10
$g^{(2)}$	0.10	110
$g^{(1)}$	0.10	111

Entropie: $H = 2.185$



Mittlere Kodewortlänge: 2.2

Statistische Abhängigkeiten

Ein Bild kann als eine Menge von Zufallsvariablen mit einer multivariaten Verteilung $p(\vec{x}) = p(x_1, x_2, \dots, x_N)$ modelliert werden.

Die exakte Verteilung ist meist unbekannt, aber Korrelationen können häufig bestimmt werden.

Korrelation von zwei Variablen:

$$E[x_i x_j] = c_{ij}$$

$$E[\vec{x} \vec{x}^T] = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \cdots \\ c_{21} & c_{22} & c_{23} & \cdots \\ c_{31} & c_{32} & c_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Korrelationsmatrix

Kovarianz von zwei Variablen:

$$E[(x_i - \mu_i)(x_j - \mu_j)] = v_{ij} \quad \text{mit } \mu_k = \text{Mittelwert von } x_k$$

$$E[(\vec{x} - \vec{\mu})(\vec{x} - \vec{\mu})^T] = \begin{bmatrix} v_{11} & v_{12} & v_{13} & \cdots \\ v_{21} & v_{22} & v_{23} & \cdots \\ v_{31} & v_{32} & v_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Kovarianzmatrix

Anmerkung: Unkorrelierte Variable müssen nicht statistisch unabhängig sein:

$$E[x_i x_j] = 0 \quad \not\Rightarrow \quad p(x_i, x_j) = p(x_i) \cdot p(x_j)$$

Aber: Unkorrelierte Gauß'sche Zufallsvariable sind statistisch unabhängig

Karhunen-Loève Transformation

(auch bekannt als Hauptkomponentenanalyse)

Bestimme unkorrelierte Variable \vec{y} von korrelierten Variablen \vec{x} durch eine lineare Transformation.

$$y = A(\vec{x} - \vec{\mu})$$

$$E[\vec{y} \vec{y}^T] = A E[(\vec{x} - \vec{\mu})(\vec{x} - \vec{\mu})^T] A^T = A V A^T = D \quad D \text{ ist eine Diagonalmatrix}$$

- Es existiert stets eine orthonormale Matrix A , die eine reelle symmetrische Kovarianzmatrix V diagonalisiert.
- A ist die Matrix der Eigenvektoren von V ,
 D ist die Matrix der zugehörigen Eigenwerte.

Rekonstruktion von \vec{x} aus \vec{y} über: $\vec{x} = A^T \vec{y} + \vec{\mu}$

Anmerkung: Betrachtet man \vec{x} als Punkt im n -dimensionalen Euklid'schen Vektorraum, dann definiert A ein rotiertes Koordinatensystem.

Kompression and Rekonstruktion mit der Karhunen-Loève Transformation

Wir nehmen an, dass die Eigenwerte λ_i und die zugehörigen Eigenvektoren von A in absteigender Reihenfolge sortiert sind: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$

$$D = \begin{bmatrix} \lambda_1 & 0 & 0 & \dots \\ 0 & \lambda_2 & 0 & \dots \\ 0 & 0 & \lambda_3 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Eigenvektoren \vec{a} und Eigenwerte λ sind definiert durch $V \vec{a} = \lambda \vec{a}$ und können bestimmt werden, indem man $\det(V - \lambda I) = 0$ löst.

Zum Bestimmen der Eigenwerte von reellen symmetrischen Matrizen gibt es spezielle Verfahren.

