



**ELECTRICAL AND  
COMPUTER ENGINEERING  
PURDUE UNIVERSITY**

Project Report:

**Audio Compression using Wavelet  
Techniques**

**Project Report.**

ECE 648 – Spring 2005

Wavelet, Time-Frequency, and Multirate Signal Processing

Professor Ilya Pollak

Matias Zanartu

ID:999 09 2426

[mzanartu@ecn.purdue.edu](mailto:mzanartu@ecn.purdue.edu)

## TABLE OF CONTENTS

---

1.-	OBJECTIVES .....	3
2.-	INTRODUCTION .....	4
2.1	Useful auditory properties .....	4
2.1.1	Non linear frequency response of the hear.....	4
2.1.2	Masking property of the auditory system .....	4
2.2	Audio compression .....	6
2.2.1	Lossless compression.....	6
2.2.2	Lossy compression .....	7
2.2.3	MPEG Audio coding standards .....	7
2.3	Speech compression .....	9
2.4	Evaluating compressed audio .....	10
3.-	DISCRETE WAVELET TRANSFORM APPROACH .....	11
3.1	Original Abstract in [1].....	11
3.2	General picture .....	11
3.3	Wavelet representation for audio signals .....	12
3.4	Psychoacoustic model .....	13
3.4.1	Simplified masking model .....	13
3.4.2	Masking constraint in the Wavelet Domain.....	14
3.5	Reducing the number of non-zero coefficients: Optimization criterion .....	15
3.6	Results Dynamic Dictionary approach.....	17
3.7	Implementation Issues .....	18
3.8	Results.....	19
4.-	WAVELET PACKET APPROACH .....	20
4.1	Original abstract in [2].....	20
4.2	General picture .....	21
4.3	Psychoacoustic model .....	22
4.3.1	Subband masking model.....	22
4.3.2	Masking constrain in the wavelet structure.....	22
4.4	Wavelet packet representation.....	23
4.5	Efficient Bit allocation.....	25
4.6	Implementation Issues .....	25
4.7	Results.....	26
5.-	SOME COMMENTS ON THE PAPERS.....	27
6.-	MATLAB SIMULATIONS .....	28
6.1	Main features of the implementation .....	28
6.2	Considerations.....	29
6.3	Results.....	31
7.-	DISCUSSION: STATE-OF-THE-ART .....	33
8.-	CONCLUSIONS .....	34
9.-	ACKNOWLEDGMENTS .....	34
10.-	BIBLIOGRAPHY AND REFERENCES .....	35
11.-	MATLAB CODE.....	36

## 1.- OBJECTIVES

The main objective of this project is to study some known audio compression techniques that use wavelets. In order to this, I have considered the following activities:

- To review and summarize general literature of audio compression techniques.
- To summarize the contributions to the field of audio compression and the relationships among these two papers:
  - (a) D. Sinha and A. Tewfik. “*Low Bit Rate Transparent Audio Compression using Adapted Wavelets*”, IEEE Trans. ASSP, Vol. 41, No. 12, December 1993.
  - (b) P. Srinivasan and L. H. Jamieson. “*High Quality Audio Compression Using an Adaptive Wavelet Packet Decomposition and Psychoacoustic Modeling*”, IEEE Transactions on Signal Processing, Vol 46, No. 4, April 1998.
- To include a brief overview of current applications of wavelets techniques in the field of audio compression.
- To simulate using MATLAB the main features of the one of the two mentioned papers.
- To facilitate the evaluation of the results of the simulation by including a CD with several audio demonstrations.

## 2.- INTRODUCTION

The purpose of this chapter is to introduce several concepts that are mentioned in the selected papers and that are used in the MATLAB simulations. This introduction covers some aspects of psychoacoustics and presents a brief summary of the current audio compression techniques.

### 2.1 Useful auditory properties

#### 2.1.1 Non linear frequency response of the hear

Humans are able to hear frequencies in the range approximately from 20 Hz to 20 kHz. However, this does not mean that all frequencies are heard in the same way. One could make the assumption that a human would hear frequencies that make up speech better than others, and that is in fact a good guess. Furthermore, one could also hypothesize that hearing a tone becomes more difficult close to the extremes frequencies (i.e. close to 20 Hz and 20kHz).

After many cochlear studies, scientists have found that the frequency range from 20 Hz to 20 kHz can be broken up into critical bandwidths, which are non-uniform, non-linear, and dependent on the level of the incoming sound. Signals within one critical bandwidth are hard to separate for a human observer. A detailed description of this behavior is described in the Bark scale and Fletcher curves.

#### 2.1.2 Masking property of the auditory system

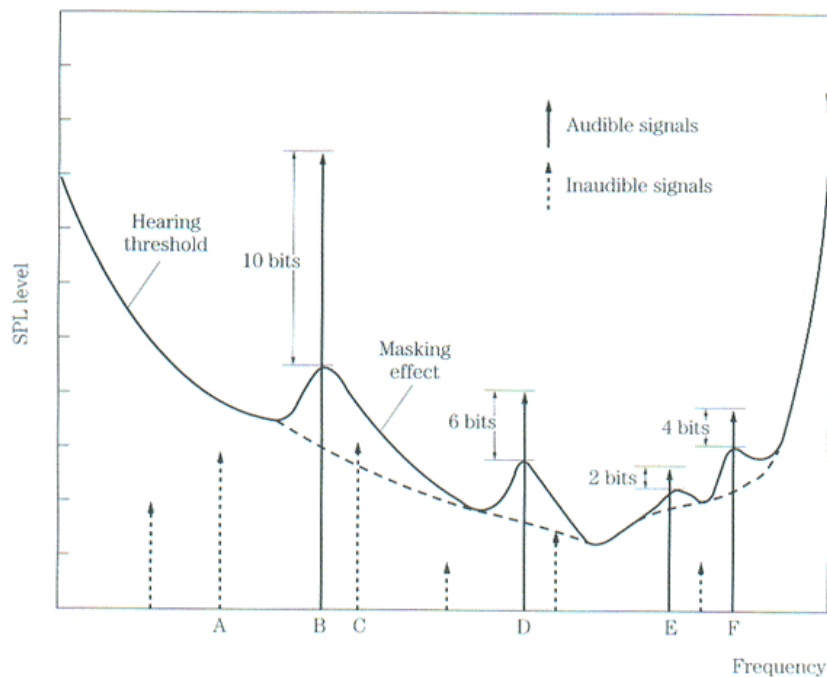
Auditory masking is a perceptual property of the human auditory system that occurs whenever the presence of a strong audio signal makes a temporal or spectral neighborhood of weaker audio signal imperceptible. This means that the masking effect can be observed in time and frequency domain. Normally they are studied separately and known as simultaneous masking and temporal masking.

If two sounds occur simultaneously and one is masked by the other, this is referred to as simultaneous masking. A sound close in frequency to a louder sound is more easily masked than if it is far apart in frequency. For this reason, simultaneous masking is also

sometimes called frequency masking. It is important to differentiate between tone and noise maskers, because tonality of a sound also determines its ability to mask other sounds. A sinusoidal masker, for example, requires a higher intensity to mask a noise-like masker than a loud noise-like masker does to mask a sinusoid. Similarly, a weak sound emitted soon after the end of a louder sound is masked by the louder sound. In fact, even a weak sound just before a louder sound can be masked by the louder sound. These two effects are called forward and backward temporal masking, respectively. Temporal masking effectiveness attenuates exponentially from the onset and offset of the masker, with the onset attenuation lasting approximately 10 ms and the offset attenuation lasting approximately 50 ms.

It is of special interest for perceptual audio coding to have a precise description of all masking phenomena to compute a masking threshold that can be used to compress a digital signal. Using this, it is possible to reduce the SNR and therefore the number of bits. A complete masking threshold should be calculated using the principles of simultaneous masking and temporal masking and the frequency response of the ear. In the perceptual audio coding schemes, these masking models are often called psychoacoustic models.

**Figure 1.- An example that shows how the auditory properties can be used to compress an digital audio signal. Source: [4]**



## 2.2 Audio compression

The idea of audio compression is to encode audio data to take up less storage space and less bandwidth for transmission. To meet this goal different methods for compression have been designed. Just like every other digital data compression, it is possible to classify them into two categories: lossless compression and lossy compression.

