Department of Signals & Systems
Chalmers University

Laboratory Exercise 2

2D IMAGE COMPRESSION USING TRANSFORM AND SUBBAND FILTERS

April 18, 2009

Submitted By

Group 43

Rayyan Ali Qureshi

Department of Signals & Systems

Chalmers University

rayyan@student.chalmers.se

Sohaib Maalik

Department of Signals & Systems

Chalmers University

sohaib@student.chalmers.se

Submitted To

Irene Y. H. Gu

Department of Signals & Systems

Chalmers University

irenegu@student.chalmers.se

Table of Contents

List of Figures

1. Introduction

This lab focuses on how first generation image compression methods can be used to achieve compression. Compression implies the removal of significantly smaller energy elements without disturbing the image properties like textures, edges etc [1].

2. Task 1 – DFT Based Image Compression

2.1. Plot of Original 2D Image

Figure 1: Plot of Original 2D Image

Plot of original 2D image of the given file is shown in Figure 1.

2.2. Plot of Intermediate Image

After applying DCT on rows of original 2D image, the resulting image is shown in Figure 2.

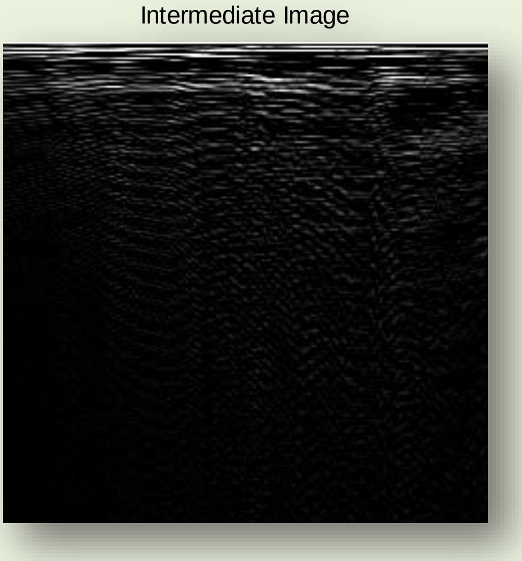

Figure 2: Plot of Intermediate Image

2.3.Plot of 2D DCT transformed Image

After applying DCT on the columns of intermediate image obtained in last step, the resulting image is the 2D DCT transformed image which is shown in Figure 3.

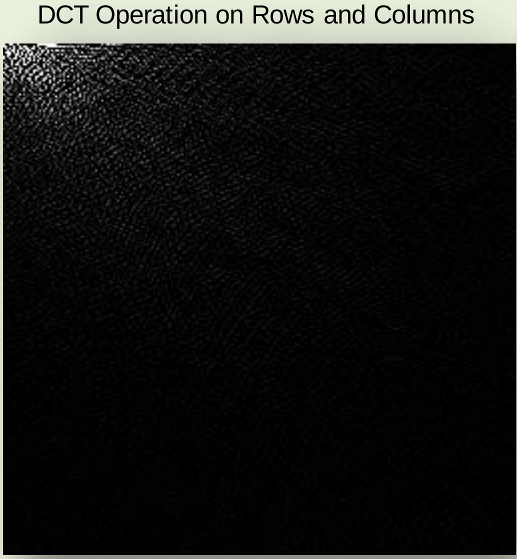

Figure 3: Plot of 2D DCT Transformed Image

2.4.Plot of IDCT on Columns

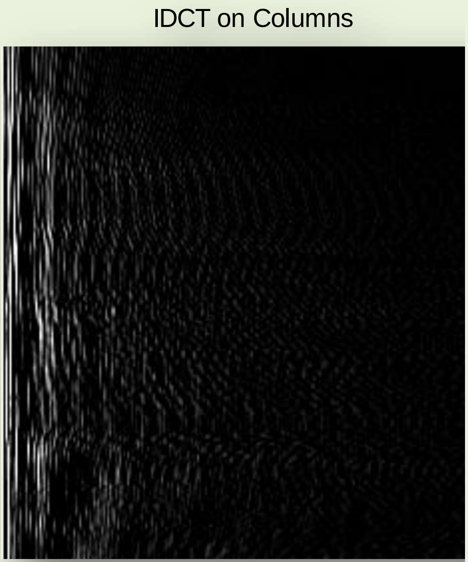

Figure 4: Plot of IDCT on Columns

After applying the inverse DCT on the columns of the 2D DCT Transformed Image, the result is shown in Figure 4.

2.5. Plot of IDCT on Rows & Columns of Transformed Image

The last step in this task was to apply IDCT on the rows of the intermediate image of Figure 4. The resulting image was the 2D inverse DCT transformed image shown in Figure 5.

