Xi’an Jiaotong-Liverpool University
西交利物浦大学

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE330 Lab Report 4

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Engineering

Student Name : Lingxuan Kong
Student ID : 1405867

Contents

1 Task 1 **2**

2 Task 2 **4**

3 Task 3 **6**

3.1 Fill table 6

3.2 PSNR vs bit rate curve 6

3.3 Rate-PSNR vs JPEG 7

3.4 Compare DCT and JPEG 8

3.5 $N = 8$ 8

3.6 Enhance the RD efficiency 9

4 Conclusion **10**

1 Task 1

This first task was about DCT compress. According to the knowledge from lecture, there are four main steps in DCT image compress:

(1) Segment the images into many blocks. For example, segment the image into 8×8 blocks. Because the details on each part of the original image are different, so if we do the DCT transform to thw whole photo without segment, the results should be bad. That's the reason why we need to block the iamge before we do the DCT.

(2) Apply DCT to the blocks. When the photo after DCT transform, the relationship between DCT coefficient will become less. Additional, most of information always concentrate on the high frequency part.

(3) Quantify to the previous photos, which should in the frequency domain. In this step, quantify decrease both size and useful information of photo.

(4) Encode the previous photo.

The main differences between this compression engine and JPEG:

- In jpeg compression, after quantization, there will be step named zig-zag reading. Additional, when using JPEG images, one can set the quality of the image from very low quality to very high quality.
- After quantization, in JPEG encoding, quantified DC values are coded by DPCM from block to block to remove the residual correlation. In AC coefficient, zig-zag reordering is performed to achieve large runs of zeros.
- In the jpeg encoding, Huffman encoding will be applied.

```
function [rate] = compress_im(im, Qmat, QP, N, file_name)
I=im;
a=size(I,1);%obtain the height of input image
b=size(I,2);%obtain the length of input image
I=double(I);%change to double

%determine the value of S
if QP>50
    S=(100-QP)/50;
else
    S=50/QP;
end
%obtain the quantize matrix, mulitipy with S
x_qmat=S.*Qmat;

for i=1:N:a
    for j=1:N:b
        P=I(i:i+N-1,j:j+N-1);%cut the input image to blocks
        K=dct2(P);%(a) apply DCT of the blocks
        I2(i:i+N-1,j:j+N-1)=K;
        K=round(K./x_qmat);%(b) quantize
        I3(i:i+N-1,j:j+N-1)=K;%obtain the blocks image in frequency ...
           domain
    end
end

y_encode=entropy_enc(I3);%(c) encode the blocks image

%%%%%(d)
y_vector=fopen(file_name,'w');%fopen
fwrite(y_vector,y_encode,'uint8');%save the entropy encoded vector ...
   as file_name
fclose(y_vector);

rate=(fsize(file_name))*8/a/b;%(e) evaluate the bit rate
```

2 Task 2

Decompress should be the inverse step of compress:

(1) Decode the images.

(2) Inverse quantization

(3) Inverse the DCT transform

(4) Combine the block into one image

```
function [imo] = decompress_im(Qmat, QP, N, file_name)

y_com=fopen(file_name,'r');%open the code
y_encode=fread(y_com,'uint8');
fclose(y_com);

I_decompress=entropy_dec(y_encode);%apply the decode
x_resize=sqrt(size(I_decompress,1));%obtain the size of input ...
    compressed image

I_decompress=reshape(I_decompress,x_resize,x_resize);%reshape the ...
    image into N*N
a=size(I_decompress,1);%obtain the height of new reshape image
b=size(I_decompress,2);

if QP>50%determine the QP
    S=(100-QP)/50;
else
    S=50/QP;
end

x_qmat=S.*Qmat;%determine the quantize matrix


for i=1:N:a
    for j=1:N:b
        P=round(I_decompress(i:i+N-1,j:j+N-1).*x_qmat);%inverse quantify
        K=idct2(P);%apply inverse DCT
        I_decompress(i:i+N-1,j:j+N-1)=K;%combine the block image
    end
```

```
end

imo=uint8(I_decompress);%output new image after compressed
```

```
%generate the Qmat
function A=Qmat_generate(N)
if(N≤0)
    disp('Please input N>0');
end

for i=12:-1:1
    B(1:2^i-1,1:2^i-1)=2^(i-1);
end
A=B(1:N,1:N);
return
```