\vec{x} kann in einen K -dimensionalen Vektor \vec{y}_K , $K < N$ transformiert werden, mit einer Transformationsmatrix A_K , die nur die ersten K Eigenvektoren von A enthält, korrespondierend zu den K größten Eigenwerten:

$$\vec{y}_K = A_K (\vec{x} - \vec{\mu})$$

Die angenäherte Rekonstruktion \vec{x}' minimiert den mittleren quadratischen Fehler (MSE – mean square error) einer Repräsentation mit K Dimensionen:

$$\vec{x}' = A_K^T y_K + \vec{\mu}$$

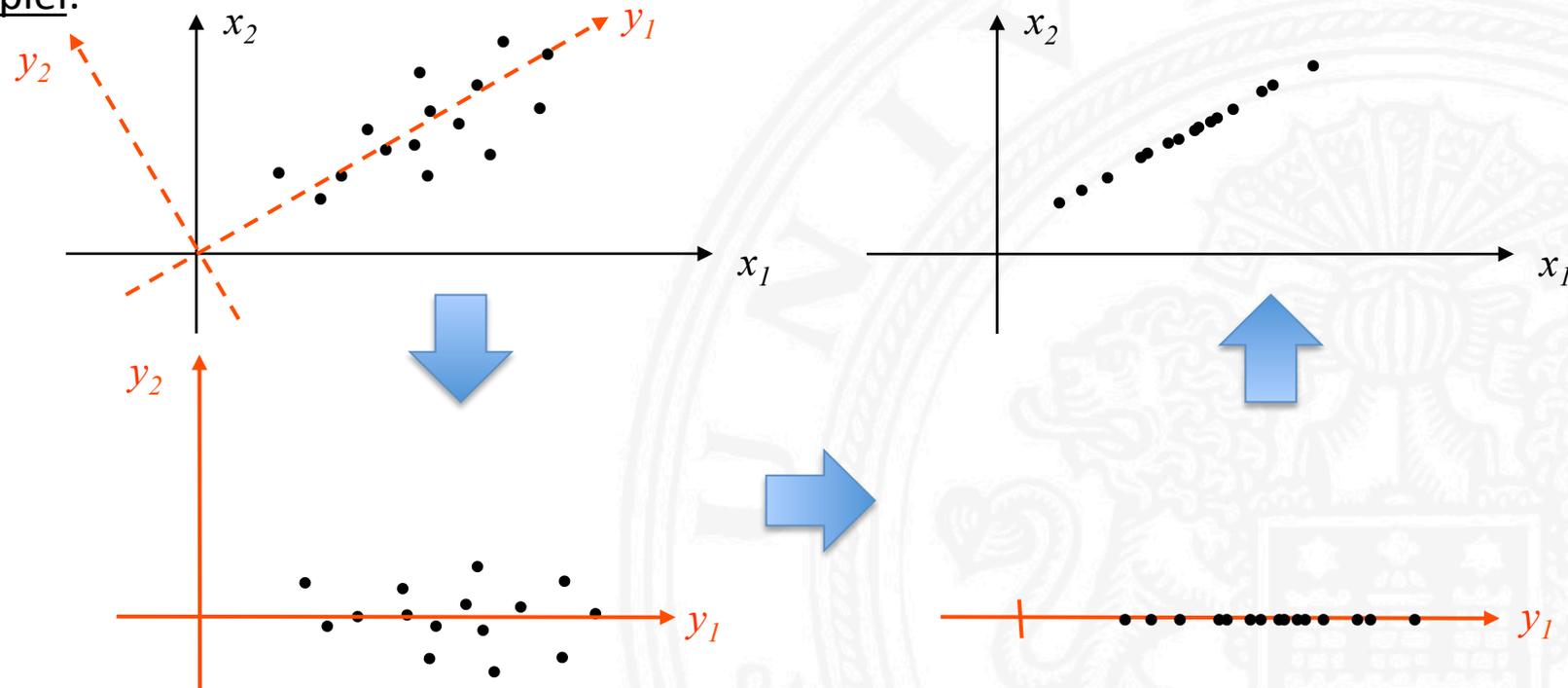
Deshalb kann \vec{y}_K zur (verlustbehafteten) Datenkomprimierung verwendet werden.

