Int. J. Adv. Res. Sci. Technol. Volume 2, Issue3, 2013, pp 130-137.

IJARST

International Journal of Advanced Research in Science and Technology

journal homepage: www.ijarst.com

ISSN 2319 – 1783 (Print)

ISSN 2320 – 1126 (Online)

Image and Audio compression by Wavelet Transform using MATLAB Wavelet tool box

P. Kanaka Raju*

Department of Electronics/Physics, Institute of Science, GITAM University, Visakhapatnam, INDIA.

****Corresponding Author's Email: p.kanakaraju@rediffmail.com***

ARTICLEINFO

Article history:

Received 28 Sept. 2013

Accepted 25 Oct. 2013

Available online 11 Nov. 2013

Keywords:

Audio Compression,
Wavelet transform,
MATLAB and Wavelet tool box

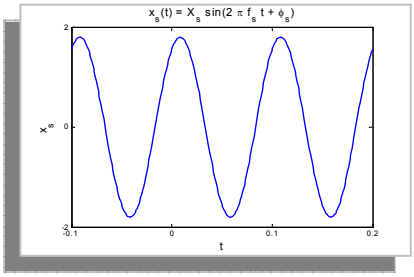
ABSTRACT

Audio and image compression has become one of the basic technologies of the multimedia. In many applications, such as the design of multimedia workstations and high quality audio transmission and storage, the goal is to achieve transparent coding of audio and speech signals at the lowest possible data rates. Bandwidth cost money, therefore the transmission and storage of information becomes costly. However, if we can use less data, both transmission and storage become cheaper. In this work, with the help of wavelet transform we can greatly reduce the storage need and transmission bandwidth when compared to WinZip. Wavelet analysis breaks a signal into shifted and scaled versions of the original wavelet, thus the advantage of wavelets is the ability to perform local analysis that is, to analyse a localized area of a larger signal.

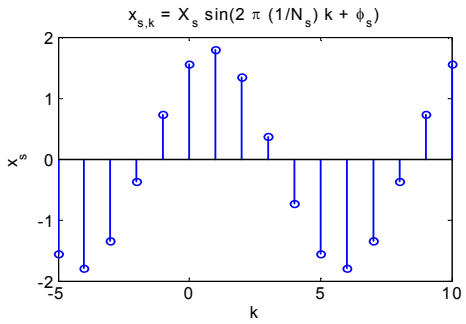
Introduction:

Signals play an important role in our daily life; examples are speech, music, picture and video signals. A signal is a function of independent variables such as time, distance, position, temperature and pressure. For example, speech and music signals represent air pressure as a function of time at a point in space. A black and white picture is a representation of light intensity as a function of two spatial coordinates. The video signal in television consists of a sequence of images called frames and is a function of three variables those are two spatial and time. Any physical quantity that carries information varies with other independent or a dependent variable is defined as a 'signal'.

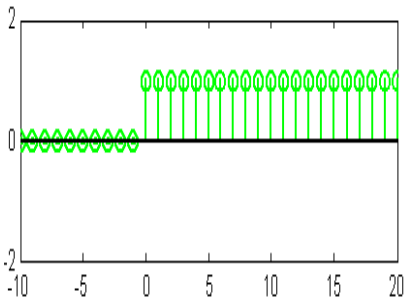
The main types of signals with respect to time as independent variable are continuous time (analog) signals and discrete time (discrete) signals. The analog signal is defined for every instant of independent variable and so magnitude of independent variable is continuous in the specified range. Here both the independent variable and magnitude are continuous. The discrete signal is a function of discrete independent variables. The independent variable is divided into uniform interval and is represented by an integer. The discrete signal is defined for every integer value of independent variable. Here both the values of signal and independent variable are discrete. The digital signal is same as discrete signal except that the magnitude of signal is quantized.



