INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, NAGPUR

DIGITAL IMAGE PROCESSING PROJECT REPORT

Image Compression Using Discrete Cosine Transform

Submitted To:
Dr. Tapan Kumar Jain

Submitted By:
Chaitali Srivastava (BT19ECE056)
Arnav Kumar (BT19ECE069)
Abhishek Tilkar (BT19ECE054)

1 Abstract

In this Project, we tried to compress the given image. We are using DCT Discrete Cosine Transform to compress the image. There are 2 methods in DCT and we are using Transformation Matrix Technique to compress the image. :- Everytime we run the program we will get 2 images which will be an original image and compressed image.

2 Introduction

The discrete cosine transform (DCT) helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). The DCT is similar to the discrete Fourier transform: it transforms a signal or image from the spatial domain to the frequency domain.
The DCT is the most widely used transformation technique in image processing, and by far the most widely used linear transform in data compression. DCT data compression has been fundamental to the Digital Revolution. Uncompressed digital media as well as lossless compression had impractically high memory and bandwidth requirements, which was significantly reduced by the highly efficient DCT lossy compression technique, capable of achieving data compression ratios from 8:1 to 14:1 for near-studio-quality, up to 100:1 for acceptable-quality content. The wide adoption of DCT compression standards led to the emergence and proliferation of digital media technologies, such as digital images, digital photos, digital video, streaming media, digital television, streaming television, video-on-demand (VOD), digital cinema, high-definition video (HD video), and high-definition television (HDTV).

3 Procedure

1) We first take the image that we have to compress in .jpg format.
2) Since we did the exp with the help of the transformation matrix we first divide the image in blocks.
3) After that, we perform compression operation on each block for compression we divide the floor pixel value to 8 and 16 not only this we divide them from any number.
4) In the conversion of the transformation matrix, we first calculate the Discrete Cosine Transform dct() of each block and corresponding Inverse Discrete Cosine Transform idct() operation of each block.
5) After that, we combine the each block and form a compressed image.
6) We write the compress image in .jpg format by imwrite() function.
7) After that, we read each image and compare their size.

4 Code

```
clc;clear all;close all

[x,y] = uigetfile('*.jpg*','select the grayscale image'); % here ...
    x=filename;y=pathname
filewithpath=strcat(y,x);
img_temp =imread(filewithpath);
original_image=img_temp;
img=original_image;

%checking for colored image
imgr = rgb2gray(original_image);
figure
imshow(img)
title('original image')

imwrite(img,'org.jpg')

data= size(img,2);
samples_divide_eighth =(data /8);

coloumn_8 = [];

for colour=1:3 %color layer
  for i=1:size(img,1)
    rowwise = dct(double(img(i,:,colour)));
    coloumn_8(i,:,colour)= idct(rowwise(1:samples_divide_eighth),data);
  endfor
endfor
dat_2 =size(img,1);
samplesEighth=floor(dat_2/8);
coloumn_8f=[];

for k=1:3 %all color layer
  for i=1:size(img,2) %all coloumn
    DCT_8 = dct(double(coloumn_8(:,i,k)));
    coloumn_8f(:,i,k)= idct(DCT_8(1:samplesEighth),dat_2);
  endfor
endfor

dct_1 =(uint8(coloumn_8));

%dct_1=rgb2gray(dct_1);
figure
imshow(dct_1)
title('COMPRESSED IMAGE');

imwrite(dct_1,'OUT.jpg')
```