3 Task 3

3.1 Fill table

Table 1: Relation between Rate and PSNR

QP	1	15	29	43	60	80	99
Rate	0.1812	0.9588	1.4485	1.8681	2.3440	3.2805	7.5717
PSNR	27.8241	36.7627	39.1779	41.1102	43.3237	47.9981	58.7071

```
IM=imread('lenna512.bmp');%read the original photo
Qmat=Qmat_generate(16);%generate the quantify matrix
QP=99;%define QP
N=16;%defien N
Z='test.txt';%input file_name
Rate=compress_im(IM,Qmat,QP,N,Z);%obtain the rate
L=decompress_im(Qmat,QP,N,Z);%apply the compressed photo
PSNR=psnr(L,IM);%calculate the psnr
```

3.2 PSNR vs bit rate curve

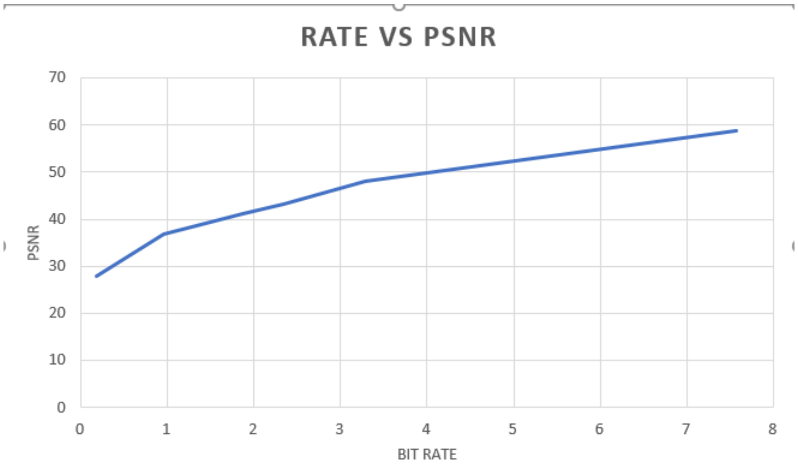

Figure 1: Bit rate vs PSNR curve

3.3 Rate-PSNR vs JPEG

Table 2: Bit Rate vs PSNR in JPEG

QP	1	15	29	43	60	80	99
Rate	0.1335	0.3076	0.4530	0.5776	0.7366	1.1593	4.6481
PSNR	34.2411	31.9271	34.1452	35.3240	36.4283	38.4917	54.5089

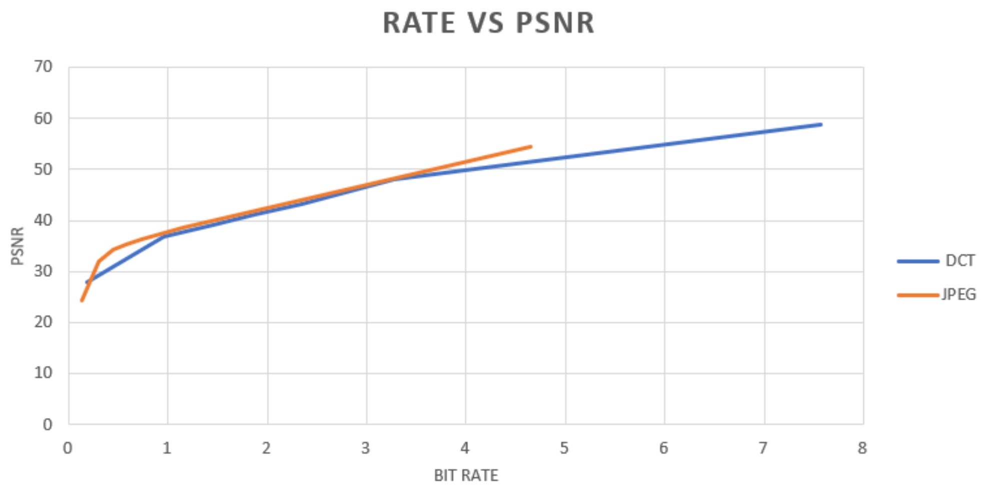

Figure 2: DCT compress vs JPEG compress

After observe the figure2, we can draw some significant conclusions:

- When QP change from 1 to 99, the Rate of DCT change from 0.18 to 7.57, but the rate of JPEG change from 0.13 to 4.64. Thus, if we only observe the rate of two compression method, we can find that the performance of JPEG is better than DCT.
- Then we observe the PSNR: The maximum psnr of DCT is a little higher than JPEG, and their starting point could be considered as the same. Additional, the change domain of two difference method is nearly same. Therefore, if we only observe the psnr, we can say that the two method can make the same difference.
- The we observe the slope of two methods, the JPEG can make a lower rate but with the high PSNR. Therefore, afte observing the 'RD-curve' of two difference method, we can find that JPEG is better than DCT compression.

```
im=imread('lenna512.bmp');%read the original photo
QP=99;
imwrite(im,'compress.jpeg','Quality',QP);%use JPEG compress
im_1=imread('compress.jpeg');
a=size(im_1,1);
b=size(im_1,2);

rate=fsize('compress.jpeg')*8/a/b;%obtain the rate
PSNR=psnr(im_1,im);%obtain the psnr
```

