8-10-2005

Exploring Discrete Cosine Transform for Multi-resolution Analysis

Safdar Ali Syed Abedi

EXPLORING DISCRETE COSINE TRANSFORM FOR MULTI-RESOLUTION ANALYSIS

by

SAFDAR ALI SYED ABEDI

Under the Direction of Saeid Belkasim

ABSTRACT

Multi-resolution analysis has been a very popular technique in the recent years. Wavelets have been used extensively to perform multi resolution image expansion and analysis. DCT, however, has been used to compress image but not for multi resolution image analysis. This thesis is an attempt to explore the possibilities of using DCT for multi-resolution image analysis. Naive implementation of block DCT for multi-resolution expansion has many difficulties that lead to signal distortion. One of the main causes of distortion is the blocking artifacts that appear when reconstructing images transformed by DCT. The new algorithm is based on line DCT which eliminates the need for block processing. The line DCT is one dimensional array based on cascading the image rows and columns in one transform operation. Several images have been used to test the algorithm at various resolution levels. The reconstruction mean square error rate is used as an indication to the success of the method. The proposed algorithm has also been tested against the traditional block DCT.

INDEX WORDS: Discrete cosine transform, Modified discrete cosine transform, Multi-resolution analysis, Noise reduction

EXPLORING DISCRETE COSINE TRANSFORM FOR MULTI-RESOLUTION ANALYSIS

by

SAFDAR ALI SYED ABEDI

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

Georgia State University

EXPLORING DISCRETE COSINE TRANSFORM FOR MULTI-RESOLUTION ANALYSIS

by

SAFDAR ALI SYED ABEDI

Major Professor:	Saeid Belkasim
Committee:	Rajshekhar Sunderraman
	A. P. Preethy

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
August 2005

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

The field of computer science is growing at the fastest pace since it started back in the early 20th century. We didn't have the actual computing machines until the mid of the 20th century but the algorithms for computing were already being developed from the beginning of the century. The advancement of the computing field has affected every imaginable field in the human lives. These fields include engineering, medicine, geology, meteorology, movies, and pictures etc. The high speed of digital computer has contributed to significant progress in the field of optics. Image processing by far has benefited significantly from the high speed digital computers [1].

Since the day the very first successful picture was taken in June of 1827, the quality and style of images have improved significantly. Niepce took this picture by using material that hardened on exposure to light [2]. There were many different ways to taking these images before the camera that we know today was created. The main problem with all of the images taken was the fact that with every picture that was taken a lot of redundant and useless information was also stored. The problems range from faded, blurry and noisy pictures. All these useless data stored with the images is termed as Noise. Noise, in technical terms is defined as the irrelevant or meaningless data.

Image restoration is one of the many branches of image processing. It is defined as a process that removes all the noise and irrelevant data from the images to make them clearer and better visible. The tools and techniques used in image processing are imported from signal processing. The main difference, however, between signal processing and image processing is

that when binary data is compressed, it is essential that we get the same data back when it is decompressed. With the images, on the other hand, it is not required. When an image is decompressed it is in most cases enough to get a replica of the image as long as the mean square error is within certain set limits and tolerable.

As mentioned above, there are a number of factors that can adversely affect the image quality. There have been a number of techniques introduced and being researched to enhance the image as well as to improve the usefulness of the data. Enhancement programs make the information more visible. One of the most popular enhancement techniques is the Histogram equalization. This technique redistributes the intensities of the image equally to the entire gray level range of the image [4].

Convolution is an image processing technique which is essentially a sequence mask operating on pixel neighborhood. Convolution involves High Pass and Low pass filters. Other image processing techniques include noise filters, trend removing filters, edge detection, and image analysis tools etc.

One of the important aspects of image processing is the reduction of the coded description of the image while keeping all the pertinent information. Data compression methods with zero information loss have been used on image data for some time. GIF, JPEG, MPEG etc are all examples of the tools used for data compression without the loss of the information.

There are a various techniques used these days to capture and store the images in a compressed format to reduce their storage sizes and use a smaller space. There are basically two categories of compression techniques; lossless and lossy. The main difference between these two categories is that in lossy compression the quality of the image is reduced in order to achieve higher compression ratio. On the other hand, in a lossless compression the quality of the image is

given a higher preference than the compression ratio. So it can be said that there is an apparent trade off between quality of the image and the compression ratio whenever we talk about the image compression. Lossy techniques include Transform Coding such as DCT, Wavelets and Gabor, Vector quantization, Segmentation and Approximation methods, Spline approximation methods i.e. Bilinear Interpolation/Regularization, Fractal coding which includes Texture synthesis, Iterative functions systems (IFS) and Recursive iterative functions systems (RIFS). Lossless compression techniques, on the other hand, includes Run length Encoding, Huffman Encoding, Entropy Coding (Lempel/Ziv) and Area coding [3].

Discrete cosine transform (DCT) has become the most popular technique for image compression over the past several years. One of the major reasons for its popularity is its selection as the standard for JPEG. DCTs are most commonly used for non-analytical applications such as image processing and. signal-processing DSP applications such as video conferencing, fax systems, video disks, and HDTV. DCTs can be used on a matrix of practically any dimension.

Mapping an image space into a frequency space is the most common use of DCTs. For example, video is usually processed for compression/decompression as 8 x 8 blocks of pixels. Large and small features in a video picture are represented by low and high frequencies. An advantage of the DCT process is that image features do not normally change quickly, so many DCT coefficients are either zero or very small and require less data during compression algorithms. DCTs are fast and, like FFTs, require calculation of coefficients. The entire standards employ block based DCT coding to give a higher compression ratio. Various different techniques and algorithms employed for DCT will be discussed in detail in section 3 where DCT will be explored.

As the topic of my thesis suggests, I will be exploring the possibilities of discrete cosine transforms for multi-resolution analysis. The aim here is to see if we can get the same results in compression using DCT as we can get by using the wavelets. The word multi-resolution refers to the simultaneous presence of different resolutions. The notion of multi-resolution was introduced by Mallat and Meyer in the years 1988-89. Multi-resolution analysis provides a convenient framework for developing the analysis and synthesis filters [5].

DCTs have been in used for compression for quite some time and have been very popular. As I mentioned earlier the main reason for DCTs popularity is the fact that it's been used as a standard in JPEGs. Wavelets are considered better than DCT when it comes to getting better results in compression. The supporters of MPEGs and JPEGs claim that DCT provides very good results as far as the compression is concerned. If this claim is in fact true then the question is that why are so many people interested in Internet protocol streaming. The reason IP streaming is really good that there are algorithms that provide a lot more compression with the same video quality as MPEG-2 does. We can take a wavelet algorithm and keep the same video quality and use 1 Mbps for a very nice high quality movie that would take us 3 Mbps with normal MPEGs with DCT [12].

The previous paragraph is good argument for choosing wavelets over DCT, however, the previous paragraph has been erroneous in the sense that researches use DCT and JPEG interchangeably. The arguments made in previous paragraph against DCT are in fact against the standard JPEG and not DCT. JPEG uses just a small part of DCT and should not be taken as standard DCT algorithms in comparisons with wavelets [13]. The basic difference between DCT and wavelets is that in wavelets rather than creating 8 X 8 blocks to compress, wavelets decompose the original signal into sub-bands. Wavelets are basically an optimizing algorithm for

representing a lot of change in the pictures. With DCT algorithm, the 8 X 8 blocks can lose their crisp edges, whereas, with wavelets the edges are very well defined. I don't really want to go in detail with the differences between wavelets and DCT but I just want to mention it to prove my case for the proposal of DCT for multi-resolution analysis. There is another compression method being developed call Fractals which is based on quadratic equations. This method is very well suitable with images which have patterns or a lot of repetitions.

Now, if the wavelets produce much better results than DCT then why do we need to try DCT for multi-resolution? The reason is that there are certain drawbacks to wavelets specially in terms of computation time required. For the highest compression rates, it takes a longer time to encode. The other reason is that MPEG is already a standard using DCTs and computer hardware comes with MPEGs built in. There is hardly any hardware available in the market these days which comes with wavelets as a built in standard.

The above arguments are my motivation for this research and this thesis to come up with an algorithm of multiresolutoin analysis for image processing which will have the efficiency and cost of the DCT and the compression results of the wavelets. The next chapter will focus on the multi-resolution analysis, its properties, benefits and its applications in the field of signal processing.

Chapter 2

MULTI-RESOLUTION ANALYSIS

The concept of multi-resolution analysis was formally introduced by Mallat [1989] and Meyer [1993]. Multi-resolution analysis provides a convenient framework for developing the analysis and synthesis filters [6]. The basic components for a multi-resolution analysis are: an infinite chain of nested linear function spaces and an inner product defined on any pair of functions. Multi-resolution has been widely used recently with great success with the wavelets. Wavelets and multi-resolution analysis have received immense attention in the recent years. There have been a lot of problems which have made use of wavelets and multi-resolution analysis and thus making it a popular scheme for compression. The basic idea behind multi-resolution analysis, as explained in [6], is to decompose a complicated function into smaller and simpler low resolution part together with wavelet coefficients. These coefficients are very important to recover the original signal when we apply the inverse.

Mallat [1989] described multi-resolution representation as a very effective method for analyzing the information content for images. Mallat and Meyer were the pioneers in the theory of multi-resolution analysis. For this reason most of this section is written under the influence of the paper written by Mallat in 1989. The original scale and size of objects in an image depends upon the distance between the image and the camera. To compress an image to a smaller size, we ought to keep the essential information of the image. In Mallat's words a multi-resolution decomposition enables us to have a scale-invariant interpretation of the image [7].

Multi-resolution analysis provides a hierarchical structure. It means that in order to get to 15 % compression, the image is not compressed directly to 15% as in block DCT, instead the image is compressed in stages; reducing the image to a half at every stage. At different resolutions the details of a signal generally characterize different physical aspects of the image or a signal per say. It is a common observation that at coarse resolutions the details correspond to larger overall aspects of the image while at fine resolutions the distinguishing features are prominent.

Some of the common applications of multi-resolution analysis are image compression, edge detection, and texture analysis. Multi-resolution analysis is not only restricted to the previously mentioned techniques but recently researchers also found some more applications of multi-resolution analysis and found good results. These applications include image restoration and noise removal. Multi-resolution analysis tries to understand the content of the image at different resolutions [8].

Wavelet analysis makes use of the notion of the multi-resolution analysis. Wavelet analysis is built on Fourier analysis and it was designed to overcome the draw backs of the Fourier analysis. According to Meyer [1992], the most powerful tool for the construction wavelets and for the implementation of the wavelet decomposition and reconstruction algorithms is the notion of multi-resolution analysis.

Fourier analysis has been a useful mathematical approach over the past several years, used in the analysis of the frequencies of the signal. Fourier analysis has been very effective in signals that statistically show the same characteristics over time and for those signals whose behavior does not change abruptly. It has been found, however, that Fourier analysis is not the best way to solve the problem of reconstruction and analysis in many aspects. One of the major

problems with using Fourier analysis is its lack of localization [9]. It is a common observation that sinusoidal waves are periodic which means that they are equal all over the spatial domain and hence it means that they are not localized. The drawback with Fourier analysis is that it is not very effective and accurate in handling abrupt peaks and sharp edges. The problem of localization is solved by wavelets. Wavelets are a small group of functions that form a good basis in certain spaces just like sines and cosines. The concept of multi-resolution is strongly related to the concept of wavelets. The main idea is to decompose the signals into a chain of nested subspaces [9].

The fundamental concept behind wavelets is to analyze according to scale [10]. As mentioned in the previous paragraph, wavelets are basically mathematical functions that satisfy the requirements of both time and frequency localization to have a better representation of data. With all the other forms of data and signal processing specially in images, either the stress is on the coarser parts or on the finer parts but none of the techniques considers the image as a whole with both the fine and coarse information. Multi-resolution analysis, wavelets in particular, looks at the images as a whole with trying to read both the coarse and fine information. If the image is of natural scenery, multi-resolution analysis would read both the smaller parts as trees, canals etc and also the main image for instance the village itself.

Based on the concept of multi-resolution, we have image pyramids. These pyramids are essentially simple structures for representing images at more than one resolution. The base of the pyramid, as we can guess, contains the image representations at the highest possible resolution, whereas the apex contains the low resolution approximation. The other technique that arises from the concept of multi-resolution analysis is sub-band coding. In this technique an image is decomposed into a set of components called the sub bands. These components are limited in their

bandwidth, which means that they cannot be greater than a certain pre-defined size. This is where the importance of multi-resolution analysis lies i.e. the reconstruction of the original image; reconstruction of the original image is done by up sampling, filtering and summing the individual sub-bands without the loss of any pertinent information [11].

As mentioned in the introduction section, researches have been erroneous in comparing JPEG with wavelets as a means of comparing DCT and wavelets [13]. Discrete cosine transforms and wavelets have been popular techniques used in signal processing for quite some time. Many researches have been comparing these two techniques to decide which one is superior. It has been an erroneous practice to compare wavelets with JPEGs as a way to compare wavelets and discrete cosine transform [13], [15]. The results of such comparisons have suggested that wavelets outperform DCT by a big margin. This is not a fair comparison due to the fact hat JPEGs only use a part of the discrete cosine transform. Discrete cosine transform is far more powerful and performance efficient than what JPEG can offer [13]. JPEG uses 8 X 8 blocks which are then transformed into 64 DCT coefficients [14]. This is a very low number because when we do the compression we make the coefficients 0s and in this case we are not left with much of the information when we convert some of the coefficients out of 64 to 0s. We can have far better results if we use larger block sizes such as 32 x 32.

It will only be fair to run a performance comparison between wavelets and discrete cosine transform because the hardware implementation of discrete cosine transform is very less expensive than wavelets and it is well acknowledged. The other important consideration while comparing wavelets and Discrete cosine transform is to keep the quantizer same. Various quantization schemes are available to be used with different transforms. A detailed analysis of these schemes can be found in [16].

Far better results can be obtained from discrete cosine transform than the standard JPEG if the new improved methods are employed. These methods include Q-matrix design, optimal thesholding and joint optimizations. The results from the experiments in [13] indicate that a gain of 1.7 dB was achieved with *Joint optimization* over the standard JPEG with the same bit rate. The standard JPEG is, in fact, not optimal and even JPEG with its limited DCT use can still be improved.

In order to stay compatible with JPEG in [13] DCT based embedded image coding is compared with wavelet based JPEG-like image coding. In wavelet based JPEG-like image coding, the DCT in baseline JPEG is replaced by a three level wavelet transform. This way the performance of the wavelet transform is achieved while staying compatible with JPEG.

Image coding mainly depends on what kind of entropy coder and quantizer is used rather than the difference between the wavelets and DCT. This observation is a benchmark in the comparison of DCT and wavelets. There is actually not a big difference between DCT and wavelets but in fact the difference lies in choice of quantizer and entropy coder. For still images, the difference between the wavelet transform and the discrete cosine transform is less than 1 dB and for video coding this difference tends to be even smaller [13].

This results obtained in [13] are very important because of the fact that JPEG and discrete cosine transform are erroneously used interchangeably in research and in comparing wavelets with DCT. DCTs can do much more and much better than what the baseline JPEG can offer. One of the major critiques of the standard DCT is the blocking effect that becomes more prominent at higher compression ratios. This blocking effect and other properties of DCT will be explored in detail in the next chapter. As already mentioned, the standard JPEG is based on DCT while an

improved version of JPEG, known as JPEG2000, is based on Wavelets and it is supposed to solve the problem of blocking artifacts in standard DCT.

There are quite many benefits of JPEG2000 over the standard JPEG. JPGE2000 makes use of the multi-resolution analysis and that's why it is pertinent to mention JPEG2000 here as well in comparison with the standard JPEG which used the standard 8x8 block DCT. The main features of JPEG2000 are: its superior low bit rate performance which enables it to attain higher compression without the loss of the information data, its multiple resolution representation, lossy to lossless compression, and region of interest (ROI) coding etc.

Below some examples for compressed images are shown at different bits per pixel to illustrate the difference between the 8x8 block DCT and wavelets based JPEG2000. It is very clear in these images that the blocking effect becomes more prominent at higher compressions and lower bit rates.

(a)

(b)

Figure 2.1: (a) block DCT at 12.5%; (b) JPEG 2000 at 12.5%

(a)

(b)

Figure 2.2: (a) block DCT at 25 %; (b) JPEG 2000 at 25%

Figures 2.1 and 2.2 clearly demonstrate the difference between the wavelets and the block DCT. It is also clear from the above images that higher the compression ratio, the more the blocking affects in the image. This should only be considered as the comparison between 8x8 block DCT and wavelets because there are several different new techniques that improve DCT considerably. These techniques will be explored in the next chapter.

Chapter 3

DISCRETE COSINE TRANSFORM (DCT)

The term “transform” means to change form or appearance. In terms of signal processing, a transform is normally a tool that is used to convert the signals from time domain or spatial domain to the frequency domain. There are various instances when it is pertinent to have the signal in the time domain and on the other instances it is important to have the signal in the frequency domain. For most of the image processing purposes it is better to have the signal in the frequency domain. In others words, a transformation can be described as the process of mapping the correlated data to no-correlated data. Each pixel in an image is correlated with its neighbor pixels. The information represented by any pixel should be predicted by its neighbors because of the fact that they are all correlated.

DCT has been a very popular transform for many years. The fact that DCT is a near optimal transform is the main reason for its popularity. The optimal transform up to now is Karhunen Loeve Transform (KLT). Even though it is optimal, it’s not widely adopted due to the fact that it is very expensive and slow. There have been some new algorithms proposed to improve the KLT to make it less expensive [24], [25], still the complexity in KLT is very high. DCT, the closest to KLT, is much faster to compute which has been the main driving force behind its popularity. DCT is also very closely related to the Discrete Foruier Transform (DFT). It is actually possible to compute DCT using DFT [15]. The main difference between DCT and DFT is that DCT only has real values and it is comparatively easier to compute.

Before formally describing the functioning to DCT, it will be beneficial to discuss the main parts of an image encoder and a decoder. Figure 3.1 shows the main parts of a transformation system.

All the parts of the transformation system will be discussed in detail later in this chapter. Some interesting new researches are on going about improving the transformation system which can be found in [17], [18], [19] and [20].

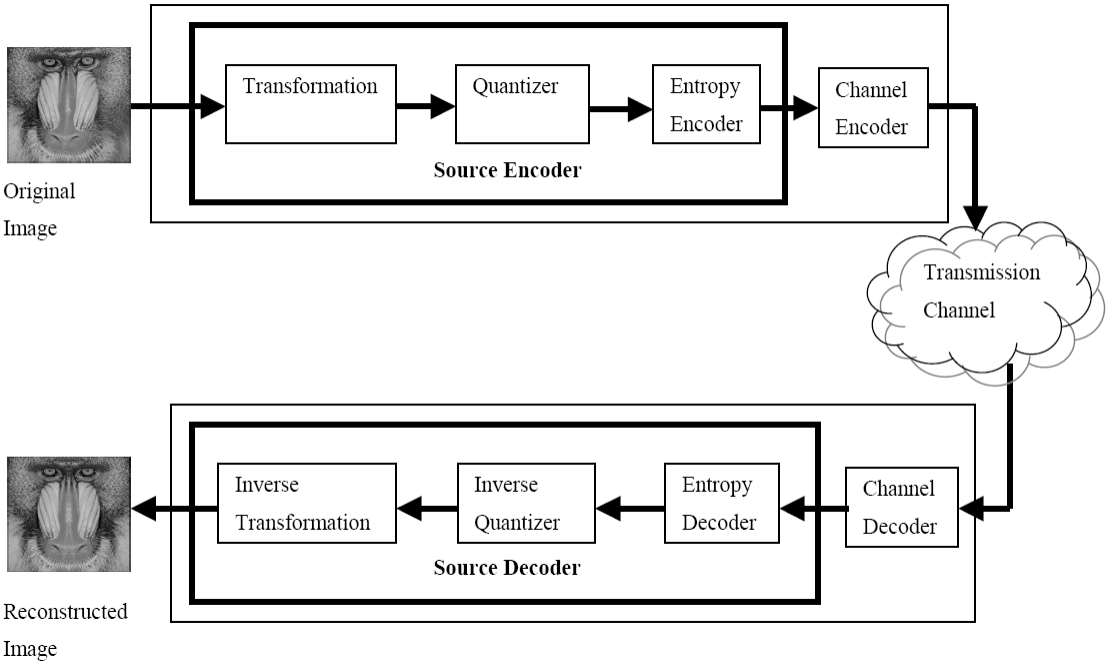


Figure 3.1. Taken from [26] shows a transformation system

As described in the introduction section, image compression is sometimes lossless and at other times lossy. In lossy compression some of the information is lost. The aim of any compression technique is to reduce the redundant amount of information from the signal during the compression process [22]. There are mainly three kinds of redundancies in images that are exploited in the process of compressing images.

3.1 Pyschovisual Redundancy

It has been observed over the years that there are certain sound frequencies that human cannot hear. For most humans the audible frequencies lie between 20 and 20,000 Hertz [23]. The sound signals outside this frequency range may still be audible to certain animals for example; dogs can hear sounds beyond the 20K Hz limit. So if the signal is meant for humans only then the signal outside this range is redundant or rather useless and can easily be ignored without the loss of any useful information. Humans have certain limitations when it comes to visuals as well. There are certain frequencies that humans can't see and thus are useless for them. This kind of redundancy is called Pyschovisual redundancy [22]. This redundant information can be eliminated from the signal because even though it reduces the quantity of the signal, it does not reduce the quality.

3.2 Coding Redundancy

Coding redundancy comes into play when the pixels are represented in binary information. The basic idea here is that the probability of the gray level is computed. The gray levels with higher probabilities are represented with fewer bits and more bits are used to represent the gray levels with lower probabilities. Compression is achieved without the loss of any information when coding redundancy is removed.

3.3 Interpixel Redundancy

This kind of redundancy deals with the fact that every pixel in an image is correlated. It is a well acknowledged fact that any pixel can be predicted with the information of the neighboring

pixels. This fact is further elaborated by the encoders such as Huffman, Run length etc. The basic idea of the run length encoder is that a sequence 1111110000000011 can be represented as 6 1s, 8 0s followed again by 2 1s. This way the space and bits needed to represent the sequence is reduced and hence the information is compressed.

3.4 DCT Formal Definition

As mentioned earlier, the DCT transform decorrelates the image data [26]. In DCT, an image is typically broken into 8x8 blocks. These blocks are each transformed into 64 DCT coefficients [14]. In [26], the most commonly used DCT definition of one dimensional sequence of length N is

$$C(u) = \alpha(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{\pi(2x+1)u}{2N}\right] \tag{3.1}$$

The above equation is defined for u = 0, 1, 2, 3…..N-1. There are two kinds of DCT coefficients; AC and DC. The DC coefficient corresponds to the value of C (u) when u = 0. In other words, DC coefficient provides the average value of the sample data [26]. The rest of the coefficients are called AC coefficients.