### 2.2.1 Lossless compression

Lossless compression in audio is usually performed by waveform coding techniques. These coders attempt to copy the actual shape of the analog signal, quantizing each sample using different types of quantization. These techniques attempt to approximate the waveform, and, if a large enough bit rate is available they get arbitrary close to it. A popular waveform coding technique, that is considered uncompressed audio format, is the pulse code modulation (PCM), which is used by the Compact Disc Digital Audio (or simply CD). The quality of CD audio signals is referred to as a standard for hi-fidelity. CD audio signals are sampled at 44.1 kHz and quantized using 16 bits/sample Pulse Code Modulation (PCM) resulting in a very high bit rate of 705 kbps.

As mentioned before, human perception of sound is affected by SNR, because adding noise to a signal is not as noticeable if the signal energy is large enough. When digitalize an audio signal, ideally SNR could to be constant for al quantization levels, which requires a step size proportional to the signal value. This kind of quantization can be done using a logarithmic compander (compressor-expander). Using this technique it is possible to reduce the dynamic range of the signal, thus increasing the coding efficiency, by using fewer bits. The two most common standards are the  $\mu$ -law and the A-law, widely used in telephony.

Other lossless techniques have been used to compress audio signals, mainly by finding redundancy and removing it or by optimizing the quantization process. Among those techniques it is possible to find Adaptative PCM and Differential quantization. Other lossless techniques such as Huffman coding and LZW have been directly applied to audio compression without obtaining significant compression ratio.

### 2.2.2 Lossy compression

Opposed to lossless compression, lossy compression reduces perceptual redundancy; i.e. sounds which are considered perceptually irrelevant are coded with decreased accuracy or not coded at all. In order to do this, it is better to have scalar frequency domains coders, because the perceptual effects of masking can be more easily implemented in frequency domain by using subband coding.

Using the properties of the auditory system we can eliminate frequencies that cannot be perceived by the human ear, i.e. frequencies that are too low or too high are eliminated, as well as soft sounds that are drowned out by loud sounds. In order to determine what information in an audio signal is perceptual irrelevant, most lossy compression algorithms use transforms such as the Modified Discrete Cosine Transform (MDCT) to convert time domain sampled waveforms into a frequency domain. Once transformed into the frequency domain, frequencies component can be digitally allocated according to how audible they are (i.e. the number of bits can be determined by the SNR). Audibility of spectral components is determined by first calculating a masking threshold, below which it is estimated that sounds will be beyond the limits of human perception (see 2.1 on this report).

Briefly, the modified discrete cosine transform (MDCT) is a Fourier-related transform with the additional property of being lapped. It is designed to be performed on consecutive blocks of a larger data set, where subsequent blocks are overlapped so that the last half of one block coincides with the first half of the next block. This overlapping, in addition to the energy-compaction qualities of the DCT, makes the MDCT especially attractive for signal compression applications, since it helps to avoid artifacts stemming from the block boundaries.

### 2.2.3 MPEG Audio coding standards

Moving Pictures Experts Group (MPEG) is an ISO/IEC group charged with the development of video and audio encoding standards. MPEG audio standards include an elaborate description of perceptual coding, psychoacoustic modeling and implementation issues. It is interesting for our report to mention some brief comments

on these audio coders, because some of the features of the wavelet-based audio coders are based in those models.

- (a) MP1 (MPEG audio layer-1): Simplest coder/decoder. It identifies local tonal components based on local peaks of the audio spectrum.
- (b) MP2 (MPEG audio layer-2): It has an intermediate complexity. It uses data from the previous two windows to predict, via linear interpolation, the component of the current window. This is based on the fact that tonal components, being more predictable, have higher tonality indices.
- (c) MP3 (MPEG audio layer-3). Higher level of complexity. Not only includes masking in time domain but also a more elaborated psychoacoustic model, MDCT decomposition, dynamic allocation and Huffman coding.

All three layers of MPEG-1 use a polyphase filterbank for signal decomposition into 32 equal width subbands. This is a computationally simple solution and provides reasonable time-frequency resolution. However it is known that this approach has three notable deficiencies:

- Equal subbands do not reflect the critical bands of noise masking, and then the quantization error cannot be tuned properly.
- Those filter banks and their inverses do not yield perfect reconstruction, introducing error even in the absence of quantization error.
- Adjacent filter banks overlap, then a single tone can affect two filter banks.

These problems have been fixed by a new format which is considered the successor of the MP3 format: AAC (Advanced Audio Coding) defined in MPEG-4 Part 3 (with an extension .m4a or namely MP4 audio).

- (d) M4A: AAC (MPEG-4 Audio): Similar to MP3 but it increases the number of subbands up to 48 and fixes some issues in the previous perceptual model. It has higher coding efficiency for stationary and transient signals, providing a better and more stable quality than MP3 at equivalent or slightly lower bitrates.



### 2.3 Speech compression

Speech signals has unique properties that differ from a general audio/music signals. First, speech is a signal that is more structured and band-limited around 4kHz. These two facts can be exploited through different models and approaches and at the end, make it easier to compress. Many speech compression techniques have been efficiently applied. Today, applications of speech compression (and coding) involve real time processing in mobile satellite communications, cellular telephony, internet telephony, audio for videophones or video teleconferencing systems, among others. Other applications include also storage and synthesis systems used, for example, in voice mail systems, voice memo wristwatches, voice logging recorders and interactive PC software.

Basically speech coders can be classified into two categories: waveform coders and analysis by synthesis vocoders. The first was explained before and are not very used for speech compression, because they do not provide considerable low bit rates. They are mostly focused to broadband audio signals. On the other hand, vocoders use an entirely different approach to speech coding, known as parametric coding, or analysis by synthesis coding where no attempt is made at reproducing the exact speech waveform at the receiver, but to create perceptually equivalent to the signal. These systems provide much lower data rates by using a functional model of the human speaking mechanism at the receiver. Among those, perhaps one of the most popular techniques is called Linear Predictive Coding (LPC) vocoder. Some higher quality vocoders include RELP (Residual Excited Linear Prediction) and CELP (Code Excited Linear Prediction). There are also lower quality vocoders that give very low bit rate such as Mixed Excitation vocoder, Harmonic coding vocoder and Waveform interpolation coders.

## 2.4 Evaluating compressed audio

When evaluating the quality of compressed audio it is also convenient to differentiate between speech signals and general audio/music signals. Even though speech signals have more detailed methods to evaluate the quality of a compressed signal (like intelligibility tests), both audio/music and speech share one of the most common methods: acceptability tests. These tests are the most general way to evaluate the quality of an audio/speech signal, and they are mainly determined by asking users their preferences for different utterances. Among those tests, Mean Opinion Score (MOS) test is the most used one. It is a subjective measurement that is derived entirely by people listening to the signals and scoring the results from 1 to 5, with a 5 meaning that speech quality is perfect or “transparent”. The test procedure requires carefully prepared and controlled test conditions. The term “transparent quality” means that most of the test samples are indistinguishable from the original for most of the listeners. The term was defined by the European Broadcasting Union (EBU) in 1991 and statistically implemented in formal listening tests since then.

Finally, it is necessary to emphasize that the fact that measures of quality of audio signal does not have an objective measure that we can extract directly from the signal (such mean square error), make it more difficult to evaluate it. This is because subjective evaluations require a large number of test samples and special conditions during the evaluation.

### 3.- DISCRETE WAVELET TRANSFORM APPROACH

This chapter summarizes the approach to audio compression using discrete wavelet transform as described in [1] “Low Bit Rate Transparent Audio Compression using Adapted Wavelets”, by D. Sinha and A. Tewfik.

In this report, this particular paper will be discussed with more detail than [2], because most of MATLAB simulations are based in the scheme that this document describes.

