Image compression using DCT

Akshat Rana
180102090

TABLE OF CONTENTS

Types of Compressions

1. **<u>Lossless Compression</u>:** The reconstructed image after compression is identical to the original image. Lossless compression can only achieve a modest amount of compression. Lossless compression is preferred for medical imaging, technical drawings, clip art or comics.

2. **<u>Lossy Compression</u>:** An image reconstructed following lossy compression contains degradation relative to the original because the compression scheme discards redundant information. Lossy schemes are capable of achieving much higher compression and are suitable for natural images such as photos and in applications where minor loss of data is acceptable.

Discrete Cosine Transform(DCT)

1. Discrete cosine transform(DCT) represents the image as a sum of cosines of different magnitudes and frequencies.

2. DCT coefficients are real valued while DFT coefficients are complex.

3. DCT is used in the international standard image compression algorithm know as **JPEG**.

4. Image compression using DCT transform is a lossy process.

5. The DCT is an orthonormal transform.

for $\alpha(0)=1/\sqrt{2}$,$\alpha(k)=1$ for $k\neq 0$

$$y(k) = \sqrt{\frac{2}{N}}\,\alpha(k)\sum_{n=0}^{N-1} x(n)\cos\frac{(2n+1)k\pi}{2N}; \quad k = 0,1,\ldots N-1$$

DCT-2 transform

$$\mathbf{b}_k = \left\{\sqrt{\frac{2}{N}}\,\alpha(k)\cos\frac{(2n+1)k\pi}{2N}\right\}_{n=1,2,\ldots,N-1} \qquad for\ k = 0,1,\ldots,N-1$$

Basis vector

Why DCT for image compression?

1. DCT has the property that most of the visually significant information about the image is concentrated in just a few coefficient of the DCT.
2. This property is known as Energy compaction property.
3. So the unnecessary coefficients can be discarded or scaled without making much difference.
4. That's why DCT is important in image and video compression techniques.

- Eyes cannot see high frequency changes in the image clearly.
- Eyes are also unable to see chrominance(colour part of image) very well.

2D Forward Discrete Cosine Transformation(FDCT)

The 2D DCT of an MxN image f(x,y) is given by the following equation:

$$F(p,q) = \alpha_p \alpha_q \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{\pi(2x+1)p}{2M}\right] \cos\left[\frac{\pi(2y+1)q}{2N}\right]$$

where 0<=p<=M-1 and 0<=q<=N-1

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & 1 \le p \le M-1 \end{cases} \qquad \alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & 1 \le q \le N-1 \end{cases}$$

2D Inverse Discrete Cosine Transformation(IDCT)

The 2D inverse DCT of an MxN image f(x,y) is given by the following equation:

$$f(x,y) = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q F(p,q) \cos\left[\frac{\pi(2x+1)p}{2M}\right] \cos\left[\frac{\pi(2y+1)q}{2N}\right]$$

where 0<=x<=M-1 and 0<=y<=N-1

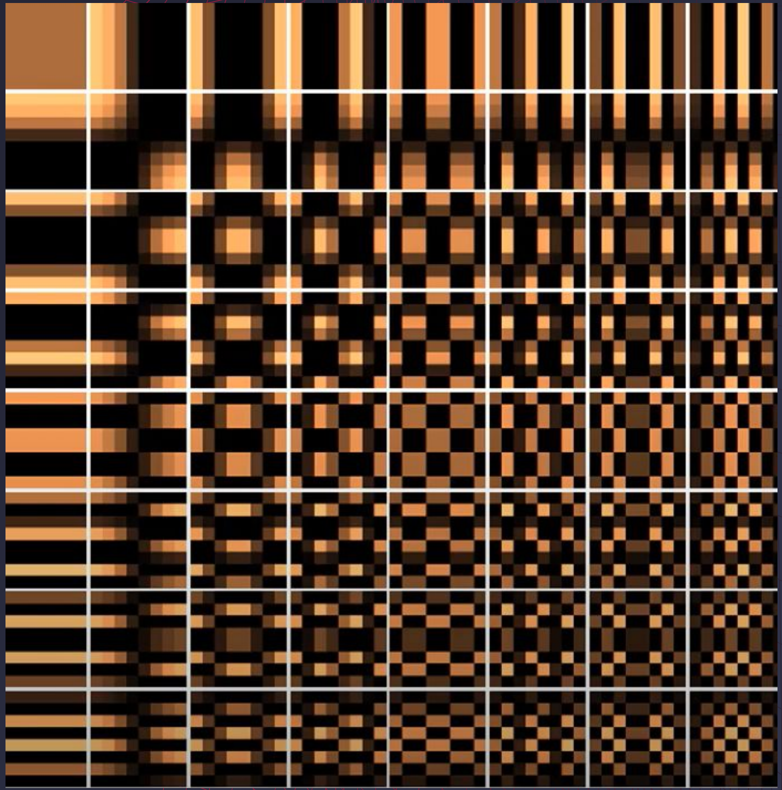
$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & 1 \le p \le M-1 \end{cases} \qquad \alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & 1 \le q \le N-1 \end{cases}$$

2D DCT Basis Function

$$\alpha_p \alpha_q \cos\left[\frac{\pi(2x+1)p}{2M}\right] \cos\left[\frac{\pi(2y+1)q}{2N}\right]$$

Basis function for 8x8 matrix is shown in the figure.