Based on the one dimensional DCT as described above, the two dimensional DCT can be achieved.

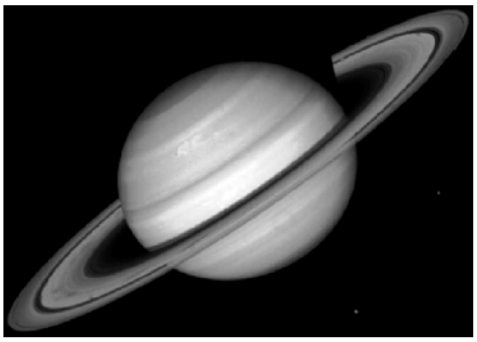
$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right] \tag{3.2}$$

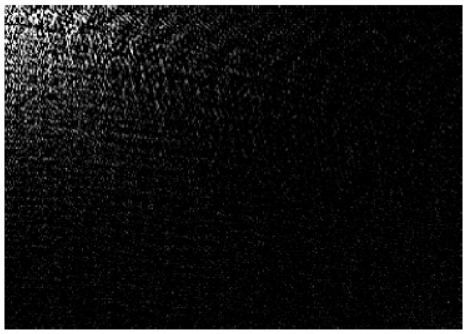
The above equation shows the two dimensional DCT. It is clear from the above equation that it is derived by multiplying the horizontal one dimensional basis function with the vertical one

dimensional basis function. Both one and two dimensional DCTs work in similar fashion. One dimensional DCT is used mainly in sound signals because of its one dimensional nature, whereas, two dimensional DCT is used in images because of their tow dimensional nature.

3.5 Advantages of DCT

DCT has many important properties that help significantly in image processing, especially in image compression. Even though, as described in [13], the transform itself does not compress the image, however, it can greatly help in achieving the desired compression or it can make the compression process considerably easier. One of these properties is *Energy Compaction.*





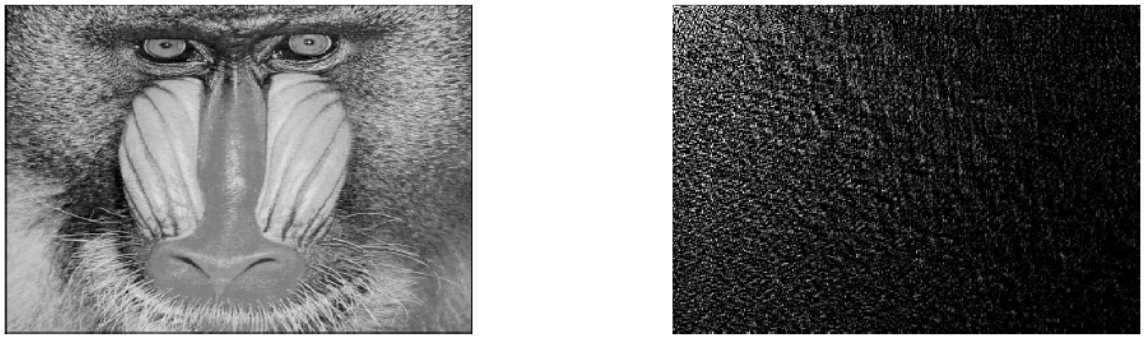


Figure 3.2. Images taken from [26]. Images show Energy Compaction Property of DCT

Energy compaction means that most of the pertinent information of the image is stored or compacted in top left corner of the image. In DCT, lower frequencies are on the top left corner and they contain most of the information. The high frequencies don't have much of the information needed to reconstruct the image. With this property it is easier for the quantizer and encoder to simply leave out the high frequencies as a means of compressing the image without losing the information. Since some of the data is lost or neglected, DCT results in Lossy compression.

Another important property of DCT is decorrelation. It was mentioned earlier that DCT is very good at removing the interpixel redundancy. This property is used in reducing the amplitude of the signals [26]. Figure 3.3 shows that the decorrelation property of DCT.

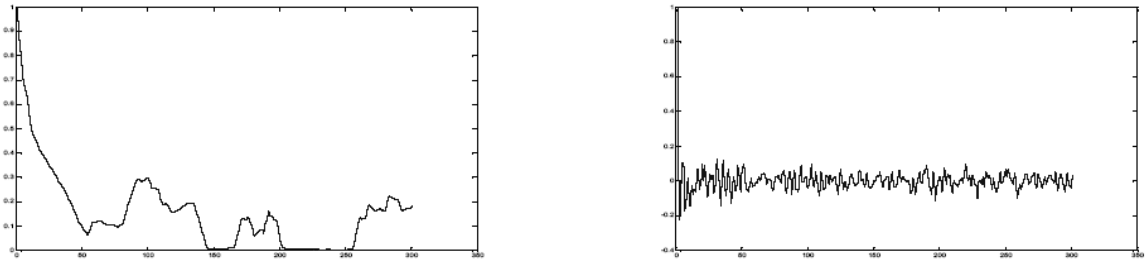


Figure 3.3: (Taken from [26]) depicts the decorrelation property of DCT

3.6 Disadvantages of DCT

One of the main problems and the criticism of the DCT is the blocking effect. In DCT images are broken into blocks 8x8 or 16x16 or bigger. The problem with these blocks is that when the image is reduced to higher compression ratios, these blocks become visible. This has been termed as the blocking effect. This is evident in figure 3.4. This image is compressed using 8x8 blocks and only 4 coefficients are retained. The blocking effect is very prominent in this image.

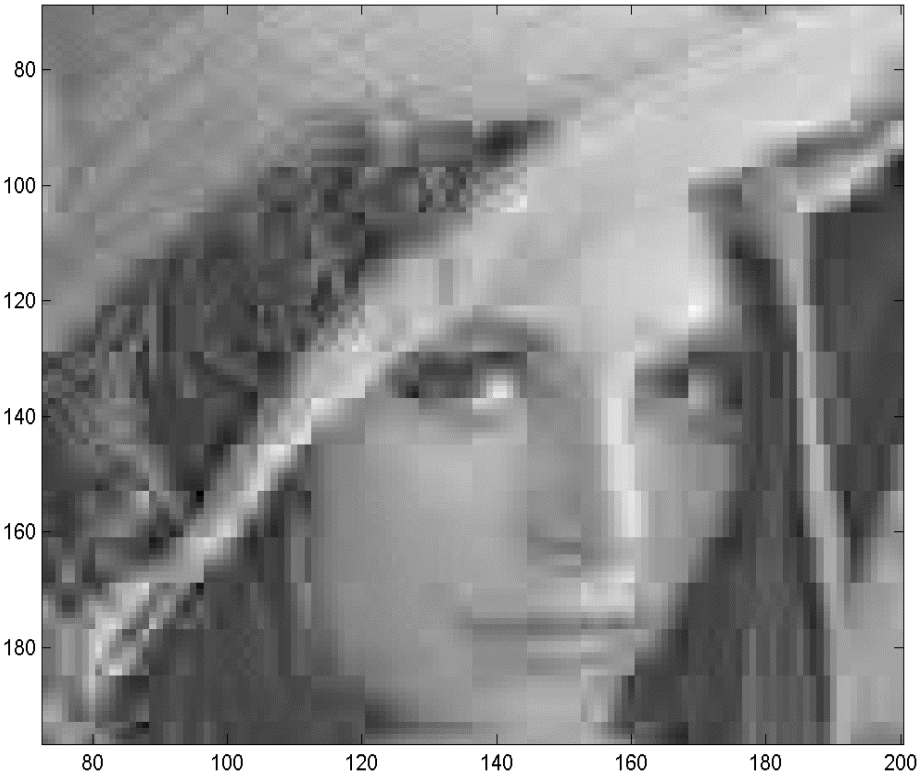


Figure 3.4: Blocking effect of DCT using 8x8 blocks. 4 coefficients retained.

This blocking effect creates a lot more problems in videos and it becomes really hard to recognize the person in the image during teleconferencing. Many techniques have been proposed to improve on the blocking artifacts but none of these techniques have been successful so far. A detailed description of these proposed methods for removing blocking artifacts can be found in [27], [28], [29].

There have also been quite a few attempts at improving on DCT. One of these methods is by using one dimensional DCT proposed by Belkasim and Bhatia in 2002 [30]. An extension of [30], also by Belkasim, is termed as “Zig-zag line method”. Basically in this method the DCT process is broken into four parts. DCT is applied to the diagonal elements of the original image matrix, then the same algorithm is applied to the other diagonal and the results are averaged. Then line DCT [30] is applied to rows and columns and at the end the four images are averaged to get the final image. The results from this method are very encouraging in reducing the blocking effects. As mentioned earlier in this thesis, it is not fair to compare block DCT of 8x8 blocks with other transforms. The reason being that with 8x8 blocks there are only 64 coefficients which is a very low number to get any compression on and that’s why there is more blocking effect.

As mentioned earlier, DCT has been experimented and explored by many researchers. In the next section another extension of DCT known as MDCT will be explored. Modified DCT or MDCT as it has been known for a while, is intended for videos for better compression.

Chapter 4

MODIFIED DISCRETE COSINE TRANSFORM (MDCT)

The concept of MDCT was first introduced by Princen and Bradley in [31]. They proposed a system that provides perfect reconstruction of a signal from a critically sampled analysis signals. This technique, in contrast with the normal DCT, allows overlapping between adjacent time windows. The overlapping in the adjacent windows introduces time domain aliasing but the process proposed in [31] suggests that this aliasing can be cancelled out in the synthesis process and the system can provide perfect reconstruction. Formally MDCT and inverse MDCT can be described by the following equations:

$$\alpha_r = \sum_{k=0}^{2N-1} \tilde{a}_k \cos\left[\pi \frac{\left(k + \left(N+1\right)/2\right)\left(r + 1/2\right)}{N}\right] \qquad (4.1)$$
$$r = 0,\ldots,N-1$$

$$\hat{a}_k = \frac{2}{N} \sum_{r=0}^{N-1} \alpha_r \cos\left[\pi \frac{\left(k + \left(N+1\right)/2\right)\left(r + 1/2\right)}{N}\right] \qquad (4.2)$$
$$k = 0,\ldots,2N-1$$

It has been explored already in this thesis that DCT introduces the blocking effect because of the discontinuity of DCT at boundaries between the blocks. MDCT introduces the concept of overlapping between the blocks and this overlapping result in aliasing where same information

appears twice in the overlapped region. Time domain alias cancellation (TDAC) is one of the most important characteristics of MDCT and it is explored in [31]. The concept of analysis/synthesis has been widely used in the speech coding [31]. The analysis part of the system breaks the signal into sub signals or bands and the synthesis part attempts to reconstruct the image back from the sub signals provided by the analysis part. MDCT is a critically sampled analysis/synthesis system which allows overlapped windows to be used. It is a well acknowledged fact that whenever a signal is reconstructed from sub-signals or band, there is always some sort of distortion or disturbance introduced in it. In MDCT the distortion is caused by the overlapping of the windows. This distortion is removed in the synthesis process by the technique termed as Time Domain Alias Cancellation (TDAC) [31].

Modified discrete cosine transform (MDCT) is based on DCT and hence it also belongs to the Fourier related transforms. The main difference between DCT and MDCT is that MDCT is applied on larger blocks contrary to DCT where the block sizes are normally smaller. The major force behind the popularity of MDCT is its use in high quality audio coding. In addition to the energy compaction property of DCT as explained in the previous chapter, MDCT also has the properties of simultaneously achieving critical sampling, flexible window switching and reduction of blocking effect [32]. Window switching allows switching from higher frequency resolution to lower frequency resolution for sub bands above a chosen index. MDCT has a special quality of canceling out time domain aliasing as well as frequency domain aliasing. MDCT uses the concept of time domain alias cancellation which is explained in [31]. DCT and DFT have been unsuccessful in providing better processing of audio signals, which led to the popularity of the modified discrete cosine transform. The blocking artifact of DCT, as explained

earlier, is even more prominent in sound signals than the images. MDCT is better than DCT for sound signals because of its overlapping technique.

In MDCT, the subsequent blocks are overlapped so that the last half of the first block is overlapped with the first half of the second block. This overlapping reduces the artifacts that are produced by block boundaries. One of the many observations about MDCT made in [33] is that MDCT is not orthogonal and does not fulfill Parsevale's theorem [33]. MDCT, like DCT is a lossy transforming process.

MDCT was developed by many researchers independently and that's why it has been also known as time domain alias cancellation (TDAC) and modulated lapped transform (MLT) [37]. It has been reiterated many times in this thesis that the biggest shortcoming of block transforms coding is the introduction of blocking artifacts in the reconstructed signal. This artifact is a lot more prominent in audio signal coding than it is in the image coding. The reason for this artifact is due to the fact that the blocks are processed independently and the quantization errors produce discontinuities in the signal at the block boundaries [37]. MLT or MDCT is a form of a lapped orthogonal transform (LOT) which was studied in detail by Malvar [29] in an attempt to reduce this blocking artifact. Lapped orthogonal transform works essentially in the same manner as the standard block transform, the only difference, however, is that in LOT the basis function of the transform extend beyond the block boundaries [37].

4.1 Properties of MDCT

The properties of MDCT are explored in detail in [33]. MDCT can achieve a perfect reconstruction of the signal by using the overlap-add process. Also, if the signal has local symmetry then MDCT and IMDCT can reconstruct the original time domain signal perfectly

[33]. MDCT, just like DCT and DFT also possesses the energy compaction property which states that all the pertinent information is stored in the top left corner of the image hence making it easier for the compression process.

The results produced in [33] conclude that MDCT is a very useful concept with its Time Domain Alias cancellation (TDAC) characteristic. As far as the energy compaction property of MDCT is concerned, it is not any better than DCT or DFT. The superiority of MDCT over DCT relies mainly in its critical sampling property, reduction of blocking effect and the possibility of window switching [33].

There has been active research going on to improve the cost of calculating the MDCT. [34] presents a fast algorithm for modified discrete cosine transform. The basic idea here is that IDMCT is converted to normal IDCT and then it is easily computed using IFFT. More information on how IDMCT is converted into normal IDCT can be further explored in detail in [34].

Chapter 5

MULTI-RESOLUTION DCT

In the previous chapters we have explored the benefits of DCT as well as Multi-resolution analysis. It has been established that DCT has been the most widely used and researched transform. It has also been established that DCT has some useful properties such as energy compaction and decorrelation properties which make DCT close to the optimal transform and none of the other transforms are as good as DCT in that sense. Nevertheless, there are still some problems with DCT such as the blocking effect which has been one of the biggest criticisms of this transform. This problem becomes more prominent when the compression ratios are higher and also when the block sizes are smaller. It has also been established in the previous sections that it is not fair to compare DCT of 8x8 blocks with other kinds of transforms because the number of DCT coefficients is very small which limits its ability to distribute information among the coefficients.

Mutiresolution analysis (MRA) has been very popular with compression specially in video streaming because it eliminates the blocking effects of the DCT which become more prominent in videos. Wavelets are based on the multi-resolution but it is well acknowledged that wavelets are far more expensive in terms of hardware and software as compared to DCT [13]. Recently there have been some attempts at combining the positive aspects of DCT and multi-resolution analysis [35], [36]. This research is fairly new and there are still a lot that can be done in this realm.

Wavelets have demonstrated that image compression carried out on sub-band decomposition can be more effective than compressing the full band image [35]. A novel approach, known as, sub-band DCT is introduced in [35]. This new DCT computation scheme consists essentially of two parts as explained in the paper. The first part of the process is the decomposition of the input data into sub-bands and the second part deals with the computation of the DCT coefficients of the input data from the sub-band samples. The concepts of the *half band approximations* and *Quarter band approximations* are also introduced in [35]. The basic idea depends on the fact that in many image coding applications, most of the relevant data is in the low frequencies and hence the other half band can be neglected and we still have a good presentation of the original image after reconstruction. The same idea is applied again in the *Quarter band approximations* and further compression is achieved. As duly noted, this kind of compression will be a lossy compression but the energy compaction property of DCT will come into play and only the redundant information will be removed from the image.

A new approach for combining discrete cosine transform (DCT) and multi-resolution is introduced in this section. As it has been mentioned repeatedly in this thesis, the main problem with the DCT is the blocking artifact. In this new approach for DCT first of all we will eliminate the blocking completely. Instead of using the blocks as done in the standard DCT, we will use the full rows and columns of the image. Once we have the rows and columns of the image, we will apply the techniques of multi-resolution on it to achieve the required compression. The idea here is not to prove that this new approach is better than the block DCT or the wavelets for compression purposes. In fact, the idea here is to show that multi-resolution analysis can be applied to DCT.

5.1 THE NEW ALGORITHM

The steps of the new algorithm are as follows:

1. Read the image and convert into a double using MTALAB function “im2double”.
2. Divide the image into rows and columns [row col] = size(a);.
3. Compute the total size of the image 2(row*col).
4. Concatenate the rows and columns to each other and make a long linear line of data.
5. Apply 1-D Dct to the data obtained from step 4.
6. Apply multi-resolution analysis and discard the 50% of the data obtained from step 5.
7. Repeat the process until the desired compression is reached.
8. Apply inverse 1-D DCT
9. Separate the rows and columns from the data received from step 8.
10. Reconstruct the original image matrix by averaging pixels from the rows and the columns. This averaging is necessary to remove any distortion or noise introduced in the process.
11. Compute the mean square error and compare it with the block DCT or various sizes.

As it has been mentioned earlier, the results obtained from the above algorithm will be compared against block DCT. The blocks used in block DCT will be at least 16x16 or more. We will not compare our new algorithm with 8x8 blocks as it is not a fair comparison because of the fewer number of coefficients.

5.2 ALGORITHM EXPLAINED

There are two main parts to this algorithm which are of immense importance. The first step deals with eliminating the blocking artifacts which has been a problem with the block DCTs. The other part deals with applying multi-resolution analysis to the coefficients obtained after applying the DCT. In steps 2 and 4 it is clear that instead of making the blocks for the DCT we are taking the whole rows and columns and making a big one dimensional array for the pixels.

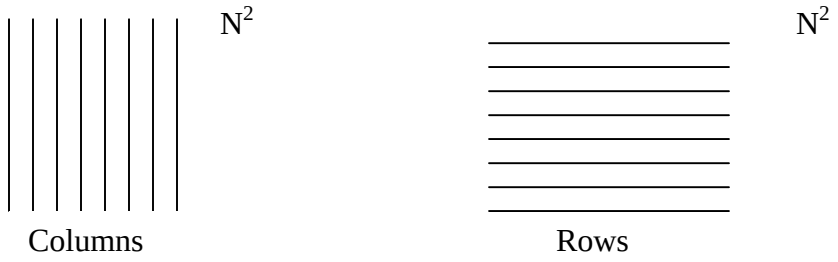


Figure 5.1: Rows and columns of the image separated.

The size of both columns and rows is $N^{2.}$ So the total size of the image will be $2N^2$. The rows are concatenated to each other to form a one big horizontal line of pixels. The columns are also concatenated to each other in the same way. Now we concatenate concatenated columns to the end of the concatenated rows forming a 1-D signal of size $2N^2$. All these steps have been taken to eliminate the need of making the blocks for running DCT. This procedure is termed as *"cascading rows and columns"*. This process is shown pictorially in the following figure 5.2.

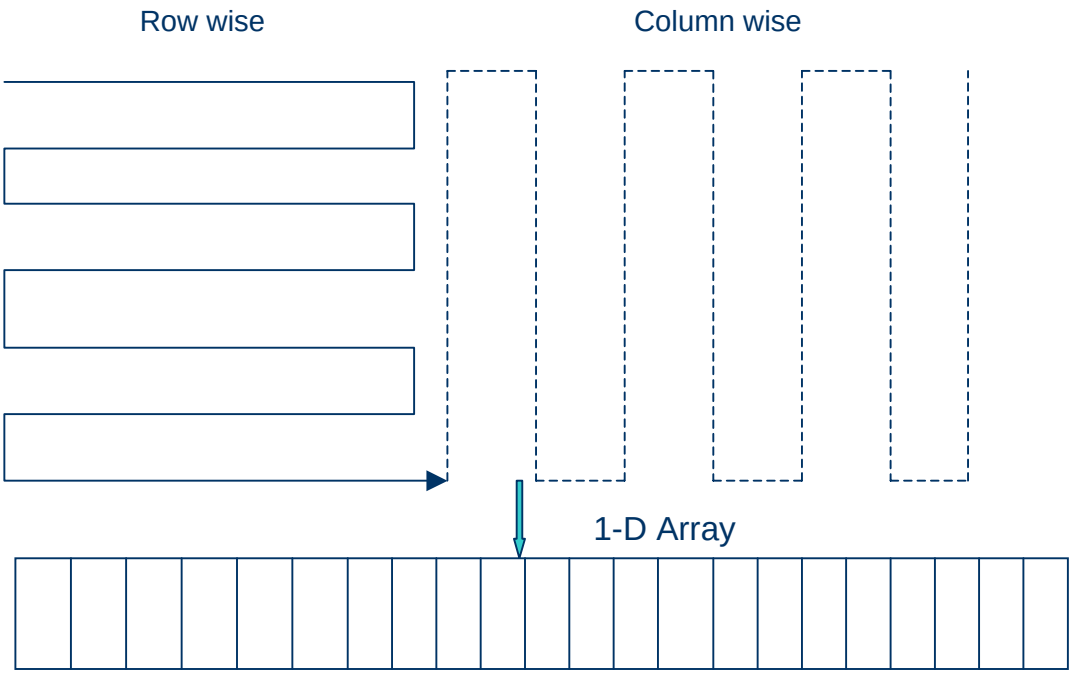


Figure 5.2: Cascading the rows and columns into a 1-D array

Once we have all the pixels arranged in a 1-D array then DCT is applied on it. We use the MATLAB built-in function “dct2”. This function computes a 2-D DCT but we can use it to compute the one dimensional DCT by using a matrix size of Nx1. After the DCT coefficients are obtained, we apply the multi-resolution analysis to the DCT coefficients to obtain the desired compression. Here the compaction energy property of the DCT is utilized. It has been established in the previous chapters that most of the information in DCT is retained at lower frequencies, meaning at the top left corner of the blocks. This property makes it easier and simpler to apply multi-resolution analysis on DCT coefficients.

