

Grundlagen der Bildverarbeitung WS 2014/15

Praxis-Aufgabe 4

"Bildkompression: Grundlagen der JPEG-Kompression"

Abgabe: 15.12.2014, 9 s.t.

Die JPEG-Kompression erfolgt durch mehrere Schritte:

- Farbraumtransformation von RGB nach YCbCr.
- Tiefpassfilterung und Unterabtastung der Farbabweichungssignale Cb und Cr.
- Einteilung in $m \times m$ -Blöcke und DCT
- Quantisierung der DCT-Koeffizienten.
- Umsortierung und Entropie-Kodierung (Huffman).

In dieser Praxisaufgabe sollen einige wichtige Schritte daraus nachimplementiert werden, um die Auswirkungen der Kompression auf ein **Grauwertbild** zu zeigen: Das zu kodierende Bild soll zunächst in gleich große Blöcke zerlegt werden, die dann mittels der diskreten Kosinustransformation (DCT) in den Frequenzraum überführt werden. Die Repräsentation im Frequenzraum macht dann eine vergleichsweise hohe Kompressionsrate des Bildes bei geringem Datenverlust möglich.

Schreiben Sie eine Funktion oder ein Script zur Durchführung folgender Schritte:

1. Lesen Sie ein Bild interaktiv (Dialog) ein und geben Sie das Bild in einem Fenster aus.
2. Unterteilen Sie das Bild in gleich große Blöcke der Größe $M \times M$ ($M = 8$ und $M = 16$)
3. Überführen Sie die einzelnen Blöcke mithilfe der DCT in den Frequenzraum.
4. Führen Sie nun eine Quantisierung der unterschiedlichen Frequenzen durch. Dazu können sie die Quantisierungsmatrizen „quantm.mat“ verwenden.
5. Anschließend sollen die Daten umsortiert (zig-zag scan) und Run Length kodiert werden. Auf die Huffman-Kodierung kann an dieser Stelle verzichtet werden. Sie können die Funktionen `rle_enc()` und `zigzag()` verwenden.
6. Implementieren Sie auch die Rückwärts-Schritte, die nötig sind, um die verlustbehafteten Daten zurück in den Ortsraum zu transferieren.
7. Visualisieren Sie das Ergebnis für die Blockgröße 8 und 16 mit den unterschiedlichen Quantisierungsmatrizen und geben sie jeweils die Kompressionsrate an.

Versuchen Sie, die Aufgabe mit möglichst wenig Schleifen zu lösen (maximal zwei Verschachtelungen). Die komprimierten Bilddaten müssen nicht abgespeichert werden können. Für die Kosinustransformation können die Matlab-Funktionen `dct2` und `idct2` verwendet werden. Sie können außerdem die mitgelieferten Funktionen `rle_dec` und `rle_enc` für die Run-Length-Kodierung und `zigzag` bzw. `zagzig` für Umsortierung verwenden.

Achten Sie bei eigenen Bildern darauf, dass die Höhe und Breite Vielfache der Blockgröße sind, da wir hier auf eine „Randbetrachtung“ verzichten.

Testfälle:

Benutzen Sie das Test-Bild „hockey.bmp“ und nutzen Sie eigene Grauwertbilder. Inwiefern unterscheiden sich die Ergebnisse mit unterschiedlichen Quantisierungsmatrizen sowie mit unterschiedlichen Bildern visuell und warum?