IDCT Operation on Rows and Columns

Figure 5: 2D IDCT Transformed Image

2.6. Mean Square Error (MSE)

Mean square error between the original and IDCT transformed image was found to be;

$$MSE = 3.45105062488544 \times 10^{-32}$$

2.7. Absolute Threshold (th)

The absolute threshold value using which 90% of the DCT coefficients were removed was found to be;

$$th = 0.1047$$

2.8. Peak Signal-to-Noise Ratio (PSNR)

The peak signal-to-noise ratio was calculated to be;

$$PSNR = 29.3335707221722\ dB$$

2.9.Plot of Original, Compressed & Error Image

Original 2D Image

Figure 6: Original 2D Image

Compressed Image

Figure 7: Compressed Image

30X Enlarged Error Image

Figure 8: 30x Enlarged Error Image

2.10. Comments

Figure 6, Figure 7 and Figure 8 shows the comparison between Original, Compressed and the Error Image respectively. It is clearly discernible from Figure 7 that the compressed image is still of very good quality with the minor difference in the details which are a bit blurred and not sharp. But nonetheless, texture is clearly recognizable. This is due to the fact that texture is created by low frequency components where as edges and sharpness is due to the presence of high frequency components. This can lead us to the result that image retains low frequency components but doesn't have high frequency components which were removed while compressing. The proof of this result is the error image which contains the high frequency components shown in Figure 8.

3. Task 2 – Block-based DCT-domain Image Compression

3.1. Plot of DCT Block Transformed Image

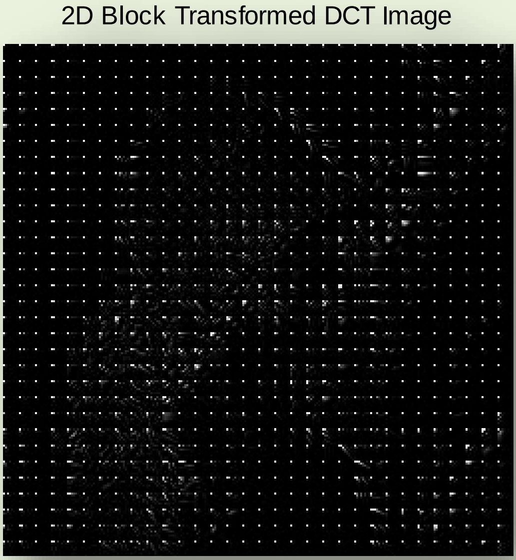

Figure 9: 2D Block Transformed DCT Image

3.2. Peak Signal-to-Noise Ratio

$$PNSR_4 = 32.4562707356365\ dB$$

3.3. Plot of Original, Compressed & Error Image Enlarged 30 Times

Figure 10: Original 2D Image

Figure 11: 2D Compressed Image

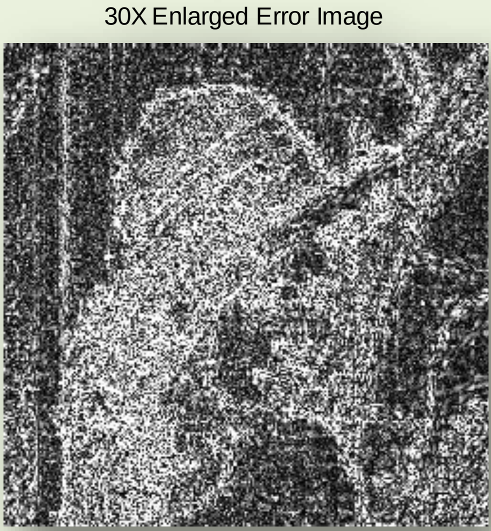

Figure 12: 30x Enlarged Error Image

3.4.Comparison & Comments

After comparing the results from both the previous tasks, it can be deduced that block based DCT gives a better result than the DCT applied to entire image. The compressed image as seen from Figure 11 is very clear and shows almost all the textures. The image contains both the low and high frequency components and is not blurred like the previous image in Task 1. Though there is one drawback of some blocked artifacts in the image of Figure 11. But the peak signal-to-noise ratio has increased by almost 3dB as compared to the Task 1. Also the error image in Figure 12 contains both the high and low frequency components whereas the error image of Task 1 contained only high frequency components.