Analog signal



Discrete-Time signal



Digital signal

Fourier analysis:

As shown in Fig: 1, Fourier analysis breaks down a signal into constituent sinusoids of different frequencies. It is as a mathematical technique for transforming the view of the signal from time-based to frequency-based. Fourier analysis is extremely useful because the signal's frequency content is of great importance. So why do we need other techniques, like wavelet analysis? Fourier analysis has a serious drawback. In transforming to the frequency domain, time information is lost. From the Fourier transform of a signal, it is impossible to tell when a particular event took place. If the signal properties do not change much over time is called a stationary signal—this drawback isn't very important. However, most of the signals contain numerous non-stationary or transitory characteristics: drift, trends, abrupt changes, beginnings and ends of events. These characteristics are often the most important part of the signal, and Fourier analysis is not suited to detecting them.

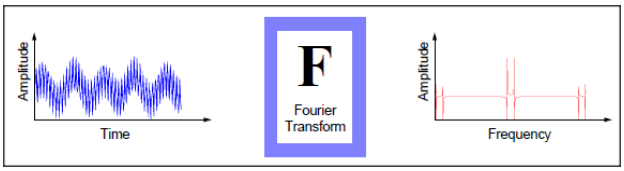


Fig: 1. Fourier analysis breaks down a signal into constituent sinusoids of different frequencies

To correct this deficiency, Dennis Gabor (1946) adapted the Fourier transform to analyze only a small section of the signal at a time-this technique called *windowing* the signal. Gabor's adaptation called the *Short-Time Fourier Transform* (STFT), which maps a signal into a two-dimensional function of time and frequency. The STFT represents a sort of compromise between the time- and frequency-based views of a signal (fig. 2). It provides some information about both, when and at what frequencies a signal event occurs. However, we can only obtain this information with limited precision, and that precision is determined by the size of the window. While the STFT compromise between time and frequency information can be useful, the drawback is that once we choose a particular size for the time window, that window is the same for all frequencies. Many signals require a more flexible approach—one where we can vary the window size to determine more accurately either time or frequency.

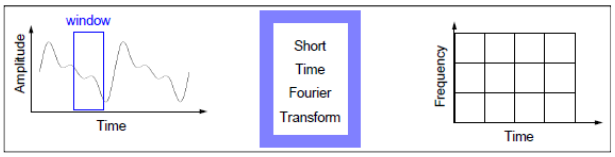


Fig: 2. Short-Time Fourier Transform maps a signal into a two-dimensional function of time and frequency

Wavelet Analysis:

A wavelet is a waveform of effectively limited duration that has an average value of zero [1]. Compare wavelets with sine waves (fig. 3). Sinusoids do not have limited duration — they extend from minus to plus infinity. Sinusoids are smooth and predictable; Wavelets tend to be irregular and asymmetric.

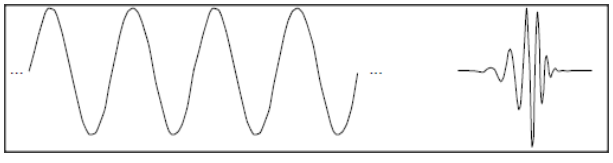


Fig: 3. Sine Wave Wavelet (db10)

Fourier analysis consists of breaking up a signal into sine waves of various frequencies. Similarly, wavelet analysis breaks a signal into shifted and scaled versions of the original (or mother) wavelet [2]. The signals with sharp changes might be better analyzed with an irregular wavelet than with a smooth sinusoid.

Wavelet analysis represents the next logical step: a windowing technique with variable-sized regions. It allows the use of long time intervals where we want more precise low-frequency information and shorter regions where we want high-frequency information (fig .4)

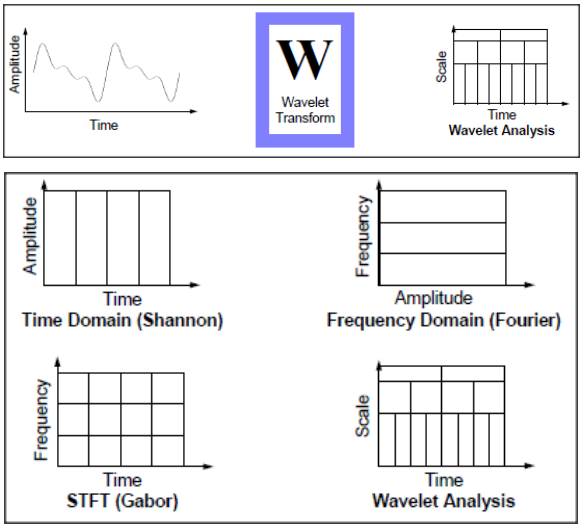


Fig: 4. Wavelet analysis representation

Wavelet analysis does not use a time-frequency region, but rather a time-scale region. Mathematically, the process of Fourier analysis is represented by the Fourier transform:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

