

数字图像处理作业五

- 7.5 设信源 $X = \{a, b, c, d\}$, 且 $p(a) = \frac{1}{8}, p(b) = \frac{5}{8}, p(c) = \frac{1}{8}, p(d) = \frac{1}{8}$, 计算各符号的自信息量和信源熵.

$$I(a) = I(c) = I(d) = \log_2\left(\frac{1}{p(a)}\right) = 3$$

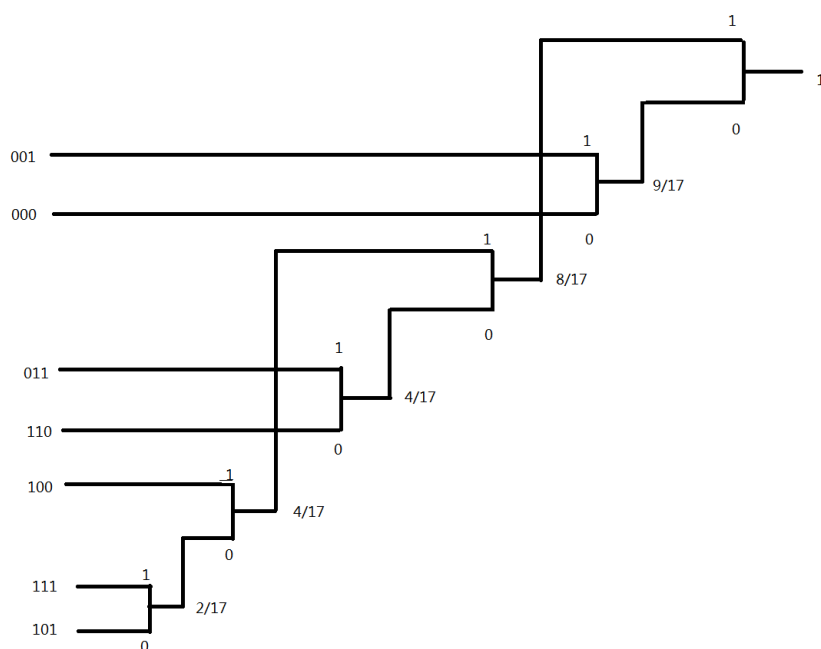
$$I(b) = \log_2\left(\frac{1}{p(b)}\right) = 0.678$$

$$H(X) = \frac{1}{8} * 3 * 3 + \frac{5}{8} * 0.678 = 1.549$$

- 7.7 根据图7.2所示的Huffman树, 下面是某图像的一段Huffman编码. 请将其解码写出来

11100100100000110000001101110000000111000000110110

$$P(001) = \frac{5}{17}, P(000) = \frac{4}{17}, P(110) = \frac{2}{17}, P(011) = \frac{2}{17}, P(100) = \frac{2}{17}, P(111) = \frac{1}{17}, P(101) = \frac{1}{17}$$



信源符号	出现概率	码字	码长
001	5/17	11	2
000	4/17	10	2
110	2/17	000	3
011	2/17	001	3
100	2/17	011	3
111	1/17	0101	4
101	1/17	0100	4

