

Codierung multimedialer Daten

Aufgaben zum nächsten Mal (AZNM 10)

Johann-Markus Batke

2023-06-07

1 Quellencodierung

Machen Sie sich mit folgenden Begriffen und Definitionen vertraut:

- <https://vfhcmd.eduloop.de/loop/Transformationscodierung> ff.
- <https://vfhcmd.eduloop.de/loop/JPEG>, insbesondere
 - <https://vfhcmd.eduloop.de/loop/Blockbildung>
 - https://vfhcmd.eduloop.de/loop/Level_Shift
 - <https://vfhcmd.eduloop.de/loop/Transformation>
 - <https://vfhcmd.eduloop.de/loop/Quantisierung/Dequantisierung>
 - <https://vfhcmd.eduloop.de/loop/Zig-Zag-Sequence>
 - <https://vfhcmd.eduloop.de/loop/Huffman-Codierung>

2 Aufgaben

2.1 Walsh-Transformation

Lösen Sie die Aufgabe in https://vfhcmd.eduloop.de/loop/Aufgabe:_2D-Walsh-Transformation.

2.2 JPEG-Codierung (26 Punkte)

2.2.1 BSB

Skizzieren Sie das Blockschaltbild des Verfahrens JFIF! Verwenden Sie dazu die Blöcke 8x8-Blockbildung, 2-DCT, Quantisierung, Differenzcodierung, RLE-Zigzag, Huffman, Packen.

2.2.2 DCT-Matrix

Stellen Sie die Matrix für den Transformationskern der 2-DCT für $N = 5$ auf, geben Sie die Matrixelemente in ihrer analytischen Form an!

Hinweis: Elemente der 2-DCT-Matrix sind

$$a(i,j) = \begin{cases} \frac{1}{\sqrt{N}} & \text{für } i = 0 \quad 0 \leq j \leq N-1 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{\pi}{2N}(2j+1)i\right) & \text{für } 1 \leq i \leq N-1 \quad 0 \leq j \leq N-1 \end{cases} \quad (1)$$

2.2.3 Chrominanzwerte

Gegeben sei ein 2×2 Bildelement mit den Chrominanzwerten

$$\mathbf{Y}_C = \begin{pmatrix} 3 & 4 \\ 2 & 1 \end{pmatrix} \quad (2)$$

Berechnen Sie die 2-DCT dieser Matrix mit Hilfe der Matrix

$$\begin{pmatrix} 0.707 & 0.707 \\ 0.707 & -0.707 \end{pmatrix}$$

2.2.4 Quantisierung

Quantisieren Sie die 2-DCT der Chrominanz-Matrix. Verwenden Sie die Quantisierungsmatrix

$$\mathbf{Y}_Q = \frac{1}{10} \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix}. \quad (3)$$

2.2.5 Zickzack

Für eine 2-DCT mit $N = 5$ haben sich nach der Quantisierung folgende Werte ergeben:

$$\begin{pmatrix} -3 & 2 & 1 & 0 & -1 \\ 4 & 2 & 3 & -3 & -2 \\ -1 & -1 & -3 & -2 & -2 \\ 1 & -3 & -2 & -2 & -4 \\ -3 & 1 & 4 & -4 & 1 \end{pmatrix}$$

Führen Sie einen Zickzack-Scan durch und stellen Sie die Matrix lauflängencodiert dar. Die Lauflängencodierung soll nach dem PackedBits-Verfahren (vgl. Tabelle 1) erfolgen.

Tabelle 1: Header-Daten für das PackedBits-Verfahren

<header>	<data>
$n = 0 \dots 127$	$(1 + n)$ Bytes Rohdaten
$n = -1 \dots -127$	$(1 - n) \times 1$ Byte
$n = -128$	no operation