Which is the sum over all time of the signal f(t) multiplied by a complex exponential. The results of the transform are the Fourier coefficients f(ω), when it was multiplied by a sinusoid of frequency ω, yield the constituent sinusoidal components of the original signal. Graphically, the process looks as shown figure 5:

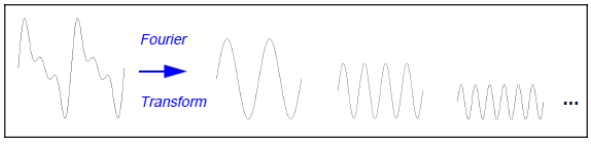


Fig: 5. Signal - Constituent sinusoids of different frequencies

Similarly, the continuous wavelet transform (CWT) is defined as the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet function Ψ:

$$C(\text{scale, position}) = \int_{-\infty}^{\infty} f(t)\,\psi(\text{scale, position, t})dt$$

The result of the CWT has many wavelet coefficients C, which are a function of scale and position. Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelets of the original signal as fig: 6.

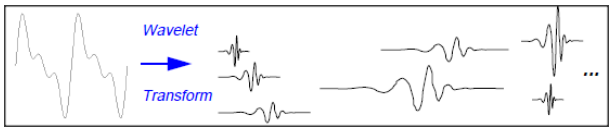


Fig: 6. *Signal - Constituent wavelets of different scales and positions*

Advantage of Wavelet Analysis:

The advantage of wavelets is the ability to perform local analysis — that is, to analyze a localized area of a larger signal. Let us consider a sinusoidal signal with a small discontinuity. Such a signal easily could be generated in the real world, perhaps by a power fluctuation or a noisy switch as shown fig: 7.

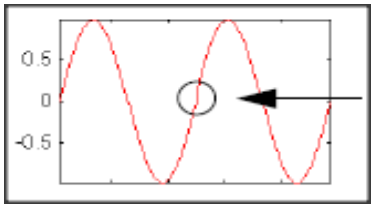


Fig: 7. Sinusoid with a small discontinuity

A plot of the Fourier coefficients (provided by FFT command) of this signal shows a flat spectrum with two peaks representing a single frequency. However, a plot of wavelet coefficients clearly shows the exact location in time of the discontinuity as shown in figure 8.

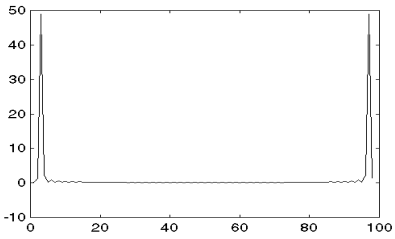


Fig: 8(a). Fourier Coefficients

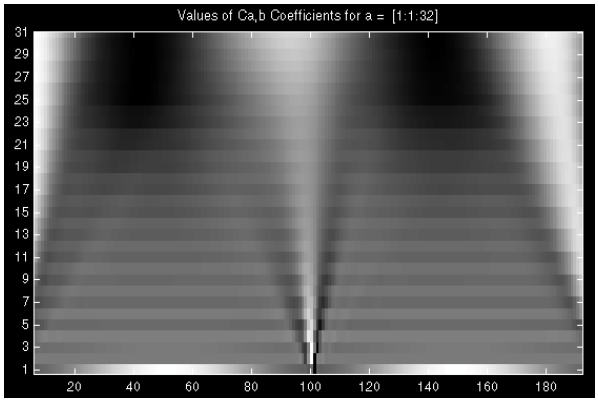
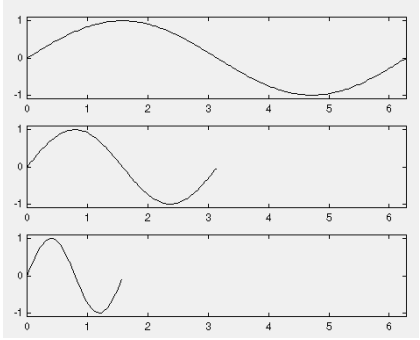


Fig: 8(b). Wavelet Coefficients

Wavelet analysis is capable of revealing aspects of data that other signal analysis techniques miss aspects like trends, breakdown points, discontinuities in higher derivatives, and self-similarity. Furthermore, wavelet analysis can often compress or de-noise a signal without appreciable degradation.