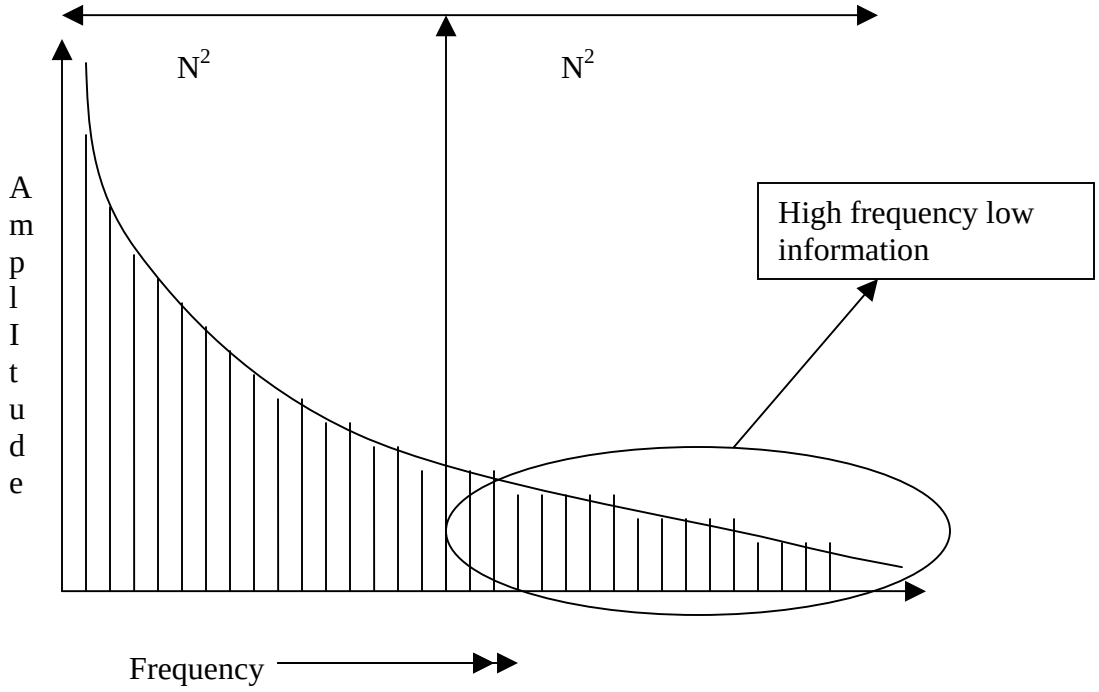


Figure 5.3: Shows the energy compaction property of DCT coefficients

Figure 5.2 shows the energy compaction property of the DCT. On the X-axis there is frequency and the amplitude is plotted on the Y-axis. It is evident from the graph that much of the image information is stored in the left corner of the graph which is the low frequency. It means that in DCT coefficients most of the information is represented in the lower frequencies. This property plays a very important role while applying the multi-resolution analysis. In multi-resolution analysis we discard the 50% of the coefficients at each level until we get to the desired compression. Based on figure 6, this process becomes easier as we can simply discard half of the coefficients at higher frequencies without losing much of the information because we have already established that most of the information is compacted at the lower frequencies.

5.3 ALTERNATIVE APPROACH

The above proposed algorithm is one way of eliminating the blocks i.e. by taking all rows and columns and concatenating them to make 1-D data. This approach has worked well and the results will be presented in the next section that consolidates our claim that the blocking artifacts can be avoided while still having acceptable errors. An alternate way of eliminating the blocks is explained as follows:

Instead of taking rows and columns of the image matrix in this approach first we take only the rows of the original image matrix. The rows are concatenated to form a 1-D array of the pixels. DCT is applied to these 1-D row-wise pixels and using the energy compaction property to the DCT, half of the pixels are discarded. Next columns are taken from this compressed matrix and the process is repeated for concatenating, applying DCT and the multi-resolution part where half of the pixels are discarded. In the next resolution, we alternate this process and this time the columns are taken first and then the rows.

This approach achieves 50% compression in two cycles; first for the rows and then for the columns as compared to the first approach in which 50% compression is achieved in one cycle because the rows and columns are taken at the same time. Figure 5.3 shows this concept pictorially.

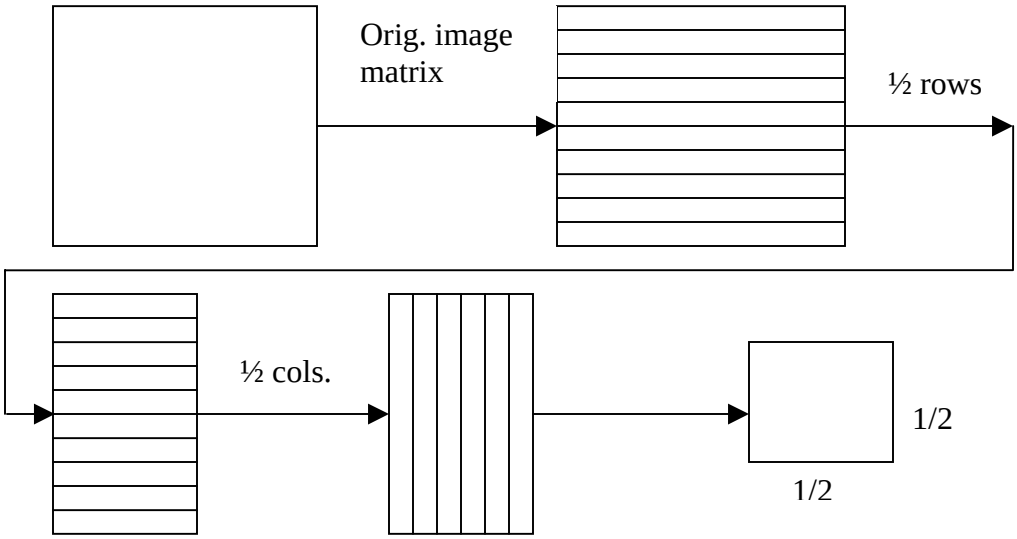


Figure 5.4: An alternate approach to eliminate blocks in DCT

It is shown in the above figure that from the original image matrix the rows are taken and reduced to half and then on the reduced coefficients the columns are taken and then reduced again to half to achieve the first 50% compression or the first resolution.The fact that this approach reduces the size of the image at each level, aliasing is introduced in this process. Since this approach is based on one dimensional DCT and we have already established in the previous chapters that MDCT also uses a similar approach. In order to use this approach properly, MDCT can be used because of its special technique of time domain alias cancellation (TDAC). Due to the simplicity of the first approach in which cascading rows and columns is used; we have implemented it and left the alternate approach for future work. Also, cascading the rows and columns lets us make full use of the energy compaction property of the DCT.

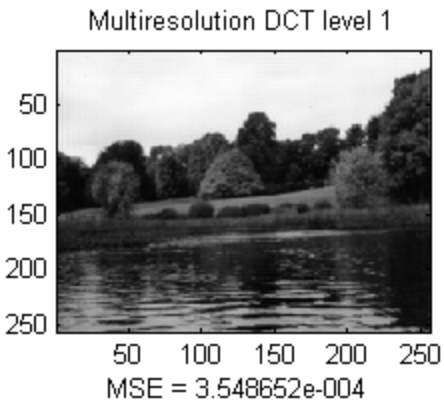
Chapter 6

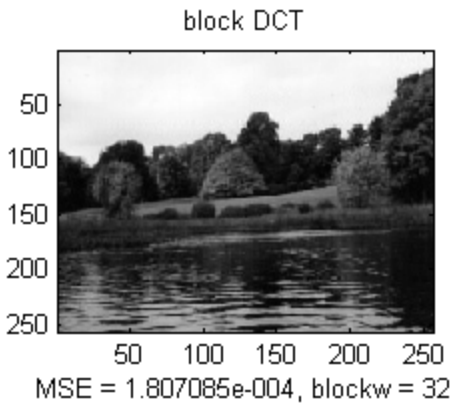
EXPERIMENTAL RESULTS

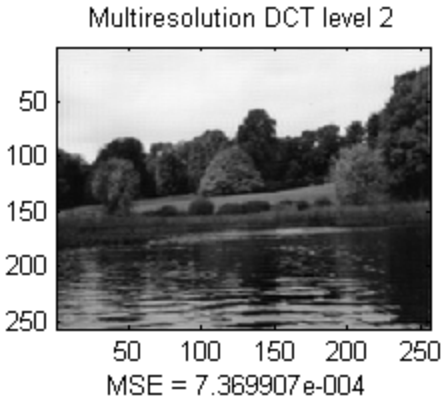
As mentioned in the previous chapters, the main idea of this thesis is to come up with a new algorithm that eliminates the blocking from the DCT as well as to show that multi-resolution analysis can be applied to the DCT. Several images have been used with our algorithm as well as the block DCT in order to compare the results. The block size used in block DCT is 32 x 32. We have not used any other block size because if a smaller block size is used then definitely the results will not be as good for the block DCT and it will not be a fair comparison.

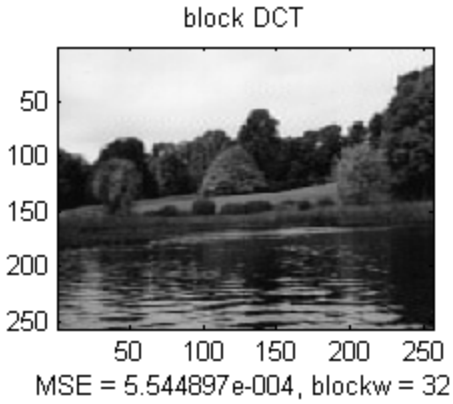
For both the images, we have run the test up to 6 resolutions. The mean square average is used to computer the errors for comparisons. Although the resolution is reduced at every level we keep the same size for the images as the original size in order to make easier comparisons. We obtain the same size as the original image in the reconstruction part by padding zeros to the reduced resolution to come up with the same size image. The results of the multi-resolution DCT as well as block DCT applied to the images can be seen in the following images.

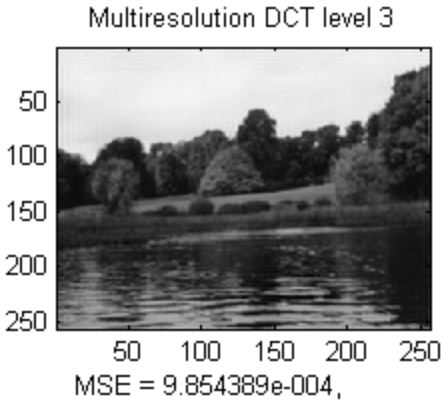
Image of Autumn at different resolutions with
Multiresolutuin DCT and Block DCT

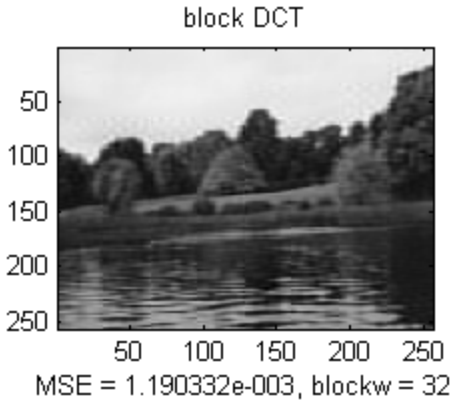


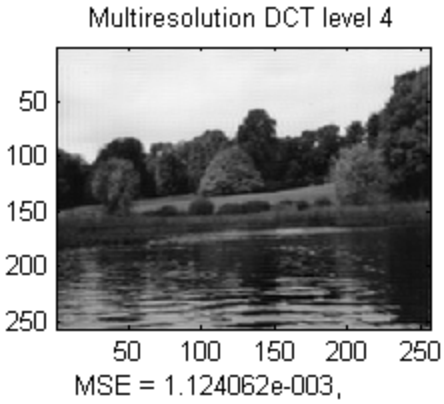


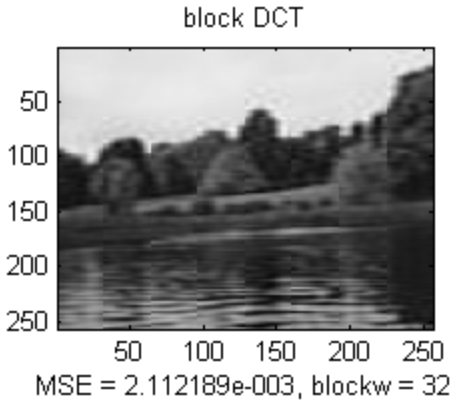


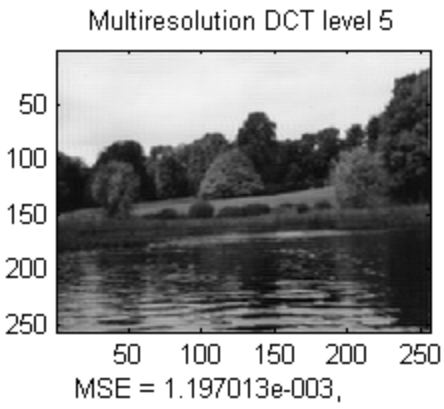


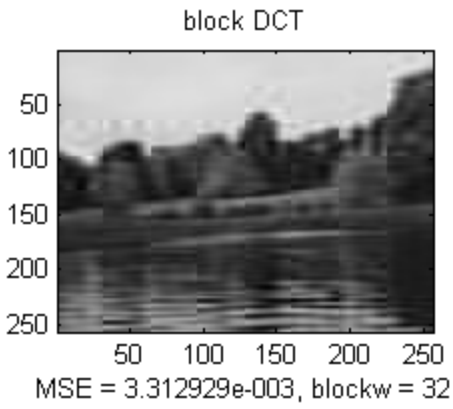

Multiresolution DCT level 3
50
100
150
200
250
50 100 150 200 250
MSE = 9.854389e-004,

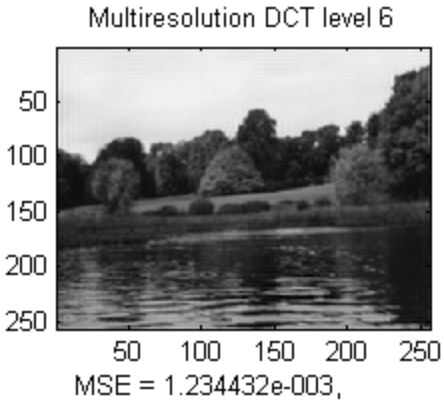

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 1.190332e-003, blockw = 32


Multiresolution DCT level 4
50
100
150
200
250
50 100 150 200 250
MSE = 1.124062e-003,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 2.112189e-003, blockw = 32


Multiresolution DCT level 5
50
100
150
200
250
50 100 150 200 250
MSE = 1.197013e-003,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 3.312929e-003, blockw = 32


Multiresolution DCT level 6
50
100
150
200
250
50 100 150 200 250
MSE = 1.234432e-003,

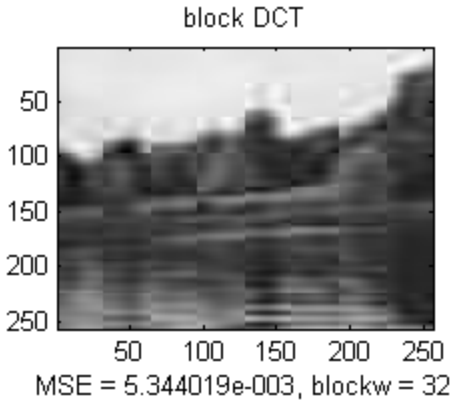
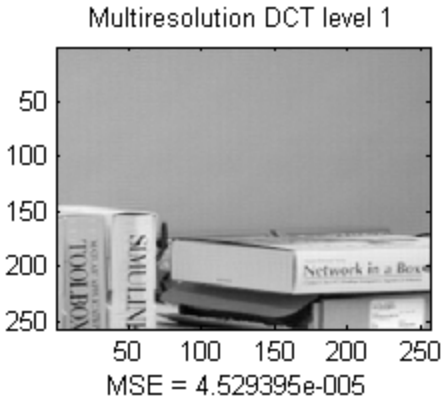
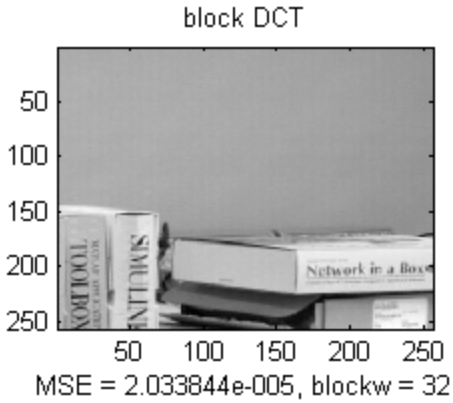
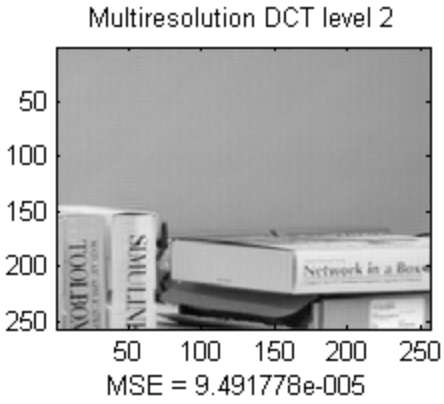
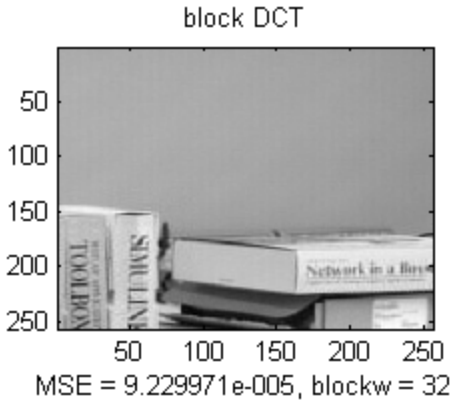

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 5.344019e-003, blockw = 32

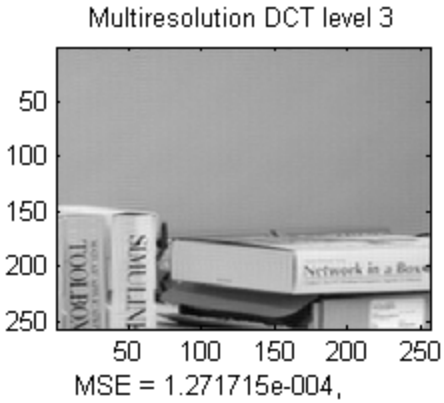
Image of Books at different resolutions with
Multiresolutuin DCT and Block DCT

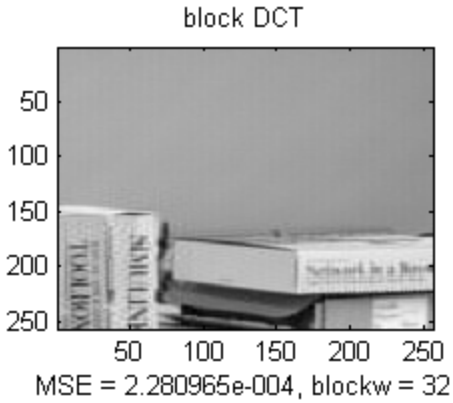


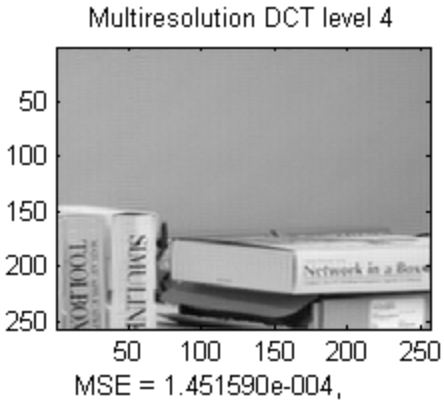


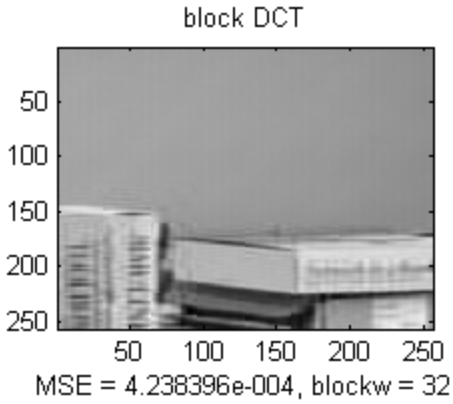


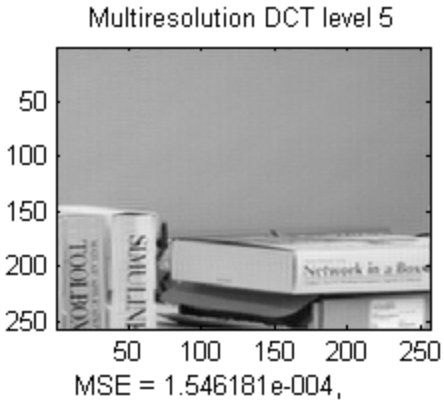


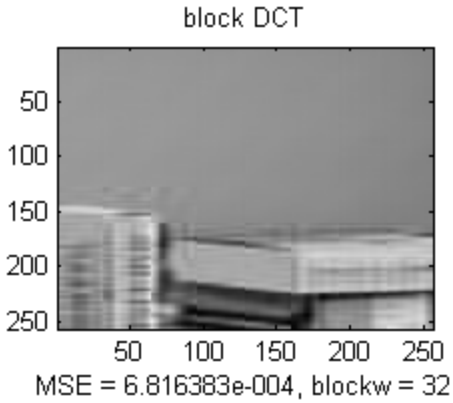

Multiresolution DCT level 3
50
100
150
200
250
50 100 150 200 250
MSE = 1.271715e-004,

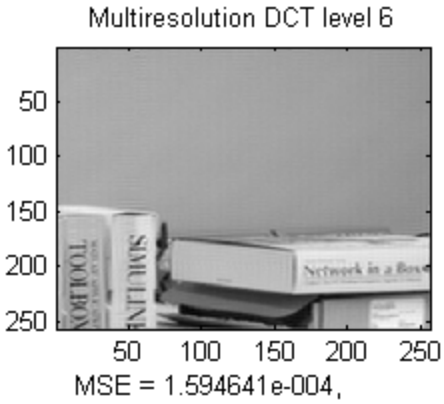

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 2.280965e-004, blockw = 32


Multiresolution DCT level 4
50
100
150
200
250
50 100 150 200 250
MSE = 1.451590e-004,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 4.238396e-004, blockw = 32


Multiresolution DCT level 5
50
100
150
200
250
50 100 150 200 250
MSE = 1.546181e-004,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 6.816383e-004, blockw = 32


Multiresolution DCT level 6
50
100
150
200
250
50 100 150 200 250
MSE = 1.594641e-004,