3.4 Compare DCT and JPEG

- We first observe the figure1, we can find that when QP=99, the rate is nearly to 8. In another word, when bit rate is equals to 8, the photo can be considered as not compressed. Then we observe table2, we can find that in jpeg compression, when QP=99, the bit rate is only 4.6, which means this compression is better than the previous one.
- According to the knowledge about jpeg, in the quantization part, quantized DC values are coded by DPCM from block to block to remove the residual correlation. However, in DCT compression, we only mulitify the photo in frequency with a matrix, without DPCM.
- In jpeg encoding, Huffman coding is applied, which should be regarded as one of the difference between jpeg and DCT compression. Also, zig-zag performed to achieve large runs of zeros.
- Because of these differences between DCT and JPEG, the compression performance of JPEG is better than DCT.

3.5 $N = 8$

Table 3: Rate vs PSNR (N=8)

QP	1	15	29	43	60	80	99
Rate	0.3219	1.4728	2.1905	2.7607	3.2166	4.1812	8.5
PSNR	29.017	39.018	41.9652	44.1515	47.0285	51.3735	58.8691

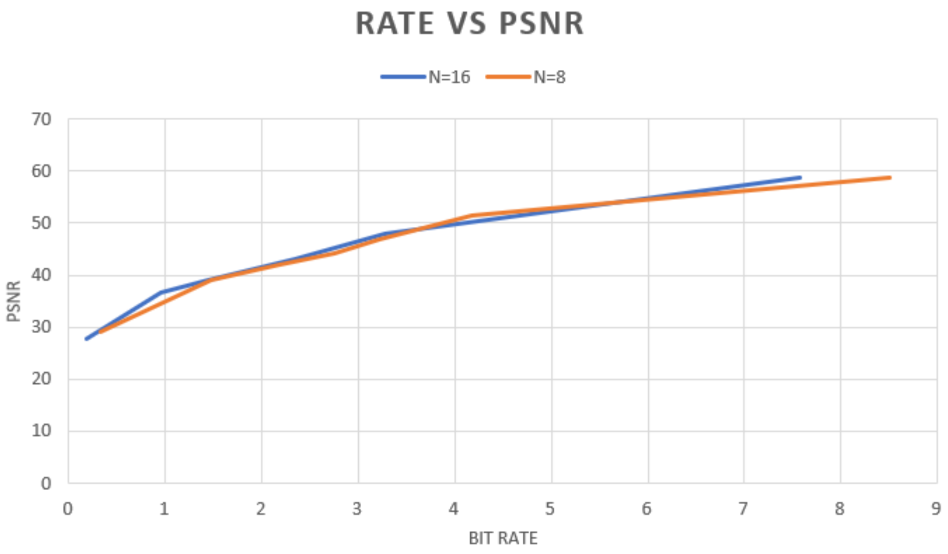

Figure 3: $N = 16$ & $N = 8$

- Observe table, we can find that when QP=99, the rate is 8.5, which is larger than 8. In another word, the size of compressed photo is bigger than original photo.

- When QP=99, the psnr is same, but the bit rate of $N = 16$ is smaller than $N = 8$. Therefore, in this DCT compression, N=16 is better than when N=8.

3.6 Enhance the RD efficiency

- In encoding part, we can use DPCM from block to block to remove the residual correlation in DC values. Using the same theory, Zig-zag reordering is performed to achieve large runs of zeros. Additional, Huffman coding could be considered as another useful method in encoding part.

- In the previous DCT compression, we can improve the value of N to improve the RD efficiency.

- Based on the previous results, the jpeg compression also plays better than DCT compression. Thus, we can use jpeg compression instead of DCT compression.

4 Conclusion

According to the previous results, we can obtain some important conclusions:

(1) Image compression is one of the most important method in image processing, it can reduce the size of image with losing a little information. In the previous task, we use two difference method to compress image, which are DCT compression and jpeg compression.

(2) There are 8 main step in DCT compression: Segment, DCT, quantify, encode, decode, inverse quantify, inverse DCT, combine blocks into one photos. The value of QP, which will make a difference to the quantify matrix. Additional, based on the taks3, we have already know that the value of N will influence the RD efficiency. When N is larger, the compression performance is better.

(3) Jpeg compression is difference from the previous DCT compression. In jpeg encoding part, quantized DC values are coded by DPCM from block to block to remove the residual correlation. And in AC values, zig-zag, encoding of zero-runs, and Hufman coding will be applied in jpeg compression. In jepg decoding, Huffman decoding, run length decoding, and coefficient de-quantization will be performed.