Scaling:

Wavelet analysis produces a time-scale view of a signal, Scaling a wavelet simply means stretching (or compressing) it. For "stretching," we introduce the scale factor, often denoted by the letter a. In the case of sinusoids, the effect of the scale factor is as shown below Fig: 9.



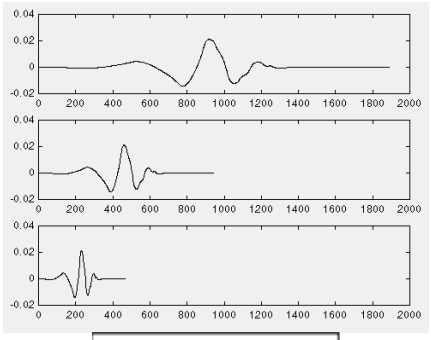
$f(t) = \sin(t);\ a = 1$

$f(t) = \sin(2t);\ a = 1/2$

$f(t) = \sin(4t);\ a = 1/4$

Fig: 9. Stretching of sinusoidal signal

The scale factor works exactly the same with wavelets. The smaller the scale factor, the more "compressed" the wavelet.



$f(t) = y(t);\quad a = 1$

$f(t) = y(2t);\quad a = 1/2$

$f(t) = y(4t);\quad a = 1/4$

Fig: 10. Scaling of Wavelet

From the Fig: 10, for a sinusoid sin (ωt), the scale factor a is related (inversely) to the radian frequency ω. Similarly, with wavelet analysis, the scale is related to the frequency of the signal.

Shifting:

Shifting a wavelet simply means delaying its onset. Mathematically, delaying a function f (t) by k is represented by f (t – k) as shown in Fig: 11.

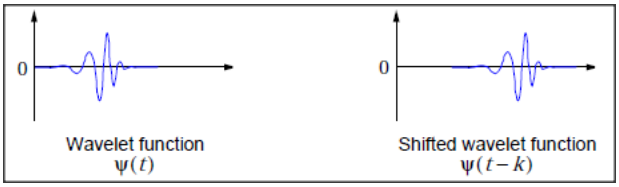


Fig: 11. Shifting of a wavelet

Scale and Frequency:

The higher scales correspond to the most "stretched" wavelets. The more stretched the wavelet, the longer the portion of the signal with which it is being compared, and thus the coarser the signal features being measured by the wavelet coefficients. Thus, there is a correspondence between wavelet scales and frequency as revealed by wavelet analysis Fig: 12.

•Low scale a ⇒ compressed wavelet ⇒ rapidly changing details ⇒ High frequency ω.
•High scale a ⇒ Stretched wavelet ⇒ slowly changing, coarse features ⇒ Low frequency ω.

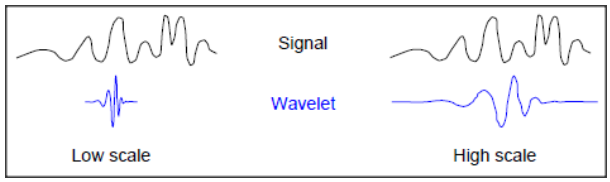


Fig: 12. Representation of low and high scales

Discrete Wavelet Transform:

If we choose scales and positions based on powers of two- called dyadic scales and positions, then the analysis will be much more efficient and accurate. We obtain such an analysis from the discrete wavelet transform (DWT). It can be implement by using filters, was developed in 1988 by Mallat. The Mallat algorithm is a classical scheme known in the signal processing community as a two-channel subband coder (the book Wavelets and Filter Banks, by Strang and Nguyen). This is a practical filtering algorithm yields a fast wavelet transform [3].