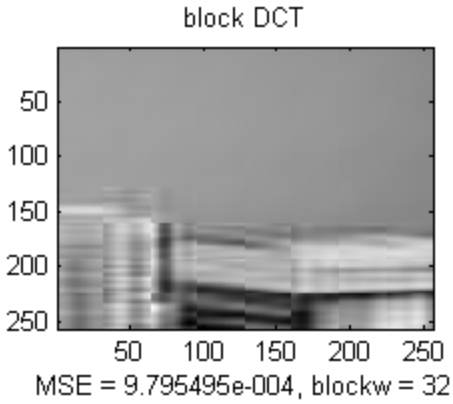
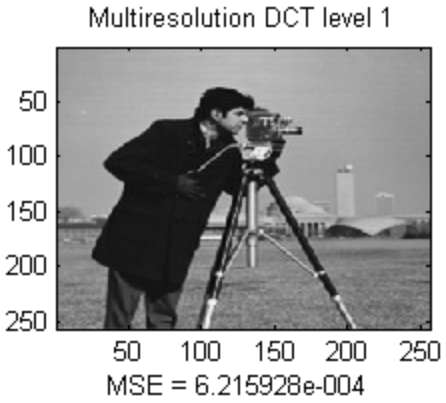
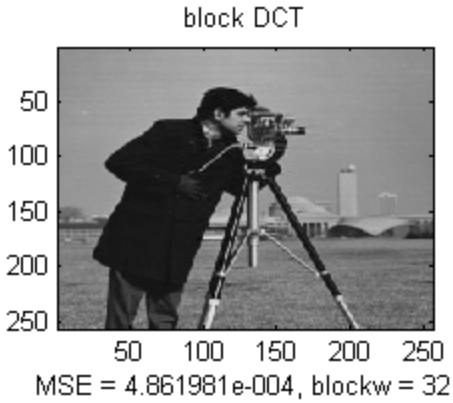
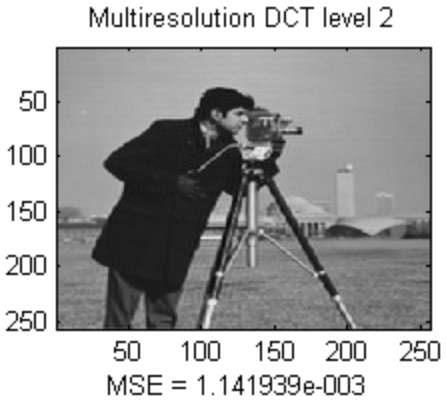
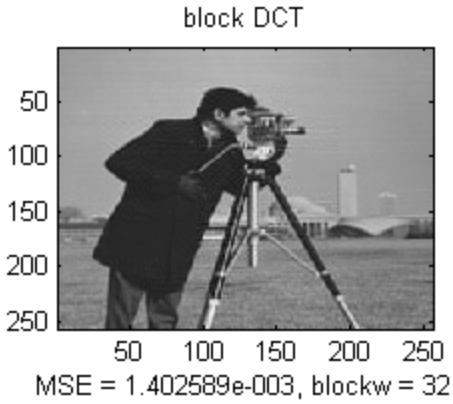

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 9.795495e-004, blockw = 32

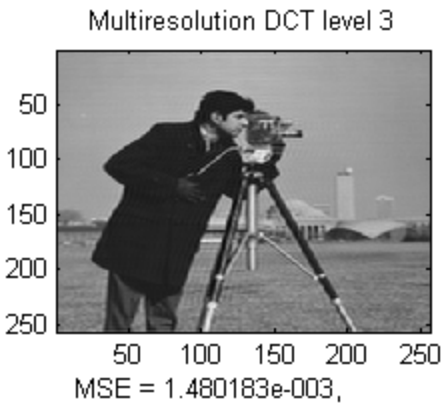
Image of Cameraman at different resolutions with
Multiresolutuin DCT and Block DCT

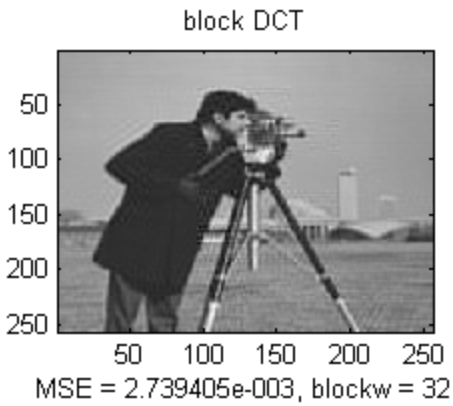


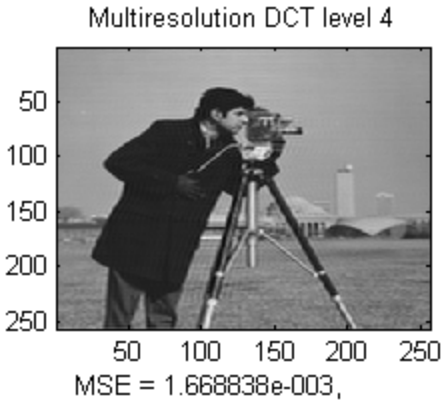


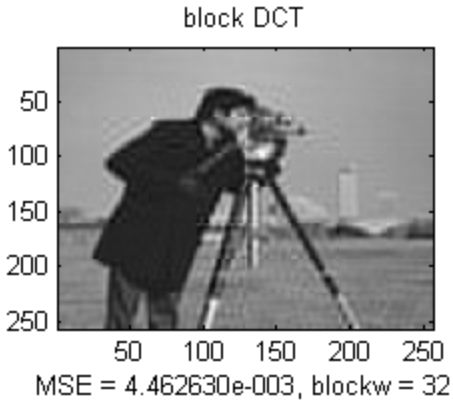


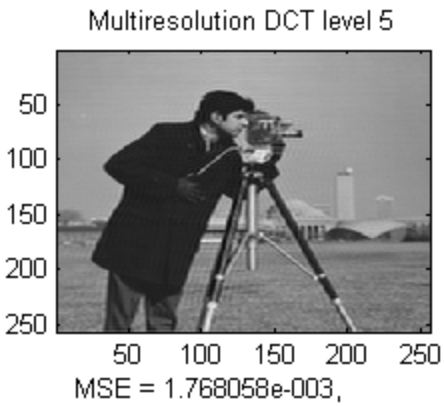


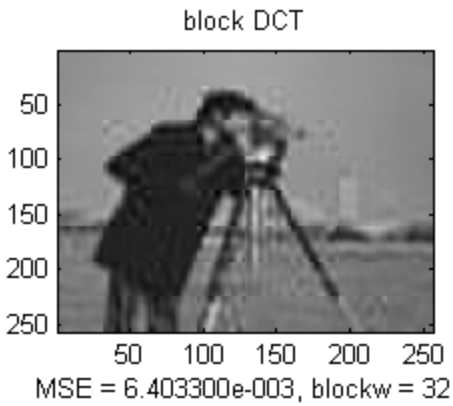

Multiresolution DCT level 3
50
100
150
200
250
50 100 150 200 250
MSE = 1.480183e-003,

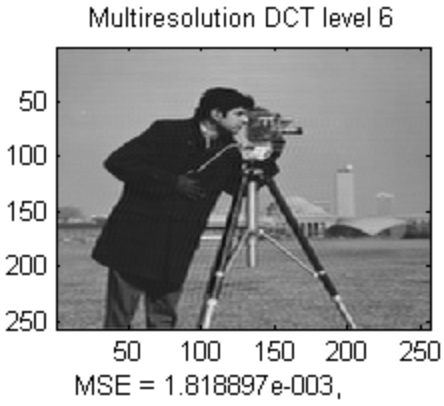

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 2.739405e-003, blockw = 32


Multiresolution DCT level 4
50
100
150
200
250
50 100 150 200 250
MSE = 1.668838e-003,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 4.462630e-003, blockw = 32


Multiresolution DCT level 5
50
100
150
200
250
50 100 150 200 250
MSE = 1.768058e-003,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 6.403300e-003, blockw = 32


Multiresolution DCT level 6
50
100
150
200
250
50 100 150 200 250
MSE = 1.818897e-003,

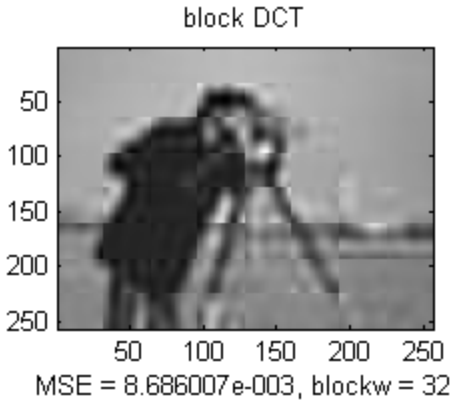
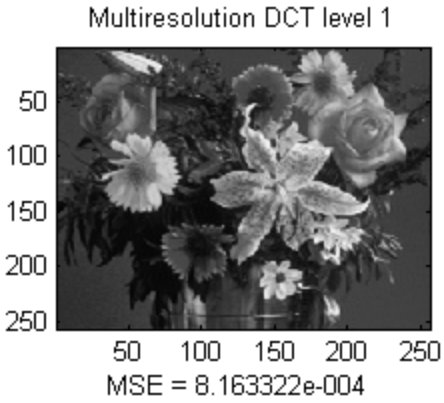
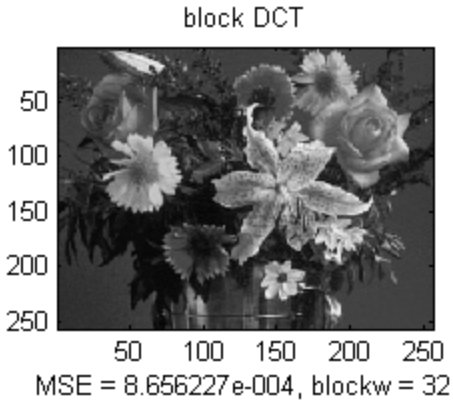
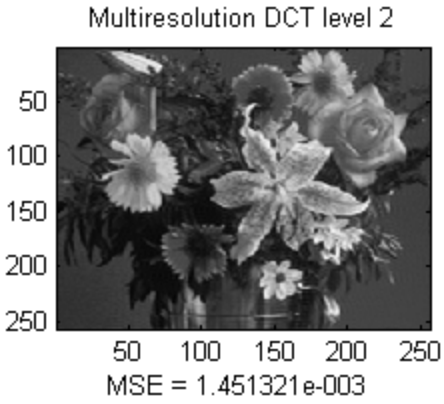
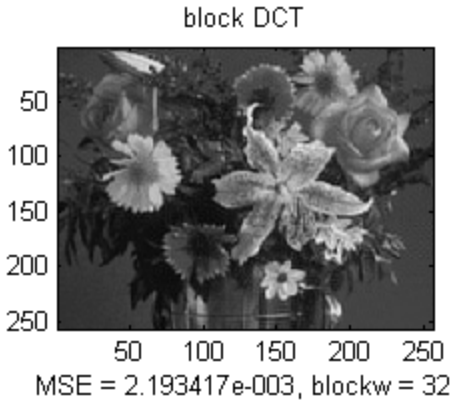

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 8.686007e-003, blockw = 32

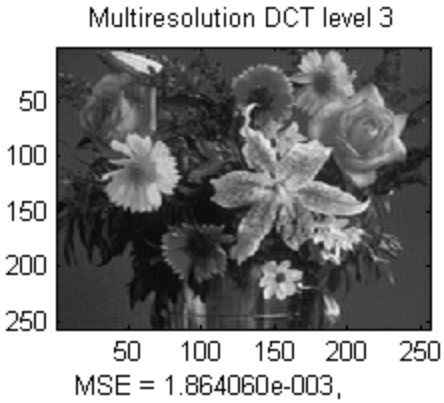
Image of Flowers at different resolutions with
Multiresolutuin DCT and Block DCT

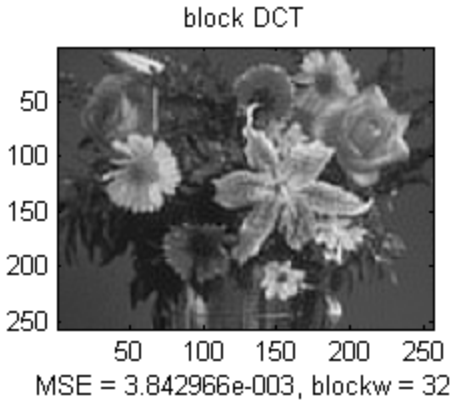


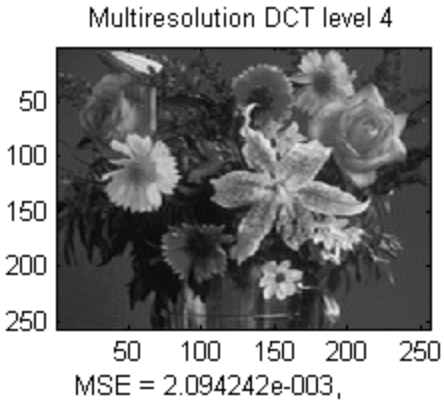


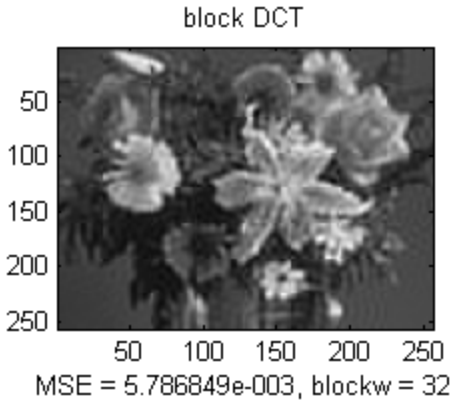


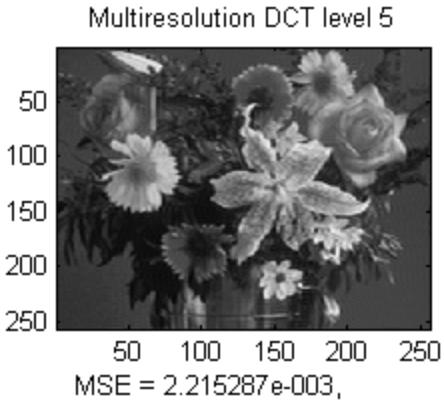


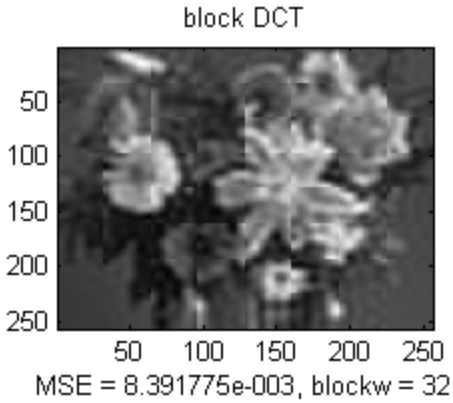

Multiresolution DCT level 3
50
100
150
200
250
50 100 150 200 250
MSE = 1.864060e-003,

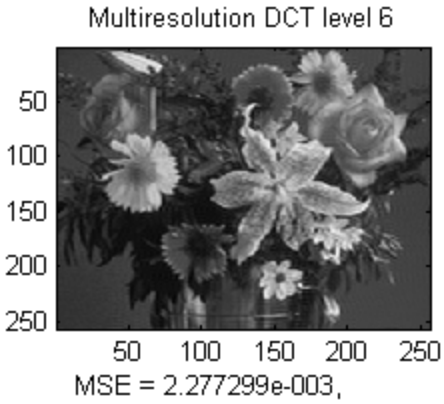

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 3.842966e-003, blockw = 32


Multiresolution DCT level 4
50
100
150
200
250
50 100 150 200 250
MSE = 2.094242e-003,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 5.786849e-003, blockw = 32


Multiresolution DCT level 5
50
100
150
200
250
50 100 150 200 250
MSE = 2.215287e-003,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 8.391775e-003, blockw = 32


Multiresolution DCT level 6
50
100
150
200
250
50 100 150 200 250
MSE = 2.277299e-003,

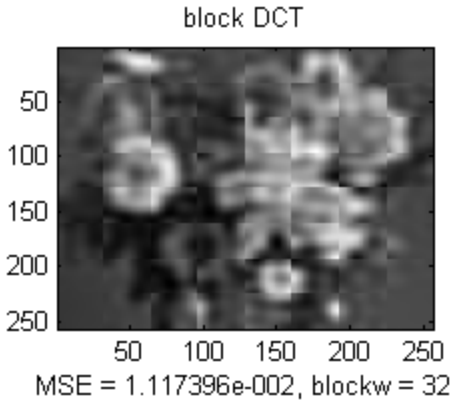
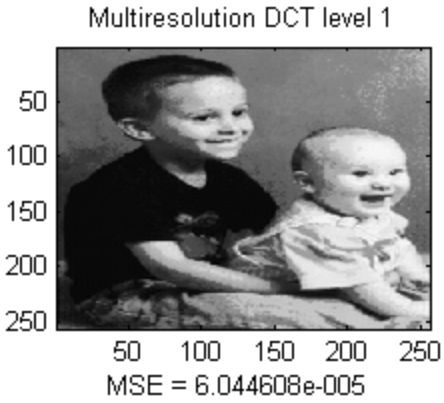
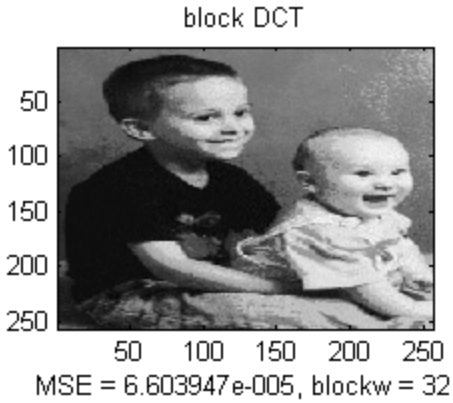
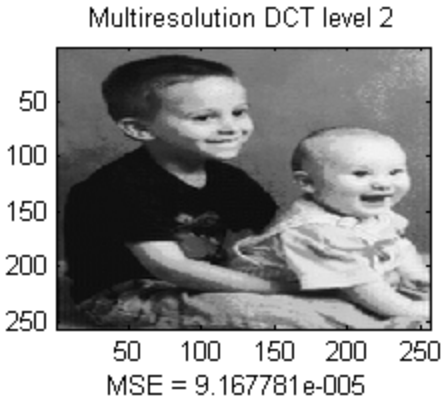
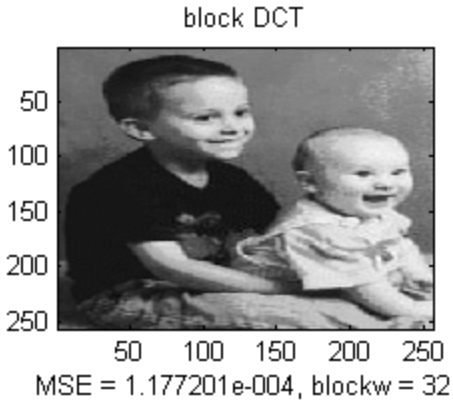

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 1.117396e-002, blockw = 32

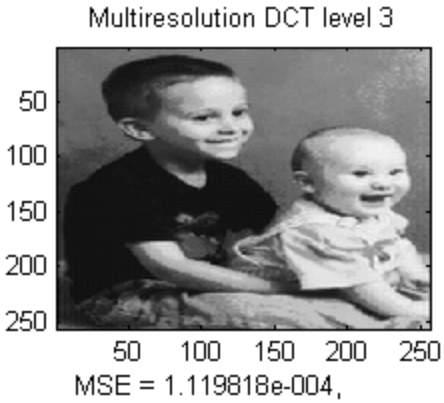
Image of Kids at different resolutions with
Multiresolutuin DCT and Block DCT

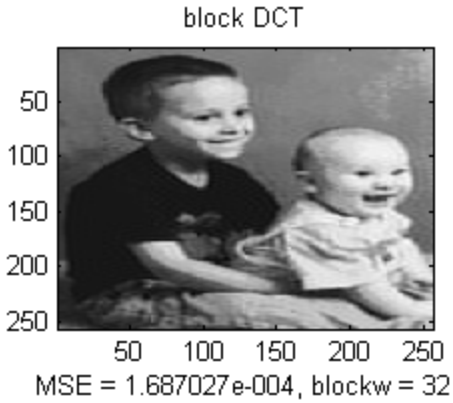


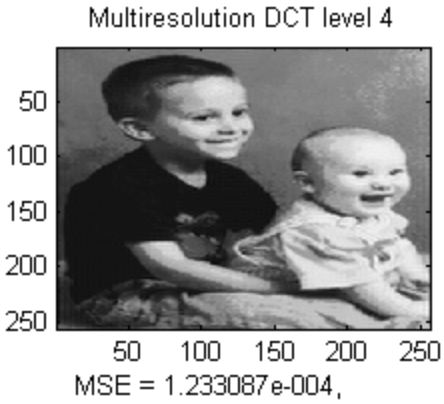


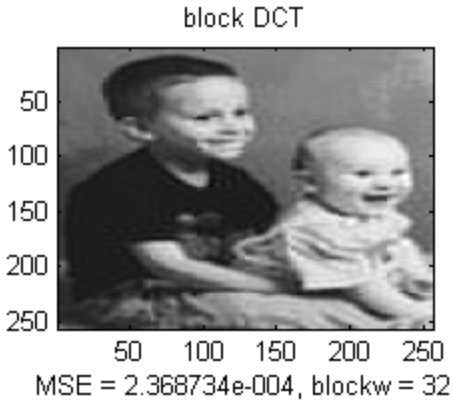


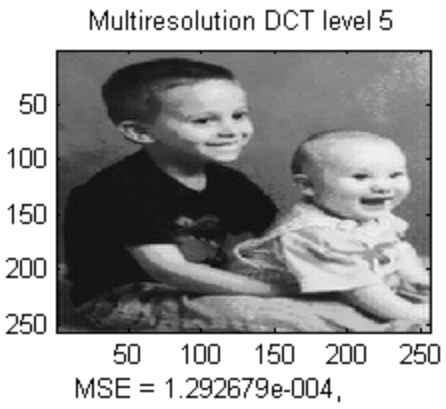


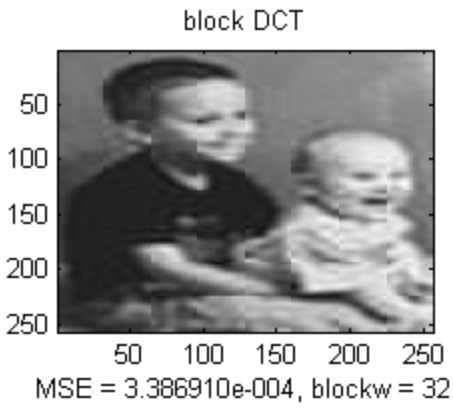

Multiresolution DCT level 3
50
100
150
200
250
50 100 150 200 250
MSE = 1.119818e-004,

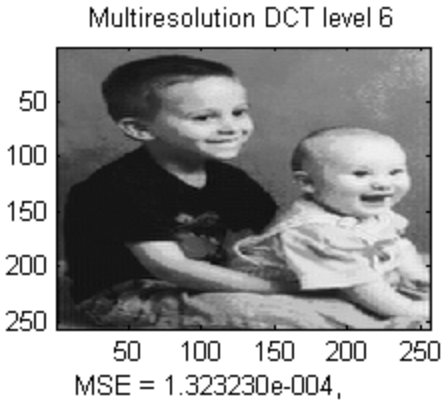

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 1.687027e-004, blockw = 32


Multiresolution DCT level 4
50
100
150
200
250
50 100 150 200 250
MSE = 1.233087e-004,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 2.368734e-004, blockw = 32


Multiresolution DCT level 5
50
100
150
200
250
50 100 150 200 250
MSE = 1.292679e-004,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 3.386910e-004, blockw = 32


Multiresolution DCT level 6
50
100
150
200
250
50 100 150 200 250
MSE = 1.323230e-004,