#### 3.1 Original Abstract in [1]

*“This paper describes a novel wavelet based audio synthesis and coding method. The method uses optimal adaptive wavelet selection and wavelet coefficients quantization procedures together with a dynamic dictionary approach. The adaptive wavelet transform selection and transform coefficient bit allocation procedures are designed to take advantage of the masking effect in human hearing. They minimize the number of bits required to represent each frame of audio material at a fixed distortion level. The dynamic dictionary greatly reduces statistical redundancies in the audio source. Experiments indicate that the proposed adaptive wavelet selection procedure by itself can achieve almost transparent coding of monophonic compact disk (CD) quality signals (sampled at 44.1 kHz) at bit rates of 64-70 kilobits per second (kb/s). The combined adaptive wavelet selection and dynamic dictionary coding procedures achieve almost transparent coding of monophonic CD quality signals at bit rates of 48-66 kb/s.”*

- *Deepen Sinha and Ahmed H. Tewfik.*

#### 3.2 General picture

The main goal of the algorithm presented in paper is to compress high quality audio maintaining transparent quality at low bit rates. In order to do this, the authors explored the usage of wavelets instead of the traditional Modified Discrete Cosine Transform (MDCT). Several steps are considered to achieve this goal:

- Design a wavelet representation for audio signals.
- Design a psychoacoustic model to perform perceptual coding and adapt it to the wavelet representation.
- Reduce the number of the non-zero coefficients of the wavelet representation and perform quantization over those coefficients.
- Perform extra compression to reduce redundancy over that representation
- Transmit or store the stream of data. Decode and reconstruct.
- Evaluate the quality of the compressed signal.
- Consider implementation issues.

In this chapter the summary of the main considerations and contributions for each of these points is presented.

### **3.3 Wavelet representation for audio signals**

The authors have chosen to implement an adaptive DWT signal representation because the DWT is a highly flexible family of signal representations that may be matched to a given signal and it is well applicable to the task of audio data compression. In this case the audio signal will be divided into overlapping frames of length 2048 samples (46 ms at 44.1 kHz). The two ends of each frame are weighted by the square root of a Hanning window of size 128 to avoid border distortions. Further comments on the frame size will be presented under *Implementation issues* in this chapter.

When designing the wavelet decomposition the authors have considered some restrictions to have compact support wavelets, to create orthogonal translates and dilates of the wavelet (the same number of coefficients than the scaling functions), and to ensure regularity (fast decay of coefficients controlled by choosing wavelets with large number of vanishing moments). In that sense the DWT will act as an orthonormal linear transform.

The wavelet transform coefficients are computed recursively using an efficient pyramid algorithm, not described in this paper. In particular, the filters given by the decomposition are arranged in a tree structure, where the leaf nodes in this tree correspond to subbands of the wavelet decomposition. This allows several choices for a basis. This

filter bank interpretation of the DWT is useful to take advantage of the large number of vanishing moments.

Wavelets with large number of vanishing moments are useful for this audio compression method, because if a wavelet with a large number of vanishing moments is used, a precise specification of the pass bands of each subband in the wavelet decomposition is possible. Thus, we can approximate the critical band division given by the auditory system with this structure and quantization noise power could be integrated over these bands.

### **3.4 Psychoacoustic model**

#### **3.4.1 Simplified masking model**

As mentioned before, auditory masking depends on time and frequency of both the masking signal and the masked signal. It is assumed in this paper that masking is additive, so they estimate the total masked power at any frequency by adding the masked power due to the components of the signal at each frequency. Thus, minimum masked power within each band can be calculated. From this, they get the final estimate of masked noise power. Then the idea is that a listener will tolerate an additive noise power in the reproduced audio signal as long as the power spectrum is less than the masked noise power at each frequency.

The authors noted that even though this masking model has several drawbacks, it yields reasonable coding gains. The main problems that this psychoacoustic model has are:

- The shape of the masking property used is valid for masking by tonal signals, which is not the same for masking by noise.
- The model is based on psychoacoustic studies for the masking of a single tone like signal (quantization error could happen if it contains several components).
- Masking is assumed to be additive (a power law rule of addition should be used instead).

### 3.4.2 Masking constraint in the Wavelet Domain

This masking model is incorporated within the framework of the wavelet transform based coder. The idea is to convert the perceptual threshold of each subband into a wavelet constrain. To do that the authors defined  $e$ , an  $N \times 1$  error vector consisting of the value of the Discrete Fourier Transform of the error in reconstructing the signal from a sequence of approximate wavelet coefficients ( $N$  is the length of the audio frame). Also  $R_D$  is defined as a diagonal matrix with entries equal to discretized value of one over the masked noise power.

The psychoacoustic model implies that the reconstruction error due to the quantization or approximation of the wavelet coefficients corresponding to the given audio signal may be made inaudible as long as

$$|e_i|^2 r_{ii} \leq N, \quad \text{for } i = 1, \dots, N,$$

where  $e_i$  is the  $i$  th component of  $e$  and  $r_{ii}$  is the  $i$  th diagonal entry of  $R_D$ . The above equation can be written in its vector form as

$$e' R_D e \leq N,$$

which is equivalent to

$$e_q' Q W' R_D W Q' e_q \leq N,$$

where  $e_q$  is the  $N \times 1$  vector consisting of the values of the error in the quantization of wavelet coefficients. Here  $Q$  and  $W$  are respectively the Wavelet Transform and the DFT matrix.  $Q'$  and  $W'$  denote respectively the complex conjugate transpose of  $Q$  and  $W$ . Note that  $Q$  is fully determined by the wavelet coefficients. Note also that this constrain represents a multidimensional rectangle that could be also simplified by an ellipsoid fitted inside the rectangle.

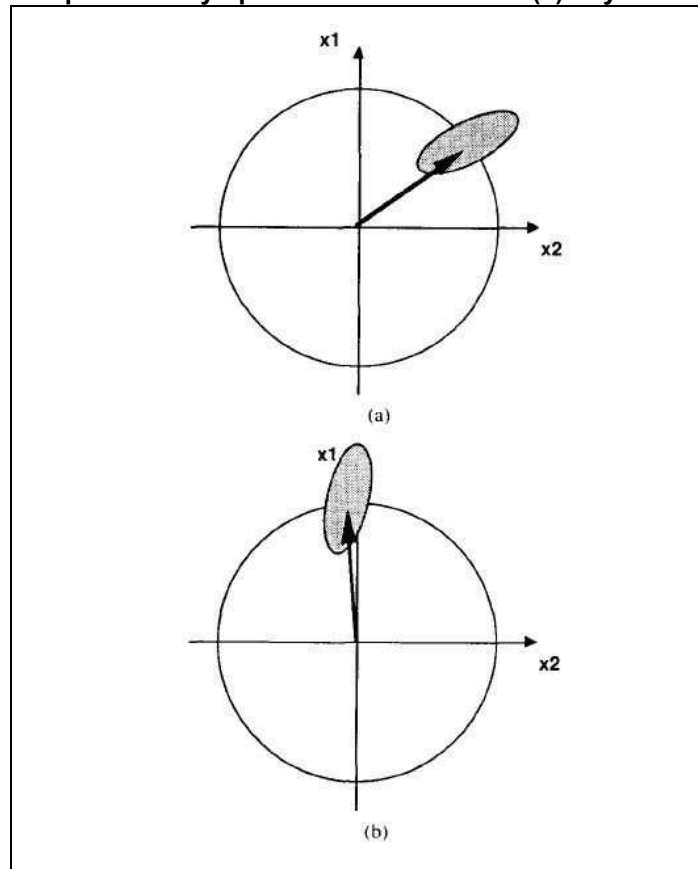
### 3.5 Reducing the number of non-zero coefficients: Optimization criterion

For each frame, an optimum wavelet representation is selected to minimize the number of bits required to represent the frame while keeping any distortion inaudible. This wavelet selection is the strongest compression technique of the paper, because it highly reduces the number of non-zero wavelet coefficients. In addition to that, those coefficients may be encoded using a small number of bits. Therefore, this technique involves choosing an analysis wavelet and allocating bits to each coefficient in the resulting wavelet representation.