One-Stage Filtering:

In wavelet analysis, we often speak of approximations and details. The approximations are the high-scale, low-frequency components of the signal. The details are the low-scale, high-frequency components. The original signal S passes through two complementary filters and emerges as two signals. The filtering process at its basic level depicted in Fig: 13.

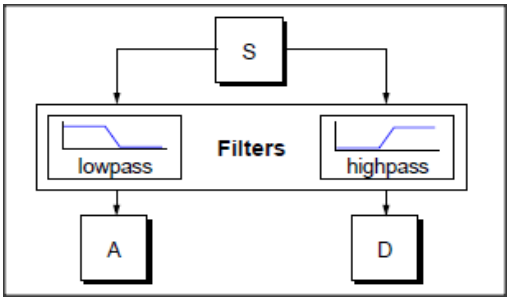


Fig: 13. Filtering process

Multiple-Level Decomposition:

The decomposition process (fig. 14) can be iterated, with successive approximations being decomposed in turn, so that one signal is broken down into many lower resolution components. This is called the wavelet decomposition tree. Since the analysis process is iterative, in theory it can be continued indefinitely. In reality, the decomposition can proceed only until the individual details consist of a single sample or pixel. In practice, we will select a suitable number of levels based on the nature of the signal, or on a suitable criterion such as entropy.

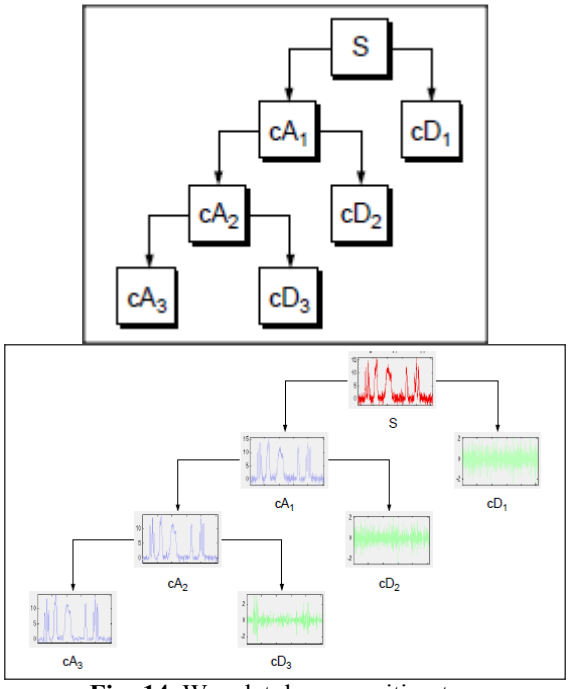


Fig: 14. Wavelet decomposition tree

Wavelet Reconstruction:

The decomposed components can be assembled back into the original signal without loss of information, is called reconstruction or synthesis [4]. The mathematical manipulation, that effects synthesis is called the inverse discrete wavelet transform (IDWT). To synthesize a signal in the Wavelet Toolbox, we reconstruct it from the wavelet coefficients as shown in Fig: 15.

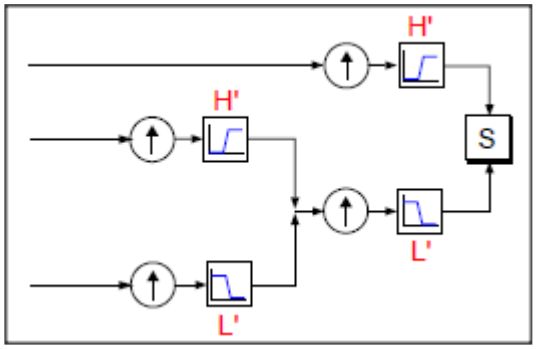


Fig: 15. Reconstruct from the wavelet coefficients

Wavelet analysis involves filtering and down sampling, the wavelet reconstruction process consists of up sampling and filtering (fig. 16). Up sampling is the process of lengthening a signal component by inserting zeros between samples:

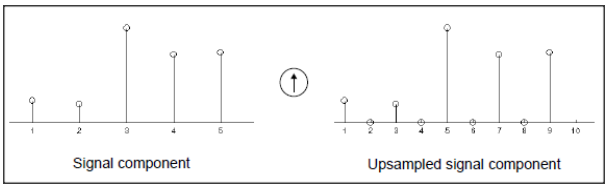


Fig: 16. Multistep Decomposition and Reconstruction

A multistep analysis- can be represented synthesis process as:

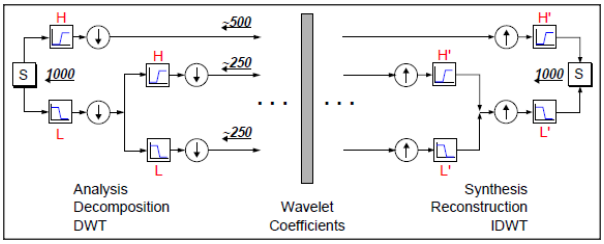


Fig: 17. Multistep analysis

This process involves two aspects: breaking up a signal to obtain the wavelet coefficients, and reassembling the signal from the coefficients. There is no point breaking up a signal merely to have the satisfaction of immediately reconstructing it. We may modify the wavelet coefficients before performing the reconstruction step.

Picture / Audio Compression:

We employed a lossy compression technique [5] for Picture / Audio compressions where we used the wavelet transform of the original signal, then calculated a threshold based on the compression ratio required.

MATLAB:

A data-manipulation software package that allows data to be analyzed and visualized, using existing functions and user-designed programs. MATLAB refers to both the numerical computing environment and to its core programming language. MATLAB allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages. Although it specializes in numerical computing, an optional toolbox [6] interfaces with the Maple symbolic engine making it a full computer algebra system.

MATLAB Source Code:

Compress.m

```
Function Compress (File Name)
[ImgData,Map] = imread(FileName);
[Coef,Pos] = wavedec2(ImgData,2,'Bior3.7');
%[thr,s_or_h,keep_app] = ddencmp('cmp','wv',ImgData);
[thr,s_or_h,keep_app] =
compthreshold(c,s,percentageCompression,keepapp);
[xcomp,CXC,LXC,PERF0,PERFL2]=
wdencmp('gbl',Coef,Pos,'Bior3.7',2,thr,s_or_h,keep_app);
MaxValue = max(CXC);
NormalizationFactor = MaxValue / 255;
NormalizedCoef = Coef / NormalizationFactor;
IntValues = fix(NormalizedCoef);
EightBitValues = uint8(IntValues);
save compressed.mat EightBitValues Pos
NormalizationFactor Map;
```

Compthreshold.m

```
Function [threshold, sorh, keepap] = compthreshold
(c, s, percentage, keepapp)
sorh = 'h';
keepap = keepapp;
if keepapp == 1
   x = abs (c(prod(s(1,:))+1:end));
   x = sort(x);
   dropindex = length(x) * percentage/100;
   dropindex = round(dropindex);
   threshold = x(dropindex);
  else %drop coefficients even from the approximation
   x = abs(c);
   x = sort(x);
   dropindex = length(x) * percentage/100;
   dropindex = round(dropindex);
   threshold = x(dropindex);
 end
if (threshold == 0)
      threshold = 0.05*max(abs(x));
   end
```

Bit-map image:

Cons.bmp

Imagecompression.m

function y = ImageCompression

(levelOfdecomposition,percentageCompression,keepapp)

%[X,map] = imread ('badcloning','jpg');

load wbarb;

IMWRITE(X,map,'OriginalImage.bmp');

figure;

colormap(map);

image(X);

save Original.mat X;

title('ORIGINAL IMAGE');

[c,s] = wavedec2(X,levelOfdecomposition,'bior3.7');

[threshold,sorh,keepapp] =

compthreshold(c,s,percentageCompression,keepapp);

[Xcomp,cxc,lxc,perf0,perfl2] = wdencmp

('gbl',c,s,'bior3.7',levelOfdecomposition,threshold,sorh,ke

epapp);

save Compressed.mat cxc;