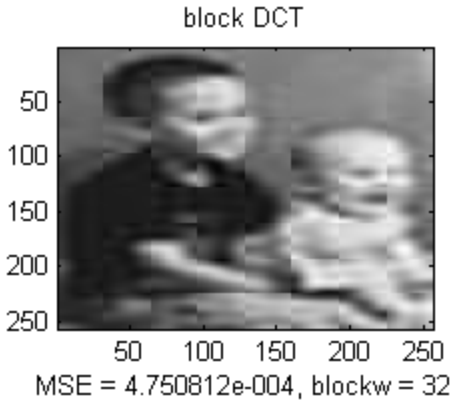
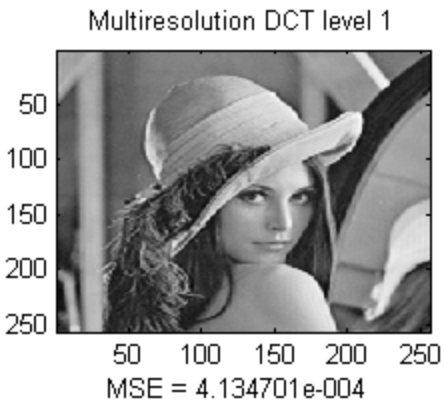
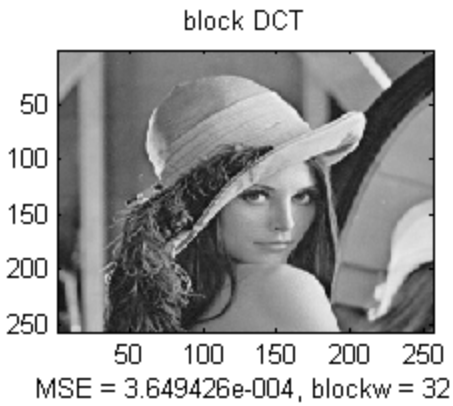
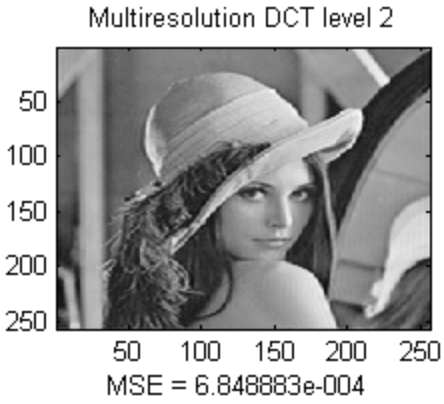
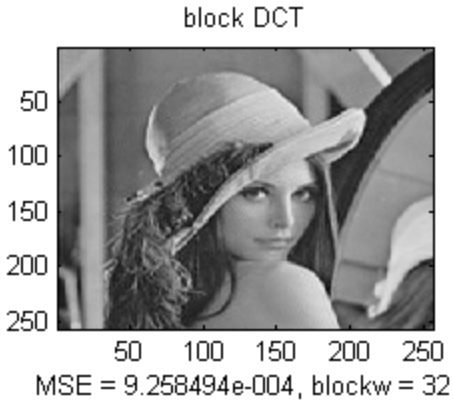

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 4.750812e-004, blockw = 32

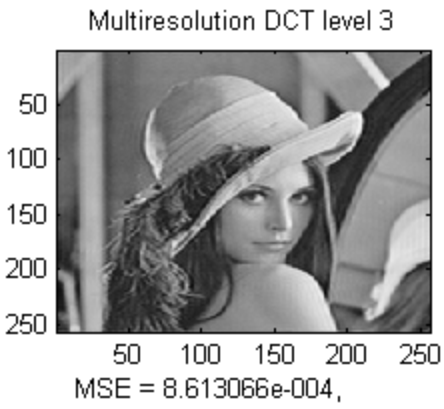
Image of Lena at different resolutions with
Multiresolutuin DCT and Block DCT

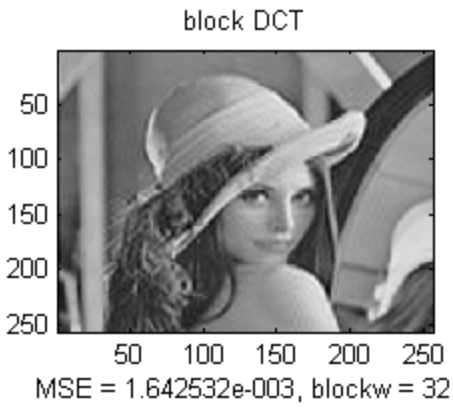


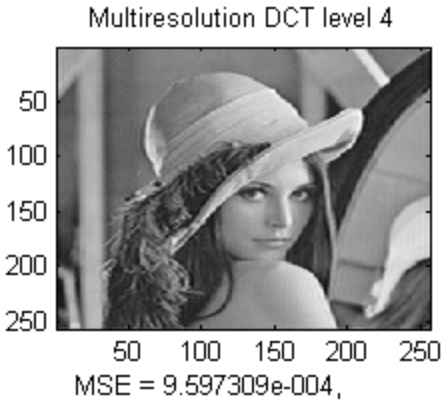


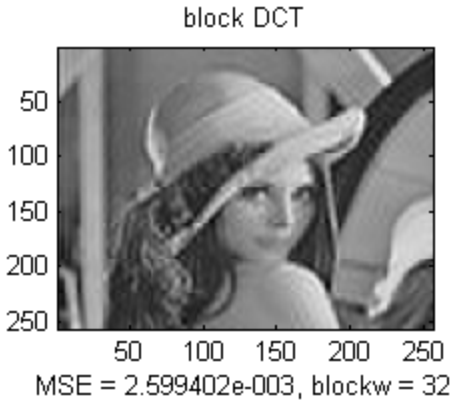


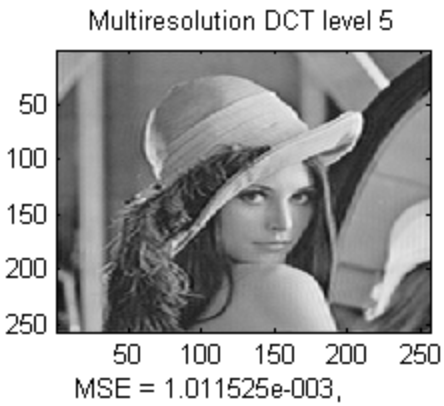


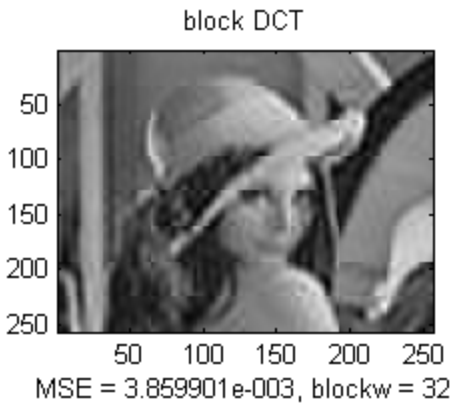

Multiresolution DCT level 3
50
100
150
200
250
50 100 150 200 250
MSE = 8.613066e-004,

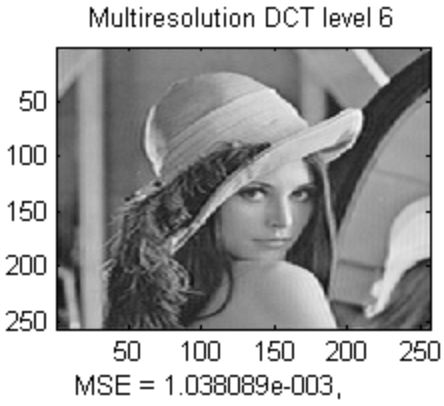

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 1.642532e-003, blockw = 32


Multiresolution DCT level 4
50
100
150
200
250
50 100 150 200 250
MSE = 9.597309e-004,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 2.599402e-003, blockw = 32


Multiresolution DCT level 5
50
100
150
200
250
50 100 150 200 250
MSE = 1.011525e-003,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 3.859901e-003, blockw = 32


Multiresolution DCT level 6
50
100
150
200
250
50 100 150 200 250
MSE = 1.038089e-003,

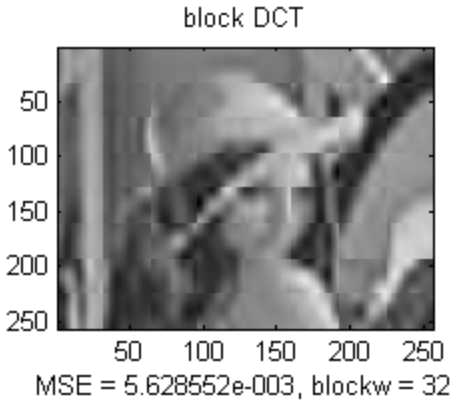
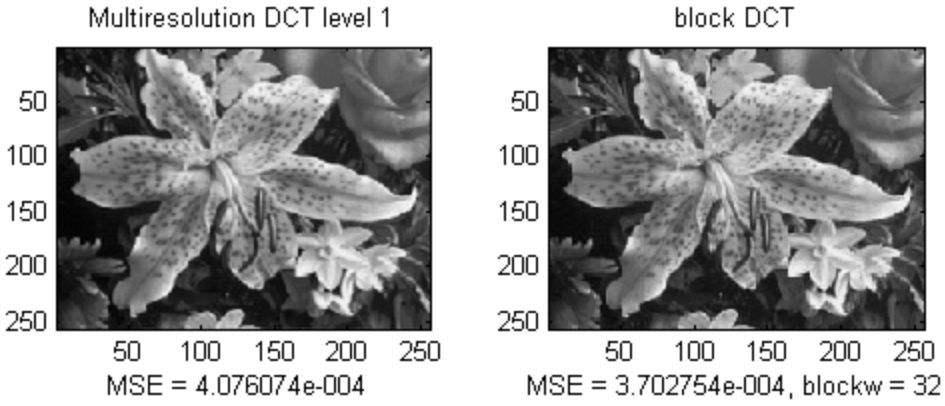
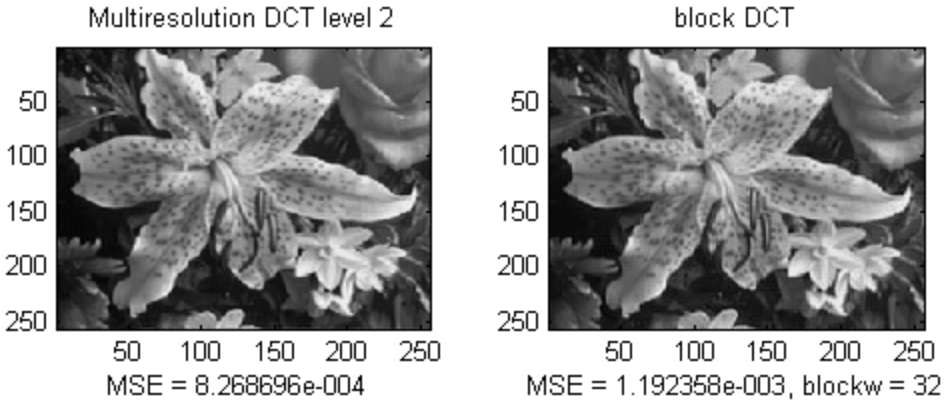
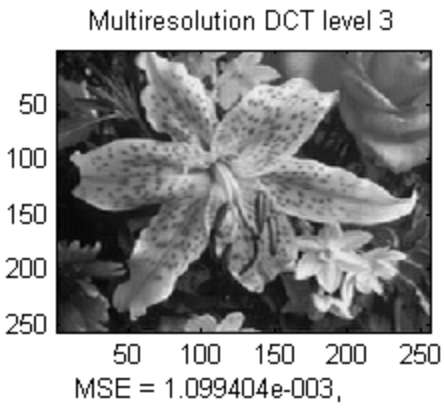
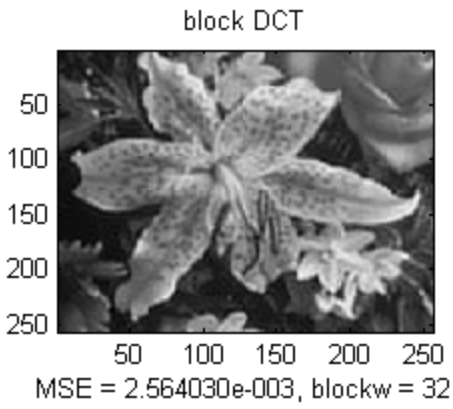

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 5.628552e-003, blockw = 32

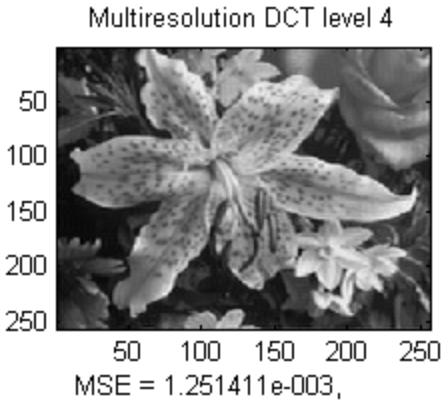
Image of Lily at different resolutions with
Multiresolutuin DCT and Block DCT

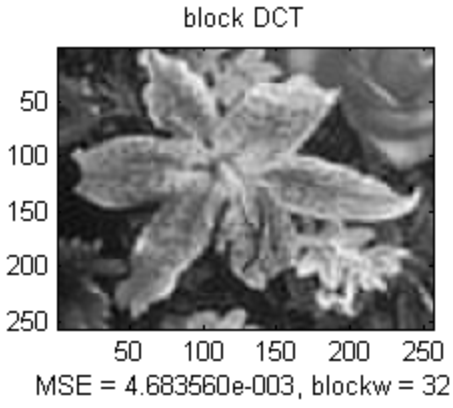


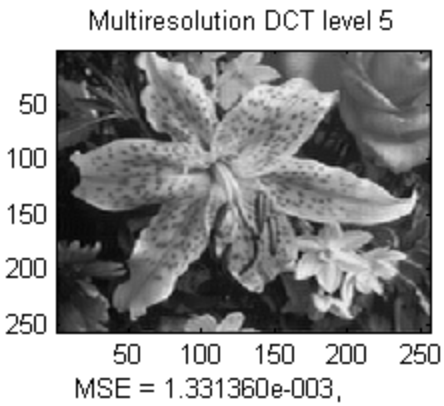


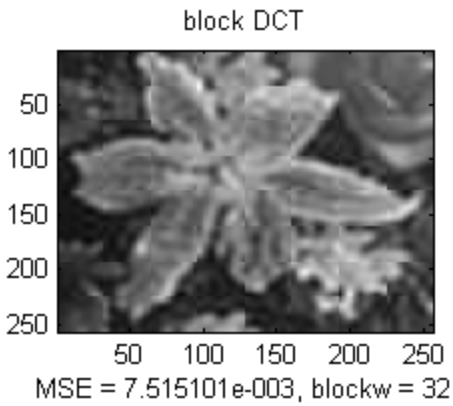

Multiresolution DCT level 3
50
100
150
200
250
50 100 150 200 250
MSE = 1.099404e-003,

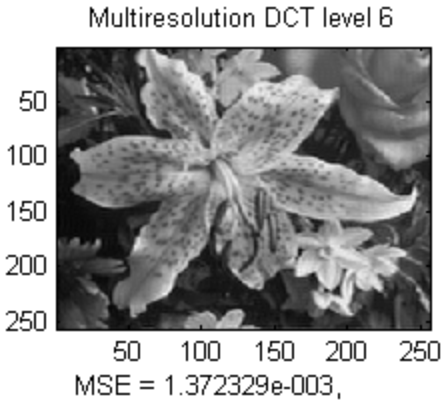

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 2.564030e-003, blockw = 32


Multiresolution DCT level 4
50
100
150
200
250
50 100 150 200 250
MSE = 1.251411e-003,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 4.683560e-003, blockw = 32


Multiresolution DCT level 5
50
100
150
200
250
50 100 150 200 250
MSE = 1.331360e-003,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 7.515101e-003, blockw = 32


Multiresolution DCT level 6
50
100
150
200
250
50 100 150 200 250
MSE = 1.372329e-003,

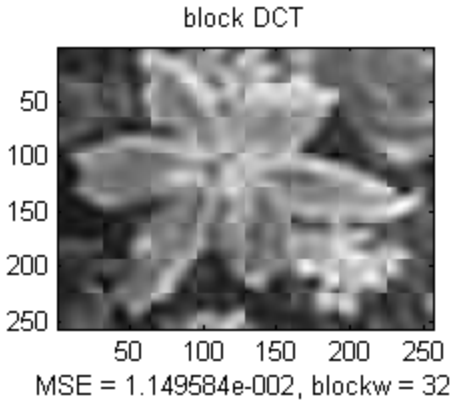
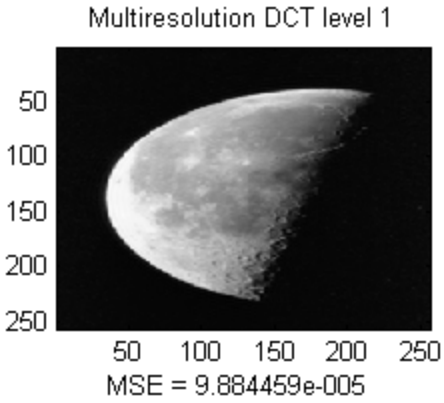
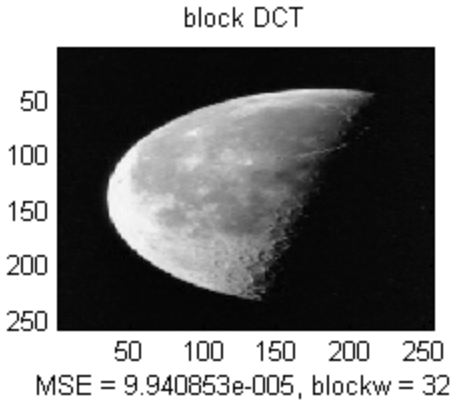
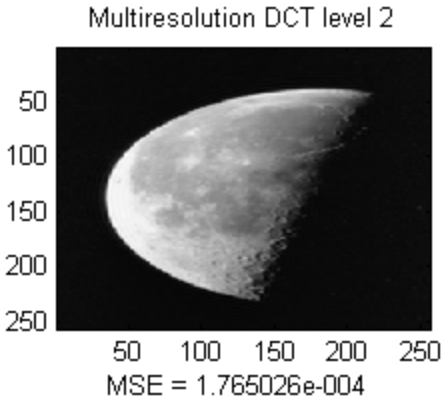
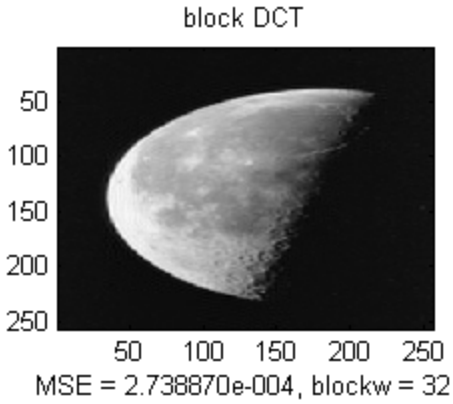

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 1.149584e-002, blockw = 32

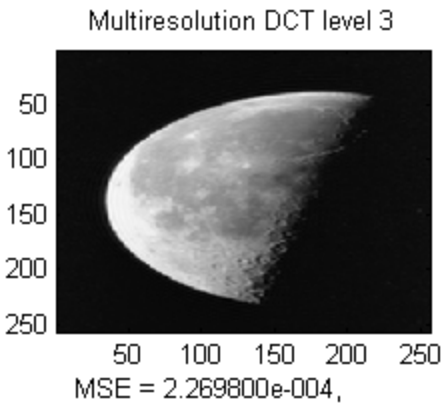
Image of Moon at different resolutions with
Multiresolutuin DCT and Block DCT

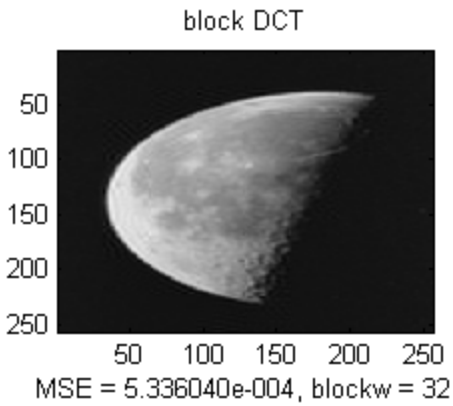


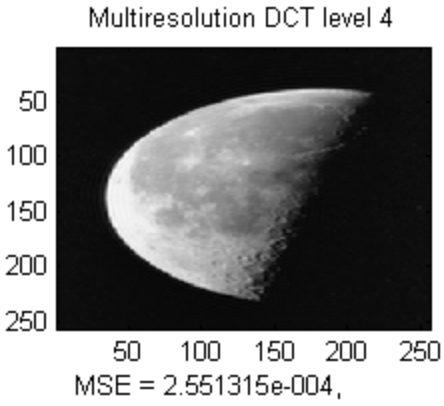


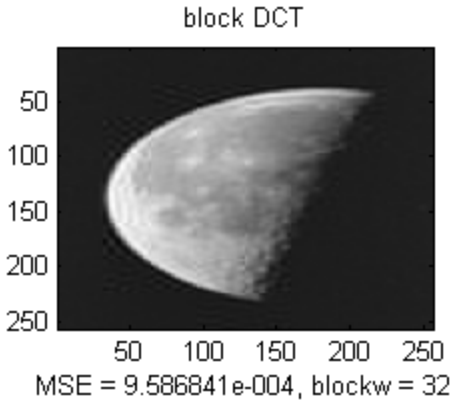


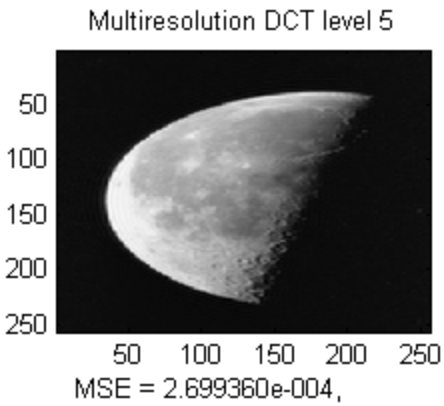


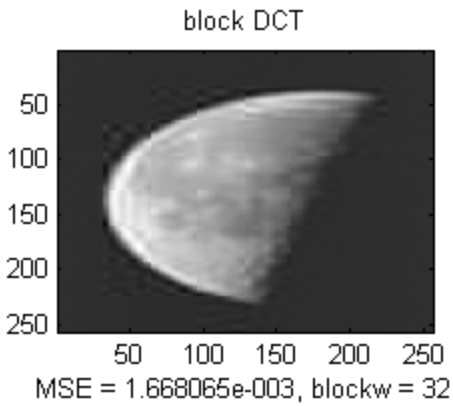

Multiresolution DCT level 3
50
100
150
200
250
50 100 150 200 250
MSE = 2.269800e-004,