The Figure N°2 explains how this technique works. It shows a signal vector representation by a particular choice of a basis. The radius of the sphere shown is equal of the norm of the time domain signal, and the error ellipse corresponds to the perceptual seminorm calculated by the psychoacoustic model. The audio segment can be represented using any vector whose tip lies inside the error ellipse with no perceptual distortion. Hence, the projection of the error ellipsoid along each coordinate axis specifies the coarsest quantization that can be used along the axis without producing any perceptual degradation. Therefore, a large projection along a particular coordinate axis implies that only a small number of bits to quantize that coordinate need to used. Exploiting this fact, a low bit rate representation of the signal can be achieved by the rotation of the vector representation of the signal via a unitary wavelet transformation.

This has two desirable results. First, the projection of the signal vector along most coordinate directions becomes same as that of the error ellipsoid. The signal vector projections along these coordinate directions can therefore, either be neglected and set to zero, or encoded using a small number of bits without producing any perceptual degradation. Second, the projection of the error ellipsoid is made large along the remaining coordinate directions. The signal vector projections along these directions can then be encoded using a small number. Since the wavelet transform is a family of orthogonal basis it provides the flexibility of choosing the unitary transformation that best achieves these two desirable results.

Figure 2.- Audio compression by optimal basis selection: (a) any basis (b) optimal basis.



To apply this technique, let  $R_k(\theta)$  be the number of bits assigned to the quantization of the  $k$ th transform coefficient  $x_k^q(\theta)$  when the wavelet identified by the vector  $\theta$  is used to decompose frame  $x$ . The goal is to minimize

$$R(\theta) = \sum_{k=1}^N R_k(\theta)$$

by properly choosing  $\theta$  and the number of bits  $R_k(\theta)$  assigned to the quantization of each transform coefficient  $x_k^q(\theta)$ . The minimization must be done under the constraint on the perceptual encoding error. It is proven in the paper that, for a particular choice of a wavelet, the bit rate requirement may be computed using the following formula directly from the transform coefficients. The best wavelet is then identified by minimizing the following over all vectors:

$$R_{\min}(\theta) = \frac{1}{2} \sum_k \log_2 \frac{(x_k^q(\theta))^2 w_{kk}(\theta)}{C},$$

where  $w_{kk}$  comes from the matrix  $W$  and  $C$  is an arbitrary constant.



Thus, this wavelet based encoding method essentially involves an optimization over all wavelets of a given support length to identify the one that minimizes the bit rate.

The authors evaluated this with the following results:

- An optimization is required, because there is no need to perform a full-blown search for the optimal basis. It is only necessary to search under wavelets with a large number of vanishing moments.
- Longer sequences yield better results: This is because longer sequences correspond to wavelet filter banks with sharper transition bandwidths. Again, this property is given by wavelets with a large number of vanishing moments.

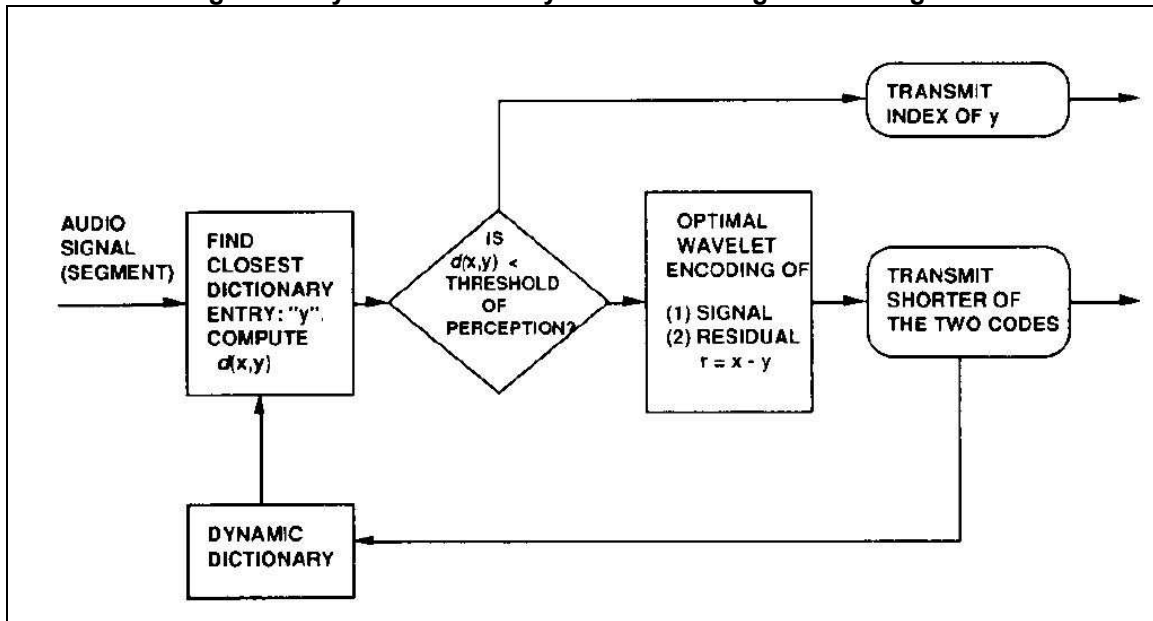
### 3.6 Results Dynamic Dictionary approach

Further reduction of the bit rate requires getting rid of statistical redundancies in the signal. A simple dynamic dictionary is used to eliminate the statistical redundancies in the signal. Both the encoder and the decoder maintain the dictionary. It is updated at the encoder and the decoder using the same set of rules and decoded audio frames.

This dictionary works using the following idea: for each frame  $x$  of the audio data, first a best matching entry  $x^D$  currently present in the dictionary is identified. Next, the residual signal  $r = x^D - x$  is calculated. Both  $x$  and  $r$  are then encoded using wavelet based method. Finally the code which requires the smaller number of bits is transmitted.

The dictionary in this coding scheme is dynamic. The minimum distance measure between the decoded signal corresponding to the frame  $x$  and the perceptually closest entry into the dictionary is compared against a preselected threshold. If it is below the threshold the dictionary remains unchanged. Otherwise the decoded signal is used to update the dictionary by replacing the last-used entry of the dictionary is replaced by decoded signal. Several improved techniques for dictionary update in the audio coder can be used.

Figure 3.- Dynamic dictionary based encoding of audio signals



### 3.7 Implementation Issues

- The technique described in this paper requires a long coding delay. Decoding on the other hand can be accomplished in real time.
- When selecting frame size it necessary to address two conflicting requirements. A larger frame size is desirable for maintaining lower bit rates, but, on the other hand, larger frames sizes may also lead to poorer quality because of audio signals are non-stationary.
- Frame size can lead also to a significant amount of pre-echoes in signals containing sudden bursts of energy. This problem is solved by using an adaptive frame size depending on the incoming audio signal, by dividing the frame and monitoring the variation of bit rate requirement to decide whether to change the size of the frame or not.
- The proposed default frame size (2048 samples) lead the best result with the current design of the algorithm.
- Side information requires just a few bits per frame.
- Wavelet transform coefficients may still contain redundancies which can be exploited using an entropy coding method, e.g., a Huffman code or a Ziv-Lempel type of encoding.

### 3.8 Results

The results of the algorithm, referred as Wavelet Technique Coder, were evaluated using subjective testing. The audio source material are of CD quality, and it contains some music signals which have been traditionally considered to be “hard” to encode (e.g., the castanets, drums, etc). Different subjective testing techniques were used to reduce the error. Some of those results are summarized in the following tables.

**Table 1.- Subjective listening test results: Transparency test**

<b>Music Sample</b>	<b>Average probability of original music preferred over WTC encode music</b>	<b>Sample Size [N° of people]</b>	<b>Comments</b>
Drums (solo)	0.44	18	Transparent
Pop (vocal)	0.58	36	Transparent
Castanets (solo)	0.61	36	Nearly Transparent
Piano (solo)	0.66	18	Original Preferred

**Table 2.- Subjective listening test results: Comparison with MPEG Coding**

<b>Music Sample</b>	<b>Average probability of original music preferred over WTC encode music</b>	<b>Sample Size [N° of people]</b>	<b>Comments</b>
Castanets (solo)	0.33	45	WTC clearly preferred
Piano (solo)	0.53	36	Same quality.

Finally, the authors claim that combined adaptive wavelet selection and dynamic dictionary coding procedures achieve almost transparent coding of monophonic CD quality signals at bit rates of 48-66 kb/s.