Illustration der Dimensionsreduktion

Mit der Karhunen-Loève-Transformation wird Datenkompression erreicht durch

- Wechseln (Drehen) des Koordinatensystems
- Weglassen der am wenigsten informativen Dimensionen

Beispiel:



Eigenfaces (1)

Turk & Pentland: Face Recognition Using Eigenfaces (1991)

Eigenfaces = Eigenvektoren der Kovarianzmatrix von
normalisierten Gesichtsbildern



Beispielbilder des Eigenface-Projekts an der Rice University

Eigenfaces (2)

Die ersten 18 Eigenfaces (= ersten 18 Eigenvektoren) der Kovarianzmatrix von 86 Gesichtsbildern:



Eigenfaces (3)

Originalbilder und Rekonstruktionen aus 50 Eigenfaces:



Diskrete Kosinus-Transformation (DCT)

Die Diskrete Kosinus-Transformation ist in Bildkompressionsverfahren verbreitet, z.B. im JPEG (Joint Photographic Expert Group) Standard.

Definition der DCT:

$$G_{00} = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} g_{mn}$$

$$G_{uv} = \frac{1}{2N^3} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} g_{mn} \cos[(2m+1)u\pi] \cos[(2n+1)v\pi]$$

Inverse DCT:

$$g_{mn} = \frac{1}{N} G_{00} + \frac{1}{2N^3} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} G_{uv} \cos[(2m+1)u\pi] \cos[(2n+1)v\pi]$$

Die DCT berechnet im Effekt die Fourier-Transformation einer Funktion, die bei N durch Ergänzung einer gespiegelten Kopie symmetrisch gemacht wurde.