5 Discussion and Observation

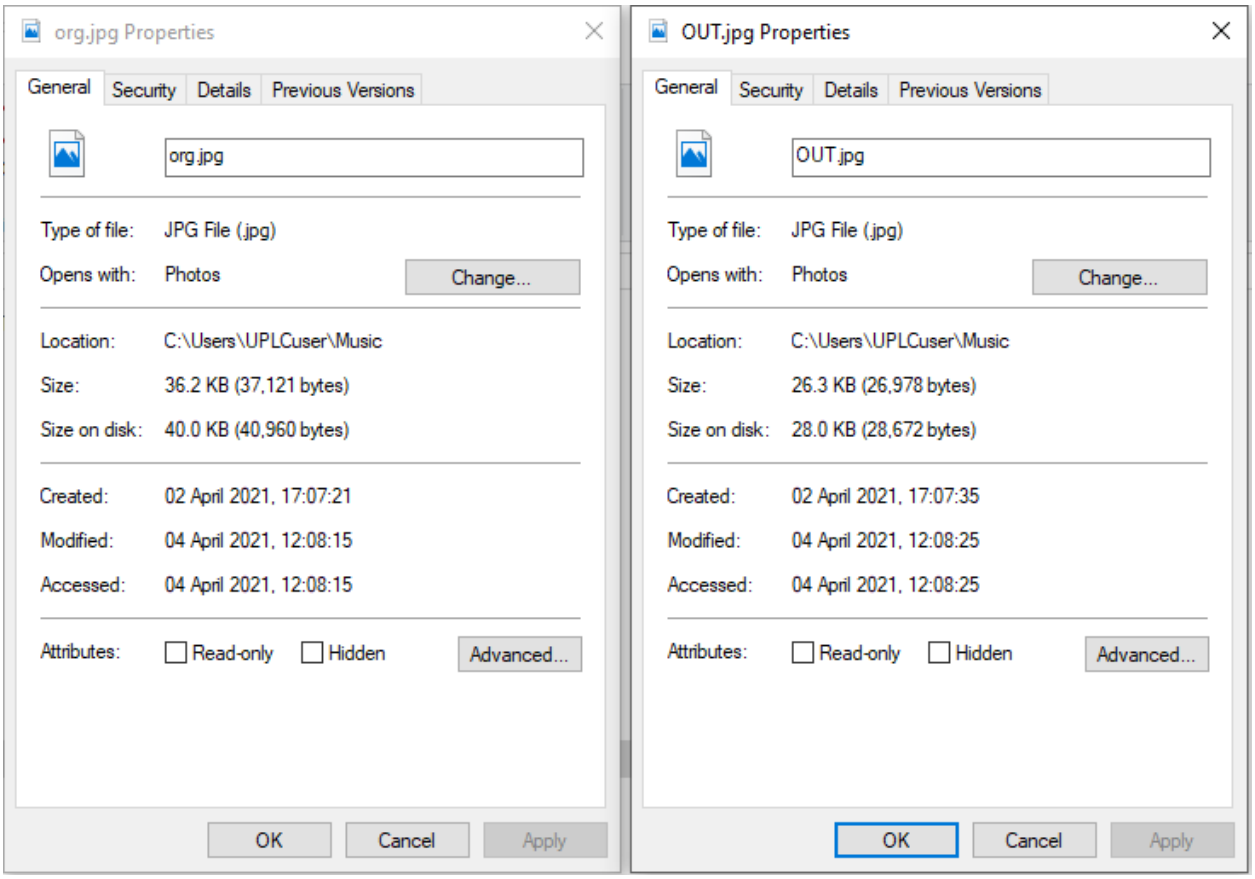

Figure 1: Size Comparison

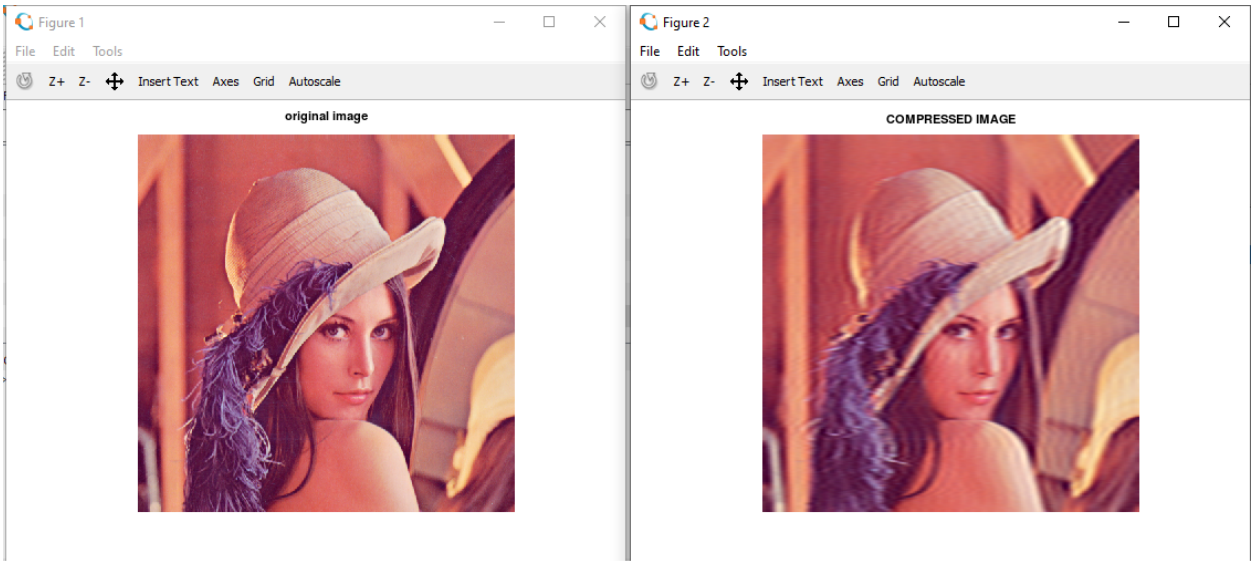

Figure 2: Image Comparison

1) By the observation of image and compress image we see that before compression the image size is something 36.8Kb and after truncation it size is 26.8kb.
2) We also observed that the quality of compress image is decreasing gradually as we increase the no. of block.
3) It looks blurry as we increase the number of block and the size is also decreasing.

6 Problems faced

1) We first started doing the experiment with the help fft (fast Fourier transform) but in this we see that it will work on only those images whose size is small.
2) Apart from this we got some errors that can only be handled by the MATLAB software but the inability of this software we cannot resolve.
3) We also tried to resolve the error with help of an online MATLAB simulation but the online MATLAB simulation was not working finely.
4) Finally we have done our exp. With help of transformation matrix in octave software.
5)In this we got an error i.e (out of memory) but we have successfully remove this error.

7 Conclusion

Image Compression using Discrete Cosine Transform (DCT) has been successfully done.

8 References

1) https://ieeexplore.ieee.org/documen t/7916121
2) https://ieeexplore.ieee.org/documen t/6037249
3) https://www.youtube.com/watch?v=zP6bqpDV_7U
4) https://www.youtube.com/watch?v=ZWQ8TbePN2U