## 4.- WAVELET PACKET APPROACH

This chapter summarizes the approach to audio compression using wavelet packet as described in [2] “High Quality Audio Compression using an Adaptive Wavelet Packet Decomposition and Psychoacoustic Modeling”, by Pramila Srinivasan and Leah H. Jamieson.

### 4.1 Original abstract in [2]

*“This paper presents a technique to incorporate psychoacoustic models into an adaptive wavelet packet scheme to achieve perceptually transparent compression of high-quality (44.1 kHz) audio signals at about 45 kb/s. The filter bank structure adapts according to psychoacoustic criteria and according to the computational complexity that is available at the decoder. This permits software implementations that can perform according to the computational power available in order to achieve real time coding/decoding. The bit allocation scheme is an adapted zero-tree algorithm that also takes input from the psychoacoustic model. The measure of performance is a quantity called subband perceptual rate, which the filter bank structure adapts to approach the perceptual entropy (PE) as closely as possible. In addition, this method is also amenable to progressive transmission, that is, it can achieve the best quality of reconstruction possible considering the size of the bit stream available at the encoder. The result is a variable-rate compression scheme for high-quality audio that takes into account the allowed computational complexity, the available bit-budget, and the psychoacoustic criteria for transparent coding. This paper thus provides a novel scheme to marry the results in wavelet packets and perceptual coding to construct an algorithm that is well suited to high-quality audio transfer for Internet and storage applications.”*

*- Pramila Srinivasan and Leah H. Jamieson.*

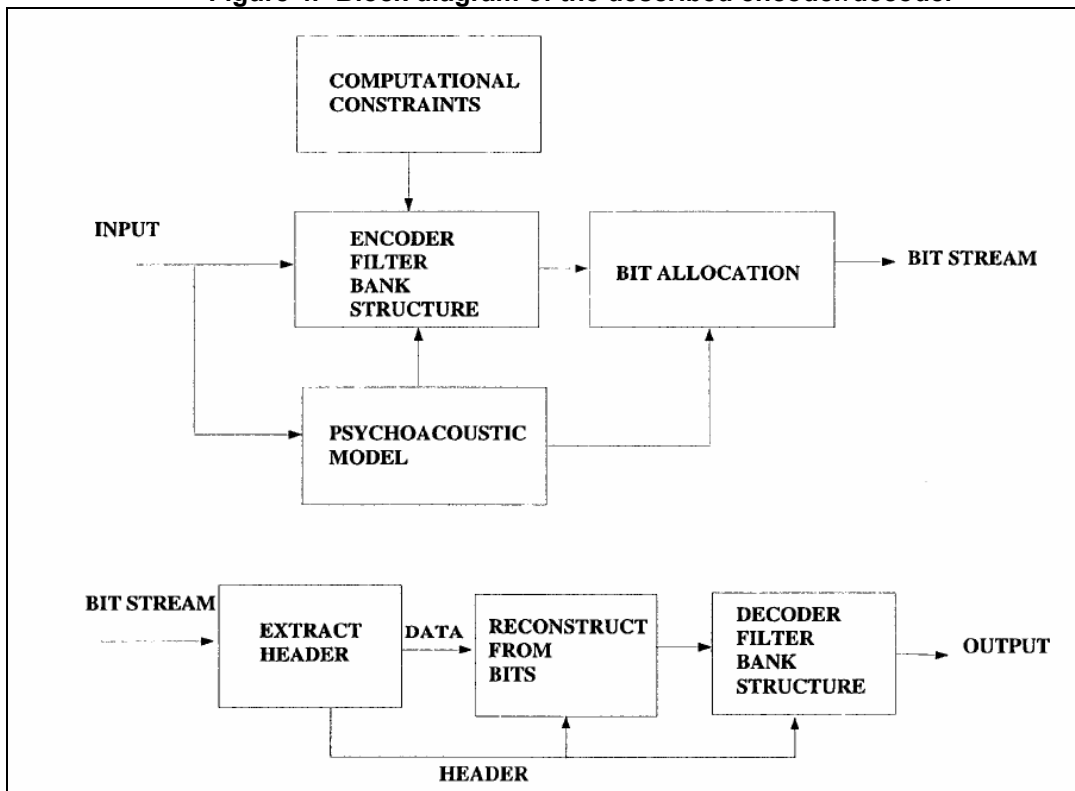
## 4.2 General picture

As in the previous paper, the main goal of this new algorithm is to compress high quality audio maintaining transparent quality at low bit rates. In order to do this, the authors explored the usage of an adaptative wavelet packet decomposition. Several key issues are considered as follows:

- Design a subband structure for wavelet representation of audio signals. This design also determines the computational complexity of the algorithm for each frame;
- Design a scheme for efficient bit allocation, which depends on the temporal resolution of the decomposition

In this chapter the summary of the main contributions for each of these points is presented

Figure 4.- Block diagram of the described encoder/decoder



### 4.3 Psychoacoustic model

#### 4.3.1 Subband masking model

The psychoacoustic model used in this paper closely resembles Model II of the ISO-MPEG specification, which means that it uses data from the previous two windows to predict, via linear extrapolation, the component values for the current window using a concept that they defined as tonality measure (it ranges from 0 to 1).

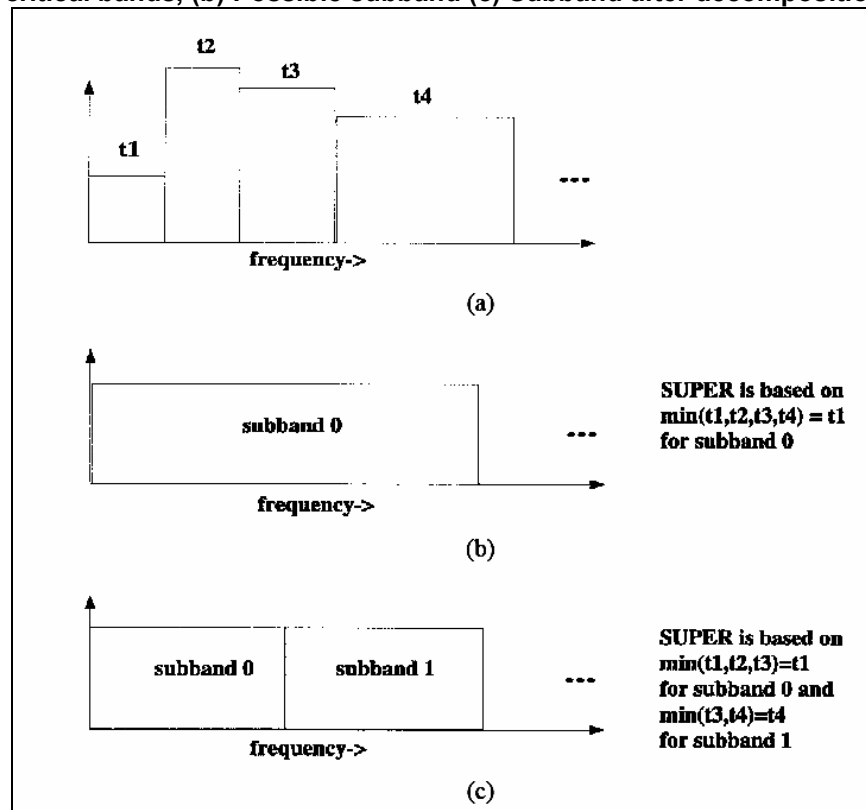
Using this concept and a spreading function that describes the noise-masking property, they compute the masking threshold in each subband given a decomposition structure. The idea is to use subbands that resemble the critical bands of the auditory system to optimize this masking threshold. This is the main reason why the authors chose the wavelet packet structure.

#### 4.3.2 Masking constrain in the wavelet structure

The main output of this psychoacoustic model block is a measure called the SUPER for a subband structure. SUPER (subband perceptual rate) is a measure that tries to adapt the subband structure to approach the PE as closely as possible. By PE (perceptual entropy) we understand the fundamental limit to which we can compress a signal with zero perceived distortion. The SUPER is the minimum number of bits in each subband (iteratively computed). It is used to decide on the need to further decompose the subband. This helps to prevent high estimates of SUPER due to several critical bands with different bit rate requirements coalescing into one subband.

