

## Bildanalyse und Bildverstehen (SoSe 2003)

### Lösungen zu Übungsblatt 1

#### Aufgabe 1

- (a)  $24\ 000\ \mu\text{m} : 30\ \mu\text{m} = 800$  Spalten  
 $36\ 000\ \mu\text{m} : 30\ \mu\text{m} = 1200$  Zeilen  
(oder, bei Querformat: 800 Zeilen, 1200 Spalten) (1 P.)
- (b)  $800 \cdot 1200 = 960\ 000$  Pixel (1 P.)
- (c) 8 bit = 1 Byte pro Farbkanal und Pixel, also 3 Byte pro Pixel (1 P.)
- (d)  $3 \cdot 960\ 000 = 2\ 880\ 000$  Byte = 2,88 MB (1 P.)

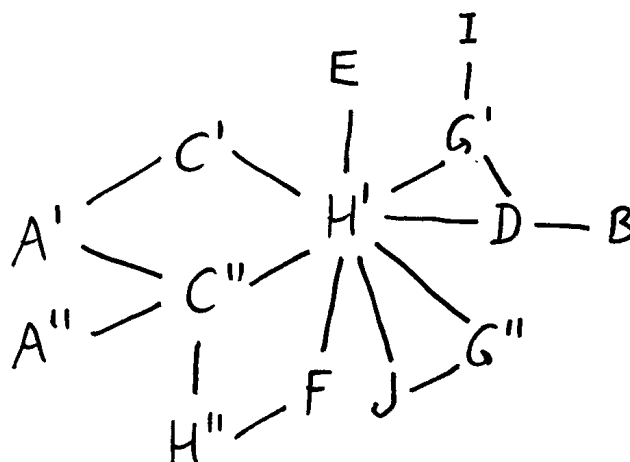
#### Aufgabe 2

- (a) Zeile 8, Spalte 7 (1 P.)
- (b) 12 Schritte der Länge 1 und 3 Diagonalschritte:  
Gesamtlänge  $12 \cdot 1 + 3 \cdot \sqrt{2} \approx 16,2426$  (2 P.)
- (c) 5 Pixel horizontal und 5 Pixel vertikal  
 $\Rightarrow$  euklidischer Abstand  $= \sqrt{(25+25)} = \sqrt{50} \approx 7,07$  (1 P.)

#### Aufgabe 3

Unter Rückgriff auf die Benennung der Regionen bzgl. 8-Nachbarschaft (erarbeitet in der 2. Übung) zerfällt A in A' und A"; C in C' und C"; G in G' und G"; H in H' und H".  $\Rightarrow$  14 Regionen.

Adjazenzgraph:



(darin bestehen A', A'', B aus Vieren, C', C'', D, E aus Dreien, F, G', G'' aus Einsen, H', H'' aus Zweien, I und J sind die isolierten Nullen.) (5 P.)

#### Aufgabe 4

$$f_{jk} = \sum_{m=0}^1 \sum_{n=0}^1 g_{mn} \cdot e^{2\pi i \left( \frac{mj}{2} + \frac{nk}{2} \right)}$$

$$j=k=0: \quad f_{00} = 2+0+0+1 = 3$$

$$j=0, k=1: \quad f_{01} = 2 + 1 \cdot e^{2\pi i \left( 0 + \frac{1}{2} \right)} = 2 + 1e^{\pi i} = 2 + \cdot 1(-1) = 1$$

$$j=1, k=0: \quad f_{10} = 1 \text{ (analog)}$$

$$j=k=1: \quad f_{11} = 1 + 2 \cdot e^{2\pi i \left( \frac{1}{2} + \frac{1}{2} \right)} = 2 + 1 \cdot 1 = 3$$

$$\text{somit } F = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix} \quad (4 P.)$$

Alternative Lösung:

$F$  direkt als Linearkombination der in Aufgabe U4(a) hergeleiteten Basismatrizen der DFT berechnen:

$$F = 2 \cdot B_{0,0} + 1 \cdot B_{1,1} = 2 \cdot \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + 1 \cdot \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}.$$

#### Aufgabe 5

Programm zum Spiegeln von PGM- (ASCII-) Dateien (10 P.)

wichtig:

- Übergabeparameter in der Kommandozeile: Quell- und Zieldatei
- Kommentare in der Quelldatei sind auszublenden

