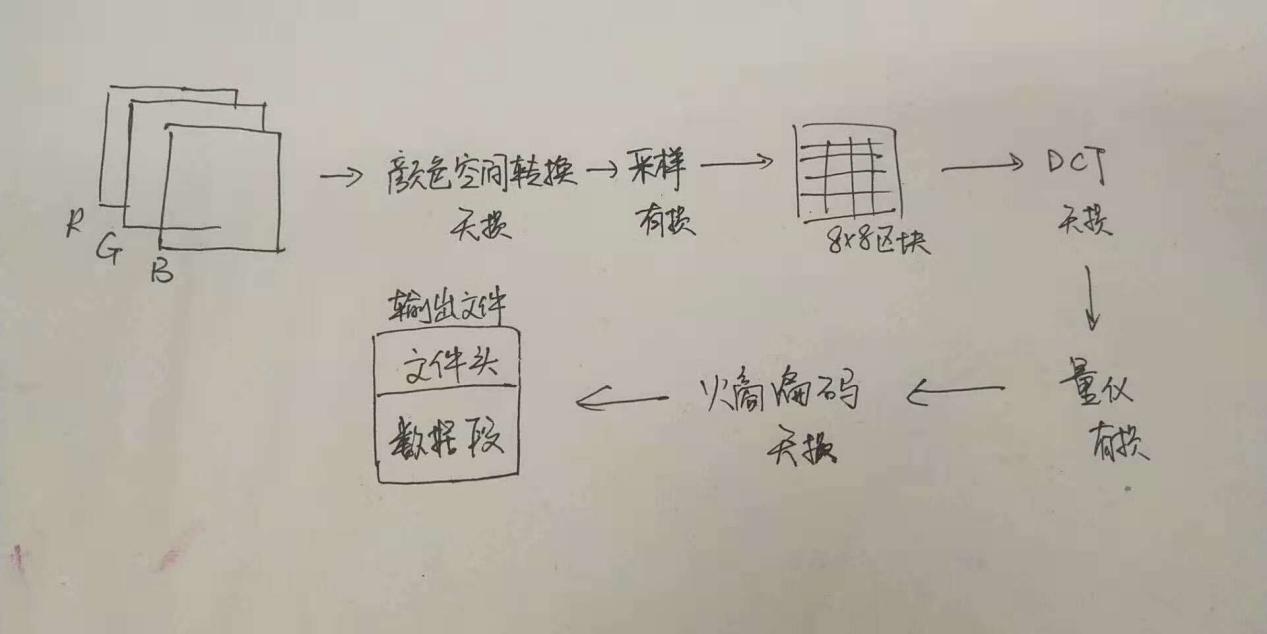
1画出JPEG压缩原理框图。哪些步骤是有损，哪些步骤是无损压缩？



2DCT系数量化的作用是什么？

降低各系数的位数，量化后增加了“0”值系数的数目，这有利于使用游程编码。

3DC系数和AC系数的编码方法各是什么？

DC：对（变长码的位数，数值）数对进行编码，查分编码。

AC：ZigZag编码

4图像分块的大小是什么？为什么是该数值？为何要分块？

分成8\*8块，因为图像分块时可能出现小数个块，为了使用其能够最大程度的显示出图像来，先将图像变成其整数倍（为什么分块）。压缩是根据像素之间的相关性来压缩的：只有比较小的邻域内的像素有比较高的相关性，相距远的话，其相关性就大大下降，所以要分块，对一个邻域内的图像根据每个像素相关的信息来压缩。随着子块尺寸的增加，算法的复杂度急剧上升。通常采用8×8的子块进行变换，但采用较大的子块可以明显减少图像分块效应。（为什么该数值？）

5为matlab代码逐行添加注释

d=imread('C:\Users\admin\Desktop\2018\_autumn\lesson 8\anna.bmp');

imshow(d)

////////读入文件并显示原始图片

%C=[0.299,0.587,0.114;-0.1687,-0.3313,0.5;0.5,-0.4187,-0.0813]);

////////%c输出构造的矩阵

%[Y;Cb,Cr]=C\*[d(:,:,1);d(:,:,2);d(:,:,3)]+[0;128;128];