- 编程实现DCT编码实例中的近似JPEG编解码过程, 并用一张图像做验证。

```

2 function res=jpegencode(x,quality)
3     %x 为输入图像
4     %quality决定了截去的系数和压缩比
5     error(nargchk(1,2,nargin));           %检查输入参数
6     if nargin<2
7         quality=1;                         %缺省的quality=1
8     end
9     x=double(x)-128;                       %像素层次移动-128
10    [xm,xn]=size(x);                       %得到图像的尺
11    t=dctmtx(8);                           %得到8*8DCT变换矩阵
12    %将图像分割成8*8子图?,进行DCT变换,然后进行量化
13    y=blkproc(x,[8 8], 'dct2(x)' ) %      'P1*x*P2',t,t');
14    m=[16 11 10 16 24 40 51 61           %JPEG标准化矩
15        12 12 14 19 26 58 60 55
16        14 13 16 24 40 57 69 56
17        14 17 22 29 51 87 80 62
18        18 22 37 56 68 109 103 77
19        24 35 55 64 81 104 113 92
20        49 64 78 87 103 121 120 101
21        72 92 95 98 112 100 103 99]*quality;
22    %用m标准化阵列对变换阵列进行量化
23    yy=blkproc(y,[8 8], 'round(x./P1)',m);
24    y=im2col(yy,[8 8], 'distinct');        %将图像块排列成向量
25    xb=size(y,2);                          %得到列数,也就子图像个数
26    order=[1 9 2 3 10 17 25 18 11 4 5 12 19 26 33 ... %变换系数排列次序
27        41 34 27 20 13 6 7 14 21 28 35 42 49 57 50 ...
28        43 36 29 22 15 8 16 23 30 37 44 51 58 59 52 ...
29        45 38 31 24 32 39 46 53 60 61 54 47 40 48 55 62 63 56 64];
30    %用Z形模板对变换系数重新排列
31    y=y(order,:);
32    eob=max(x(:))+1;
33    num=numel(y)+size(y,2);
34    r=zeros(num,1);
35    count=0;
36    %将非零元素重新排列
37    for j=1:xb
38        i=max(find(y(:,j)));               %找最后一个非零元素
39        if isempty(i)                      %没有非零元素
40            i=0;
41        end
42        p=count+1;
43        q=p+i;
44        r(p:q)=[y(1:i,j);eob];            %截去0 并加上结束符号
45        count=count+i+1;
46    end
47    r((count+1):end)=[];                  %删除没有用的部分
48
49    r=r+128;
50    %保存编码信息
51    res.size=uint16([xm,xn]);
52    res.numblocks=uint16(xb);
53    res.quality=uint16(quality*100);
54    %对r进行huffman编码
55    [res.huffman res.info]=huffencode(uint8(r));
56
57
58    %jpegdecode函数jpegdecode的解码程序
59 function x=jpegdecode(y)
60     error(nargchk(1,1,nargin));           %检查输入参数