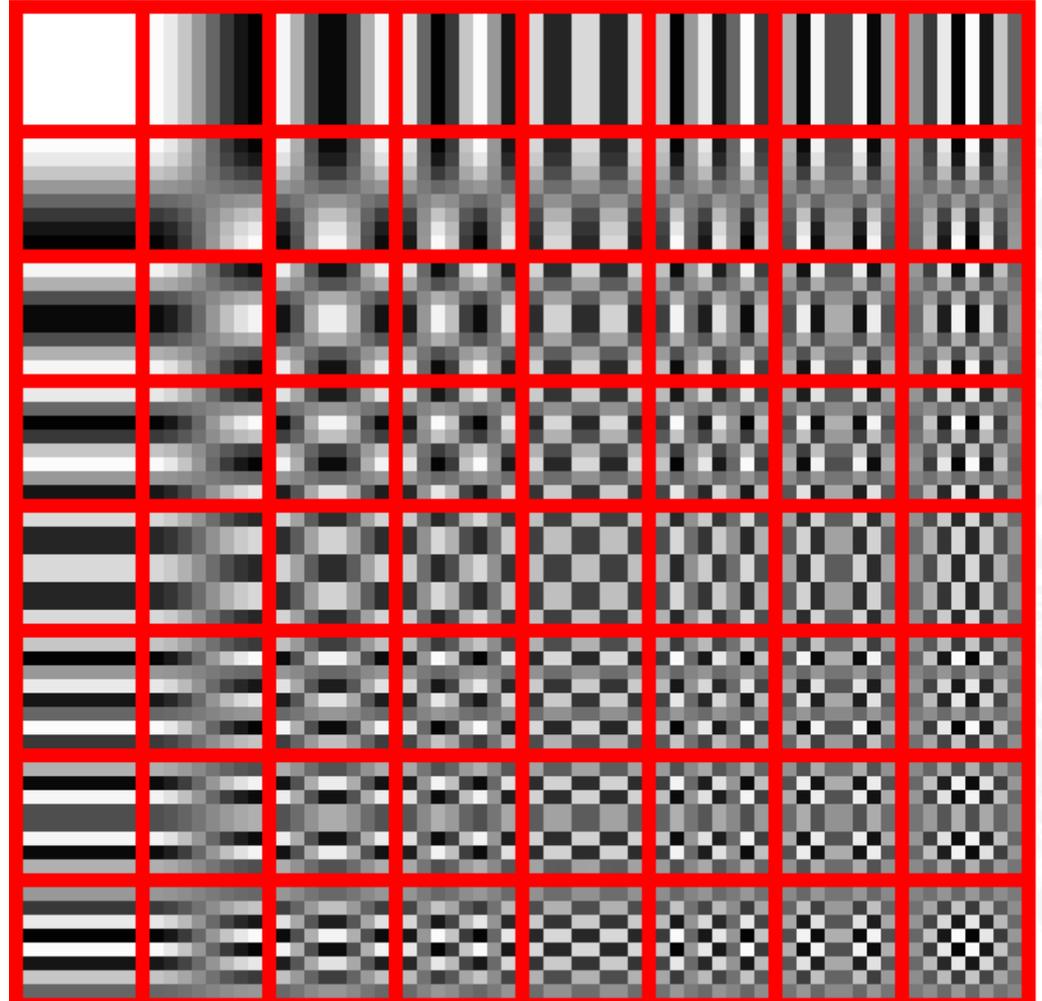


1. Resultat enthält keine Sinusterme
2. Keine Fehler durch periodische Fortsetzung.

Eine Kompression erfolgt erst durch Beschränkung auf niedrigere Frequenzen oder größere Kodierung höherer Frequenzen.

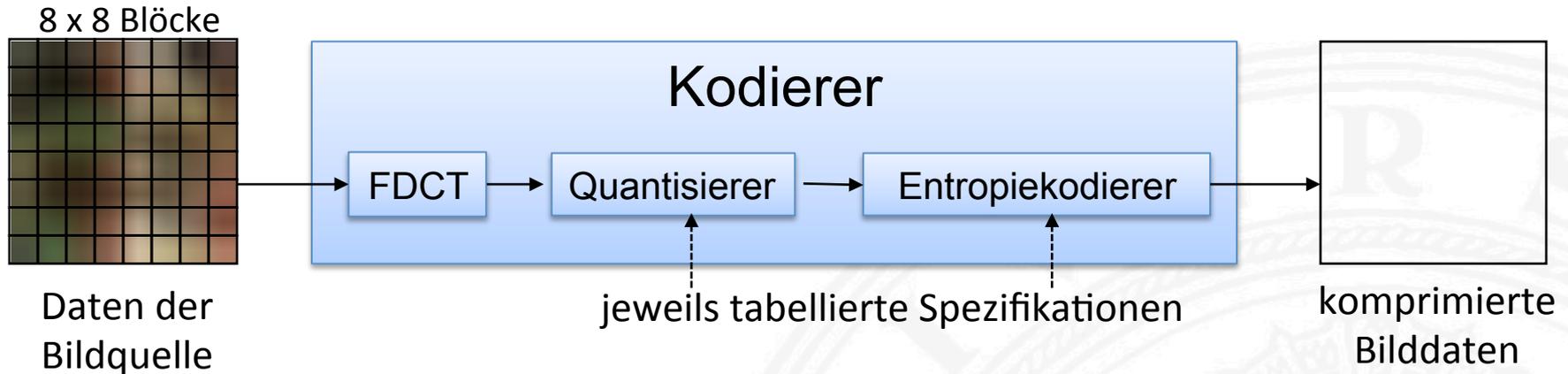
Koeffizienten der 2D-Kosinustransformation

Die Koeffizienten einer 8x8-DCT-Transformation repräsentieren Kosinus-Bestandteile mit Längen von Vielfachen von π :



Prinzip von "Baseline JPEG" (JPG Grundformat)

(Quelle: Gibson et al., Digital Compression for Multimedia, Morgan Kaufmann 98)



Vorgehensweise im Detail:

- von RGB nach YCbCr transformieren, Farbinformation unterabtasten
- Bild in 8 x 8 Blöcke partitionieren, von links nach rechts, von oben nach unten
- Diskrete Kosinus-Transformation (DCT) von jedem Block berechnen
- Koeffizienten der DCT entsprechend psychovisuellen Tabellen quantisieren
- DCT Koeffizienten im Zickzack anordnen
- Lauflängenkodierung des Bitstroms aller Koeffizienten eines Blocks
- Huffman-Kodierung für Bitmuster eines Blocks

YCbCr Farbmodell für JPEG

Menschliche Augen sind gegenüber Luminanzschwankungen (Helligkeit) empfindlicher als gegenüber Chrominanzschwankungen (Farbe). YCbCr Farbkodierung verwendet für Chrominanz weniger Bits als für Luminanz.