Therefore, the problem is to adaptively decide the optimal subband structure that achieves the minimum SUPER, given the maximum computational limit and the best temporal resolution possible (that renders the bit allocation scheme most efficient).

Figure 5.-Calculation of SUPER based on the subband structure. (a) Threshold fro the critical bands, (b) Possible subband (c) Subband after decomposition



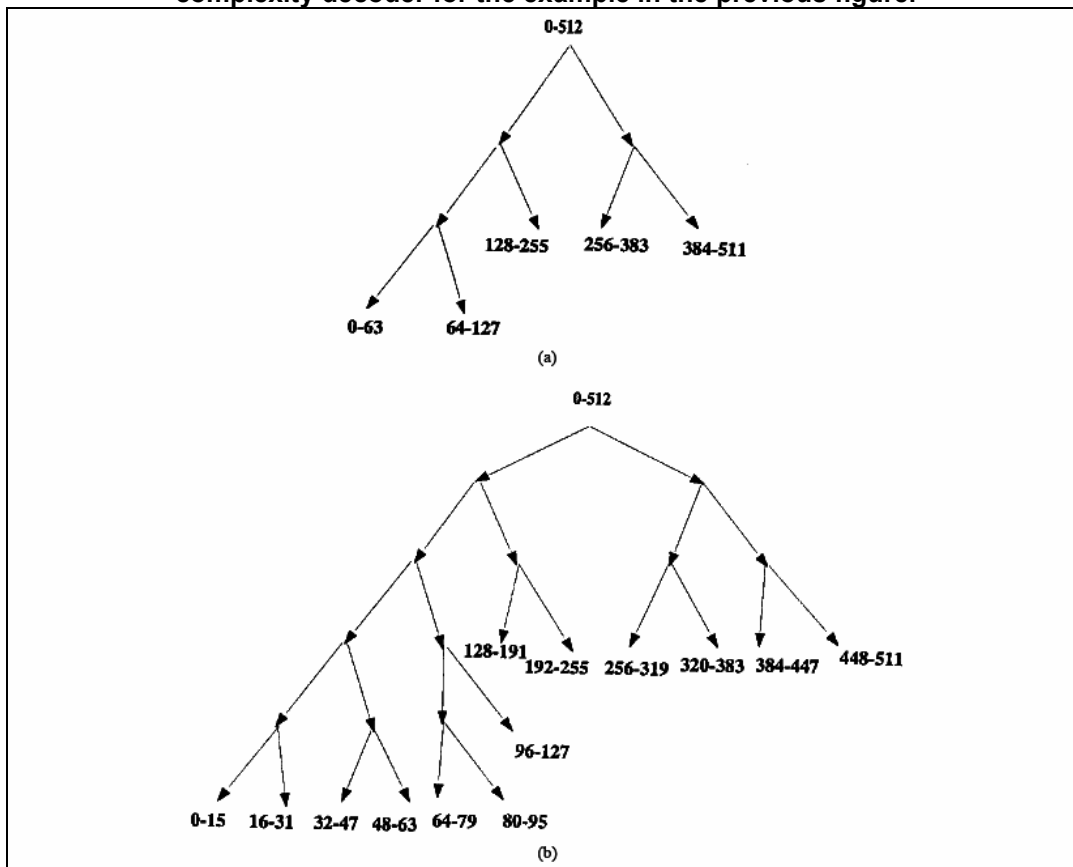
#### 4.4 Wavelet packet representation

Given a wavelet packet structure, a complete tree structured filter bank is considered. Once we find the "best basis" for this application, a fast implementation exists for determining the coefficients with respect to the basis. However, in the "best basis" approach, they do not subdivide every subband until the last level. The decision of whether to subdivide is made based on a reasonable criterion according to the application (further decomposition implies less temporal resolution).

The cost function, which determines the basis selection algorithm, will be a constrained minimization problem. The idea is to minimize the cost due to the bit rate given the filter bank structure, using as a variable the estimated computational complexity at a particular step of the algorithm, limited by the maximum computations permitted. At every stage, a decision is made whether to decompose the subband further based on this cost function.

Another factor that influences this decomposition is the tradeoff in resolution. If it is decomposed further down, it will sacrifice temporal resolution for frequency resolution. The last level of decomposition has minimum temporal resolution and has the best frequency resolution. The decision on whether to decompose is carried out top-down instead of bottom-up. Following that way, it is possible to evaluate the signal at a better temporal resolution before the decision to decompose. It is proved in this paper that the proposed algorithm yields the “best basis” (minimum cost) for the given computational complexity and range of temporal resolution.

**Figure 6.- Example of adaptation of the tree for a (a) low-complexity and (b) high-complexity decoder for the example in the previous figure.**





## 4.5 Efficient Bit allocation

The bit allocation proceeds with a fixed number of iterations of a zero-tree algorithm before a perceptual evaluation is done. This algorithm organizes the coefficients in a tree structure that is temporally aligned from coarse to fine. This zero-tree algorithm tries to exploit the remnants of temporal correlations that exist in the wavelet packet coefficients. It has been used in other wavelets applications, where its aims has been mainly to exploit the structure of wavelet coefficients to transfer images progressively from coarse to fine resolutions. In this case, a one-dimensional adaptation has been included with suitable modifications to use the psychoacoustic model. This algorithm is discussed neither in this paper nor in this report.

## 4.6 Implementation Issues

- Most of the implementation details of the psychoacoustic model essentially follow the MPEG specification. For the bit allocation scheme, implementation details follow the zero-tree algorithm.
- After the zero-tree algorithm, a lossless compression technique is used. No details are mentioned about this final compression.
- All results presented in this paper use filter banks that implement the spline-based biorthogonal wavelet transform (order 5 was considered the optimum).
- The filters are FIR, yield perfect reconstruction and avoid aliasing.
- Due to the bit allocation technique used, this method is amenable to progressive transmission, that is, it can achieve reconstruction considering the size of the bit stream available at the encoder.

## 4.7 Results

The results of the algorithm were also evaluated using subjective testing, but with a different methodology than the previous paper. This evaluation was performed with less test subjects. The audio source material is of CD quality. The results are summarized in the following table.

**Table 3.- Subjective listening test results.**

<b>Music Sample</b>	<b>Likelihood of listener preferring the original over the reconstructed (0-1)</b>
Violin	0.5
Violin and viola	0.4962
Flute	0.5*
Sitar	0.5
Film tune	0.5
Saxophone	0.5072

*\* The author mentioned that the algorithm did not perform with the flute as well as in the other cases, but this observation is not reflected in the presented results.*

Finally, the authors claim that perceptually transparent compression of high-quality (44.1 kHz) audio signals at about 45 kb/s. They also mentioned that the computational adaptation of the proposed algorithm and its progressive transmission property make this scheme very suitable for internet applications.

## 5.- SOME COMMENTS ON THE PAPERS

Even though these two papers have the same objective, to perform compression of high quality audio maintaining transparent quality at low bit rates, and they define almost the same steps to achieve that goal (e.g., both uses wavelets, both define a psychoacoustic model, and both perform efficient bit allocation, both perform quality measures), the approaches are completely different. It is possible to compare a few ideas observed in the summaries:

- First, [1] uses a discrete wavelet decomposition of the audio signal and [2] uses a wavelet packet approach. Therefore, [2] is based in the frequency domain behavior and [1] perform most of the steps in time domain.
- The psychoacoustic model defined in [1] is simpler (does not consider the behavior of previous frames) than the one in [2], and it is designed in time domain instead of frequency domain, to be congruent with the previous point.
- Exploiting the previous idea, tree structured filter banks yield better descriptions of the psychoacoustic model.
- The efficiency of the bit allocation in [1] is based in a novel technique, first presented in this paper. It uses an optimal basis to reduce the number of non-zero wavelet coefficients. On the other hand [2] uses a known algorithm (zero-tree algorithm) to perform the bit allocation.
- It is notable that [1] presents more insights when describing the approaches and considerations of the algorithm. On the other hand [2] uses most of its ideas from other authors, it only describe some minor details.
- The final evaluation is best presented in [1], because it considers more test subjects and shows more details and discussion.