///////颜色饱和度设置

Q=[16 11 10 16 24 40 51 61

12 12 14 19 26 58 60 55

14 13 16 24 40 57 69 56

14 17 22 29 51 87 80 62

18 22 37 56 68 109 103 77

24 35 55 64 81 194 113 92

49 64 78 87 103 121 120 101

72 92 95 98 121 100 103 99]; % Quantization Table for luminance

///////色度量化值表

ac\_huffman\_Luminance={

'00' '01' '100' '1011' '11010' '1111000' '11111000' '1111110110' '1111111110000010' '1111111110000011';...

'1100' '11011' '1111001' '111110110' '11111110110' '1111111110000100' '1111111110000101' '1111111110000110' '1111111110000111' '11111111100001000';...

'11100' '11111001' '1111110111' '111111110100' '1111111110001001' '1111111110001010' '1111111110001011' '1111111110001100' '1111111110001101' '1111111110001110';...

'111010' '111110111' '111111110101' '1111111110001111' '1111111110010000' '1111111110010001' '1111111110010010' '1111111110010011' '1111111110010100' '1111111110010101';...

'111011' '1111111000' '1111111110010110' '1111111110010111' '1111111110011000' '1111111110011001' '1111111110011010' '1111111110011011' '1111111110011100' '1111111110011101';...

'1111010' '11111110111' '1111111110011110' '1111111110011111' '1111111110100000' '1111111110100001' '1111111110100010' '1111111110100011' '1111111110100100' '1111111110100101';...

'1111011' '111111110110' '1111111110100110' '1111111110100111' '1111111110101000' '1111111110101001' '1111111110101010' '1111111110101011' '1111111110101100' '1111111110101101';...

'11111010' '111111110111' '1111111110101110' '1111111110101111' '1111111110110000' '1111111110110001' '1111111110110010' '1111111110110011' '1111111110110100' '1111111110110101';...

'111111000' '111111111000000' '1111111110110110' '1111111110110111' '1111111110111000' '1111111110111001' '1111111110111010' '1111111110111011' '1111111110111100' '1111111110111101';...

'111111001' '1111111110111110' '1111111110111111' '1111111111000000' '1111111111000000' '1111111111000010' '1111111111000011' '1111111111000100' '1111111111000101' '1111111111000110';...

'111111010' '1111111111000111' '1111111111001000' '1111111111001001' '1111111111001010' '1111111111001011' '1111111111001100' '1111111111001101' '1111111111001110' '1111111111001111';...

'1111111001' '1111111111010000' '1111111111010001' '1111111111010010' '1111111111010011' '1111111111010100' '1111111111010101' '1111111111010110' '1111111111010111' '1111111111011000';...

'1111111010' '1111111111011001' '1111111111011010' '1111111111011011' '1111111111011100' '1111111111011101' '1111111111011110' '1111111111011111' '1111111111100000' '1111111111100001';...

'11111111000' '1111111111100010' '1111111111100011' '1111111111100100' '1111111111100101' '1111111111100110' '1111111111100111' '1111111111101000' '1111111111101001' '1111111111101010';...

'1111111111101011' '1111111111101100' '1111111111101101' '1111111111101110' '1111111111101111' '1111111111110000' '1111111111110001' '1111111111110010' '1111111111110011' '1111111111110100';...

'1111111111110101' '1111111111110110' '1111111111110111' '1111111111111000' '1111111111111001' '1111111111111010' '1111111111111011' '1111111111111100' '1111111111111101' '1111111111111110'

};

///////亮度AC系数的Huffman编码表,赋给变量ac\_huffman\_Luminance

dc\_huffman\_Luminance = {'00', '010', '011', '100', '101', '110', '1110', ...

'11110', '111110', '1111110', '11111110', '111111110'};

///////色差DC系数的Huffman编码表，赋给变量dc\_huffman\_Luminance

z=[...

9 2 3 10 17 25 18 11 4 5 12 19 26 ...

33 41 34 27 20 13 6 7 14 21 28 35 ...

42 49 57 50 43 36 29 22 15 8 16 23 ...