CCIR-601 Schema:

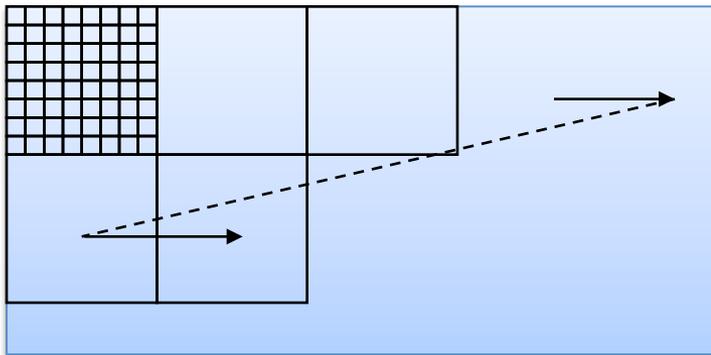
$$\begin{array}{llllll} Y = & 0.299 R + & 0.587 G + & 0.144 B & \text{"Luminanz"} \\ Cb = & 0.169 R - & 0.331 G + & 0.500 B & \text{"Blaugehalt"} \\ Cr = & 0.500 R - & 0.419 G - & 0.081 B & \text{"Rotgehalt"} \end{array}$$

In JPEG:

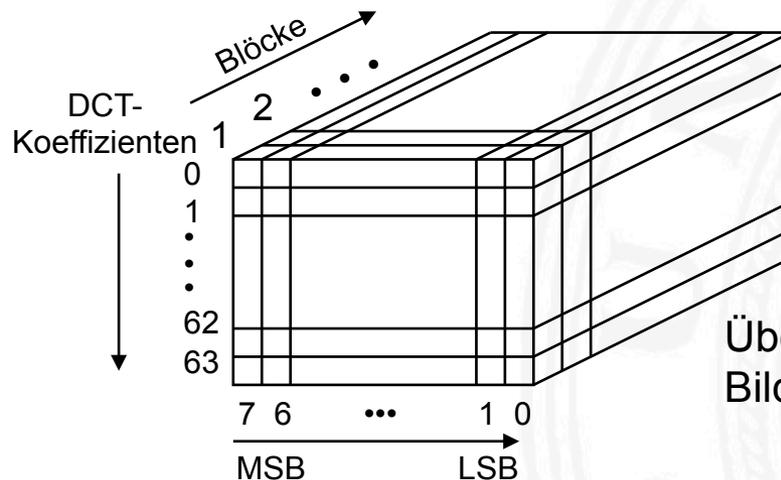
- 1 Cb, 1 Cr und 4 Y Werte für jedes 2 x 2 Teilbild

➡ 6 anstelle von 12 Werten!