## **6.- MATLAB SIMULATIONS**

The objective of this simulation is to demonstrate some of the features of one of the papers. This simulation is not intended to replicate the whole algorithm but to show how some of the ideas are used in practice. The simulation was design in MATLAB 6.5 and using its Wavelet Toolbox 2.2.

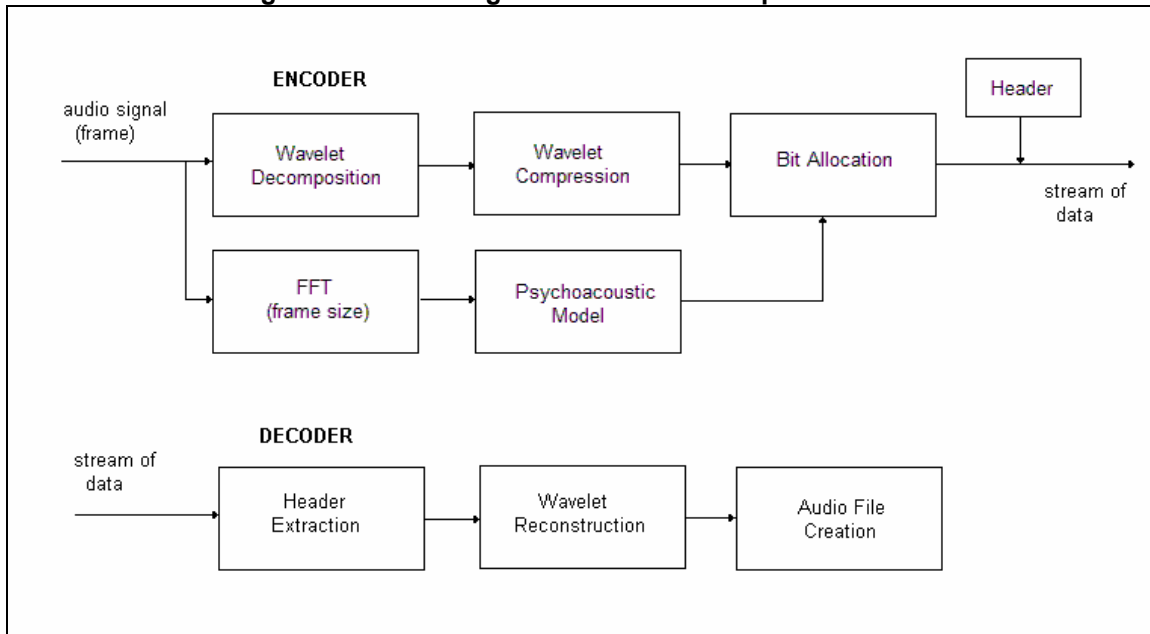
Due to the fact that a number of implementation details are not included in the paper that describes the wavelet packet approach [2], it is more convenient to design this implementation based on the features described in the discrete wavelet transform approach [1].

### **6.1 Main features of the implementation**

The Matlab implementation of [1] includes the following features:

- (a) Signal division and processing using small frames
- (b) Discrete wavelet decomposition of each frame
- (c) Compression in the wavelet domain
- (d) A psychoacoustic model
- (e) Non linear quantization over the wavelet coefficient using the psychoacoustic model
- (f) Signal reconstruction
- (g) Main output: Audio files

**Figure 7.- Block Diagram of the Matlab Implementation**



## 6.2 Considerations

Even though it is more convenient to implement the ideas described in [1], some of the suggested steps require a complicated implementation. Therefore, a few modifications and considerations have been included to the design of this MATLAB simulation:

- (a) No search for optimal basis is performed

Even though this is one of the key point of the paper, its implementation is requires a large programming design, and that is out of the scope of this demonstration. To compensate that, another compression technique has been used. This is based in the known discrete wavelet decomposition compression that uses an optimal global threshold. This technique has been successfully used in audio compression and it is described in [3]. Using the recommendations of that paper, the best results were observed when using a *Daubechies* wavelet with 10 vanishing moments (dB10 in matlab) and 5 levels of decomposition. These choices will overcome the lack of an optimal basis search.

(b) Non overlapping frames are included

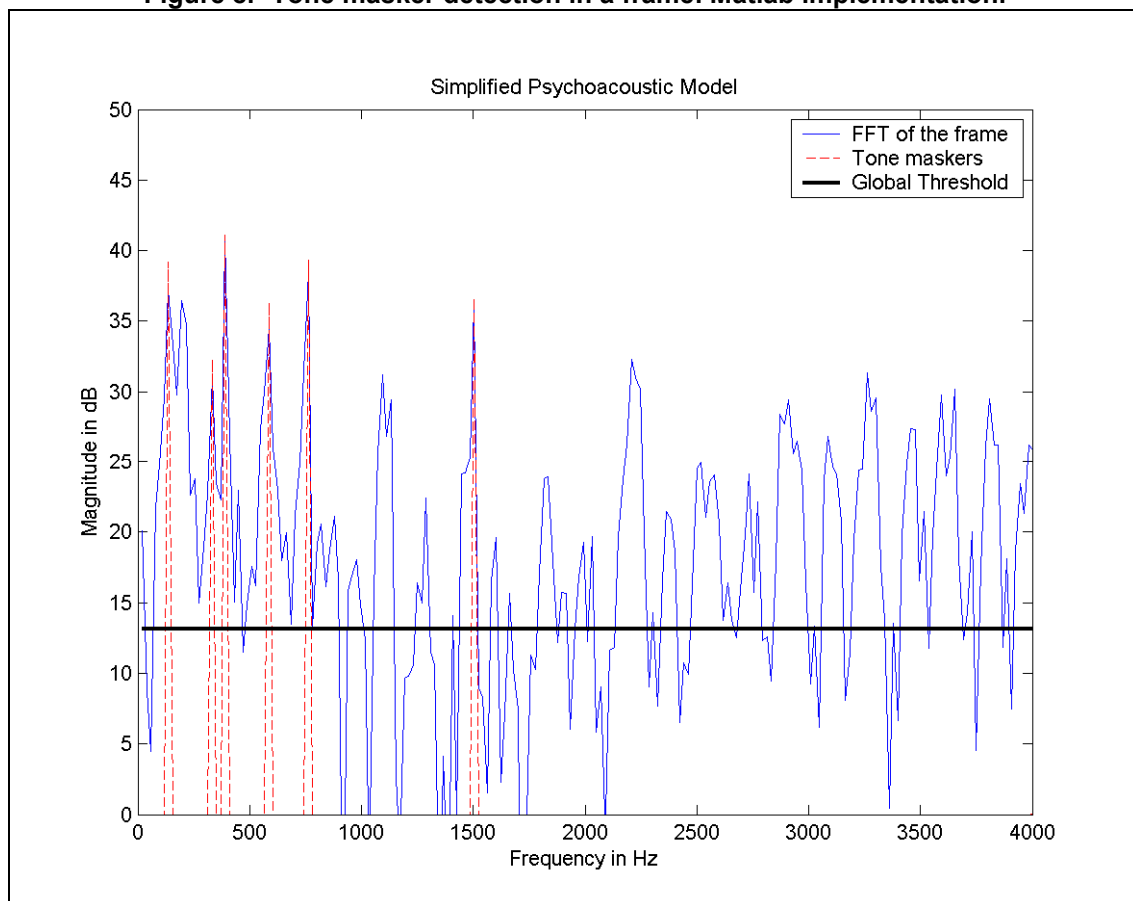
This implementation does not have overlapping frames to avoid computational complexity. The frame size is given by the recommendations in [1] corresponding to 2048 samples per frame.

(c) The psychoacoustic model is simplified

Due to the complexity associated with the construction of a psychoacoustic model, a simplified version was considered. This model can only detect masking tones in the signal, and gives a general threshold for all the frequencies. The model in [1] process also noise and gives a threshold for each subband.

An example of this simplified psychoacoustic model is shown in the following figure. The main tonal components are detected. The power average of this components is used as masking threshold for every frequency.