30 37 44 51 58 59 52 45 38 31 24 32 ...

39 46 53 60 61 54 47 40 48 55 62 63 56 64];

/////////亮度量化值表

R=d(:,:,1);G=d(:,:,2);B=d(:,:,3);

Y=0.299\*R+0.587\*G+0.114\*B;

//////颜色的亮度（纯度）的数学公式

Cb=-0.1687\*R-0.3313\*G+0.5\*B+128;

//////颜色的色度（纯度）的数学公式

Cr=0.5\*R-0.4187\*G-0.0813\*B+128;

//////颜色的饱和度（纯度）的数学公式

Cb=Cb(1:4:512,:);

Cr=Cr(1:4:512,:);

chunk=8;

Dc=zeros(prod([size(Y,1)/chunk,size(Y,2)/chunk]),1);

Ac=zeros(prod([size(Y,1)/chunk,size(Y,2)/chunk]),63);

///////zeros（m，n）是指建立一个，m\*n的0矩阵，size(Y,1)就是指Y矩阵的列数，/chunk(chunk=8)分为8\*8的区块

k=0;

for n=1:size(Y,1)/chunk

for m=1:size(Y,2)/chunk

data=Y(((n-1)\*chunk+1):n\*chunk,((m-1)\*chunk+1):m\*chunk); %chunk

data=int8(data)-128;%128 Shift

//////进行128平移

dct\_data=dct2(data); % DCT

//////进行DCT变换

q\_data=round(dct\_data./Q);% Quantization

///////取整

k=k+1;

Dc(k)=q\_data(1,1);

/////调用Dc函数

Ac(k,:)=q\_data(z);

//////调用Ac函数

end

end

%Coding Dc////////以下代码进行Dc编码

Dc\_diff=[Dc(1);diff(Dc)];

///////diff函数式用于求导数和差分的.

Dc\_encoding='';

for n=1:length(Dc\_diff)

/////遍历矩阵

if Dc\_diff(n)==0

dhl=dc\_huffman\_Luminance(1);

Dc\_encoding=[Dc\_encoding,cell2mat(dhl)];

else

dhl=dc\_huffman\_Luminance(length(dec2bin(abs(Dc\_diff(n))))+1);

if Dc\_diff(n)<0

cc=dec2bin(abs(Dc\_diff(n))); %inverse code

b=char('0'+'1'-cc);

b=strrep(b,' ','');

else

b=dec2bin(Dc\_diff(n));

//////计算差值m的二进制位数，n。如差值≥0，则用原码，否则用反码。原码最高位是1就是正，反码是0就是负。 end

Dc\_encoding=[Dc\_encoding,cell2mat(dhl),b];

///////编码为（Dc\_encoding,cell2mat(dhl),b）

end

end

% Ac////////以下代码进行Ac编码

Ac\_encoding='';

for n=1:size(Ac,1)

notzeros=find(Ac(n,:)~=0);

if length(notzeros)==0

Ac\_encoding=[Ac\_encoding,'1010'];

else

zeros\_num=diff([0,notzeros])-1;

notzero\_val=Ac(n,notzeros);

for k=1:length(zeros\_num)

len=length(dec2bin(abs(notzero\_val(k))));

zn=floor(zeros\_num(k)/16);

zn\_c=mod(zeros\_num(k),16);

ahl=ac\_huffman\_Luminance(zn\_c+1,len);

if notzero\_val(k) <0

cc=dec2bin(abs( notzero\_val(k)));

b=char('0'+'1'-cc);

b=strrep(b,' ','');

else

b=dec2bin(notzero\_val(k));

end

if zn>0

zc='';

for m=1:zn

zc=[zc,'11111111001'];

end

Ac\_encoding=[Ac\_encoding,zc,cell2mat(ahl),b];

else

Ac\_encoding=[Ac\_encoding,cell2mat(ahl),b];

end

End

//////计算非零数值前零的个数，n；该非零系数需二进制位数m；该非零系数值k，负数仍使用反码；

Ac\_encoding=[Ac\_encoding,'1010'];

///////得到编码（n，m，k），前两分量查表得到码字，后一个转写为二进制

end

end