

## Einführung in Matlab

### Lösungen 9

#### Aufgabe 1:

Z.B. für das Trailer-Bild von Matlab aus der Vorlesung.

```
>> X=imread('trailer.jpg');
>> Xg=rgb2gray(X);
>> c=gray(256);
>> figure(1); image(Xg); colormap(c);
>> Xg=double(Xg(1:680,1:1024));
>> Y=HWT2D(Xg);
>> Z=Y; Z(1:340,1:512)=0; D=abs(IHWT2D(Z));
>> Dmax=max(D(:)); tau=0.1*Dmax; K=D>tau;
>> figure(2); image(~K); colormap(gray(2));
```

#### Aufgabe 2:

```
(a) function [e,ind]=energie(x)
    x=x(:);
    [v,ind]=sort(abs(x),'descend');
    n=numel(x);
    e=zeros(1,n);
    s=0;
    for j=1:n
        s=s+v(j)^2;
        e(j)=s;
    end
    e=e/e(n);
    ind=ind';
end
```

(b) **Test** (weiter zu Bild aus A1)

```
>> [e,ind]=energie(Xg);
>> [eHWT,indHWT]=energie(Y);
>> figure(3); n=numel(e); plot(1:n,e,1:n,eHWT); legend('original','HWT')
>> eHWT(160000)
ans =
    0.9935
>> Y2=Y; Y2(indHWT(160001:end))=0; X2=IHWT2D(Y2);
>> figure(4); imagesc(X2); colormap(c)
>> q=160000/numel(Xg)
q =
    0.2298
```

Beim komprimierten HWT-transformierte Bild Y2 müssen also theoretisch nur ca. 23 Prozent der Pixel des Originalbildes gespeichert werden (Werte ungleich Null).

```
(c) function [x,q]=aufNull(x,p)
    [e,ind]=energie(x);
    n=numel(x);
    j=1;
    while e(j)<p
        j=j+1;
    end
    if j+1 <= n
        x(ind(j+1:end))=0;
    end
    q=j/n;
end
```

```
(d) function [X,q]=komprimiere(X,p)
    X=double(X);
    xmin=min(X(:));
    xmax=max(X(:));
    Y=HWT2D(X);
    [Y,q]=aufNull(Y,p);
    X2=IHWT2D(Y);
    x2min=min(X2(:));
    x2max=max(X2(:));
    X=uint8(xmin+(xmax-xmin)*(X2-x2min)/(x2max-x2min));
end
```

Test (weiter zu Bild aus A1)

```
>> [X3,q]=komprimiere(Xg,0.999); q
q =
    0.2488
>> figure(5); image(X3); colormap(c)
```

### Aufgabe 3:

```
function y=HWT1DK(x,k)
n=numel(x);
y=x;
for i=1:k
    y(1:n)=HWT1D(y(1:n));
    n=n/2;
end
end
```

```
function x=IHWT1DK(y,k)
n=numel(y);
x=y;
n=n/2^(k-1);
for i=1:k
    x(1:n)=IHWT1D(x(1:n));
    n=n*2;
end
end
```

```
function Y=HWT2DK(X,k)
[m,n]=size(X);
Y=X;
```

```

for i=1:k
    Y(1:m,1:n)=HWT2D(Y(1:m,1:n));
    m=m/2;
    n=n/2;
end
end

```

```

function X=IHWT2DK(Y,k)
[m,n]=size(Y);
X=Y;
m=m/2^(k-1);
n=n/2^(k-1);
for i=1:k
    X(1:m,1:n)=IHWT2D(X(1:m,1:n));
    m=m*2;
    n=n*2;
end
end

```

```

function [X,q]=komprimiereK(X,p,K)
X=double(X);
xmin=min(X(:));
xmax=max(X(:));
Y=HWT2DK(X,K);
[Y,q]=aufNull(Y,p);
X2=IHWT2DK(Y,K);
x2min=min(X2(:));
x2max=max(X2(:));
X=uint8(xmin+(xmax-xmin)*(X2-x2min)/(x2max-x2min));
end

```

**Test** (weiter zu Bild aus A1)

```

>> K=3; size(Xg)/2^K
ans =
    85    128    % die 3-fache geht also noch
                % ohne Zeilen- oder Spaltenzahl weiter zu kürzen
>> figure(6); imagesc(HWT2DK(Xg,3)); colormap(c)
>> [X4,q]=komprimiereK(Xg,0.9999,K); q
q =
    0.0578
>> figure(7); image(X4); colormap(c)

```