RESEARCH PAPER | Engineering | Volume : 5 | Issue : 11 | November 2015 | ISSN - 2249-555X

Comparative Analysis of FFT and DCT Performances in Image Compression and Evaluation of Their Performances

KEYWORDS: Discrete Fourier Transform, Fast Fourier Transform, DCT, Image compression , Border Distortion,dataRedundancy

Dr NSSR Murthy	Dr IV Muralikrishna
Professor Mumtaz college of engg & technology, Malakpet, Hyderabad	Former director, IST JNTU University , Hyderabad

ABSTRACT *Compression of images is an important task for Data compression in digital image processing . It reduces redundancies in data so that data storage requirements will be minimized and there by cost for storing data also will be reduced . This is equivalent to increasing capacity of the storage medium and helps in efficient transmission of data at lower costs. . Development of efficient compression techniques is very important for this . There are so many tools to compress data efficiently. This paper tries to analyse and study various image compression techniques . There are different techniques in image compression of an digital image which includes DFT, FFT, DCT , DWT.Wavelets and fractal compression etc, each one of them has an advantage in their domain. This paper tries to compare and analyse DFT and FFT transforms and their applications to image compression .*

INTRODUCTION Data compression on digital images can be achieved by image compression. The objective of image compression is to reduce redundancy of the image data so that storage or transmition of data can be done efficiently than in original un compressed image. Image compression is basically lossy or lossless. Lossless compression is much preferred at low bit rates . Lossy compression methods when used at low bit rates, introduce compression artifacts. Lossless compression is also preferred for high value content like medical imagery. Lossy methods are more suited natural images such as photos. In lossless method Loss less compression can be achieved by Run-length encoding and entropy encoding. Lossy compression can be done by Transform coding, where applying Fourier-related transform such as DFT or FFT or the wavelet transform followed by quantization A general compression model is shown in figure 1.

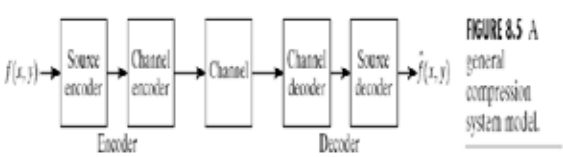


FIGURE 8.5 A general compression system model.

Figure1

It contains an encoder and decoder sub blocks . The encoder is made up of source encoder, which removes input redundancies, and a channel encoder, which increases the noise immunity of the source encoder output. The decoder contains a channel decoder followed by a source decoder. If the channel between the encoder and decoder is noise free, the channel encoder and decoder are omitted, and the general encoder and decoder is noise free, the channel encoder and decoder are omitted, and the general encoder and decoder become the source encoder and decoder, respectively. The source encoder is responsible for reducing coding, interpixel, or psychovisual redundancies in the input image. The approach can be modeled by a series of three independent operations. Source encoder and Source decoder In the first stage of the source encoding process, the mapper transforms the input data into a (usually non-visual) format designed to reduce inter pixel redundancies in the input image. This is reversible and may or may not reduce directly the amount of data required to represent to represent the image. The second stage or quantizer block reduces the accuracy of the mapper's output in accordance with some pre fixed fidelity criterion. This stage reduces the Psychovisual redundancies of the input image. This is irreversible. The final stage of source encoding processes, the symbol coder creates a fixed or variable-length code to represent the quantizer output and maps the output in accordance with the code In most cases, a variable length code is used to represent the mapped and quantized data set. It assigns the shortest code words to the most, frequently occurring output values and thus reduces coding redundancy. This is completely reversible. After symbol coding, the input image has been processed to remove each of the three redundancies. It is shown that the source encoding processes consist three successive operations, but all three operations are not necessarily included in every compression. The source decoder shown in figure contains only two components: a symbol decoder and an inverse mapper. These blocks perform the inverse operations of the source encoder's symbol encoder and mapper blocks. No inverse quantization is applied as it is irreversible .