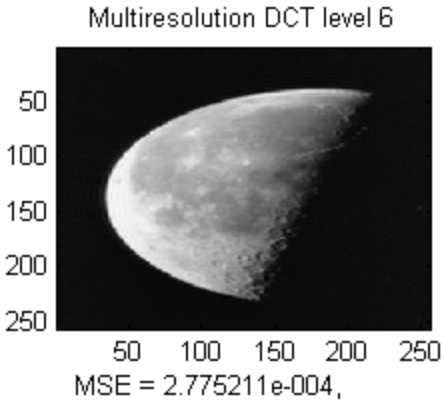

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 5.336040e-004, blockw = 32


Multiresolution DCT level 4
50
100
150
200
250
50 100 150 200 250
MSE = 2.551315e-004,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 9.586841e-004, blockw = 32


Multiresolution DCT level 5
50
100
150
200
250
50 100 150 200 250
MSE = 2.699360e-004,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 1.668065e-003, blockw = 32


Multiresolution DCT level 6
50
100
150
200
250
50 100 150 200 250
MSE = 2.775211e-004,

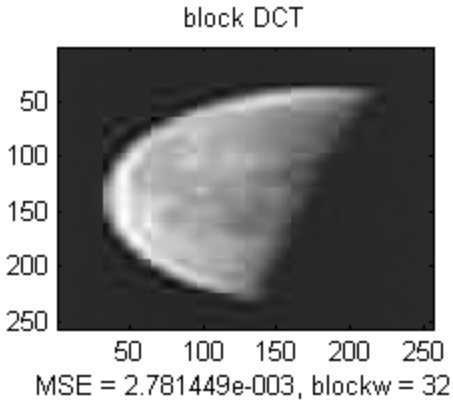
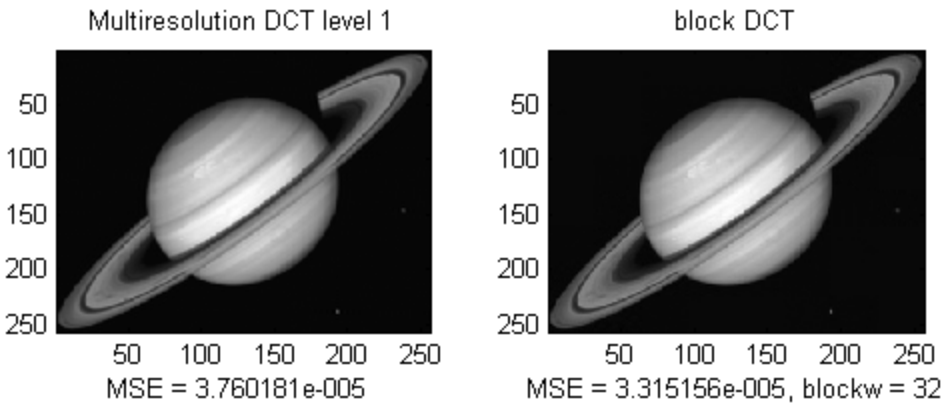
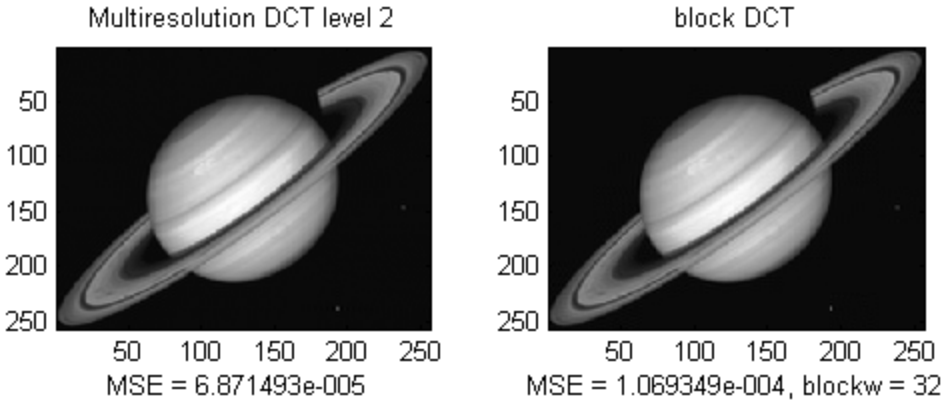
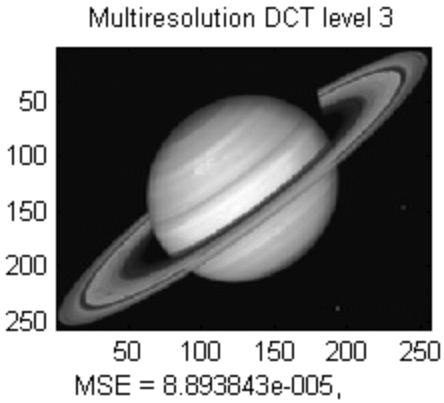
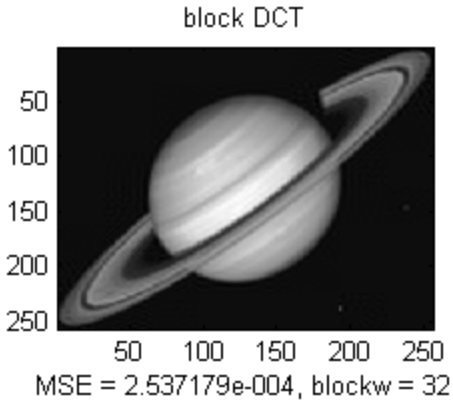

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 2.781449e-003, blockw = 32

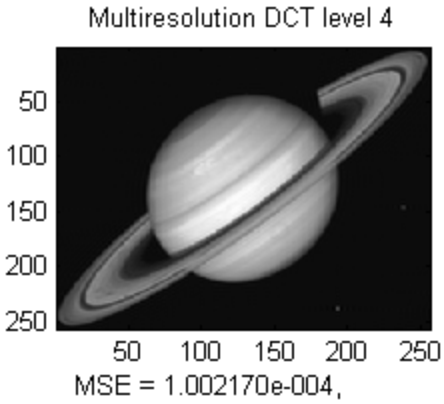
Image of Saturn at different resolutions with
Multiresolutuin DCT and Block DCT

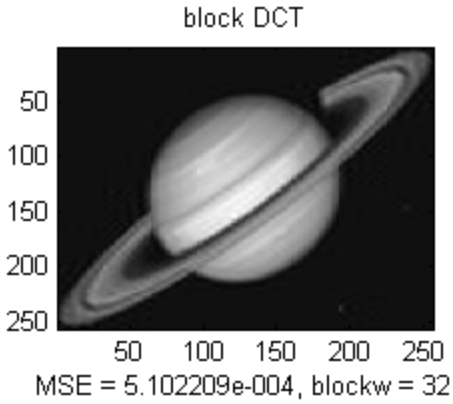


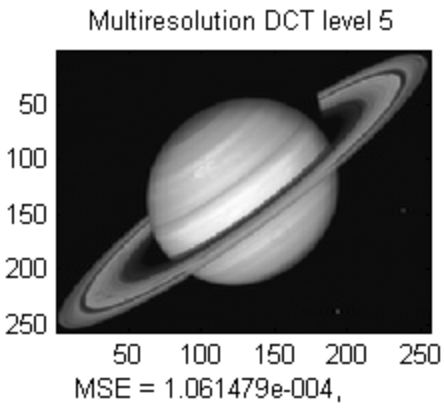


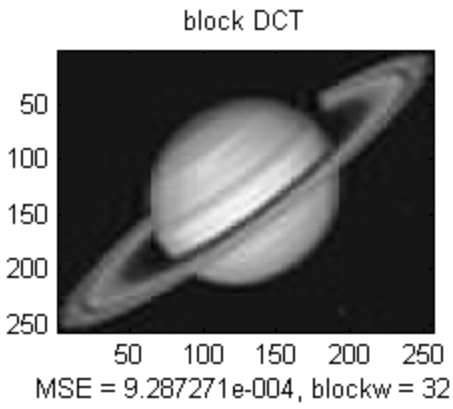

Multiresolution DCT level 3
50
100
150
200
250
50 100 150 200 250
MSE = 8.893843e-005,

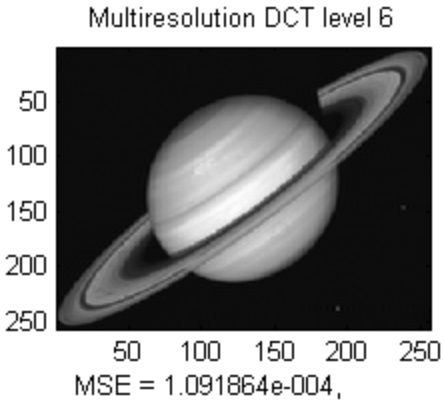

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 2.537179e-004, blockw = 32


Multiresolution DCT level 4
50
100
150
200
250
50 100 150 200 250
MSE = 1.002170e-004,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 5.102209e-004, blockw = 32


Multiresolution DCT level 5
50
100
150
200
250
50 100 150 200 250
MSE = 1.061479e-004,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 9.287271e-004, blockw = 32


Multiresolution DCT level 6
50
100
150
200
250
50 100 150 200 250
MSE = 1.091864e-004,

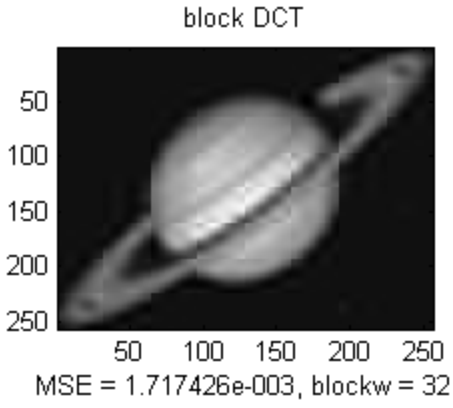
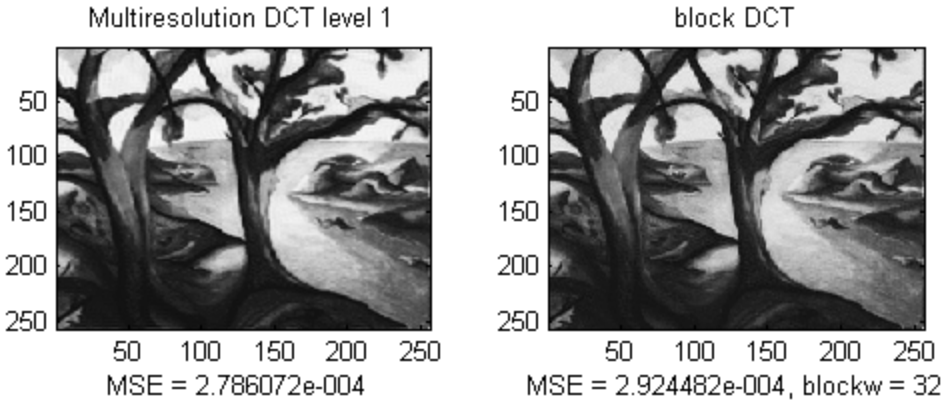
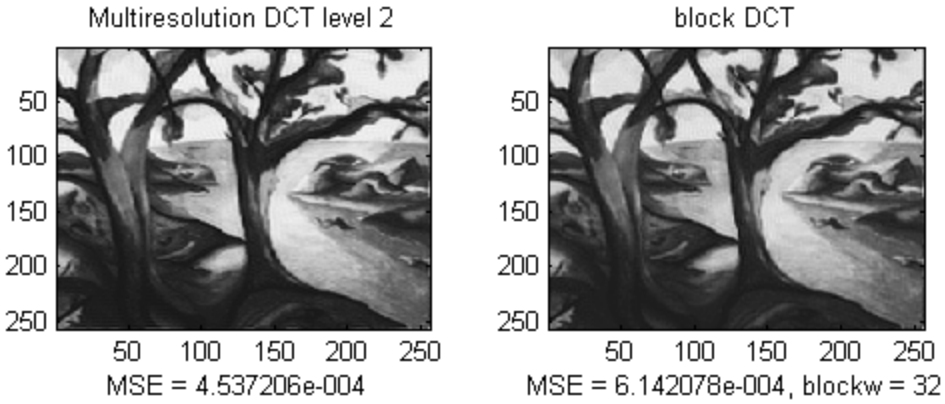
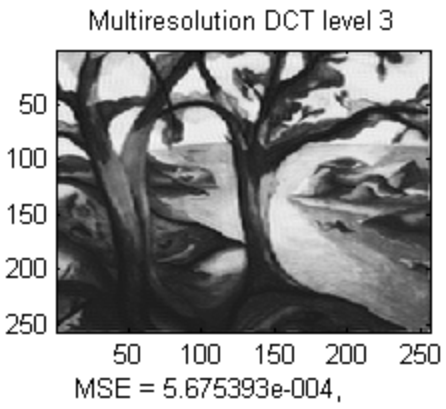
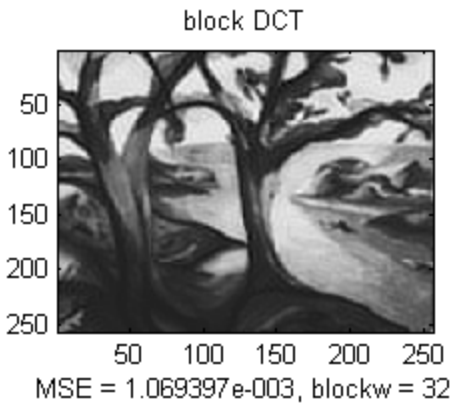

block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 1.717426e-003, blockw = 32

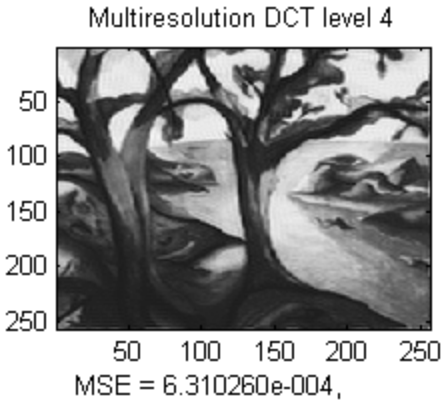
Image of Trees at different resolutions with
Multiresolutuin DCT and Block DCT

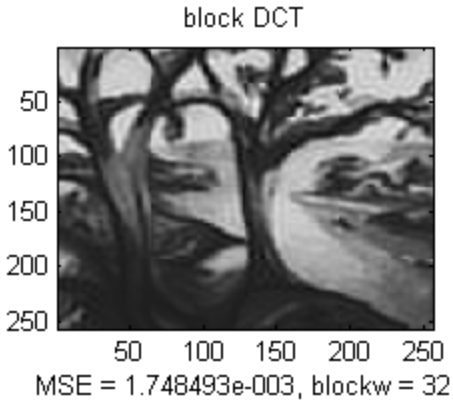





Multiresolution DCT level 3
50
100
150
200
250
50 100 150 200 250
MSE = 5.675393e-004,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 1.069397e-003, blockw = 32


Multiresolution DCT level 4
50
100
150
200
250
50 100 150 200 250
MSE = 6.310260e-004,


block DCT
50
100
150
200
250
50 100 150 200 250
MSE = 1.748493e-003, blockw = 32

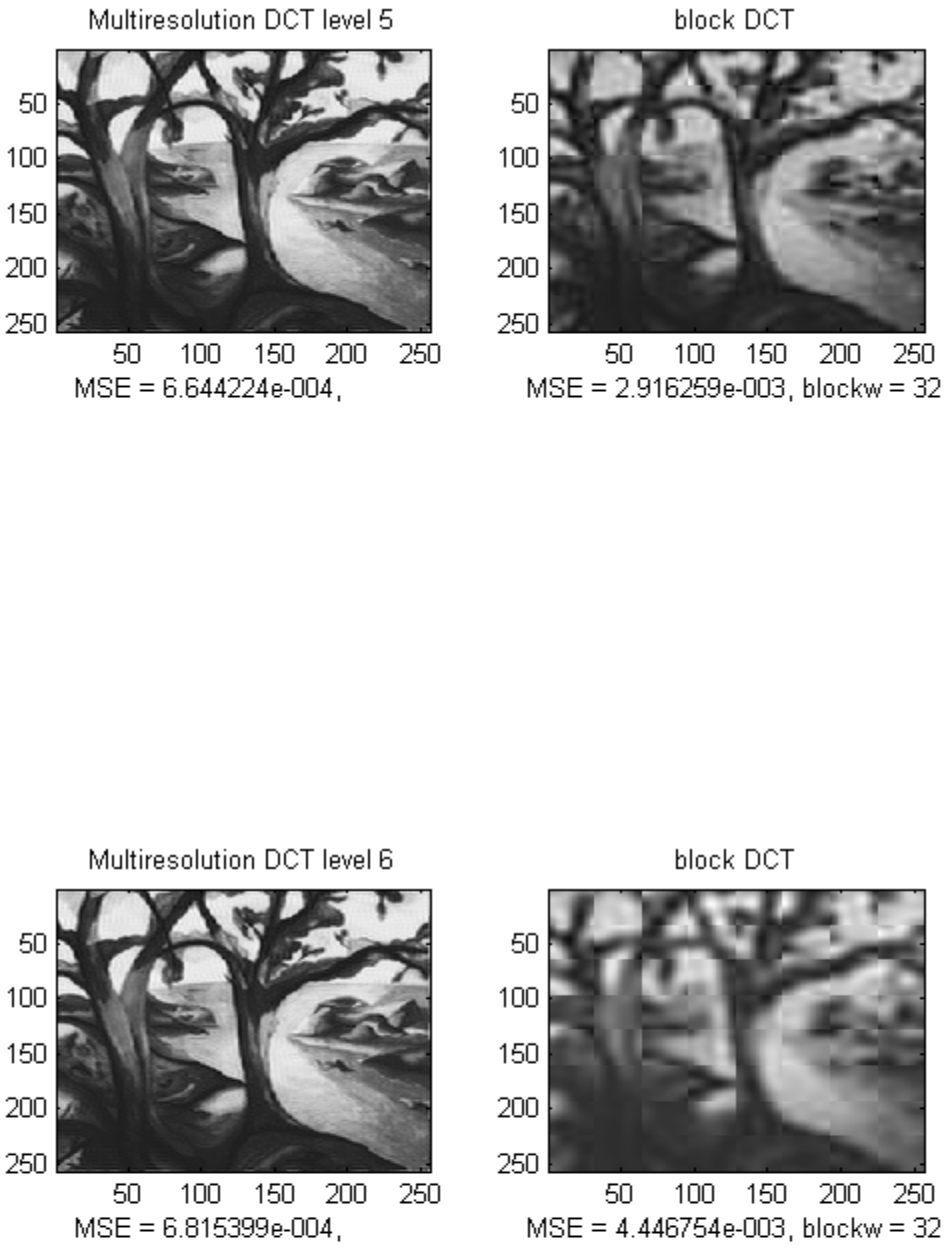


Figure 6.1: From page 35 – 64. Various images at different resolutions with two different approaches

The above images represent the comparison between our new multi-resolution DCT algorithm and the block DCT at different levels. At each level after the DCT is taken, 50% of the high frequency DCT coefficients and discarded and replaced by zeros to keep the original size. IDCT is taken after that to reconstruct image. At the next level the DCT is taken again for the reconstructed image and the DCT coefficients are spread again across the image. The sharp edges will be in the high frequency range and the other important information in the lower frequency range. It is evident from the above images that; since we keep discarding the high frequency coefficients, meaning the edges, the images at each subsequent level get blurrier. The claim that the DCT coefficients are spread nicely with most of the information being compacted in the lower frequency can be proven by the following graph of the DCT coefficients.

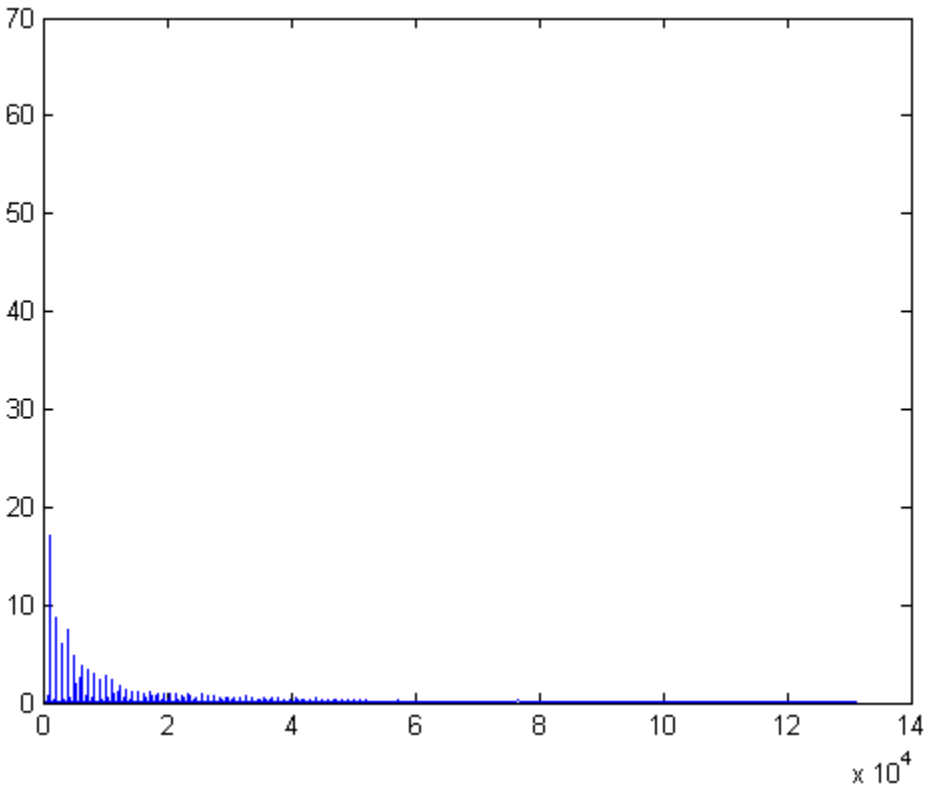


Figure 6.2: The DCT coefficients of a 256 x 256 image

Figure 6.2 represents the distribution of the DCT coefficients of a 256x256 image. The X axis represents the frequency and the Y axis represents the amplitude. It is clear from the above graph that the high frequency DCT coefficients do not contain much of the information about the image and hence can be discarded in the compression process.

We stopped our analysis at the sixth resolution mainly because the block DCT image at higher levels is almost unrecognizable and the blocking effect is more prominent than the actual image. It is clear from the images shown above that at every resolution the images with block DCT gets more blocking effects than the previous level. One the other hand, the images that have been applied the multi-resolution analysis, even though they get blurry on each resolution, the results are still much better than the block DCT mainly because of no blocking effect. The tables below show the mean square errors for various images at different resolutions with multi-resolution DCT and also with Block DCT.

