

## Inhaltsbasierte Bildsuche

# J.P.E.G = Joint Photographic Expert Group

Informatica Feminale  
Universität Bremen, Aug. 2005  
Maja Temerinac  
Albert-Ludwigs-Universität Freiburg

## J.P.E.G. Standard

- Standard zur Kompression von digitalen Standbildern
- Seit 1993 unter dem Namen ISO 10918 als Standard definiert
- Zusammenarbeit zwischen CCITT (International Telegraph and Telephone Consultative Committee) und ISO/IEC (International Organization for Standardization/International Electrotechnical Commission).

21/08/2005

JPEG, Maja Temerinac

2

## J.P.E.G. Eigenschaften

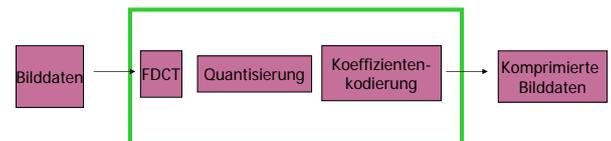
- Ziele der Bildkompression
  - akzeptable Komplexität
  - Unabhängigkeit von der Bildbeschaffenheit
  - vom Benutzer beeinflussbare Bildqualität/Kompression
- 2 Arten der Kompression:
  - verlustfreie
  - **verlustbehaftete**

21/08/2005

JPEG, Maja Temerinac

3

## J.P.E.G. System



21/08/2005

JPEG, Maja Temerinac

4

## JPEG Verschlüsselungsschritte

- Konvertierung des Bildes in ein geeignetes Farbmodell
- Diskrete Cosinus Transformation
- Quantisierung der DCT-Koeffizienten
- Kodierung der Koeffizienten

21/08/2005

JPEG, Maja Temerinac

5

## Farbmodell

- menschliches Auge erkennt, geringe Helligkeitsunterschiede besser als Farbveränderungen
- Helligkeit-Farbigkeit-Modell YUV ist für die Kompression besonders gut geeignet
- $Y = 0,299 R + 0,587 G + 0,114 B$
- $V = R - Y$
- $U = B - Y$
- Grün nimmt der Mensch besonders gut wahr!

21/08/2005

JPEG, Maja Temerinac

6

## Diskrete Cosinus Transformation

- **64 Basisfunktionen**, die aus 8x8 Pixeln bestehen
- Nimmt man jede dieser Funktionen als 64-stelligen Vektor, bilden diese 64 Vektoren eine **Orthonormalbasis** bezüglich eines Vektorraums
- Wird der zu kodierende Block nach einer Basistransformation durch die DCT bezüglich der eben beschriebenen Basis dargestellt, erhält man **64 Koeffizienten**, die angeben, wie stark die einzelnen Basisfunktionen an dem zu kodierenden Block beteiligt sind
- Diese Koeffizienten heißen DCT-Koeffizienten.

21/08/2005

JPEG, Maja Temerinac

7

## Berechnung der DCT Koeffizienten

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

$$\text{mit: } \left. \begin{array}{l} C(u), C(v) = \frac{1}{\sqrt{2}}, \text{ für } u, v = 0 \\ C(u), C(v) = 1, \text{ sonst} \end{array} \right\} \text{ Normierungskonstanten}$$

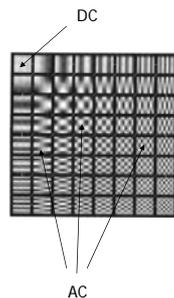
21/08/2005

JPEG, Maja Temerinac

8

## Eigenschaften DCT

- Betrachtet man die Basisfunktionen der 2-dimensionalen DCT, die eigentlich ein Produkt zweier eindimensionaler DCT ist, stellt man fest, dass sich in der linken oberen Ecke die einzige konstante Basisfunktion befindet
- Der Koeffizient dieser Funktion, deren Frequenz in beiden Dimensionen Null ist, heißt DC-Koeffizient (direct-current terms), während die übrigen 63 Koeffizienten AC-Koeffizienten (alternating-current terms) heißen
- DCT ist **verlustfrei!**



21/08/2005

JPEG, Maja Temerinac

9

## Quantisierung der DCT-Koeffizienten

- Im eigentlich verlustbehafteten Prozeß der Quantisierung werden die 64 DCT-Koeffizienten nun dividiert und das Ergebnis auf eine ganze Zahl gerundet.
- Die Division geschieht anhand einer Quantisierungstabelle, die die Farb- und Helligkeitsempfindlichkeit des Auges berücksichtigt
- Das menschliche Auge reagiert auf niedrige Frequenzen viel empfindlicher als auf hohe.
- Die Koeffizienten höherer Frequenzen werden deshalb durch höhere Zahlen geteilt als die der niedrigeren -> Verkleinerung des Wertebereichs

21/08/2005

JPEG, Maja Temerinac

10

## Berechnung der Quantisierung

- Viele Koeffizienten werden zu Null!