61     m=[16 11 10 16 24 40 51 61           %JPEG标准化矩阵
62        12 12 14 19 26 58 60 55
63        14 13 16 24 40 57 69 56
64        14 17 22 29 51 87 80 62
65        18 22 37 56 68 109 103 77
66        24 35 55 64 81 104 113 92
67        49 64 78 87 103 121 120 101
68        72 92 95 98 112 100 103 99];
69    order=[1 9 2 3 10 17 25 18 11 4 5 12 19 26 33 ... %变换系数排列次序
70        41 34 27 20 13 6 7 14 21 28 35 42 49 57 50 ...
71        43 36 29 22 15 8 16 23 30 37 44 51 58 59 52 ...
72        45 38 31 24 32 39 46 53 60 61 54 47 40 48 55 62 63 56 64];
73    rev=order;                             %计算逆序
74    for k=1:length(order)
75        rev(k)=find(order==k);

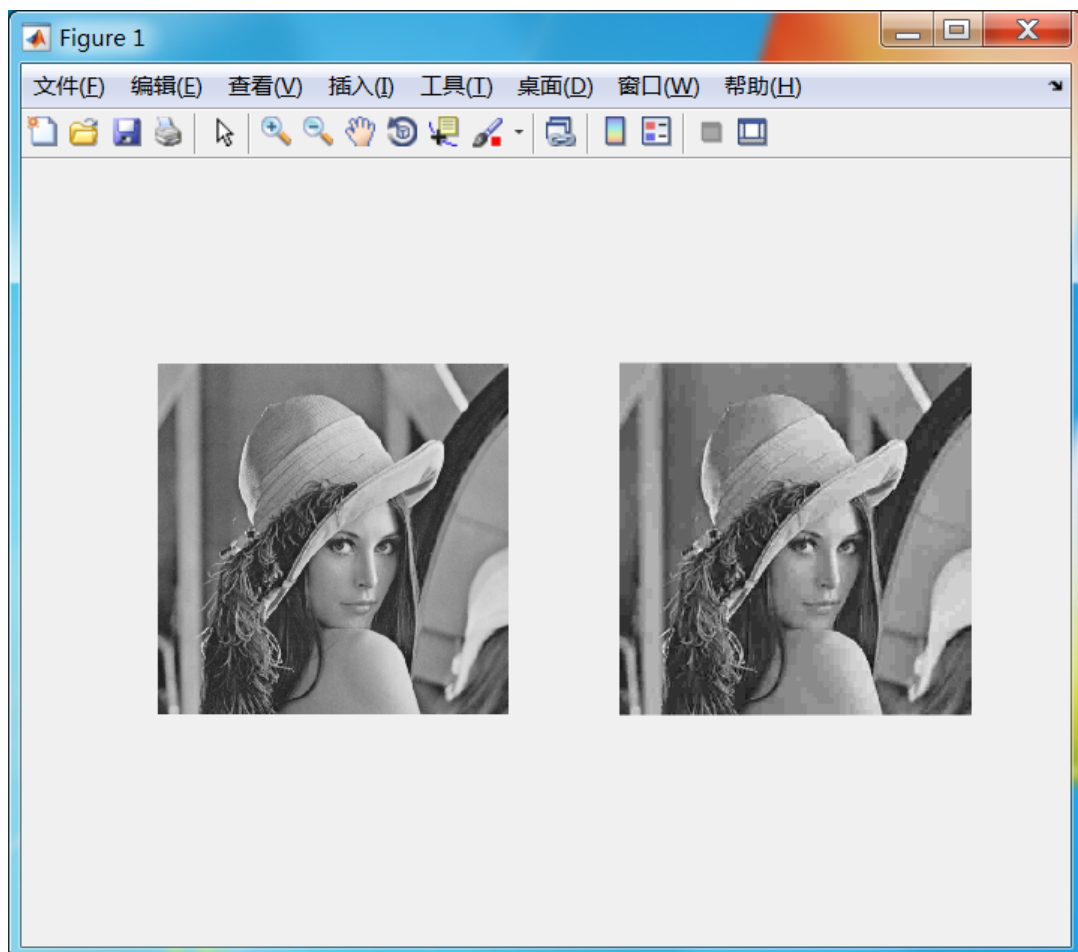
```

```

76     end
77     %ff=max(rev(:))+1;
78     m=double(y.quality)/100*m;
79     xb=double(y.numblocks);           %得到图像块数
80     sz=double(y.size);
81     xn=sz(1);                         %得到行数
82     xm=sz(2);                         %得到列数
83     x=huffdecode(y.huffman,y.info);   %huffman解码
84     x=double(x)-128;
85     eob=max(x(:));                    %得到块结束符
86     z=zeros(64,xb);k=1;
87     for j=1:xb
88         for i=1:64
89             if x(k)==eob
90                 k=k+1;break;
91             else
92                 z(i,j)=x(k);
93                 k=k+1;
94             end
95         end
96     end
97     z=z(rev,:);                       %恢复次序
98     x=col2im(z,[8 8],[xm xn],'distinct');%重新排列成图像块
99     x=blkproc(x,[8 8],'x.*P1',m);     %反标准化
100    t=dctmtx(8);
101    x=blkproc(x,[8 8], 'P1*x.*P2',t,t); %DCT反变换
102    x=uint8(x+128);                    %进行位移
103
104    %测试函数
105    function DCT_Test
106        clear all
107        x = imread('lena.bmp');
108        figure(1);
109        subplot(121);
110        imshow(x);
111        y = jpegencode(x, 5);
112        X = jpegdecode(y);
113        subplot(122);
114        imshow(X);
115
116        e = double(x) - double(X);
117        [m, n] = size(e);
118        erms = sqrt(sum(e(:).^2) / (m * n))
119        cr = imageratio(x, y)
120

```

- 运行结果



```
erms =
```

```
7.7002
```

```
cr =
```

```
31.3719
```