- Horizontal frequencies increase from left to right.
- Vertical frequencies increase from top to bottom.
- The constant valued basis function at the upper left corner is the DC basis function.
- Rest of the coefficients are called AC coefficients.



Computing DCT of an image (Transformation matrix approach)

- Original image is divided into blocks of 8 x 8.
- Then we compute DCT transformation matrix T for MxM segment by the following equation:-

$$T = \begin{cases} \dfrac{1}{\sqrt{M}} & p = 0, \quad 0 \le q \le M-1 \\ \sqrt{\dfrac{2}{M}} \cos\left[\dfrac{\pi(2q+1)p}{2M}\right] & 1 \le p \le M-1, \quad 0 \le q \le M-1 \end{cases}$$

- DCT is applied to each block by multiplying the modified block with DCT matrix on the left and transpose of DCT matrix on its right:-

F(p,q)=T*f(x,y)*T’

Transformation matrix approach(Continued)

- Each block is then compressed by completely removing the high frequency components.
- Since T is orthonormal, it's inverse is equivalent to its transpose.
- Inverse DCT is used for decompression. IDCT is applied to each block by multiplying the modified block with matrix T' on the left and matrix T on its right.

$$f(x,y)=T'*F(p,q)*T$$

T =

0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536
0.4904	0.4157	0.2778	0.0975	-0.0975	-0.2778	-0.4157	-0.4904
0.4619	0.1913	-0.1913	-0.4619	-0.4619	-0.1913	0.1913	0.4619
0.4157	-0.0975	-0.4904	-0.2778	0.2778	0.4904	0.0975	-0.4157
0.3536	-0.3536	-0.3536	0.3536	0.3536	-0.3536	-0.3536	0.3536
0.2778	-0.4904	0.0975	0.4157	-0.4157	-0.0975	0.4904	-0.2778
0.1913	-0.4619	0.4619	-0.1913	-0.1913	0.4619	-0.4619	0.1913
0.0975	-0.2778	0.4157	-0.4904	0.4904	-0.4157	0.2778	-0.0975

Transformation matrix approach(Continued)

As the DCT has very good energy compaction capability, it means the information is carried out by fewer DCT coefficients, therefore large amount of coefficients can be neglected to achieve image compression.

45	18	47	41	14	11	37	32
13	11	43	12	26	8	10	15
20	19	31	39	17	12	34	47
27	15	28	33	5	17	27	35
45	34	26	19	1	49	39	21
13	7	1	46	4	21	22	17
40	8	12	41	40	28	38	13
47	43	5	26	1	2	6	11

8x8 Luminance matrix

189.3750	11.2625	14.2389	-11.7633	12.1250	39.3032	0.7317	-11.4883
13.0005	-9.0938	-2.5408	-34.7074	-7.2808	-0.7041	18.5921	27.2826
-5.1750	25.9718	-3.8029	-0.9106	14.8950	-3.1129	-8.3295	15.2390
7.4729	-8.1367	-9.8750	15.0937	-6.6267	17.6210	4.7798	-4.7374
12.3750	23.6339	30.5381	2.1903	-12.3750	3.7889	4.8043	-10.2701
20.5578	-16.2111	-7.0980	-3.7616	19.3353	18.7785	-5.6627	4.5097
-3.6743	-3.2712	9.9205	-19.1920	8.8485	6.8259	-10.6971	-21.4293
32.7236	-2.6000	9.6278	14.0576	-8.3389	3.7411	0.0111	3.7216

8x8 matrix T*f(x,y)*T’

MATLAB Code

```
%select the file
[file,path]=uigetfile('*.*',"Select image");
getfile=strcat(path,file);
img=double(imread(getfile));

%performing the DCT on the image blockwise
T=dctmtx(8);
dct= @(block_struct) T*(block_struct.data)*T';
D=blockproc(img,[8 8],dct);

% masking the higher frequency component
mask_matrix=[ 1 1 1 1 0 0 0 0
              1 1 1 0 0 0 0 0
              1 1 0 0 0 0 0 0
              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];
```

```
%performing the DCT on the image blockwise
C=blockproc(D,[8 8],@(block_struct)(mask_matrix.*block_struct.data));
invdct=@(block_struct) T' * (block_struct.data)*T;
invD=blockproc(C,[8 8],invdct);

%saving the images
imwrite(uint8(img),"original.jpg",'quality',100);
imwrite(uint8(invD),"compressed.jpg",'quality',100);

%displaying them together for comparison
figure
imshowpair(img,invD,'montage')
title('Original(left) and Compressed(right)');
```