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

- Q = Quantisierungstabelle
- F = transformiertes Bild
- Q und F sind 8 x 8 Matrizen
- Quantisierungstabelle muss bei der Codierung mitgeliefert werden

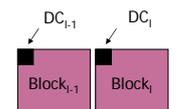
21/08/2005

JPEG, Maja Temerinac

11

## DCT Koeffizienten

- Nach der Quantisierung werden die DC- und AC Koeffizienten getrennt voneinander betrachtet.
- Starke Korrelation zwischen benachbarten 8 x 8 Blöcken
- DC Koeffizienten werden relativ zum vorhergehenden Block abgespeichert



$$\text{DIFF} = DC_i - DC_{i-1}$$

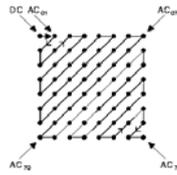
21/08/2005

JPEG, Maja Temerinac

12

## DCT Koeffizienten (2)

- Dieser Wert wird danach zusammen mit den AC-Koeffizienten in eine Reihe gebracht, wobei sich die AC-Koeffizienten in einem Zick-Zack-Kurs anschließen. So erreicht man, dass man so gleiche Werte eines zusammenhängenden Abschnitts hintereinander bekommt, was für die anschließende Codierung vor allem bei den oft auftretenden Nullen von Vorteil sein wird



21/08/2005

JPEG, Maja Temerinac

13

## Koeffizientenkodierung

- Im letzten Kodierschritt werden dann die DCT Koeffizienten kodiert, wodurch noch eine weitere Kompression (ohne Verluste) erreicht wird
- Ausnutzen der charakteristischen Verteilung der DCT-Basisfunktionen
- Huffmancodierung
- Tabelle für Huffman-Codierung muss mitgeliefert werden

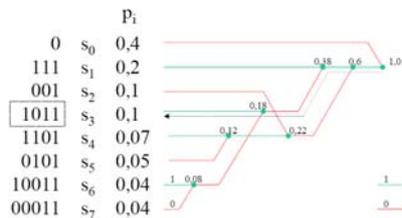
21/08/2005

JPEG, Maja Temerinac

14

## Huffman-Codierung

- Fasse in jeder Stufe die zwei kleinsten Wahrscheinlichkeiten zusammen und führe zur Unterscheidung eine Bitstelle ein.



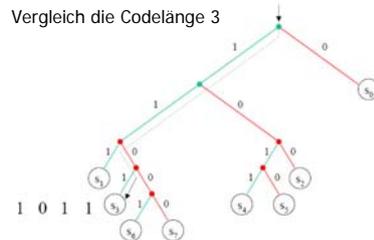
21/08/2005

JPEG, Maja Temerinac

15

## Huffman-Decodierung

- Der Huffman-Code minimiert die mittlere Codelänge
- Ein Blockcode von 8 Zeichen hätte zum Vergleich die Codelänge 3



21/08/2005

JPEG, Maja Temerinac

16

## Übung

### Aufgabe 1:

- Berechne die zweidimensionale DCT für die folgende Matrix:

$$\begin{bmatrix} 35 & 116 & 46 & 15 & 12 & 11 & 8 & 6 \\ 19 & 86 & 75 & 14 & 12 & 9 & 11 & 55 \\ 24 & 52 & 102 & 17 & 7 & 9 & 13 & 68 \\ 57 & 66 & 141 & 49 & 6 & 10 & 52 & 127 \\ 18 & 28 & 123 & 42 & 7 & 37 & 117 & 112 \\ 8 & 9 & 80 & 81 & 25 & 109 & 117 & 84 \\ 8 & 4 & 27 & 84 & 98 & 129 & 88 & 89 \\ 6 & 5 & 14 & 67 & 110 & 83 & 83 & 140 \end{bmatrix}$$

21/08/2005

JPEG, Maja Temerinac

17

## Übung (2)

- Aufgabe 2:
  - Berechne die DC-Koeffizienten eines Bildes und speichere sie in einer Matrix. Skaliere die Matrix, indem Du sie durch den maximalen DC-Wert teilst. Stelle, die so entstandene Matrix, grafisch da!
  - Berechne die AC-Koeffizienten: AC01 und AC10. Welche Linien des Bildes stellen diese Koeffizienten grafisch da?
  - Hinweis:** Bilder mit einem Bus sind hierfür besonders geeignet!

21/08/2005

JPEG, Maja Temerinac

18

## Inhaltsbasierte Bildsuche Segmentieren

Informatica Feminale  
Universität Bremen, Aug. 2005  
Maja Temerinac  
Albert-Ludwigs-Universität Freiburg

## Was ist Segmentieren?

- Nicht überwachtes Lernproblem:
  - *Finde eine Struktur in einem Haufen von unsortierten Daten*
- Ein **Segment** ist eine Ansammlung von Datenpunkten, die möglichst gleich zu einander und unterschiedlich zu den Datenpunkten eines anderen Segmentes sind

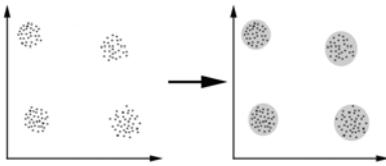
21/08/2005

JPEG, Maja Temerinac

20

## Segmentieren graphisch

- Man kann hier die Daten sehr leicht in 4 Segmente unterteilen:



- Das Hauptmerkmal der Unterteilung ist die **Distanz** zwischen den Datenpunkten. Punkte mit kleiner Distanz sind ähnlich; Punkte mit großer Distanz sind unterschiedlich.