IMWRITE(Xcomp,map,'compressedImage.bmp');

figure;

%colormap(map);

colormap(map);

image(Xcomp);

title('COMPRESSED IMAGE');

runproject.m

run project

ImageCompression (3,70,1);

Compress('cons.bmp',70,1);

Temp.m

function y = ImageCompression

(levelOfdecomposition,percentageCompression,keepapp)

load wbarb;

figure;

title('ORIGINAL IMAGE');

colormap(map);

image(X);

[c,s] = wavedec2(X,levelOfdecomposition,'bior3.7');

[threshold,sorh,keepapp] =

compthreshold(c,s,percentageCompression,keepapp);

[Xcomp,cxc,lxc,perf0,perfl2] = wdencmp

('gbl',c,s,'bior3.7',levelOfdecomposition,threshold,sorh,ke

epapp);

figure;

title('COMPRESSED IMAGE');

colormap(map);

image(Xcomp);

test.m

x = 1:9;

h = 1:9;

y = dwt (x,'db1');

TwoDimDWT.m

function TwoDimDWT

ImgData = imread('cons.bmp');

[a1,h1,v1,d1] = dwt2(ImgData,'Bior3.7');

load map;

figure;

image(ImgData);

title('Original Image');

colormap(map);

pause;

subplot(221);

image(a1);

title('Approximations');

subplot(222);

```
image(h1);
title('Horizontal Details');
subplot(223);
image(v1);
title('Vertical Details');
subplot(224);
image(d1);
title('Diagonal Details');
ReConsImg = idwt2(a1,h1,v1,d1,'Bior3.7');
pause;
subplot(111);
image(ReConsImg);
colormap(map);
title('Re-Constructed Image');
```

Uncompress.m

```
function Uncompress
load compressed.mat;
ActualCoef = double(EightBitValues);
ActualCoef = ActualCoef .* NormalizationFactor;
ImgData = waverec2(ActualCoef,Pos,'Bior3.7');
imwrite(ImgData,Map,'aRecon.bmp');
```

Results:

The results of the compression achieved were as follows.

Image:

Original Image Size: 100kb (Bitmap)
Compressed Image Size (using WinZip): 76kb
Compressed Image Size (using Wavelets and then WinZip): 37kb

Sound:

Original File Size: 200kb (wav)
Compressed File Size (using WinZip): 189kb
Compressed File Size (using Wavelets and then WinZip): 22kb

Discussion and Conclusion:

The demand for compression technology increases every year in parallel with the increase in aggregate bandwidth for the transmission of audio and video signals. As a result, the Wavelet-based approach plays an important role in the scheme of things as Perceptual coding of audio signals found its way to a growing number of consumer applications. The foremost criterion for audio compression technology is to achieve a certain signal quality at a given bit-rate as this directly translates to cost savings by getting a higher compression ratio at the same quality of service. Wavelet-based compression is claimed to be more efficient at low bit rates but are actually less successful than discrete cosine transform (DCT) [7] -based systems in achieving good efficiency at near-transparent compression ratios. It is notable that Wavelet compression does require more computational power than DCT-based compression. In addition, using wavelets, the compression ratio can be easily varied, while most other compression techniques have fixed compression ratios.

References:

1. Robi Polikar, .The Wavelet Tutorial. .
2. H. L. Resnikoff and R.O. Wells, Jr. .Wavelet Analysis The Scalable Structure of Information. Springer .Verlag New York, Inc. 1998
3. Mulcahy, Colm. .Image compression using the Haar wavelet transform.. Spelman Science and Math Journal. Found at: http://www.spelman.edu/~colm/wav.html
4. S. G. Chang, B Yu and M Vetterli. .Adaptive Wavelet Thresholding for image Denoising and Compression.. IEEE Transactions on Image Processing, Vol. 9, No. 9, September 2000
5. S. G. Chang, B. Yu and M Vetterli. .Image Denoising via Lossy Compression and Wavelet Thresholding.
6. Hubbard, Barbara Burke. The World According to Wavelets. A.K Peters Ltd, 1995.
7. Saha, Subhasis. .Image Compression. from DCT to Wavelets : A Review.