4. Task 3 – Sub-band Image Compression Using Wavelets

4.1. Compression Ratio

$$Compression\ Ratio = 90\%\ or\ 0.9$$

4.2. PNSR Values

$$PNSR_{Wavlets} = \ 30.6360\ dB$$

$$PNSR_{Block\ DCT} = 32.4563\ dB$$

4.3. Plot of Original, Wavelet & Block DCT Compressed Image

Original Image

Figure 13: Original 2D Image

Wavelet Compressed Image

Figure 14: Wavelet Compressed Image

Block-based DCT Compressed Image

Figure 15: Block-based DCT Compressed Image

4.4.Comparison between Wavelet & DCT Compressed Images

Wavelet compressed image and block based DCT compressed image are shown in Figures 14 & 15 respectively. Following comments can be made by the comparison between two images;

Compared Element / Compression Type	Wavelet Based Compression	Blocked-based DCT Compression
Peak Signal-to-Noise Ratio (PSNR)	PSNR is worse	PSNR is better than the Wavelet based compression
Image Sharpness	Image obtained using this compression lacks high frequency components as a result of which the image is blurred and the edges are not clear	When compared the image obtained using this compression to the Wavelet based, the image is more clear and contains both high and low frequency components
Blocked Artifacts	There are no blocked artifacts in this image compression	A closer look reveals some blocked artifacts in this type of compression

4.5.Plots of Different Values of Threshold and Compression Ratios

Wavelet Compressed Image with CR=80% and TH=8.98

Figure 16: Wavelet Compressed Image

Wavelet Compressed Image with CR=20% and TH=.7715

Figure 17: Wavelet Compressed Image

4.6.Comments

It is very obvious from Figures 16 & 17 that with decrease in threshold, the compression ratio decreases and as a result the image quality gets better. But to a naked eye, this comparison is not much differentiable. Now by comparing Figure 16 with the compression ratio of 80%, to the image in Figure 17 with the compression ratio of only 20%, the quality of both the image seems almost the same when viewed at a scale of 65% to 100%.

5. Summary & Results

The purpose of this lab was to give deep insight of the first generation image compression methods. In Task 1 the compression was achieved by applying Discrete Cosine Transform (DCT) to the whole image. A discrete cosine transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies [2]. Using this method energy was compacted to lower frequencies in the transform domain. However the high frequency components with very less energy were discarded. Afterwards the image was reconstructed by taking the inverse DCT of the transformed image. The resulting image was a compressed one with blurred edges and diffused color but still recognizable. The resulting image lacked high frequency components as they were removed during the compression. Moreover there were no blocked artifacts in the compressed image.

In Task 2 compression was achieved using the technique called Block-based DCT. In this type of technique the image was divided into many small blocks and DCT was then applied to each and every block rather than to the whole image as in previous case. Compression was achieved in the same way by removing the high frequency components with comparatively lower energy from each block. Since there was very less variation of color in each block, the blocks were considered stationary. This lead to the fact that most of the energy will be contained in low frequency components rather than the high frequency components which contained almost zero energy. And since the number of low frequency components was very less, fewer coefficients were required to store them giving much better compression. The reconstructed image in this case, aside from the fact that it was highly compressed, was much better in quality with sharp and clear edges. The only drawback while using this method was the blocked artifacts in the compressed image.

In Task 3 another method for image compression was implied namely Sub-band compression using wavelets. This method utilizes the fact that most of the image energy is contained in low frequencies. So the image was filtered using sub-band filters into high frequency and low frequency bands containing the high frequency and low frequency components respectively. Afterwards each band was allocated with different bits with higher number of bits to the low frequency and lesser number of bits to the higher frequency band depending upon the information they contained. The analysis was performed afterwards using wavelets and DCT. The results comparison between the wavelets and DCT with same compression ratio showed the superiority of DCT. But with the increase in the number of levels for splitting the band and decreasing the compression ratio also showed good results in wavelets compression. One advantage of wavelet over the DCT was no blocked artifacts in the compressed image.

References

[1] Irene Y. H. Gu, Laboratory Exercise 2: *2D Image Compression Using Transform and Sub-band Filters,* Chalmers University, 2009