Disadvantages of using DCT compression

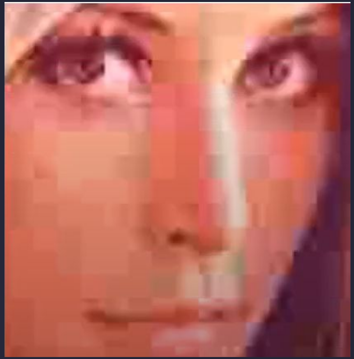
1. **Blocky compression artifacts**: The DCT algorithm can cause block based artifacts when heavy compression is applied.

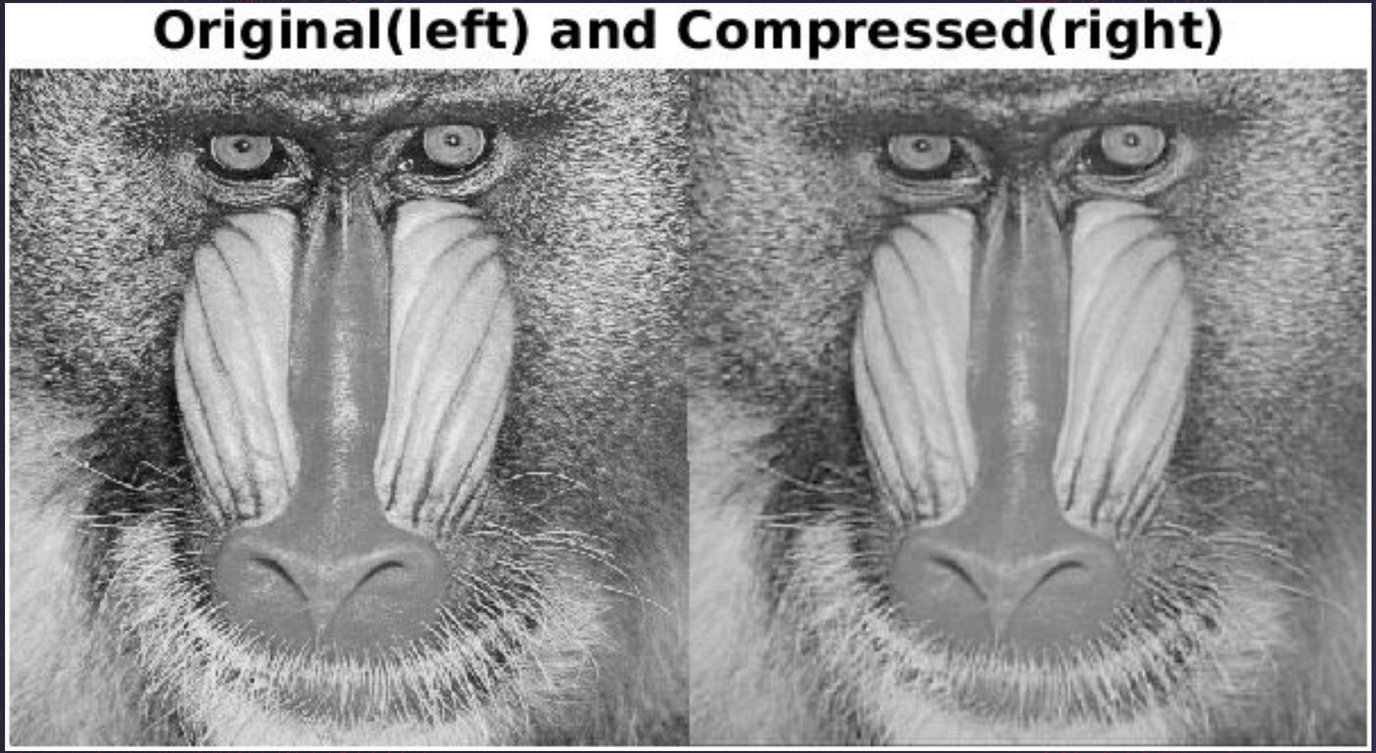
2. Truncation of the higher spectral coefficients results in blurring of the images, especially whenever the details are high.

3. Coarse quantization of some of the low spectral density introduces graining in the smooth portion of the image.

4. JPEG algorithm does not work well for text.

Blocky Compression Artifacts




Original(left) and Compressed(right)

THANKS!

Does anyone have any questions?

Bibliography

- https://matlab.mathworks.com/
- http://www-math.mit.edu/~gs/papers/dct.pdf
- http://ethesis.nitrkl.ac.in/1731/1/project.pdf
- https://www.youtube.com/watch?v=LFXN9PiOGtY&list=PLzH6n4zXuckoAod3z31QEST1ZaizBuNHh