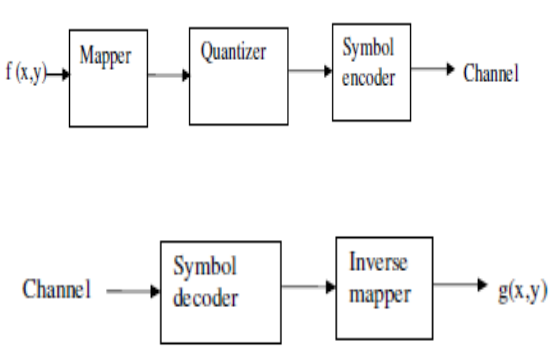


Figure 2 source encoder & Decoder

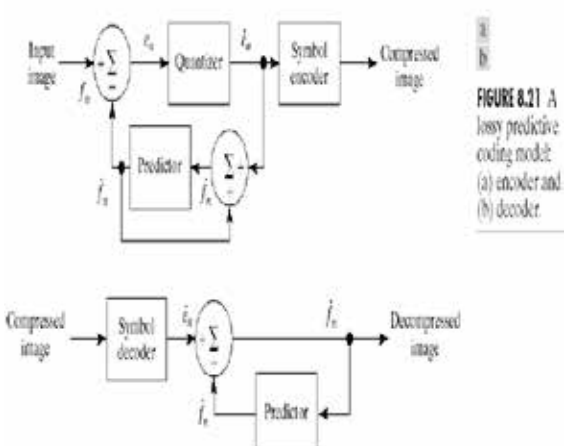
Channel Encoder and Decoder : The channel encoder and decoder play an important role in the overall encoding-

Volume : 5 | Issue : 11 | November 2015 | ISSN - 2249-555X

decoding process when the channel of above figure 1 is noisy or prone to error. They are designed to reduce the impact of channel noise by inserting a controlled form of redundancy into the source-encoded data. As the output of the source encoder contains little redundancy,

Lossless Image compression : A loss of information can be avoided totally in lossless compression, where image data are reduced while image information is totally preserved. It uses the predictive encoding which uses the gray level of each pixel to predict the gray value of its right neighbour. Only the small deviation from this prediction is stored. This is a first step of lossless data reduction. Its effect is to change the statistics of the image signal drastically. Statistical encoding is another important approach to lossless data reduction.Statistical encoding can be especially successful if the gray level statistics of the images has already been changed by predictive coding. The overall result is redundancy reduction, that is reduction of the reiteration of the same bit patterns in the data. Of course, when reading the reduced image data, these processes can be performed in reverse order without any error and thus the original image is recovered. Lossless compression is therefore also called reversible compression. A loss of information is, however, totally avoided in lossless compression, where image data are reduced while image information is totally preserved.

Lossy image compression : Lossy data compression has of course a strong negative connotation and sometimes it is doubted quite emotionally that it is at all applicable in medical imaging. In transform encoding one performs for each image run a mathematical transformation that is similar to the Fourier transform thus separating image information on gradual spatial variation of brightness (regions of essentially constant brightness) from



a
b

FIGURE 8.21 A lossy predictive coding model: (a) encoder and (b) decoder.

Figure 3. lossy compression model

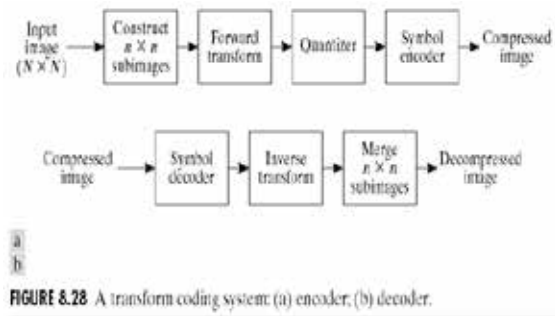
information with faster variation of brightness at edges of the image (compare: the grouping by the editor of news according to the classes of contents). In the next step, the information on slower changes is transmitted essentially lossless (compare: careful reading of highly relevant pages in the newspaper), but information on faster local changes is communicated with lower accuracy (compare: looking only at the large headings on the less relevant pages). In image data reduction, this second step is called quantization. Since this quantization step cannot be reversed when decompressing the data, the overall compression is 'lossy' or 'irreversible'.

