

## Serie 5

1.

Im Matlab-Programm schreiben Sie eine  $6 \times 6$  Matrix, die den grauen Farbton darstellt. Erklären Sie den Unterschied zwischen `DOUBLE` und `UINT8`. Zeigen Sie diese Matrix als Bild.

2.

Mit dem Matlab-Befehl `IMREAD` schreiben Sie das Bild `LENA.JPG` als Matrix (eine Erklärung über dieses Bild findet man auf `GOOGLE.CH` ...). Geben Sie die Grösse der Matrix und erklären Sie es. Zählen Sie die Anzahl von weissen Pixels (255). Probieren Sie es auch mit der Inverse von dem grauen Farbton (z.B. in Matlab, `Bnew=255-B`).

3.

Öffnen Sie das Bild `NEWSPAPERPICTURE.JPG`. Mit dem Matlab-Befehl `CONV2`, stellen Sie eine Faltung zwischen das Bild und die folgende Matrix her:

$$f = 1/9 \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Normalisieren Sie Ihr Ergebnis. Diese Faltung soll das originelle Bild glätten. Vergleichen Sie das ergebende Bild mit dem originellen Bild.

4.

In dieser Übung werden Sie ein Programm um ein Bild zu komprimieren. Einige Teile von dem Programm finden Sie in der File `SERIE5DATA.TXT`.

- Schreiben Sie eine Funktion `MYDCT2` (bzw. `MYIDCT2`), um die DCT (bzw IDCT) von einer Matrix  $B$  zu berechnen:

$$Z_{k,\ell} = 1/4 C_k C_\ell \sum_{i=0}^7 \sum_{j=0}^7 B_{i,j} \cos(kx_i) \cos(\ell x_j)$$

und

$$B_{i,j} = 1/4 \sum_{k=0}^7 \sum_{\ell=0}^7 Z_{k,\ell} C_k C_\ell \cos(kx_i) \cos(\ell x_j),$$

mit  $C_0 = 1/\sqrt{2}$  und  $C_\ell = 1$  für  $\ell = 1, \dots, 7$  (siehe die Matlab Funktion `DCT2`).

Hinweis:

```
function dctB=mydct2(B)
%% dct von 8x8-Matrix B
global mNORM mYV mUX (siehe serie5data.txt)
...
```

- Schreiben Sie eine Funktion QUANT (bzw UNQUANT) um die Quantisierung (bzw Unquantisierung) einer Matrix  $B$  zu berechnen:

```
function BQ=quant(B,Qt)
...
```

- Testen Sie Ihre Funktionen auf den Test-Block. Das Ergebnis mit QT50 ist:

```
79    0    -1    0    0    0    0    0
-2   -1    0    0    0    0    0    0
-1   -1    0    0    0    0    0    0
-1    0    0    0    0    0    0    0
 0    0    0    0    0    0    0    0
 0    0    0    0    0    0    0    0
 0    0    0    0    0    0    0    0
 0    0    0    0    0    0    0    0
```

- Versuchen Sie die gegebene Kodierung-Funktion KODI zu verstehen. Was bekommt man für unsere Test-Block-Matrix ? Erklären Sie dies.
- Schreiben Sie einen Code MAIN.M, um die gegebenen Bilder zu komprimieren.

Hinweis:

```
clear all
%% INTITIALISIERUNG
...
%% Bild
B=double(imread('bild1.tif'));
figure,imagesc(B),colormap('gray'); % oder imageview

[nl,nc]=size(B);
%% Anzahl Blocks
nbx=floor(nc/8);nby=floor(nl/8);

%% KOMPRESSION
nbbits=0;meanval=0;
frameB=[];
for indy=1:nby
    idy=indy*8;
    for indx=1:nbx
        ... % DCT und Quantisierung
        [framec, valm]=kodi(dctB,meanval,zick);
        frameB=[frameB framec];
        meanval=valm;
```

```

    end
end

%% DEKOMPRESSION
BK=zeros(8*nby,8*nbx);pix0=0;idx=1;
for indy=1:nby
    id8=indy*8;
    for indx=1:nbx
        kB=zeros(1,64);kz=1;
        idx=idx+1;a=frameB(idx);
        a=a+pix0;kB(zick(kz))=a;pix0=a;
        idx=idx+1;kz=kz+1;
        % laest den Frame fuer diesem Block
        while (frameB(idx)~=0|frameB(idx+1)~=0)
            if (frameB(idx)~=0), kz=kz+frameB(idx);end
            kB(zick(kz))=frameB(idx+2);
            kz=kz+1;idx=idx+3;
        end
        idx=idx+2;
        kB=reshape(kB,8,8);
        uQB=unquant(kB,Qt);
        pix=fix(iDCTp(coeff));
        BK(id8-7:id8,indx*8-7:indx*8)=pix;
    end
end

figure,imagesc(BK),colormap('gray');

```