Xi'an Jiaotong-Liverpool University

西交利物浦大學

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE412 Image and Video Processing

Lab4 - Image compression

Student Name Zhiyu.Xu

Student ID 1926905

November 2019

Contents

Task 1

1.1 Entropy of an image

1.1.1 Result and code

Images	Original image	Half size image	16 gray level image
Entropy	7.3775	7.4231	3.4846

Table 1 Entropy result

```
img = imread('lenna512.bmp');
foo = @(mat)mean(mat.data,"all");
img_half = blockproc(img,[2 2],foo);%reduce the original
image to half size;
img_half = uint8(img_half);
img_16_gray_level = floor(img/16);%change 265 gray level
to 16 gray level;
entropy_OGimg = Entropy(img);
entropy_half_OGimg = Entropy(img_half);
entropy_img_16 = Entropy(img_16_gray_level);
```

Code 1 Main code

```
function [out] = Entropy(im)
%Calculating the entropy of an image
%   Detailed explanation goes here
im = uint8(im);
[r,c] = size(im);
count = imhist(im);
p = count/(r*c);
f = @(x) x*log2(1/x);%set entropy function
out = arrayfun(f,p);
%Since there is p=0 in the array, matlab calculates log0
to return NaN, so
%you need to propose NaN value when calculating entropy;
out = sum(out(~isnan(out)));
end
```

Code 2 Function of calculating entropy

1.1.2 Founding and explanation

According to the Table 1, the entropy of original image and half size image is roughly the same(7.3775 and 7.4321 respectively). Compared with entropy of original image, the entropy of 16-gray-level image is quiet small(3.4846). This is because that the formula of calculating entropy is according to the proportion of gray value in the image, changing the size of image can not change the entropy, Only change the

proportion of gray value in the image can change the entropy.

1.2 DPCM coding

Images	Difference of output
Entropy	2.3425

Table 2 Entropy result

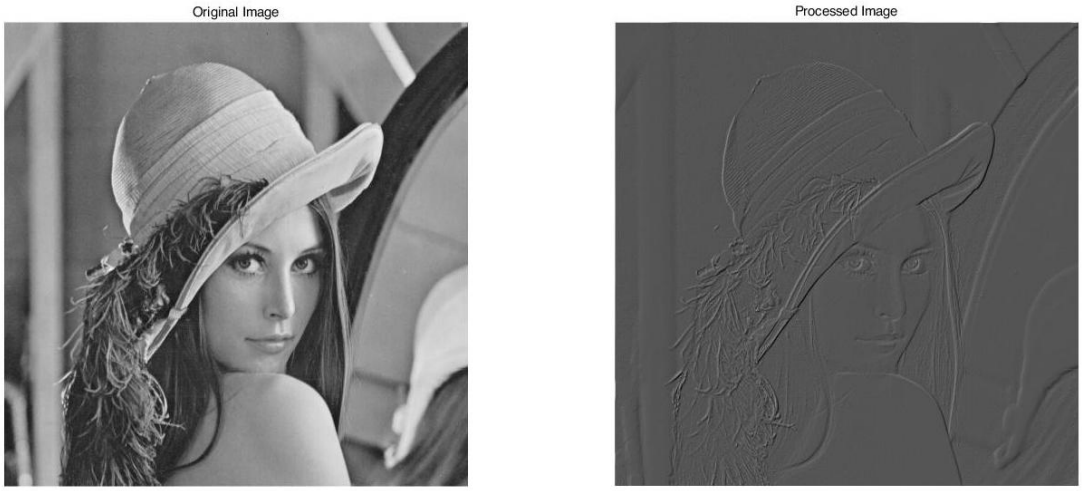

Figure 1 Processed image by DPCM coding

```
[im_DPCM_e,im_DPCM_p] = DPCM(img);
subplot(1,2,1);imshow(img);title('Original Image');
subplot(1,2,2);imshow(im_DPCM_e,[]);title('Processed
Image');
truesize;
entropy_img_DPCM_e = Entropy(im_DPCM_e);
```

Code 3 Main code

```
function [e,p] = DPCM(im)
%This function is the weighted average value of the
%neighboring pixels (left, left-top, and top);
im = double(im);
h = [0.2,0.4;0.4,0];%Set filter coefficients;
p = imfilter(im,h,'replicate','same','corr');
e = im-p;
end
```

Code 4 Function of DPCM

1.2.2 Explanation

The processed image by DPCM coding is easier to be compressed since it has smaller entropy value. And the larger entropy value means that the more information the image contains, which will make image more difficult to compress.