Need for compression : Compression is necessary in modern data transfer and processing whether it is performed on data or an image/video file as transmission and storage of uncompressed video would be extremely costly and impractical.. Hence, Compression of images while maintaining the image quality is must for digital data, image or video file transfer in fast way and lesser amount of time.

Image Compression Transforms :

2-D transforms:

A reversible linear transform (such as Fourier Transform) is used to map the image into a set of transform coefficients. These coefficients are then quantized and coded.The goal of transform coding is to decorrelate pixels and pack as much information into small number of transform coefficients.Compression is achieved during quantization not during the transform step



a
b

FIGURE 8.28 A transform coding system: (a) encoder; (b) decoder.

Figure 4. 2D transform and inverse transform model

Discrete Fourier Transform (DFT)

The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the *Fourier* or frequency domain, while the input image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image. The Fourier Transform is used in a wide range of applications, such as image analysis, image filtering, image reconstruction and image compression. The DFT is the sampled Fourier Transform and therefore does not contain all frequencies forming an image, but only a set of samples which is large enough to fully describe the spatial domain image. The number of frequencies corresponds to the number of pixels in the spatial domain image, *i.e.* the image in the spatial and Fourier domain are of the same size. For a square image of size N×N, the two-dimensional DFT is given by:

$$F(K,L) = \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} f(a,b) e - i2\Pi(ki/N + Lj/N)$$

where f(a,b) is the image in the spatial domain and the exponential term is the basis function corresponding to each point F(k,l) in the Fourier space. The equation can be interpreted as: the value of each point F(k,l) is obtained by multiplying the spatial image with the corresponding base function and summing the result. In a similar way, the Fourier image can be re-transformed to the spatial domain. The inverse Fourier transform is given by

Volume : 5 | Issue : 11 | November 2015 | ISSN - 2249-555X

$$f(a,b) = 1/N^2$$
$$\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} F(k,L)e2\Pi i(ka/N + lb/N)$$

Fast Fourier transform (FFT)
Fast fourier transform is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse. There are many distinct FFT algorithms involving a wide range of mathematics, from simple arithmetic to group theory and number theory. A DFT decomposes a sequence of values into components of different frequencies. This operation is useful in many fields but computing it directly from the definition is often too slow to be practical. An FFT is a way to compute the same result more quickly. The difference in speed can be substantial, especially for long data sets where N may be in the thousands or millions—in practice, the computation time can be reduced by several orders of magnitude in such cases, and the improvement is roughly proportional to N / log (N). This huge improvement made many DFT-based algorithms practical; FFTs are of great importance to a wide variety of applications, from digital and solving partial differential equations to algorithms for quick multiplication of large integers. It may be noted that the number of complex multiply and add operations required by the simple forms both the DFT and IDFT is of order N^2. This is because there are N data points to calculate, each of which requires N complex arithmetic operations. In computer science jargon, we say they have algorithmic complexity $O(N^2)$. This is not good news. If we can't do any better than this then the DFT will not be very useful for the majority of practical DSP applications. Fortunately, there are a number of different 'Fast Fourier Transform' (FFT) algorithms that enable us to do very much better than this.

For example, the popular 'Radix 2' algorithms are useful if N is a regular power of 2 ($N=2^p$). These algorithms have complexity of only $O(N.logN)$. If we (naively) assume that algorithmic complexity provides a direct measure of execution time (and that the relevant logarithm base is 2) then the ratio of execution times for the (DFT) vs. (Radix 2 FFT) can be expressed:

$$\frac{N^2}{Nlog_2N} = \frac{N}{log_2N} = \frac{2^p}{p}$$

Descrete cosine transform (DCT) :

$$\mathbf{D}_x\{f\}(\omega) = a(\omega)\sum_{x=0}^{N-1} f(x)\cos\left(\frac{(2x+1)\omega\pi}{2N}\right) \;;\; a(\omega) = \begin{cases}\sqrt{\frac{1}{N}} & \omega = 0\\ \sqrt{\frac{2}{N}} & else\end{cases}$$

