

Die Begriffe *codieren* und *decodieren* stehen hier synonym für *komprimieren* (*deflate*) und *dekomprimieren* (*inflate*).

1. Allgemeines

- Du kannst die beiden Grundformen der Datenkompression (*verlustfrei* und *verlustbehaftet*) aufzählen.
- Du kannst angeben, für welche Anwendungsbereiche die oben genannten Grundtypen geeignet bzw. nicht geeignet sind.

2. Lauflängencodierung (*Run length encoding*):

- Du kannst eine Zeichenkette mittels der Lauflängencodierung codieren und decodieren. Insbesondere kannst du damit Umgehen, wenn Steuerzeichen in den Daten vorkommen.
- Du kannst den Kompressionsgrad (=Kompressionsfaktor) an konkreten Beispielen bestimmen.
- Du kannst mindestens zwei Anwendungsbereiche dieses Verfahrens nennen.

3. Huffman-Codierung (*Huffman encoding*)

- Du kannst eine Zeichenkette nach Huffman codieren. Dazu gehört die Analyse der Zeichenhäufigkeiten und der Aufbau des Codebaums.
- Du kannst eine nach Huffman codierte Zeichenkette mit Hilfe des entsprechenden Codebaums decodieren.
- Du kannst (nicht) präfixfreie Codes erkennen.
- Du kannst den Kompressionsgrad an konkreten Beispielen bestimmen.
- Du kannst je mindestens ein Vor- und ein Nachteil des Verfahrens aufzählen.

4. LZW-Codierung (*Lempel Ziv Welch-Codierung*)

- Du kannst eine Zeichenkette und eine gegebene Basistabelle mit dem LZW-Verfahren codieren. Dazu gehören der Aufbau der erweiterten Tabelle und die Bestimmung der Codefolge bzw. der Codebits.
- Du kannst einen LZW-Code decodieren. Dazu gehört auch die Behandlung des Spezialfalls, wenn es zu einem Code noch keinen Eintrag in der erweiterten Tabelle gibt.
- Du kannst den Kompressionsgrad an konkreten Beispielen bestimmen.
- Du kannst je ein Vor- und ein Nachteil des Verfahrens aufzählen.

5. JPEG-Format

- Du kannst das Akronym JPEG ausschreiben.
- Du kennst die folgenden Verfahrensschritte in der richtigen Reihenfolge grob wiedergeben, verstehst sie und kannst eine Begründung dafür angeben.
 - (a) Trennung des Bildes in seine Luminanzkomponente (Helligkeit) und die Cb- und Cr-Chrominanzkomponente (b=blau/gelb und r=rot/türkis).
Grund: siehe Schritt (b)
 - (b) In den Chrominanzkomponenten wird jeder 2×2 -Teilblock durch Mittelwertbildung auf ein Pixel reduziert.
Grund: Wir nehmen Unterschiede in der Helligkeit stärker wahr in der Chrominanz.
 - (c) Luminanzblöcke und Chrominanzblöcke werden in 8×8 Teilblöcke zerlegt. Anschliessend von jedem Luminanz bzw. Chrominanzwert (von 0 bis 255) die Zahl 128 subtrahiert (von -128 bis $+127$).
Grund: technisches „Detail“; wird nicht verlangt
 - (d) Durch eine diskrete Cosinustransformation werden die 64 Helligkeitswerte in 64 Frequenzwerte zerlegt.
Grund: Siehe Schritt (e).
 - (e) Die Frequenzwerte werden durch die entsprechenden Elemente einer 8×8 -Quantisierungsmatrix gedämpft.
Grund: Wir reagieren weniger sensibel auf hohe Helligkeitsfrequenzen und können diese daher ohne grosse Qualitätseinbusse stärker dämpfen.
 - (f) Linearisieren der quantisierten Matrix im Zickzack-Schema.
Grund: Vorbereitung für die Lauflängencodierung
 - (g) Lauflängencodierung des Bytestroms.
Grund: Kompression (verlustfrei)
 - (h) Huffman-Codierung des Bytestroms.
Grund: zusätzliche Kompression (verlustfrei)