Realisierung in C:

```
/* Bildanalyse und Bildverstehen, Uebungsbl.1, Aufg. 5
   pgmread.c
   Aufruf: pgmread Eingabedatei Ausgabedatei
   liest pgm-Datei und spiegelt Grauwerte an der Diagonalen */

#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <string.h>
#define MAXSIZE 30
/* global: */
int bild[MAXSIZE][MAXSIZE];

int main(int argc, char *argv[])
{
    FILE *ifile;
    FILE *zfile;
    FILE *ofile;
    char achar, bchar;
    long breite, hoehe, maxgrau, wert;
    char ifilename[60];
    char zfilename[10];
    char ofilename[60];
    int status;

    if ((argc > 1) && (strlen(argv[1]) < 60))
    {
        strcpy(ifilename, argv[1]);
        strcpy(zfilename, "help@.pgm");
        if ((argc > 2) && (strlen(argv[2]) < 60))
            strcpy(ofilename, argv[2]);
        else
            strcpy(ofilename, "output.pgm");
    }
    else
    {
        printf("\nProgramm abgebrochen:"
            " Kommandozeile nicht o.k.! <return>:\n");
        achar = getchar();
        exit(0);
    }
    if ((ifile = fopen(ifilename, "r")) == NULL)
    {
        printf("\nEingabedatei nicht in Ordnung! <return>:\n");
        achar = getchar();
        exit(0);
    }
    if ((zfile = fopen(zfilename, "w")) == NULL)
    {
```

```

    printf("\nZwischendatei nicht in Ordnung! <return>:\n");
    achar = getchar();
    exit(0);
}

if ((ofile = fopen(ofilename, "w")) == NULL)
{
    printf("\nAusgabedatei nicht in Ordnung! <return>:\n");
    achar = getchar();
    exit(0);
}
/* 1. Durchlauf: Eliminieren von Kommentaren */
status = 0;
while (fscanf(ifile, "%c", &achar) != EOF)
{
    if (status == 0)
    {
        if (achar == '#')
            status = 1;
    }
    if (status == 1)
    {
        if ((achar == '\n') || (achar == '\r'))
            status = 0;
    }
    if ((achar == '\n') || (achar == '\r')
        || (achar == '\t'))
        achar = ' ';
    if (status == 0)
        fprintf(zfile, "%c", achar);
}
fclose(ifile);
fclose(zfile);
if ((zfile = fopen(zfilename, "r")) == NULL)
{
    printf("\nZwischendatei (Lesen)"
           " nicht in Ordnung! <return>:\n");
    achar = getchar();
    exit(0);
}
/* 2. Durchlauf */
fscanf(zfile, "%c%c %ld %ld %ld",
        &achar, &bchar, &breite, &hoehe, &maxgrau);
if (!(achar == 'P' && bchar == '2'))
{
    printf("\nFalsches Format oder falscher Datei-"
           "Header! <return>:\n");
    achar = getchar();
    exit(0);
}
fprintf(ofile, "P2\n %ld %ld %ld\n",
        breite, hoehe, maxgrau);

```

```

for (int y1=0; y1<hoehe; y1=y1+1)
{
    for (int x1=0; x1<breite; x1=x1+1)
    {
        fscanf(zfile, "%ld", &wert);
        bild[x1][y1]=wert;
    }
}
for (int y2=0; y2<hoehe; y2=y2+1)
{
    for (int x2=0; x2<breite; x2=x2+1)

    {
        wert=bild[hoehe-1-y2][breite-1-x2];
        fprintf(ofile, " %ld", wert);
    }
    fprintf(ofile, "\n");
}

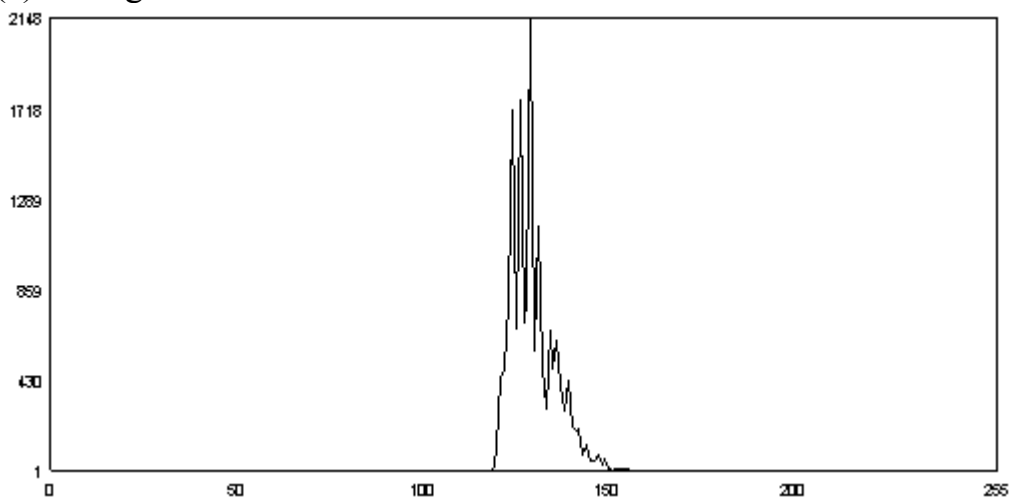
fclose(zfile);
fclose(ofile);
return 0;
}

```

### Aufgabe 6

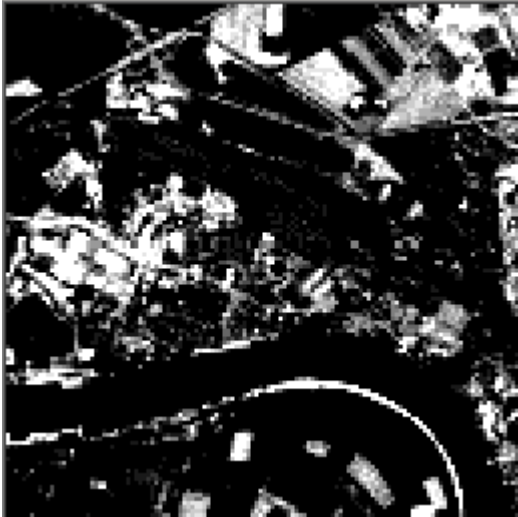
(a), (b) ohne vorzuweisendes Ergebnis und ohne Punkte

(c) Histogramm:



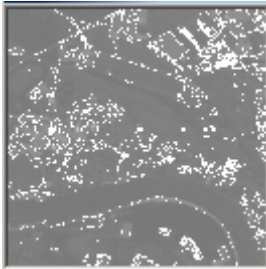
(1 P.)

(d) Grauwerte spreizen, Bild vergrößern (2 Schritte):



(2 P.)

(e) Grauwertbereich markieren:



(1 P.)

(f) Histogramm ebnen:



(1 P.)