This is similar to ID DFT But a 2D DCT is defined using the separability property as 1D transform on the rows and on the columns, applied separately

$$F(u,v) = \mathbf{D}_Y\left\{\mathbf{D}_X\left\{f(x,y)\right\}\right\}$$

One of the advantages of DCT is the fact that it is a real transform, whereas DFT is complex. This implies lower computational complexity, which is sometimes important for real-time applications. DCT is used in some lossy compression algorithms, including JPEG. DC is the matrix element (1,1), corresponding to transform value X(0,0). High spatial X and Y frequencies correspond to high column and row indexes, respectively. According to the DCT properties, a DC is transformed to discrete delta-function at zero frequency. Hence, the transform image contains only the DC component. This can be seen in the transform image. The DC value is a sum over the whole image. The majority of the DCT energy is concentrated on low frequencies. The reason is the fact that natural images possess mostly low-frequency features and high-frequency features (edges) are rarely encountered. The advantages of DCT compression are based on the fact that most natural images have sparse edges. Hence, most blocks contain primarily low frequencies, and can be represented by a small number of coefficients without significant precision loss. Edges are problematic since are associated with high spatial frequency. Consequently, the DCT at blocks where the edges pass has high-amplitude coefficients at high frequencies, which cannot be removed without significant distortion. This effect was seen on the coin image, where small number of coefficients resulted in very significant distortion of the edges.

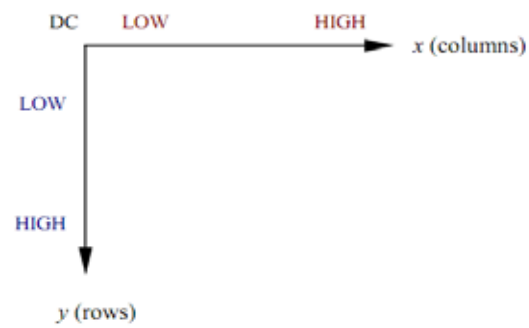


Fig 5 DCT

Methodology: The present work applies FFT and DCT and analyses output quality of image ofter compression with the above methods .

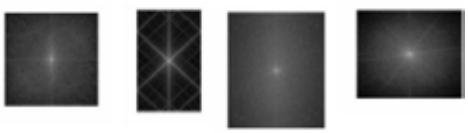
Step 1. First, convert the input MXN color image to the grayscale format. Plot the 2-D log magnitude of the 2D FFT and DCT of the grayscale image, with center shifted. Visually compare and comment on the similarity/differences among the images using the two transforms.

Strep2. Apply the truncation windows to keep 20% and 5% of the DFT and DCT coefficients, i.e. two different ratios for each transform. This truncation is done by keeping the coefficients of the lowest frequencies (those within a centered smaller rectangle of (M/2)x(N/2) and (M/4)x(N/4) on the shifted FFT, respectively). Apply the 2D inverse FFT to reconstruct the image for each of the truncated spectra. Compute the Signal-to-Noise-Ratio (SNR) value for each of the reconstructed images.

Baboon Monkey King Sunflower hexagon

Volume : 5 | Issue : 11 | November 2015 | ISSN - 2249-555X

Fig 6. Gray scale images of the above



FFT

DCT

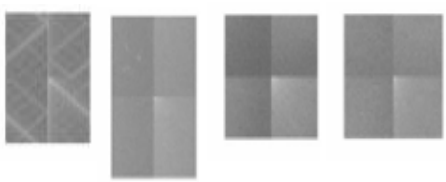
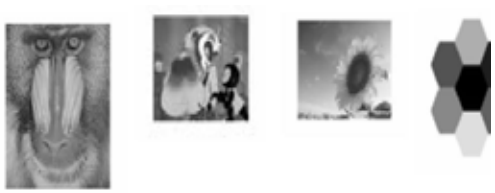