Image Autum

Level	**MSE for Multi-resolution DCT**	**MSE for Block DCT**
1	3.548652e-004	1.807085e-004
2	7.369907e-004	5.544897e-004
3	9.854389e-004	1.190332e-003
4	1.124062e-003	2.112189e-003
5	1.197013e-003	3.312929e-003
6	1.234432e-003	4.344019e-003

Table 6.1 MSE for multi-resolution DCT and Block DCT at different levels for Autumn

Image Books

Level	MSE for Multi-resolution DCT	MSE for Block DCT
1	4.529395e-005	2.033844e-005
2	9.491778e-005	9.229971e-005
3	1.271715e-004	2.280965e-004
4	1.45190e-004	4.238396e-004
5	1.546181e-004	6.816383e-004
6	1.594641e-004	9.795495e-004

Table 6.2 MSE for multi-resolution DCT and Block DCT at different levels for Books

Image Cameraman

Level	MSE for Multi-resolution DCT	MSE for Block DCT
1	6.215928e-004	4.861981e-004
2	1.141939e-003	1.402589e-003
3	1.480183e-003	2.739405e-003
4	1.668838e-003	4.462630e-003
5	1.768058e-003	6.403300e-003
6	1.818897e-003	8.686007e-003

Table 6.3 MSE for multi-resolution DCT and Block DCT at different levels for Cameraman

Image flowers

Level	MSE for Multi-resolution DCT	MSE for Block DCT
1	8.163322e-004	8.656227e-004
2	1.451321e-003	2.193417e-003
3	1.864060e-003	3.842966e-003
4	2.094242e-003	5.786849e-003
5	2.215287e-003	8.391775e-003
6	2.277299e-003	1.117396e-003

Table 6.4 MSE for multi-resolution DCT and Block DCT at different levels for Flowers

Image Kids

Level	MSE for Multi-resolution DCT	MSE for Block DCT
1	6.044608e-005	6.603947e-005
2	9.167781e-005	1.177201e-004
3	1.119818e-004	1.687027e-004
4	1.233087e-004	2.368734e-004
5	1.292679e-004	3.386910e-004
6	1.323230e-004	4.750812e-004

Table 6.5 MSE for multi-resolution DCT and Block DCT at different levels for Kids

Image Lena

Level	MSE for Multi-resolution DCT	MSE for Block DCT
1	4.134701e-004	3.649426e-004
2	6.848883e-004	9.258494e-004
3	8.613066e-004	1.642532e-003
4	9.597309e-004	2.599402e-003
5	1.011525e-003	3.859901e-003
6	1.038089e-003	5.628552e-003

Table 6.6 MSE for multi-resolution DCT and Block DCT at different levels for Lena

Image Lily

Level	MSE for Multi-resolution DCT	MSE for Block DCT
1	4.076074e-004	3.702754e-004
2	8.268696e-004	1.192358e-003
3	1.099404e-003	2.564030e-003
4	1.251411e-003	4.683560e-003
5	1.331360e-003	7.51501e-003
6	1.372329e-003	1.149584e-002

Table 6.7 MSE for multi-resolution DCT and Block DCT at different levels for Lily

Image Moon

Level	MSE for Multi-resolution DCT	MSE for Block DCT
1	9.884459e-005	9.940853e-005
2	1.765026e-004	2.738870e-004
3	2.269800e-004	5.336040e-004
4	2.551315e-004	9.586841e-004
5	2.699360e-004	1.668065e-003
6	2.775211e-004	2.781449e-003

Table 6.8 MSE for multi-resolution DCT and Block DCT at different levels for Moon

Image Saturn

Level	MSE for Multi-resolution DCT	MSE for Block DCT
1	3.760181e-005	3.315156e-005
2	6.871493e-005	1.069349e-004
3	8.893843e-005	2.537179e-004
4	1.002170e-004	5.102209e-004
5	1.061479e-004	9.287271e-004
6	1.091864e-004	1.717426e-003

Table 6.9 MSE for multi-resolution DCT and Block DCT at different levels for Saturn

Image Trees

Level	MSE for Multi-resolution DCT	MSE for Block DCT
1	2.786072e-004	2.924482e-004
2	4.537206e-004	6.142078e-004
3	5.675393e-004	1.069397e-003
4	6.310620e-004	1.748493e-003
5	6.644224e-004	2.916259e-003
6	6.815399e-004	4.446754e-003

Table 6.10 MSE for multi-resolution DCT and Block DCT at different levels for Trees

The above tables from 6.1 to 6.10 represent the MSE for different images at different levels with two different approaches: the mutiresolution analysis for DCT and the block DCT. In order to present the results in a more meaningful manner the next table provides an average of the MSE of all the images at the different levels. These results will also be presented in a graphical manner.

Average MSE for All the images at each level

Level	Average MSE for Multi-resolution DCT	Average MSE for Block DCT
1	3.1347e-004	2.7791e-004
2	5.7275e-004	7.4738e-004
3	7.4130e-004	0.0014
4	8.3532e-004	0.0024
5	8.8476e-004	0.0036
6	9.1011e-004	0.0042

Table 6.11: Average MSE for all images for both Multi-resolution DCT and Block DCT

Table 6.11 provides an average of the MSE for each image at each level. It is clear from the above table that the MSE for Multi-resolution DCT does not change by a big margin at each level. One the other hand, it can be seen that the MSE for the block DCT changes with quite a big difference at each level. This is the reason we see very prominent blocking effects at higher levels in the images presented in figure 6.1. The results achieved in table 6.11 are also shown graphically in figure 6.2 below.

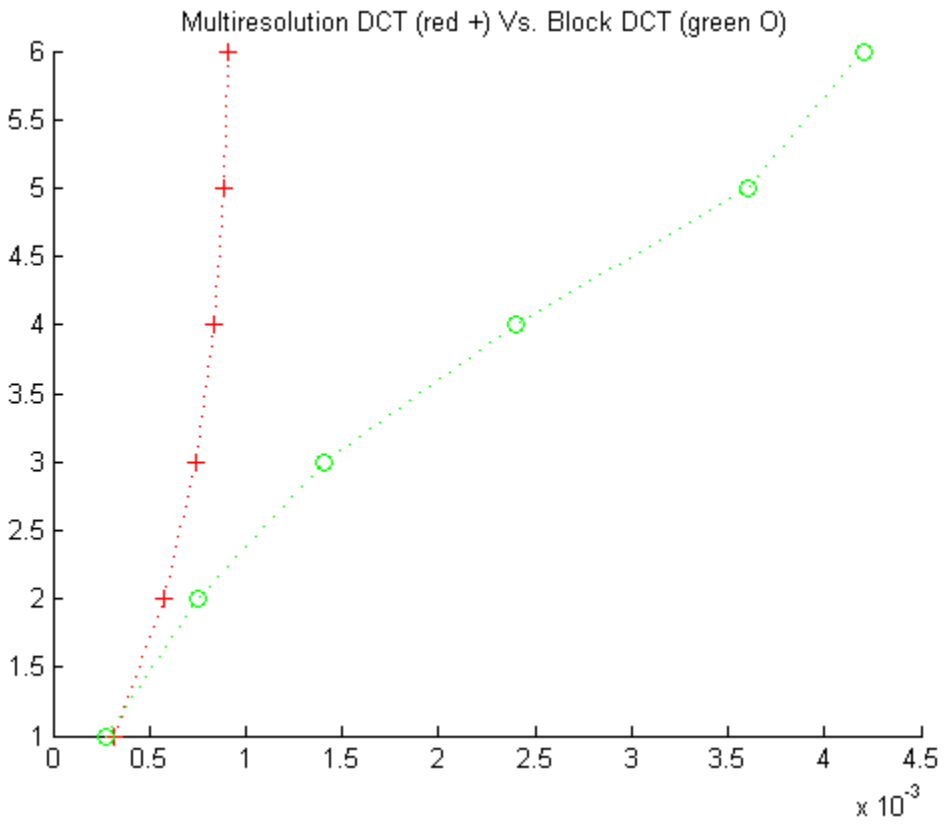


Figure 6.3: Multi-resolution vs. Block DCT (Y-axis represents the resolution and X-axis represents the MSE)

Figure 6.3 further elaborates our claim that the Multi-resolution analysis for DCT provides better results than the block DCT. The red line with "+" represents the MSE for Multi-resolution DCT while the "o" represents the MSE for block DCT at different levels. The graph for Multi-resolution DCT is very steep as compared to the graph of Block DCT which means that the MSE for Multi-resolution DCT at higher resolution levels is not as much as in Block DCT. For the first two resolutions, the MSE for both the Multi-resolution DCT and Block DCT are comparable and not that far off. But from the 3rd resolution, the differences between the two algorithms start to get bigger and bigger. This observation proves that to go to various levels of compressions, it is better to use Multi-resolution analysis than the block DCT.

Chapter 7

CONCLUSION AND FUTURE WORK

The purpose of this thesis was to come with a new algorithm that would eliminate the blocking from the DCT as well as apply multi-resolution analysis to the DCT. The experimental results obtained in the previous chapter proves our claims in the sense that the images reconstructed after compression do not have blocking artifacts and also we have shown that multi-resolution analysis can be applied to the DCT and, in fact, the results are very encouraging. We compared our algorithm with the block DCT using 32x32 blocks. The reason for choosing this block size is that any blocks smaller would produce worse results and any blocks bigger would take quite longer time to do the processing.

7.1 Future Work

The possibility of using multi-resolution analysis for DCT opens many doors for future research. The suggested future work is as follows:

7.1.1 Noise Reduction using multi-resolution analysis for DCT

Our new introduced algorithm of applying multi-resolution analysis on DCT can be used to achieve noise reduction from the images. At each resolution level the coefficients are averaged which perform the noise reduction for us. This process can be further improved by some more work. Instead of padding with zeros we can actually select the coefficients as done in the wavelets and this will further improve the quality of the noise reduction in the images.

Just to give an idea of what can be done, Gaussian noise was introduced in a coupe of images and then our algorithm was run on them. The effects and the MSE for the two images are shown below:

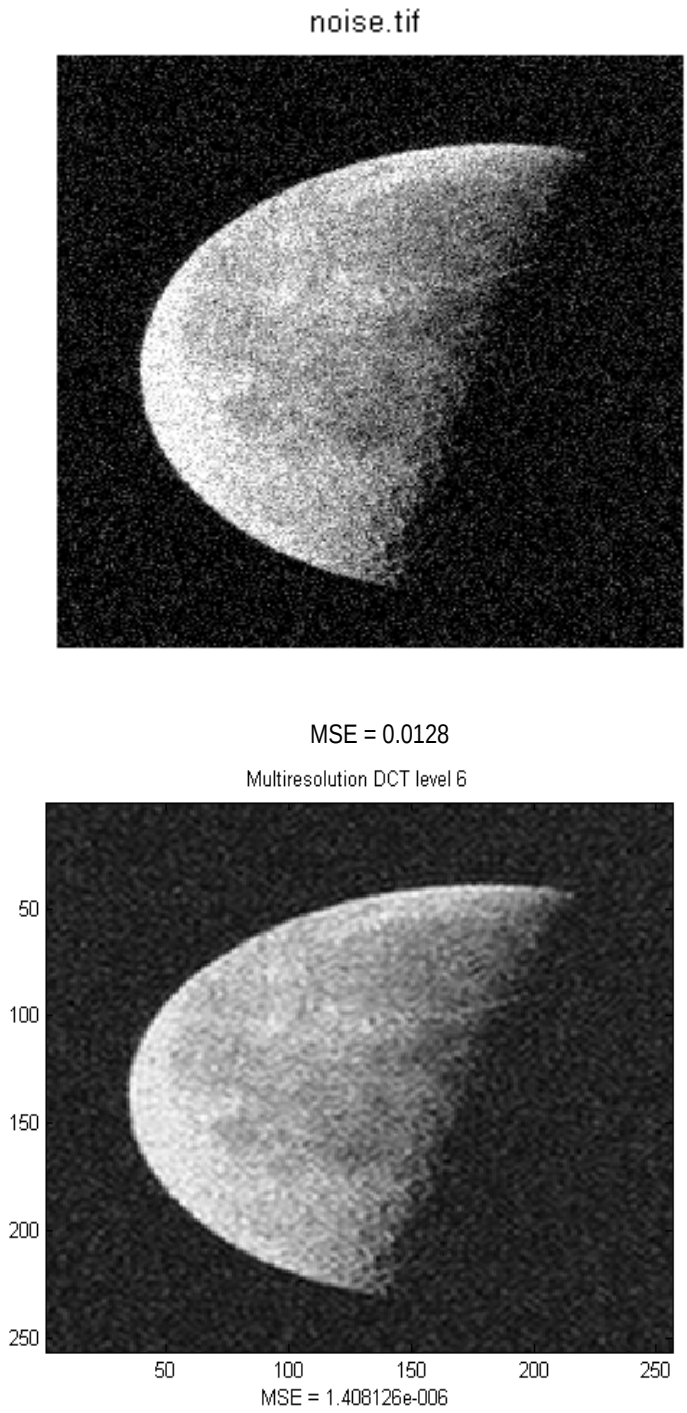


Figure 7.1: Images of moon with Noise: The top figure shows the image with Gaussian noise and the bottom figure shows the sixth level of resolution with noise reduction

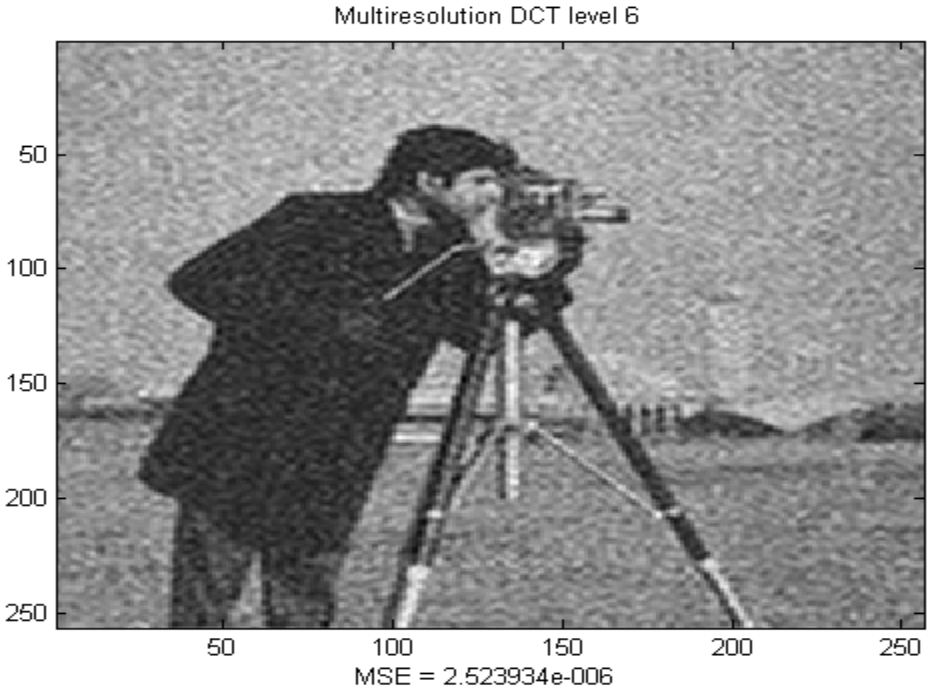


Figure 7.2: Images of cameraman with Noise: The top figure shows the image with Gaussian noise and the bottom figure shows the sixth level of resolution with noise reduction

Figures 7.1 and 7.2 clearly show the MSE at each level is reduced considerably which means that the noise is reduced at each level. The results from this are very encouraging and the future work can be done to improve it and also to compare it with the wavelets to see if better results can be obtained with DCT than with wavelets.

7.1.2 Lossless Compress using DCT

By using the proposed technique "*Lossless"* compression is possible from the DCT. In our algorithm, at each resolution, after taking the DCT of the coefficients, we discard the high resolution 50% DCT coefficients. These coefficients mainly contain the edges of the images. If we save all the coefficients that we discard in our algorithm, then at the lowest resolution we can try to reverse the process and build up from the last resolution but taking the DCT of the last resolution image and replacing the high frequency 50% of the DCT coefficients with the coefficients that we saved (instead of discarding) while going down the resolution level. This would make sure that we are not loosing any information and hence the compression would be Lossless instead of the default lossy DCT compression.

7.1.3 MDCT for Image compression

MDCT has been proven to be the best technique so far for sound compression because of its technique for time domain alias cancellation. The alternate approach mentioned in the previous chapter can be implemented using MDCT since both use one dimensional DCT.

References:

[1]. I.H. Barkdoll, B.L. McGlamery: "An online Image processing system". *Proceedings of the 1968 23rd ACM national conference Pages: 705 - 716,* Year of Publication: 1968

[2]. Robert Leggat: *A History of Photography.*

http://www.rleggat.com/photohistory/index.html

[3] Bottou, L., Howard, P. G., and Bengio, Y. "The Z-Coder Adaptive Binary Coder", *Proc. IEEE DCC*, Mar. 1998, pp. 13-22, http://www.research.att.com/~leonb/PS/bottou-howard-bengio.ps.gz.

[4] Pavel A. Chochia, "Image enhancement using sliding histograms", *Computer Vision, Graphics, and Image Processing*, v.44 n.2, p.211-229, Nov. 1988

[5] http://www.quantlet.com/mdstat/scripts/wav/html/wavhtmlnode13.html

[6] MICHAEL LOUNSBERY, TONY D. DEROSE, and JOE WARREN, "Multiresolution Analysis for Surfaces of Arbitrary Topological Type", *ACM Transactions on Graphics*, Vol. 16, No. 1, January 1997.

[7] Stephane G. Mallat **A, "**Theory for Multi-resolution Signal Decomposition: The Wavelet Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 2. NO. **7.** JULY 1989

[8] Hairong Qi, Wesley E. Snyder, Griff L. Bilbro ; "Comparison of Mean Field Annealing and Multi-resolution Analysis in Missing Data Estimation", *Electrical and Computer Engineering Department, Box 7911 North Carolina State University, Raleigh*, NC 27695-7911, U.S.A

[9] **Mauro Maggioni** , Department of Mathematics, Yale University http://www.math.yale.edu/%7Emmm82/hrmwav.htm

[10] An Introduction to Wavelets
IEEE Computational Science and Engineering, Summer 1995, vol. 2.
http://www.amara.com/IEEEwave/IEEEwavelet.html

[11] Jackie (Jianhong) Shen; "Wavelets: Motivation, Construction, & Application", *School of Math, University of Minnesota, Minneapolis*

[12] http://www.ct-magazine.com/archives/ct/0300/feature2.htm

[13] Zixiang Xiong, Kannan Ramchandran, Michael T. Orchard, and Ya-Qin Zhang; "A comparative study of DCT and wavelet based image coding:, *IEEE Transactions on circuits and systems for video technology,* VOL. 9, NO. 5, AUGUST 1999

[14] Ahumada, Albert J., Jr. Peterson, Heidi A.: "A visual detection model for DCT coefficient quantization", *AIAA Computing in Aerospace Conference, 9th, San Diego, CA, Oct. 19-21, 1993, Technical Papers.* Pt. 1 (A94-11401 01-62), Washington, American Institute of Aeronautics and Astronautics, 1993, p. 314-318

[15]. Andrew B. Watson, "Image Compression Using the Discrete Cosine Transform", *NASA Ames Research Center.*

[16]. A. Gresho, R. M. Gray, "Vector Quantization and Signal Compression", *Kluwer Academic Publishers, 1991.*

[17] Kannan Ramchandran, Antonio Ortega, K. Metin Uz, and Martin Vetterli, "Multi-resolution broadcast for digital HDTV using joint source/channel coding." *IEEE Journal on Selected Areas in Communications,* 11(1):6-23, January 1993.

[18] Mark W. Garrett and Martin Vetterli, "Joint Source/Channel Coding of Statistically Multiplexed Real Time Services on Packet Networks," *IEEE/ACM Transactions on Networking,* 1993.

[19] S. McCanne and Martin Vetterli, "Joint source/channel coding for multicast packet video,"International Conference on Image Processing (Vol. 1)-Volume 1, October 1995, Washington D.C**.**

[20] Khalid Sayood and Jay C. Borkenhagen, "Use of residual redundancy in the design of joint source/channel coders," *IEEE Transactions on Communications,* 39(6):838-846, June 1991.

[21] J.Modestino, D.G. Daut, and A. Vickers, "Combined source channel coding of images usingthe block cosine transform," *IEEE Transactions on Communications,* vol. 29, pp.1261-1274,September 1981.

[22] Irina Popovici and Wm. Douglas Withers, "The Eidochromatic Transform for Color Image Coding", *IEEE Transactions on Image Processing,* vol. 14, no. 3, MARCH 2005

[23] Cutnell, John D. and Kenneth W. Johnson. *Physics*. 4th ed. New York: Wiley, 1998: 466.

[24] Julio Enrique Castrillon-Candas and Kevin Amaratunga, "Fast Estimation of Karhunen-Loeve Eigen Function usin Wavelets," Submitted to *IEEE Transactions on Signal Processing,* 2001.

[25] A. Jain, "A Fast Karhunen-loeve Transform for Digital Restoration of Images Degraded by White and Colored Noise," *IEEE Transactions on Computers,* vol. 26, number 6, June 1977.

[26] Syed Ali Khayam, "The Discrete Cosine Transform: Theory and Application", Michigan State University, March 2003

[27] Hoon Paek; Sang-Uk Lee, "A projection-based post-processing technique to reduce blocking artifact using a priori information on DCT coefficients of adjacent blocks", *Image Processing, 1996. Proceedings., International Conference on* , Volume: 1 , 16-19, Sept. 1996.

[28] Hoon Paek; Rin-Chul Kim; "Sang-Uk Lee, A DCT-based spatially adaptive post-processing technique to reduce the blocking artifacts in transform coded images", *Circuits and Systems for Video Technology, IEEE Transactions on* , Volume: 10 Issue: 1, Feb. 2000.

[29] Malvar, H.S. "Signal Processing with Lapped Transforms", Artech House (1992).

[30] Belkasim, S., and Bhatia, P., *Image compression using one-dimensional DCT*, 2002.

[31] J. Princen and A. Bradley, "Analysis/Synthesis Filterbank Design Based on Time Domain Aliasing Cancellation," *IEEE Trans. on* ***Acoust. Speech,*** *and Signal Process.* Vol. ASSP-34, pp.1153-1161, 1986.