21/08/2005

JPEG, Maja Temerinac

21

## Ziel der Bildsegmentierung

- Objekte im Bild zu finden
- Hintergrund vom Objekt zu trennen
- Homogene Gebiete in einem Bild zu orten



21/08/2005

JPEG, Maja Temerinac

22

## Überblick-Segmentieren

- Zwei Arten von Segmentierungs-Algorithmen:
- **partitionierende Algorithmen (P)**
- **hierarchische Algorithmen (H)**:
  - Anhäufende Verfahren
  - Teilende Verfahren

21/08/2005

JPEG, Maja Temerinac

23

## Algorithmen

- Segmentierungs- oder Clustering-Algorithmen:
  - K-means (1967) (P)
  - Single Linkage (1967) (H)
  - EM (Expectation-Maximization) (1972) (P)
  - PAM (Partitioning around medoids) (1987) (H)
- Sollten möglichst schnell und „gut“ sein!

21/08/2005

JPEG, Maja Temerinac

24

## K-means

- Beim k-means Algorithmus ist eine gewünschte Anzahl  $k$  von Clustern und eine Funktion zur Bestimmung des Mittelpunktes eines Clusters bekannt.
- Algorithmus:

**(Initialisierung)** (Zufällige) Auswahl von  $k$  Clusterzentren  
**(Zuordnung)** Jedes Objekt wird dem ihm am nächsten liegenden Clusterzentrum zugeordnet  
**(Neuberechnung)** Es werden für jedes Cluster die Clusterzentren neu berechnet  
**(Wiederholung)** Falls sich nun die Zuordnung der Objekte ändert, weiter mit Schritt 2, ansonsten Abbruch

21/08/2005

JPEG, Maja Temerinac

25

## Eigenheiten von K-means

- Muss nicht konvergieren
- Ein Cluster kann leer bleiben  
-> **Abhilfe:** k-means neu starten!
- Trotzdem: „billig“ und „gut“

21/08/2005

JPEG, Maja Temerinac

26

## EM-Algorithmus

- Basiert auf der Idee von k-means
- Voraussetzung: Alle Objekte können als Vektoren der Dimension  $n$  dargestellt werden!
- $n$  kann beliebig gewählt werden
- Es muss eine Funktion existieren, um den Mittelwert zweier solcher Vektoren zu berechnen

21/08/2005

JPEG, Maja Temerinac

27

## EM-Algorithmus (2)

- Der Clusteralgorithmus selbst durchläuft zwei Schritte:
  - 1) **Estimation**  
Bestimme für jedes Objekt nach einer Wahrscheinlichkeitsverteilung Deiner Wahl (beliebig ist hier z.B. die Normalverteilung), mit welcher Wahrscheinlichkeit es zu jedem der Cluster gehört und speichere diese Wahrscheinlichkeiten für alle Objekte und Cluster.
  - 2) **Maximization**  
Bestimme anhand der ermittelten Zuordnungswahrscheinlichkeiten die Parameter neu, die die Cluster bestimmen (z.B. die Mittelwertvektoren).

21/08/2005

JPEG, Maja Temerinac

28

## EM-Algorithmus (3)

- Die Iteration wird abgebrochen, wenn entweder die Änderung der Likelihood der Daten gegeben die Cluster unter einen vorgegebenen Schwellwert sinkt, oder die ebenfalls vorgegebene maximale Anzahl von Iterationen erreicht ist.
- Im Gegensatz zu k-means "**weiche**" Clusterzuordnung
- Mit einer gewissen Wahrscheinlichkeit gehört jedes Objekt zu jedem Cluster. Jedes Objekt beeinflusst so die Parameter jedes Clusters entsprechend dieser Wahrscheinlichkeit.
- Der Erfolg des Algorithmus hängt stark von der gewählten Wahrscheinlichkeitsverteilung ab.

21/08/2005

JPEG, Maja Temerinac

29

## Übung

- Aufgabe 1:
  - Wende die Funktion *kmeans* auf ein RGB Bild an. Färbe die resultierenden Segmente mit unterschiedlichen Farben ein und zeige das so segmentierte Bild an.
  - **Anleitung:** Die Bildmatrix mit Größe  $n \times m \times 3$  muss erst in eine  $(n * m) \times 3$  Matrix umgewandelt werden. Diese ist die Eingabe für die *kmeans* Funktion. Danach bekommt man einen Vektor der für jeden Pixel angibt, zu welchem Segment er gehört. Dieser Vektor muss dann noch in eine Matrix umgewandelt werden. Durch eine colormap können die Farben für das Bild gewählt werden.

21/08/2005

JPEG, Maja Temerinac

30