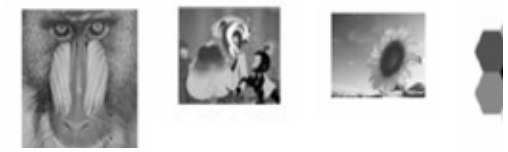
Fig 7. FFT and DCT of original images

A

B



C



D

Fig 8. Images after Inverse FFT coefficients A- 20% B-5% and
Inverse DCT coefficients C-20% and D -5%

Table1: SNR Values for (A),(B),(c),(D) in fig 8

Coefficient value %	Baboon	Monkey king	Sunflower	Hexagon
FFT with 20%(A)	16.9516	24.1643	22.3129	27.8421
FFT with 5% (B)	13.9172	18.1158	18.4930	23.8111
DCT with 20%(C)	14.4037	18.1723	18.9681	24.3310
DCT with 5%(D)	12.7320	13.8664	17.3661	19.9434

Conclusions

it is very clear that the larger the truncation window is, the better is the recovered image This is because more information is kept when the window is larger. We can also observe that the less texture image will have higher quality (higher SNR) in the recovered image , because we only keep the low frequency part and texture usually causes high frequency. We can also see clear pattern in both FFT and DCT of the hex image. The FFT and DCT of Baboon is most close to a uniform distribution, because it contains much texture so that the high frequency part is also quite bright. In general recovery performance of FFT is a little better than that of DCT.It reduces the data storage requirements Data security can also be greatly enhanced by encrypting the decoding parameters and transmitting them separately from the compressed database files to restrict access of proprietary information The rate of input-output operations in a computing device can be greatly increased due to shorter representation of data Data Compression obviously reduces the cost of backup and recovery of data in computer systems by storing the backup of large database files in compressed form Future Scope:

In this work we have considered only a comparative study of performances of FFT and DCT. This can be extended further to other transforms like wavelet transforms and and fast wavelet transforms to avoid macro blocking problems also.

REFERENCE [1] Implementation of Data Compression and FFT on TinyOS: Ning Xu, Embedded Networks Laboratory, Computer Science Dept. USC. Los Angeles [2] Hardware Implementation of a Lossless Image Compression Algorithm Using a Field Programmable Gate Array M. Klimesh,1 V. Stanton,1 and D. Watola1 (2001) [3] Implementation of Image Compression Algorithm using Verilog with Area, Power and Timing Constraints (2007-2009) : by Arun Kumar PS ,Dept. of ECE, NIT Rourkela. [4] Image Compression Using the Discrete Cosine Transform By Andrew B. Watson, NASA Ames Research Center (1994) [5] Comparative Analysis of Various Compression Methods for Medical Images : Rupinder Kaur, Nisha Kaushal (NITTTR, Chd.) 2007 [6] Fourier Analysis and Image Processing by Earl F. Glinn (2007) [7] Image compression using Fast Fourier Transform by Parminder Kaur, Thapar Institute of Engineering and Technology, Patiala. [8] Image Compression Using Curvelet, Ridgelet and Wavelet Transform, A Comparative Study by M.S. Joshi # , R.R. Manthalkar and Y.V. Joshi # Government College of Engineering, Aurangabad, (M.S. India), madhuris.joshi@gmail.com SGGS Institute of Engineering & Technology, Nanded , (M.S. India) [9] T.Sreenivasulu reddy, K.Ramani, S.Varadarajan and B.C.Jinaga, (2007, July). Image Compression Using Transform Coding Methods, IJCSNS International Journal of Computer Science and Network Security, VOL. 7 No. 7. [10] Ken Cabeen and Peter Gent, Image Compression and the Discrete Cosine Transform, Math 45, College of the Redwoods. [11] Mathematica Journal, (4)1, 1994, p.81-88 http://vision.arc.nasa.gov/ Image Compression Using Discrete Cosine Transform,Andrew B.Waston publications/mathjournal