Task 2 Image DCT and quantization

2.1 2D DCT

2.1.1 Result and code

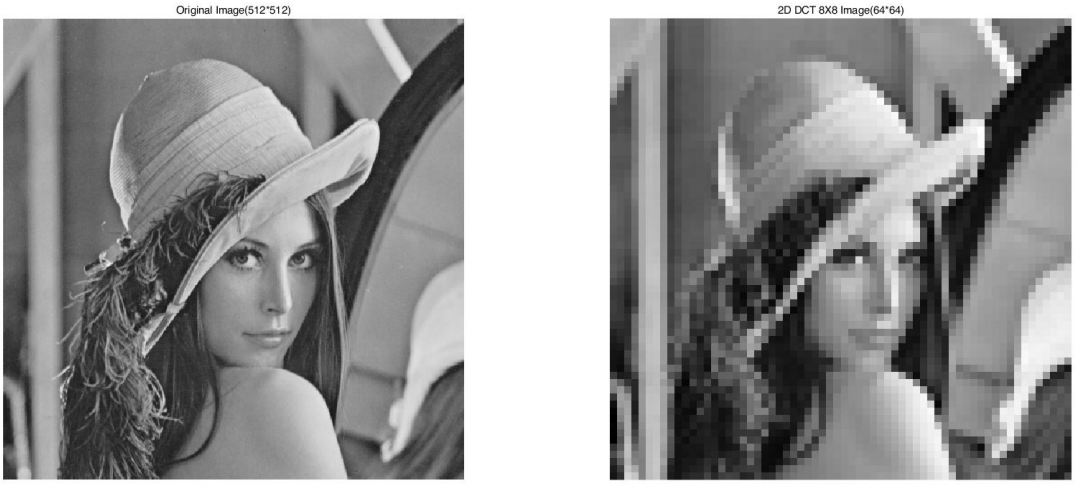

Figure 2 Processed image by 2D DCT

```
im = double(imread('lenna512.bmp'));
ims = DCT2D(im);
[r1, c1] =size(im);
[r2, c2] =size(ims);
figure(1)
subplot(1,2,1);imshow(uint8(im));title(['Original
Image','(',num2str(r1),'*',num2str(c1),')'])
subplot(1,2,2);imshow(ims,[]);title(['2D DCT 8X8
Image','(',num2str(r2),'*',num2str(c2),')'])
```

Code 5 Main code

```
function [im_output] = DCT2D(im_input)
%This function is to do the 2D DCT of all the 8X8
non-overlapping blocks of
%the image im_input, and merge the left-top pixel of all
blocks after the
%DCT transformation to get a smaller image im_output;
B =[1 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0;];
```

```
f1 = @(block_struct)dct2(block_struct.data);
f2 = @(block_struct)(sum(sum((block_struct.data).*B)));
im = blockproc(im_input,[8 8],f1);
im_output = blockproc(im,[8 8],f2);
end
```

Code 6 Function of 2D DCT by 8X8 non-overlapping blocks

2.1.2 Explanation

According to figure 2, we can clearly see that the processed image is obviously compressed, from 512X512 to 64X64, although the approximate appearance of the image can still be seen, but the image quality is greatly degraded.

This is because the function divides the original image into a plurality of 8x8 sub-images by performing block processing, and then performs DCT processing on the sub-images, so that the value of the upper left corner of each sub-image can roughly represent the content of the sub-image. Finally, extracting the value of the upper left corner of each sub-image into a new image, the compressed image is obtained.

2.2 Quantization

2.2.1 Code

```
function [im_out] = quantize(im_in,QP)
%this function is to quantize each 8 X 8 block using the
following formula;
Qmat =[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 81 104 113 92;
       49 64 78 87 103 121 120 101;
       72 92 95 98 112 100 103 99;];
if (QP>50)
     S= (100-QP)/50;
elseif (QP<=50)
     S= 50/QP;
end
f1 = @(block_struct)dct2(block_struct.data);
f2 =
@(block_struct)round((block_struct.data)./(Qmat*S));
im = blockproc(im_in,[8 8],f1);
im_out = blockproc(im,[8 8],f2);
end
```

Code 7 Function of DCT by quantized block

2.3 Decompression

2.3.1 Code

```
function [im_out] = Decompress(im_in,QP)
%Decompress image
Qmat =[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 81 104 113 92;
       49 64 78 87 103 121 120 101;
       72 92 95 98 112 100 103 99;];
if (QP>50)
     S= (100-QP)/50;
elseif (QP<=50)
     S= 50/QP;
end
f2 =
@(block_struct)round((block_struct.data).*(Qmat*S));
f3 = @(block_struct)idct2(block_struct.data);
im_out = blockproc(im_in,[8 8],f2);
im_out = blockproc(im_out,[8 8],f3);
end
```

Code 8 Function of decompressing image

2.4 Decompression with different QP values

2.4.1 Result and code

QP	1	15	29	43	57	74	85	99
PSNR(dB)	17.2018	31.9229	34.1514	35.3247	36.2191	37.3842	39.3897	53.6954

Table 3 PSNR result

```
im = double(imread('lenna512.bmp'));
QP = [1 15 29 43 57 71 85 99];
f = @(QP)
psnr(uint8(im),uint8(Decompress(quantize(im,QP),QP)))
;
PSNRS = arrayfun(f,QP);
```

Code 9 Main code

2.4.2 Comments.

According to the Table 3, as the QP value increases, the PSNR values between the original image with the decompressed imag become larger and larger, which

indicates that the larger the QP value selected by the system, the better the decompression effect. This is because there is a process of round() in the compression process that rounds to the smallest integer, so when the coefficient is large, more details are saved without being ignored.

And there is also a second case here. In the above case, the elements in the upper left corner of each 8X8 block are not selected as a compressed image when the image is compressed. But after the above operation, the PSNR values are both around 23.6, This shows that there are always some differences between the original image and the decompressed image.