

Einführung in Matlab

Lösungen 8

Aufgabe 1:

(a) mit **Huffman-Baum** als Matrix **t**

```
function cw=codewort2(t,z)
n=numel(z);
cw=struct('Zeichen',[],'Codewort',[]);
k=1; % Wurzel
folge_zweig(t(k,1),0);
folge_zweig(t(k,2),1);
% eingeneistete Funktion teilt cw, t, z und n mit Hauptfunktion
    function folge_zweig(k,c)
        if k<n
            folge_zweig(t(k,1),[c,0]);
            folge_zweig(t(k,2),[c,1]);
        else
            cw(k-n+1).Zeichen=z(k-n+1);
            cw(k-n+1).Codewort=c;
        end
    end
end
```

```
(b) >> X=imread('coins.png');
>> image(X); colormap(gray(256))
>> v=X(:);
>> z=0:255; w=haeufigkeit(v,z);
>> t=huffman_baum2(w);
>> cw=codewort2(t,z);
>> c=kodiere(v,cw);
>> c_bits=numel(c), X_bits=numel(X)*8, c_bits/X_bits
c_bits =
    468884
X_bits =
    590400
ans =
    0.7942 % also ca. 80 Prozent des ursprünglichen Speicherbedarfs
>> t_uint16_bits=numel(t)*16
t_uint16_bits =
    8160
>> (c_bits+t_uint16_bits)/X_bits
ans =
    0.8080 % macht mit Baum also nicht viel mehr aus
>> vd=dekodiere2(c,t,z);
>> Xd=reshape(vd,size(X));
>> image(Xd); colormap(gray(256))
```

```

(c) >> zq=0:127; vq=round((double(v)+1)/2)-1;
>> wq=haeufigkeit(vq,zq);
>> tq=huffman_baum2(wq);
>> cwq=codewort2(tq,zq);
>> cq=kodierte(vq,cwq);
>> cq_bits=numel(cq), X_bits=numel(X)*8, cq_bits/X_bits
cq_bits =
    396234
X_bits =
    590400
ans =
    0.6711
>> tq_uint8_bits=numel(tq)*8
tq_uint8_bits =
    2032
>> (cq_bits+tq_uint8_bits)/X_bits
ans =
    0.6746 % also nur noch knapp 70 Prozent
>> vqd=dekodiere2(cq,tq,zq); vqd=uint8(round(vqd*255/127));
>> Xqd=reshape(vqd,size(X));
>> figure(1); image(X); colormap(gray(256))
>> figure(2); image(Xqd); colormap(gray(256))

(d) function x=quantisiere(x,n)
for k=1:n
    x=round((double(x)+1)/2)-1;
end
end

>> X=imread('trailer.jpg');
>> n=5; Xq=uint8(round(quantisiere(X,n)*255/(256/2^n-1)));
>> figure(1); image(X); figure(2); image(Xq)

```