[32] Y Wang, Miika Vilermo, "The Modified Discrete Cosine Transform, Coding and Error Concealment", *Nokia Research Center, P.O. Box 100, FIN-33721 Tampere, Finland*

[33] Wang, Y., Yaroslavsky, L., Vilermo, M., and Vaananen, M.: 'Some peculiar properties of the MDCT'. Proc. ICSP 2000

[34]. Wang Jianxin, Dong Zaiwang, "A fast algorithm for discrete cosine transform", *Department of Electrical Engineering, Tsinghua University Beijing, China.*

[35] Sung-Hwan Jung, Sanjit K. Mitra, and Debargha Mukherjee, "Subband DCT: Definition, Analysis, and Applications", *IEEE Transactions on circuits and system for video technology*, VOL. 6, NO. 3, JUNE 1996

[36] Jie Huang, Wu-chi Feng, Jonathan Walpole, and Wilfried Jouve, "An experimental analysis of DCT-based approaches for fine-grain multi-resolution video", *An experimental analysis of DCT-based approaches for fine-grain multi-resolution video*

[37] Seymour Shlien, "The Modulated Lapped Transform, Its Time-Varying Forms, and Its Applications to Audio Coding Standards", *IEEE Transactions on speech and audio processing*, Vol. 5, No. 4, July 1997

APPENDIX A

There are three files used in this thesis. All the code has been written in MATLAB 7.0. The files with their codes are listed below.

DCT_multires.m

```
function DCT_multires(ratio)

%compress an image in two opposite directions of rows and columns in 1d dct
%cascade the rows with columns in horizontal and vertical zigzag wave.
%ie the last element in a row or
% a column will be next to the last element of the next row or column.
%read an image and convert it to double
% discard elements and then reconstruct the image and repeat the process in
% several resolution levels
%there is no ordering of coefficients for the cascaded method
%the block dct still uses the ordering which is results in unfair comparison.
% this function calls blockdct_comp(a,ratio,8,8,rowoverlap,coloverlap);
% and bidirection(a) where a is an image to be wave zigaged.
% We may be able to plot the error for changing the lenght of the vector from
% one row to 2,3 up to the whole image size. compare this to the block dct
% by also changing the block dct size. from 8 to 16, 32...
```

```
% for high compression ratios we used 32x32 due the fact that the number of coefficients
% in the 8x8 dct does not allow low bit rate or high compression.
% We can also use the zigzag waving on the blocks and try the 2d zigzag waves.
a = imread('trees1.tif');
a = im2double(a);
ao=a;
%a = a(32:159, 64:191);%get 128 by 128 windnow
% get the size of the image
[row col] = size(a);

% Threshold is decided by the percentage of compression desired 0.90 means 90% compression

if(nargin <1)
   error('Please enter the percentage of compression');
end
rowoverlap=0;
coloverlap=0;
window_rows=row;
window_cols=col;
totalsize=row*col;
totalsize=totalsize+totalsize;
thr = totalsize-round(totalsize*(ratio));
%get the row array
```

```
fwave=bidirection(a);%to get the rows reversed
onedarrayr= im2col(fwave,[row,col]);

%the column array;
fwavec=bidirection(a');
onedarrayc= im2col(fwavec,[row,col]);

onedarray(1:(totalsize/2),1)=onedarrayr;
onedarray(((totalsize/2)+1):totalsize,1)=onedarrayc;

%disp('the dimension of the array is'); row,size(onedarray),size(onedarrayr)
dct_onedarray=dct2(onedarray,[totalsize,1]);%use 2d dct to compute 1d
%dim=size(dct_onedarray);
%disp('the dimension of the array is'); dim
%sort the coefficients based on magnitude
%[y,i] = sort(abs(dct_onedarray));
c1 = dct_onedarray;
%dim=size(i);
%disp('the dimension of the i array is'); dim
%set the smallest ratio(90% for example) to zero.
c1((thr:totalsize),1) = 0;
%inverse one d dct
```

```
idct_onedarray = idct2(c1,[totalsize,1]);
onedarrayro=idct_onedarray(1:(totalsize/2));
onedarrayco=idct_onedarray(((totalsize/2)+1):totalsize);
bidimager=col2im(onedarrayro,[1,1],[row,col]);
%The wave inverse transform
%get the even rows reversed
fw=bidirection(bidimager);%to get the rows reversed
fin=zeros(row,col);
fin=fw;
idct_roww=fin;
%get the column-wise dct coefficients
bidimagec=col2im(onedarrayco,[1,1],[row,col]);

fw=bidirection(bidimagec);%
fin=zeros(row,col);
fin=fw;
idct_colw=fin';

for i=1:row
    for j=1:col
       %i,j
      all=[idct_roww(i,j),idct_colw(i,j)];
```

```
        finI(i,j) =sum(all)/2.0; %median(all);
        %finI(i,j)=min(idct_row(i,j)+idct_col(i,j));
    end
end
fin=finI;
%error for average
finI=fin;
error1 = a-finI;
error1 = error1.^2;
MSE_avg = sum(error1(:))/prod(size(a));
%ROW ERROR
finI=idct_roww;
error1 = a-finI;
error1 = error1.^2;
MSE_row = sum(error1(:))/prod(size(a));
%column ERRO
finI=idct_colw;
error1 = a-finI;
error1 = error1.^2;
MSE_col = sum(error1(:))/prod(size(a));
figure;
colormap(gray(256));
%subplot(2,2,1);imagesc(a); title('Original');
```

```
subplot(2,2,1);imagesc(fin);title('Multiresolution DCT level 1');
s = sprintf('MSE = %d ',MSE_avg);
xlabel(s);
clear s;
%coloverlap=1;
%8X8
window_rows=32;
window_cols=32;
orig_ratio = ratio;
ratio_dct = ratio;
[Res,MSE] = blockdct_comp(a,ratio,window_rows,window_cols,rowoverlap,coloverlap);

% Simple block processing

subplot(2,2,2);imagesc(Res); title('block DCT');%
clear s;
s = sprintf('MSE = %d, blockw = %d',MSE,window_rows);
xlabel(s);

%level 2 resoluution
a=fin;
%get the row array
fwave=bidirection(a);%to get the rows reversed
```

```
onedarrayr= im2col(fwave,[row,col]);


%the column array;
fwavec=bidirection(a');
onedarrayc= im2col(fwavec,[row,col]);


onedarray(1:(totalsize/2),1)=onedarrayr;
onedarray(((totalsize/2)+1):totalsize,1)=onedarrayc;


%disp('the dimension of the array is'); row,size(onedarray),size(onedarrayr)
dct_onedarray=dct2(onedarray,[totalsize,1]);%use 2d dct to compute 1d
%dim=size(dct_onedarray);
%disp('the dimension of the array is'); dim
%sort the coefficients based on magnitude
%[y,i] = sort(abs(dct_onedarray));
c1 = dct_onedarray;
%dim=size(i);
%disp('the dimension of the i array is'); dim
%set the smallest ratio(90% for example) to zero.
c1((thr:totalsize),1) = 0;
%inverse one d dct
idct_onedarray = idct2(c1,[totalsize,1]);
onedarrayro=idct_onedarray(1:(totalsize/2));
```

```
onedarrayco=idct_onedarray(((totalsize/2)+1):totalsize);
bidimager=col2im(onedarrayro,[1,1],[row,col]);
%The wave inverse transform
%get the even rows reversed
fw=bidirection(bidimager);%to get the rows reversed
fin=zeros(row,col);
fin=fw;
idct_roww=fin;
%get the column-wise dct coefficients
bidimagec=col2im(onedarrayco,[1,1],[row,col]);

fw=bidirection(bidimagec);%
fin=zeros(row,col);
fin=fw;
idct_colw=fin';

for i=1:row
    for j=1:col
       %i,j
      all=[idct_roww(i,j),idct_colw(i,j)];

      finI(i,j) =sum(all)/2.0; %median(all);
      %finI(i,j)=min(idct_row(i,j)+idct_col(i,j));
```

```
    end
end
fin=finI;
%error for average
finI=fin;
error1 = ao-finI;
error1 = error1.^2;
MSE_avg = sum(error1(:))/prod(size(ao));
%ROW ERROR
finI=idct_roww;
error1 = ao-finI;
error1 = error1.^2;
MSE_row = sum(error1(:))/prod(size(ao));
%column ERRO
finI=idct_colw;
error1 = ao-finI;
error1 = error1.^2;
MSE_col = sum(error1(:))/prod(size(ao));
figure;
colormap(gray(256));
%subplot(2,2,1);imagesc(a); title('Original');
subplot(2,2,1);imagesc(fin);title('Multiresolution DCT level 2');
s = sprintf('MSE = %d ',MSE_avg);
```

```
xlabel(s);

clear s;

%coloverlap=1;

%8X8


ratio_dct = ratio_dct + (orig_ratio/2);


window_rows=32;

window_cols=32;

[Res,MSE] = blockdct_comp(ao,ratio_dct,window_rows,window_cols,rowoverlap,coloverlap);


% Simple block processing


subplot(2,2,2);imagesc(Res); title('block DCT');%

clear s;

s = sprintf('MSE = %d, blockw = %d',MSE,window_rows);

xlabel(s);



%level 3 resoluution

a=fin;

%get the row array

fwave=bidirection(a);%to get the rows reversed
```

```
onedarrayr= im2col(fwave,[row,col]);


%the column array;
fwavec=bidirection(a');
onedarrayc= im2col(fwavec,[row,col]);


onedarray(1:(totalsize/2),1)=onedarrayr;
onedarray(((totalsize/2)+1):totalsize,1)=onedarrayc;


%disp('the dimension of the array is'); row,size(onedarray),size(onedarrayr)
dct_onedarray=dct2(onedarray,[totalsize,1]);%use 2d dct to compute 1d
%dim=size(dct_onedarray);
%disp('the dimension of the array is'); dim
%sort the coefficients based on magnitude
%[y,i] = sort(abs(dct_onedarray));
c1 = dct_onedarray;
%dim=size(i);
%disp('the dimension of the i array is'); dim
%set the smallest ratio(90% for example) to zero.
c1((thr:totalsize),1) = 0;
%inverse one d dct
idct_onedarray = idct2(c1,[totalsize,1]);
onedarrayro=idct_onedarray(1:(totalsize/2));
```

```
onedarrayco=idct_onedarray(((totalsize/2)+1):totalsize);
bidimager=col2im(onedarrayro,[1,1],[row,col]);
%The wave inverse transform
%get the even rows reversed
fw=bidirection(bidimager);%to get the rows reversed
fin=zeros(row,col);
fin=fw;
idct_roww=fin;
%get the column-wise dct coefficients
bidimagec=col2im(onedarrayco,[1,1],[row,col]);

fw=bidirection(bidimagec);%
fin=zeros(row,col);
fin=fw;
idct_colw=fin';

for i=1:row
    for j=1:col
       %i,j
      all=[idct_roww(i,j),idct_colw(i,j)];

      finI(i,j) =sum(all)/2.0; %median(all);
      %finI(i,j)=min(idct_row(i,j)+idct_col(i,j));
```

```
    end
end
fin=finI;
%error for average
finI=fin;
error1 = ao-finI;
error1 = error1.^2;
MSE_avg = sum(error1(:))/prod(size(ao));
%ROW ERROR
finI=idct_roww;
error1 = ao-finI;
error1 = error1.^2;
MSE_row = sum(error1(:))/prod(size(ao));
%column ERRO
finI=idct_colw;
error1 = ao-finI;
error1 = error1.^2;
MSE_col = sum(error1(:))/prod(size(ao));
figure;
colormap(gray(256));
%subplot(2,2,1);imagesc(a); title('Original');
subplot(2,2,1);imagesc(fin);title('Multiresolution DCT level 3');
s = sprintf('MSE = %d,stitch',MSE_avg);
```

```
xlabel(s);

clear s;


%coloverlap=1;

%8X8

ratio_dct = ratio_dct + (orig_ratio/4);

window_rows=32;

window_cols=32;

[Res,MSE] = blockdct_comp(ao,ratio_dct,window_rows,window_cols,rowoverlap,coloverlap);


% Simple block processing


subplot(2,2,2);imagesc(Res); title('block DCT');%

clear s;

s = sprintf('MSE = %d, blockw = %d',MSE,window_rows);

xlabel(s);




%level 4 resoluution

a=fin;

%get the row array

fwave=bidirection(a);%to get the rows reversed
```

```
onedarrayr= im2col(fwave,[row,col]);


%the column array;
fwavec=bidirection(a');
onedarrayc= im2col(fwavec,[row,col]);


onedarray(1:(totalsize/2),1)=onedarrayr;
onedarray(((totalsize/2)+1):totalsize,1)=onedarrayc;


%disp('the dimension of the array is'); row,size(onedarray),size(onedarrayr)
dct_onedarray=dct2(onedarray,[totalsize,1]);%use 2d dct to compute 1d
%dim=size(dct_onedarray);
%disp('the dimension of the array is'); dim
%sort the coefficients based on magnitude
%[y,i] = sort(abs(dct_onedarray));
c1 = dct_onedarray;
%dim=size(i);
%disp('the dimension of the i array is'); dim
%set the smallest ratio(90% for example) to zero.
c1((thr:totalsize),1) = 0;
%inverse one d dct
idct_onedarray = idct2(c1,[totalsize,1]);
onedarrayro=idct_onedarray(1:(totalsize/2));
```

```
onedarrayco=idct_onedarray(((totalsize/2)+1):totalsize);
bidimager=col2im(onedarrayro,[1,1],[row,col]);
%The wave inverse transform
%get the even rows reversed
fw=bidirection(bidimager);%to get the rows reversed
fin=zeros(row,col);
fin=fw;
idct_roww=fin;
%get the column-wise dct coefficients
bidimagec=col2im(onedarrayco,[1,1],[row,col]);

fw=bidirection(bidimagec);%
fin=zeros(row,col);
fin=fw;
idct_colw=fin';

for i=1:row
    for j=1:col
       %i,j
      all=[idct_roww(i,j),idct_colw(i,j)];

      finI(i,j) =sum(all)/2.0; %median(all);
      %finI(i,j)=min(idct_row(i,j)+idct_col(i,j));
```

```
    end
end
fin=finI;
%error for average
finI=fin;
error1 = ao-finI;
error1 = error1.^2;
MSE_avg = sum(error1(:))/prod(size(ao));
%ROW ERROR
finI=idct_roww;
error1 = ao-finI;
error1 = error1.^2;
MSE_row = sum(error1(:))/prod(size(ao));
%column ERRO
finI=idct_colw;
error1 = ao-finI;
error1 = error1.^2;
MSE_col = sum(error1(:))/prod(size(ao));
figure;
colormap(gray(256));
%subplot(2,2,1);imagesc(a); title('Original');
subplot(2,2,1);imagesc(fin);title('Multiresolution DCT level 4');
s = sprintf('MSE = %d,stitch',MSE_avg);
```

```
xlabel(s);

clear s;

%coloverlap=1;

%8X8

ratio_dct = ratio_dct + (orig_ratio/8);

window_rows=32;

window_cols=32;

[Res,MSE] = blockdct_comp(ao,ratio_dct,window_rows,window_cols,rowoverlap,coloverlap);


% Simple block processing


subplot(2,2,2);imagesc(Res); title('block DCT');%

clear s;

s = sprintf('MSE = %d, blockw = %d',MSE,window_rows);

xlabel(s);



%level 5 resoluution

a=fin;

%get the row array

fwave=bidirection(a);%to get the rows reversed

onedarrayr= im2col(fwave,[row,col]);
```

```
%the column array;
fwavec=bidirection(a');
onedarrayc= im2col(fwavec,[row,col]);

onedarray(1:(totalsize/2),1)=onedarrayr;
onedarray(((totalsize/2)+1):totalsize,1)=onedarrayc;

%disp('the dimension of the array is'); row,size(onedarray),size(onedarrayr)
dct_onedarray=dct2(onedarray,[totalsize,1]);%use 2d dct to compute 1d
%dim=size(dct_onedarray);
%disp('the dimension of the array is'); dim
%sort the coefficients based on magnitude
%[y,i] = sort(abs(dct_onedarray));
c1 = dct_onedarray;
%dim=size(i);
%disp('the dimension of the i array is'); dim
%set the smallest ratio(90% for example) to zero.
c1((thr:totalsize),1) = 0;
%inverse one d dct
idct_onedarray = idct2(c1,[totalsize,1]);
onedarrayro=idct_onedarray(1:(totalsize/2));
onedarrayco=idct_onedarray(((totalsize/2)+1):totalsize);
bidimager=col2im(onedarrayro,[1,1],[row,col]);
```

```
%The wave inverse transform
%get the even rows reversed
fw=bidirection(bidimager);%to get the rows reversed
fin=zeros(row,col);
fin=fw;
idct_roww=fin;
%get the column-wise dct coefficients
bidimagec=col2im(onedarrayco,[1,1],[row,col]);

fw=bidirection(bidimagec);%
fin=zeros(row,col);
fin=fw;
idct_colw=fin';

for i=1:row
    for j=1:col
       %i,j
      all=[idct_roww(i,j),idct_colw(i,j)];

      finI(i,j) =sum(all)/2.0; %median(all);
      %finI(i,j)=min(idct_row(i,j)+idct_col(i,j));
   end
end
```

```
fin=finI;
%error for average
finI=fin;
error1 = ao-finI;
error1 = error1.^2;
MSE_avg = sum(error1(:))/prod(size(ao));
%ROW ERROR
finI=idct_roww;
error1 = ao-finI;
error1 = error1.^2;
MSE_row = sum(error1(:))/prod(size(ao));
%column ERRO
finI=idct_colw;
error1 = ao-finI;
error1 = error1.^2;
MSE_col = sum(error1(:))/prod(size(ao));
figure;
colormap(gray(256));
%subplot(2,2,1);imagesc(a); title('Original');
subplot(2,2,1);imagesc(fin);title('Multiresolution DCT level 5');
s = sprintf('MSE = %d,stitch',MSE_avg);
xlabel(s);
clear s;
```

```
%coloverlap=1;
%8X8
ratio_dct = ratio_dct + (orig_ratio/16);
window_rows=32;
window_cols=32;
[Res,MSE] = blockdct_comp(ao,ratio_dct,window_rows,window_cols,rowoverlap,coloverlap);

% Simple block processing

subplot(2,2,2);imagesc(Res); title('block DCT');%
clear s;
s = sprintf('MSE = %d, blockw = %d',MSE,window_rows);
xlabel(s);


%level 6 resoluution
a=fin;
%get the row array
fwave=bidirection(a);%to get the rows reversed
onedarrayr= im2col(fwave,[row,col]);

%the column array;
fwavec=bidirection(a');
```

```
onedarrayc= im2col(fwavec,[row,col]);


onedarray(1:(totalsize/2),1)=onedarrayr;
onedarray(((totalsize/2)+1):totalsize,1)=onedarrayc;


%disp('the dimension of the array is'); row,size(onedarray),size(onedarrayr)
dct_onedarray=dct2(onedarray,[totalsize,1]);%use 2d dct to compute 1d
%dim=size(dct_onedarray);
%disp('the dimension of the array is'); dim
%sort the coefficients based on magnitude
%[y,i] = sort(abs(dct_onedarray));
c1 = dct_onedarray;
%dim=size(i);
%disp('the dimension of the i array is'); dim
%set the smallest ratio(90% for example) to zero.
c1((thr:totalsize),1) = 0;
%inverse one d dct
idct_onedarray = idct2(c1,[totalsize,1]);
onedarrayro=idct_onedarray(1:(totalsize/2));
onedarrayco=idct_onedarray(((totalsize/2)+1):totalsize);
bidimager=col2im(onedarrayro,[1,1],[row,col]);
%The wave inverse transform
%get the even rows reversed
```

```
fw=bidirection(bidimager);%to get the rows reversed
fin=zeros(row,col);
fin=fw;
idct_roww=fin;
%get the column-wise dct coefficients
bidimagec=col2im(onedarrayco,[1,1],[row,col]);

fw=bidirection(bidimagec);%
fin=zeros(row,col);
fin=fw;
idct_colw=fin';

for i=1:row
    for j=1:col
        %i,j
       all=[idct_roww(i,j),idct_colw(i,j)];

       finI(i,j) =sum(all)/2.0; %median(all);
       %finI(i,j)=min(idct_row(i,j)+idct_col(i,j));
   end
end
fin=finI;
%error for average
```

```
finI=fin;
error1 = ao-finI;
error1 = error1.^2;
MSE_avg = sum(error1(:))/prod(size(ao));
%ROW ERROR
finI=idct_roww;
error1 = ao-finI;
error1 = error1.^2;
MSE_row = sum(error1(:))/prod(size(ao));
%column ERRO
finI=idct_colw;
error1 = ao-finI;
error1 = error1.^2;
MSE_col = sum(error1(:))/prod(size(ao));
figure;
colormap(gray(256));
%subplot(2,2,1);imagesc(a); title('Original');
subplot(2,2,1);imagesc(fin);title('Multiresolution DCT level 6');
s = sprintf('MSE = %d,stitch',MSE_avg);
xlabel(s);
clear s;

%coloverlap=1;
```

```
%8X8
ratio_dct = ratio_dct + (orig_ratio/32);
window_rows=32;
window_cols=32;
[Res,MSE] = blockdct_comp(ao,ratio_dct,window_rows,window_cols,rowoverlap,coloverlap);

% Simple block processing

subplot(2,2,2);imagesc(Res); title('block DCT');%
clear s;
s = sprintf('MSE = %d, blockw = %d',MSE,window_rows);
xlabel(s);
```

blockdct_comp.m

```
function [Res,mserror] = blockdct_comp(amatrix,ratio,wr,wc,ro,co)

%read an image and convert it to double
a = amatrix;

[row col] = size(a);

% Threshold is decided by the percentage of compression desired


if(nargin <1)
   error('Please enter the percentage of compression');
end
rowoverlap=ro;
coloverlap=co;
window_rows=wr;
window_cols=wc;
thr = round(window_rows*window_cols*(ratio));
%get the rowwise dct coefficients
dct_rc = blkproc(a,[window_rows,window_cols],[rowoverlap,coloverlap],'dct2');
```

```
%find variance
c = im2col(dct_rc,[window_rows,window_cols],'distinct');
var_c = var(c');
%sort the variance of dct coeff. using "sort"
[y,i] = sort(var_c);
c1 = c;
c1(i(1:thr),:) = 0;
%rearrange each column of c1 as an 8 by 8 dct block
f=col2im(c1,[window_rows,window_cols],[row,col],'distinct');
%inverse row dct
idct_rc = blkproc(f,[window_rows,window_cols],[rowoverlap,coloverlap],'idct2');

finI=idct_rc;

error1 = a-finI;
error1 = error1.^2;
MSE_avg = sum(error1(:))/prod(size(a));
Res = finI;
mserror = MSE_avg;
```

bidirection.m

```
% This is a function which transforms a quadratic matrix into a
% matrix (of the same size) in which columns are transformed into
% diagonals, starting from the upper left corner
% i.e., for example matrix
%  1 4 7
%  2 5 8
%  3 6 9
%
% is transformed into matrix
% 1 2 6
% 3 5 7
% 4 8 9

function new=bidirection(q)

% to test how this transformation works, initialize a quadratic matrix, say ma
% then call this function by: col_to_diag(ma)
% output shoud give you transformed matrix in which
% coulmns are transformed into diagonals, starting for a lower left corner

a=q;
```

```
[row col]=size(a);


% initialize a new matrix
for m=1:row;
   for n=1:col;
      if mod(m,2)==0 % for even rows reverse direction
        q(m,n)=a(col-n+1) ;
      end


   end;
end;


new=a;
```