

Übungen zur Vorlesung Wissenschaftliches Rechnen

C++ Mini-Projekt Nummer 2

Aufgabe:

Schreiben Sie ein Bildkompressionsprogramm mit diskreter Kosinustransformation (DCT II).

Das Programm soll Folgendes tun:

1. Eines der beiden Bilder im `pgm`-Format auf der Webseite als Eingabe nehmen.
2. Die ersten fünf Zeilen lesen: Zeile 3 ist die Anzahl N der Pixel pro Bildzeile, Zeile 4 ist die Anzahl M der Bildzeilen des Bildes. Diese fünf Zeilen werden in die Ausgabedatei (z.B. `leberblume-k.pbm`) geschrieben.
3. Die nächsten N Zeilen der Eingabedatei sind die Grauwerte der N Pixel der ersten Bildzeile, die nächsten N Zeilen der Eingabedatei sind die Grauwerte der N Pixel der zweiten Bildzeile, usw. (0=schwarz, 255=weiß).
4. Für jede Bildzeile lesen wir deren Grauwerte v_0, v_1, \dots, v_{N-1} in einen Vektor v der Länge N ein.
5. Es wird die DCT II von v berechnet und das Ergebnis in einen Vektor \hat{v} geschrieben.
6. Die letzten 80% der Einträge von \hat{v} werden auf 0 gesetzt.
7. Es wird die inverse DCT II (also $2/N \cdot$ DCT III) von \hat{v} berechnet. Das liefert einen neuen Vektor \tilde{v} .
8. Die Einträge von \tilde{v} werden zeilenweise in die Ausgabedatei (z.B. `leberblume-k.pgm`) geschrieben.

Auch hier müssen Sie sich nicht um fehlerhafte Eingaben kümmern: der Header ist immer genau fünf Zeilen lang, alle Zeilen nach der fünften sind vom Typ `Integer`, usw. (Im Allgemeinen können `pgm`-Dateien leicht anders aussehen, siehe wikipedia).

Beispielaufruf: Wenn das Programm `kompress` heißt, und z.B. das Bild `leberblume.pgm` im aktuellen Verzeichnis liegt, dann soll ein Aufruf von der Kommandozeile z.B. so aussehen:

```
$ ./kompress
```

```
Name des Eingabebilds: [Hier gebe ich ein: leberblume.pgm]
```

```
Name des Ausgabebilds: [Hier gebe ich ein: leberblume-k.pgm]
```

Nun soll es im aktuellen Verzeichnis eine Datei namens `leberblume-k.pgm` geben, die ich als `pgm` in einem Bildbetrachter ansehen kann.