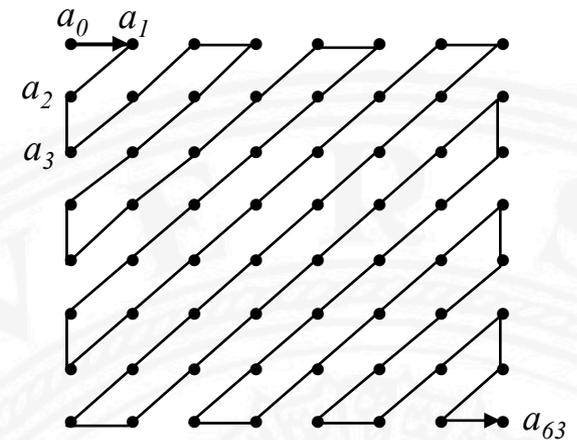
Illustrationen zur Grundform von JPEG



Zerlegen eines Bildes in Blöcke



Übertragungsreihenfolge von Bildblöcken



Reihenfolge der DCT-Koeffizienten zur effizienten Lauflängenkodierung

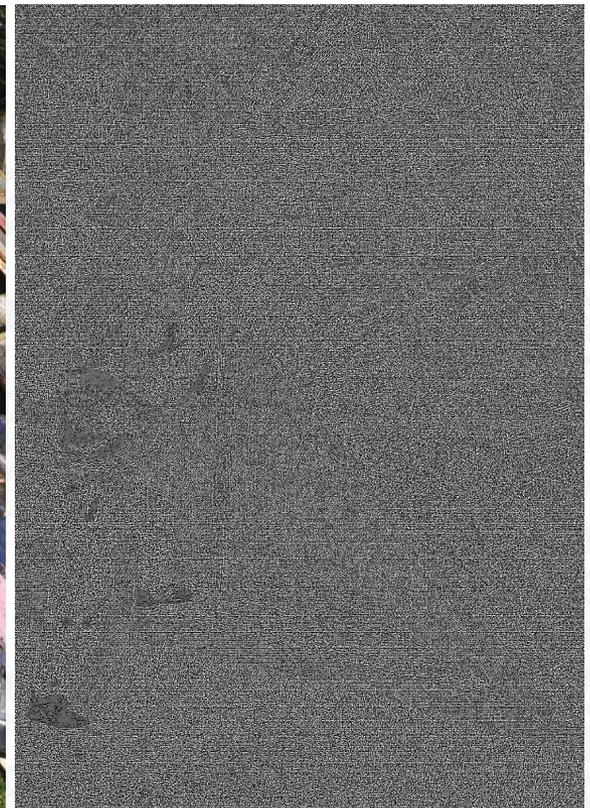
JPEG-komprimiertes Bild



Original mit
5.8 MB

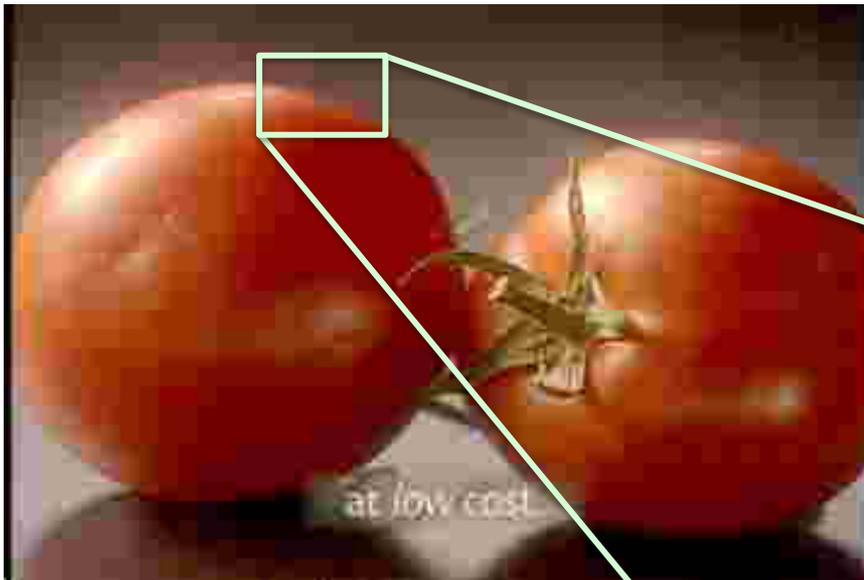


JPEG-komprimiert auf
450 KB



Differenzbild
Standardabweichung der
Luminanzwerte: 1.44

Probleme mit der Blockstruktur von JPEG



JPEG-Kodierung mit einer Kompressionsrate von 1:70

Blockgrenzen sind deutlich erkennbar!