[2] http://en.wikipedia.org/wiki/Discrete_cosine_transform

Appendix – MATLAB Code

```
close all;
clc;
clear all;
%% Task 1
% step 1.1.1
orgimg=imread('lena.bmp');                              %reading image
%figure;imshow(orgimg);title('Original Image')          %displaying image
% step 1.1.2
cnvimg=mat2gray(orgimg);                                %converting image format
figure;imshow(cnvimg);title('Converted Image')          %converted image
[R C]=size(cnvimg);
imdimg=dct(cnvimg.');                                   %applying dct on rows
figure;imshow(imdimg);title('DCT on rows');
% step 1.1.3
fnlimg=dct(imdimg.');                                   %applying dct on coloumns
figure;imshow(fnlimg);title('DCT on rows and columns');
% step 1.2.1
invimdimg=idct(fnlimg);                                 %applying idct on columns
figure;imshow(invimdimg);title('IDCT on Columns');
diff=invimdimg-imdimg;                                  %yes they have small values
% step 1.2.2
invfnlimg=idct(invimdimg.').';                          %applying idct on rows
figure;imshow(invfnlimg);title('IDCT on rows and columns');
% step 1.2.3
error=abs(cnvimg-invfnlimg);                            %difference bw original and final image
MSE = sum(sum(error.^2))/(C*R);                         %MSE(3.4511e-032)
% step 1.3
dctcoeff=reshape(fnlimg,1,R*C);
[srtdctcoeff Indx]=sort(abs(dctcoeff));                 %sorting dct coefficients in ascending order
Th=srtdctcoeff(floor(R*C*0.9));                         %setting the threshold(Threshold value=0.1047)
cmpfnlimg=fnlimg;
cmpfnlimg( abs(dctcoeff)<=Th ) = 0;                     %compressing the image by removing small
                                                        %value dct coefficients
% step 1.4
invcmpimdimg=idct(cmpfnlimg);                           %applying idct on columns
%figure;imshow(invcmpimdimg);title('IDCT on Columns');
%diff=invcmpimdimg-imdimg;                              %yes they have small values
% step 1.4.2
invcmpfnlimg=idct(invcmpimdimg.').';                    %applying dct on rows
figure;imshow(invcmpfnlimg);title('Compressed image');
% step 1.4.3
errorimage=abs(cnvimg-invcmpfnlimg);                    %difference bw original and compressed image
figure;imshow(30*errorimage);title('30 times enlarged error image');
MSE1 = sum(sum(abs(errorimage).^2))/(C*R);              %computing psnr(MSE1=0.0012)
PSNR = 10*log10(1/MSE1);                                %29.3336db
%% Task 2
% step 2.1.1
orgimg=imread('lena.bmp');                              %reading image
%figure;imshow(orgimg);title('Original Image')          %displaying image
% step 2.1.2
cnvimg=mat2gray(orgimg);                                %converting image format
figure;imshow(cnvimg);title('Converted Image')          %converted image
[R C]=size(cnvimg);
% step2.2
Blocksize=8;
RB=R/Blocksize;                                         %TO make 8*8 blocks
CB=C/Blocksize;
for i=1:RB
    for j=1:CB
       temp=cnvimg((i-1)*Blocksize+1:i*Blocksize,(j-1)*Blocksize+1:j*Blocksize);
       tempDCT = dct2(temp);                            %applying 2d dct to all 8*8 blocks and saving it in
bbi
       bbi((i-1)*Blocksize+1:i*Blocksize,(j-1)*Blocksize+1:j*Blocksize) = tempDCT;
    end
end
figure;imshow (bbi);title('2D block based transformed DCT image');
% step2.3
dctcoeff1=reshape(bbi,1,R*C);
[srtdctcoeff1 Indx]=sort(abs(dctcoeff1));               %sorting dct coefficients in ascending order
Th1=srtdctcoeff1(floor(R*C*0.9));                       %setting the threshold(Threshold value=0.0946)
cmpbbi=bbi;
cmpbbi( abs(dctcoeff1)<=Th1 ) = 0;
%step2.4
for i=1:RB
    for j=1:CB
       temp=cmpbbi((i-1)*Blocksize+1:i*Blocksize,(j-1)*Blocksize+1:j*Blocksize);
       tempIDCT = idct2(temp);%applying inverse 2d dct to all 8*8 blocks and saving it in invbbi
       invbbi((i-1)*Blocksize+1:i*Blocksize,(j-1)*Blocksize+1:j*Blocksize) = tempIDCT;
    end
end
figure;imshow (invbbi);title('2D inverse block transformed DCT image');%Block based compressed image
%step2.5
errorimage=abs(cnvimg-invbbi);                          %difference bw original and final image
```

```
figure;imshow(30*errorimage);title('30 times enlarged error image');
MSE2 = sum(sum(abs(errorimage).^2))/(C*R);              %computing psnr(MSE2=5.6803e-004)
PSNR1 = 10*log10(1/MSE2);                               %32.4563db

%% Task 3
close all;
clc;
clear all;
load ('waveletcompressedimage.mat')
orgimg=imread('lena.bmp');                              %reading image
%figure;imshow(orgimg);title('Original Image')          %displaying image
cnvimg=mat2gray(orgimg);                                %converting image format
figure;imshow(cnvimg);title('Converted Image')          %displaying image
[R C]=size(cnvimg);
wci = mat2gray(X);                                      %changing format of wavelet compressed image
figure;imshow(wci);title('Wavelet compressed Image')    %displaying image
errorimage=abs(cnvimg-wci);                             %error
%figure;imshow(30*errorimage);title('30 times enlarged error image')
MSE2=sum(sum(abs(errorimage).^2))/(R*C);                %calculating PSNR(MSE2=8.6377e-004)
PSNR = 10*log10(1/MSE2);                                %30.6360db
figure;
load wci207715.mat
wci95 = mat2gray(X);
imshow(wci95);title('wavelet compressed image with CR=20% and TH=.7715')
figure;
load wci80898.mat
wci9550 = mat2gray(X);
imshow(wci9550);title('wavelet compressed image with CR=80% and TH=8.98')
```