Progressives Kodieren

Progressive Kodierung ermöglicht es, zuerst eine Grobversion eines Bildes und dann fortschreitende Verfeinerungen zu übertragen (angenehm für schnelle Bildinspektion im Internet).

- Spektrale Auswahl

1. Übertragung: DCT-Koeffizienten $a_0 \dots a_{k1}$

2. Übertragung: DCT-Koeffizienten $a_{k1} \dots a_{k2}$

etc.

niederfrequente
Koeffizienten zuerst

- Auswahl von signifikanten Bits

1. Übertragung: Bits $7 \dots n_1$

2. Übertragung: Bits $n_1+1 \dots n_2$

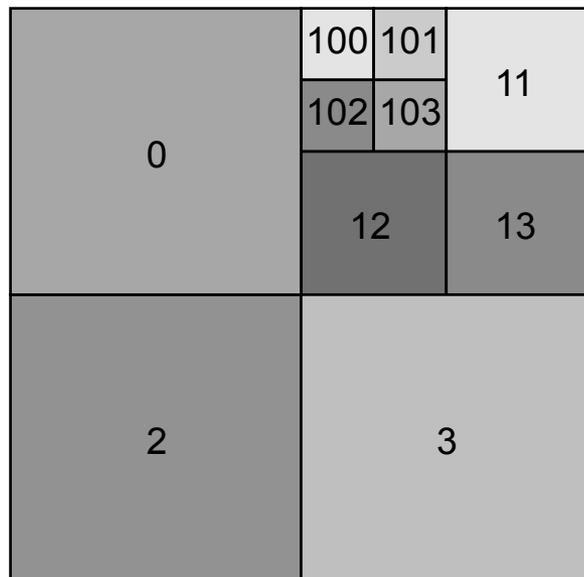
etc.

signifikante Bits
zuerst

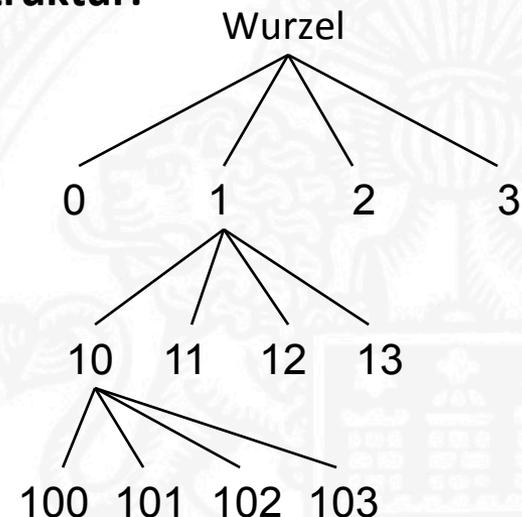
Bildrepräsentation als Quadbaum

Eigenschaften:

- Jeder Knoten repräsentiert eine rechteckige Bildfläche, z.B. durch dessen Mittelwert
- Jeder Knoten hat 4 Kinder, mit Ausnahme der Blattknoten
- Kinder eines Knoten repräsentieren gleichgroße Teilrechtecke
- Knoten können bei Bedarf weiter verfeinert werden
- Teilrechtecke werden entsprechend der Zerlegungsschritte adressiert



Baumstruktur:



Bildkompression mit einem Quadbaum

Ein vollständiger Quadbaum repräsentiert ein Bild mit $N = 2^K \times 2^K$ Pixeln durch $1 + 4 + 16 + \dots + 2^{2K} \approx 1.33 N$ Knoten.

Ein Bild kann komprimiert werden,

- indem jeder Kindknoten nur die Differenz zum Elternknoten speichert
- indem Teilbäume mit (annähernd) gleichen Werten weggelassen werden

Quadbaum-Bildkompression unterstützt progressive Bildübertragung:

- Bilder werden mit steigender Quadbaumtiefe übertragen, d.h. Bilder werden fortschreitend verfeinert
- Zwischenrepräsentationen bieten interessante Näherungsinformationen, z.B. zum Abruf von ähnlichen Bildern