**Figure 8.- Tone masker detection in a frame. Matlab implementation.**



*Note: Global Threshold in this plot is referred to the making level for all frequencies. With this value the SNR and the number of bits are computed. Note also that the FFT is performed over the whole frequency range. That frequency range that is shown in this figure was selected only for visual purposes.*

(d) No new audio format was design

Even though this simplified matlab implementation performs compression over the audio signal, this is not reflected in the size of the new audio files. This is due to the fact that a new format design was not considered, so the *wavwrite* command was used to create the audio files (.wav). The compression ratio for each case is calculated using other variables of the simulation.

(e) A lossless compression at the end was tested and suppressed

Arithmetic compression (similar to Huffman coding) was tested in this simulation but was suppressed at the end because it made the simulation too slow. However its performance is considered in the results.

### 6.3 Results

As explained before, no objective parameters (e.g. mean square error) are used when evaluating the quality of compressed audio. Therefore the results of this implementation were also evaluated using subjective testing. However these results must be considered only as a reference because they don not provide any statistical support. The audio source material was monophonic CD audio.

In a general appreciation, the quality reached by this implementation is not transparent, with an audible distortion that resembles the typical hiss noise from a vinyl disc. However, the quality reached is better than the one provided by a telephone line but worst than the one provided by a FM radio; which in any case is worse than the typical mp3 quality. This result is due to the considerations and simplifications that were taken into account. However, if we compare the results from this compression scheme with the ones obtained by only reducing the number of bit directly over the audio signal (namely “blind compression”), the quality is considerable better. For auditory test please refer to the CD attached with this report.

To identify the behavior of the compression scheme several audio files were tested.. The results of the evaluation are summarized in the following table.

**Table 4.- Subjective listening test results.**

<b>Music Style / Instrument</b>	<b>Comments</b>
Jazz – Saxophone	No special comments
Classic - Orchestra	No special comments
Contemporanean – Flute	Typical flute blows are more notorious
Contemporanean – Piano	Medium and high frequencies are slightly more notorious
Ragga – Sitar	Medium and high frequencies are slightly more notorious
Pop – Vocal	Plosive and fricative consonants are slightly more notorious

Another way to evaluate our implementation is by measuring how much it did compress signals and what are the resulting bit rates. To do this we measure the average bit per sample on each tested signal. It is useful to include here the extra reduction attained by the lossless technique that our implementation tested but did not include at the end for computational constrains.

**Table 5.- Summary of the compression. Bit rates and compression ratios**

<b>Average Number of bits used per sample</b>	<b>Current scheme</b>		<b>With the lossless compression**</b>	
	<b>Bit Rate</b>	<b>Compression Ratio</b>	<b>Bit Rate</b>	<b>Compression Ratio</b>
5*	141kb/s	5:1	84kb/s	9:1

*\* Based on the average of all the test files \*\*40% of extra compression based on a few tests*

Finally, it can stated that even though the quality of this implementation is not comparable with the current standardized compressed formats, the scheme is compressing the signal efficiently with a bit rate a comparable with the one obtained in the original paper [1].

It is also necessary to mention that this Matlab implementation is not very efficient in terms of computational complexity but, fortunately, that is out of the scope for this project.



## 7.- DISCUSSION: STATE-OF-THE-ART

This chapter presents a brief appreciation on the current state-of-the-art of wavelets in audio compression. In order to do this, it is convenient to divide again between general audio/music signal and speech signals.

Let's start with audio and music. Historically, in the 1970s and 1980s the idea of subband audio coding was very popular and widely investigated. This technique has been refined over the years and now has morphed into what it is called perceptual audio coding which is the basis for the MPEG audio coding standards. By using this idea, lossy compression has reached transparent quality (see 2.4 in this report) at low bits rates. However, lossy compressed files are unsuitable for professional audio engineering applications such as room acoustics studies, sound editing or multitrack recording.

As observed in [2], tree structured filter banks yield better descriptions of the psychoacoustic model. However this structure does not lead to the best representation of the auditory channel model, mainly because such filter banks give power of two decompositions and they do not approximate the Bark Scale very well. The critical bands associated with human hearing (the same Bark Scale) are roughly uniform to a first order. The bands are smaller at low frequencies and get slightly larger as we move to higher frequencies. Furthermore, tree structures result in the equivalent of very long filters with excessive delay. Often when coding the subbands, long filters can result in pre-echo distortions which are audible sometimes.

On the other hand, speech is a signal that is more structured. This structure can be exploited through models like LPC, vocoders, and HMMs. Again, tree structures do not help much relative to the competing methods that have been developed.

Due to all the previous ideas, wavelet techniques are not included in any current standard in audio coding. Current technology is based on MDCT which describes better the psychoacoustic model and have fewer implementation problems. However it is possible to see some application of wavelets on audio and speech. Some authors have successfully applied wavelets for watermarking in audio and speech signals.

## 8.- CONCLUSIONS

- A brief summary of the current compression techniques and the main considerations that people do when evaluating their results has been presented.
- Two papers that use wavelet technique were studied, with particular interest in the one based on discrete wavelet transform decomposition [1].
- A MATLAB simulation of the selected paper was successfully implemented, simplifying some of its features, but keeping its main structure and contributions.
- Even though some simplifications were considered, the MATLAB implementation met the objective of showing the main features of the algorithm.
- The quality of the compressed signal obtained with the MATLAB implementation is lower than any current standard compressing schemes (e.g. mp3), but considerable better than one obtained just by “blindly compressing” the signal.
- Further upgrades can be considered to the MATLAB implementation to obtain better results.

## 9.- ACKNOWLEDGMENTS

The author of this report wants to acknowledge Prof. Mark Smith for his contributions to the better understanding of the current state-of-the-art of wavelets in audio compression.

## 10.-BIBLIOGRAPHY AND REFERENCES

- [1] D. Sinha and A. Tewfik. "Low Bit Rate Transparent Audio Compression using Adapted Wavelets", IEEE Trans. ASSP, Vol. 41, No. 12, December 1993.
- [2] P. Srinivasan and L. H. Jamieson. "High Quality Audio Compression Using an Adaptive Wavelet Packet Decomposition and Psychoacoustic Modeling", IEEE Transactions on Signal Processing, Vol 46, No. 4, April 1998.
- [3] J.I. Agbinya, "Discrete Wavelet Transform Techniques in Speech Processing", IEEE Tencon Digital Signal Processing Applications Proceedings, IEEE, New York, NY, 1996, pp 514-519.
- [4] Ken C. Pohlmann "Principles of Digital Audio", McGraw-Hill, Fourth edition, 2000.
- [5] X. Huang, A. Acero & H-W. Hon "Spoken Language Processing: A Guide to Theory, Algorithm and System Development", Pearson Education, 1st edition 2001.
- [6] S.G. Mallat. "A Wavelet Tour of Signal Processing." 2nd Edition. Academic Press, 1999. ISBN 0-12-466606-X
- [7] J.G. Proakis and D.G. Manolakis, Digital Signal Processing: Principles, Algorithms, and Applications, Prentice-Hall, NJ, Third Edition, 1996.
- [8] Mathworks, Student Edition of MATLAB, Version 6.5, Prentice-Hall, NJ.

### Websites:

- [9] <http://www.m4a.com/>, April 20, 2005.
- [10] <http://www.vialicensing.com/products/mpeg4aac/standard.html> , April 24, 2005.
- [11] <http://is.rice.edu/%7Ewelsh/elec431/index.html>, April 24, 2005.
- [12] <http://perso.wanadoo.fr/polyvalens/clemens/wavelets/wavelets.html>, April 24, 2005.
- [13] <http://www.mp3developments.com/article4.php>, April 24, 2005.

## 11.-MATLAB CODE

```

%Final Project: MATLAB Code
%ECE 648 - Spring 2005
%Wavelet, Time-Frequency, and Multirate Signal Processing
%Professor: Ilya Pollak
%Project Title: Audio Compression using Wavelet Techniques
%Matias Zanartu - mzanartu@ecn.purdue.edu
%ID:999-09-2426
%Purdue University

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This one-file code performs wavelet compression over a .wav file. The
scheme is %a simplified version of the one described on the paper "Low
Bit Rate %Transparent Audio Compression using Adapted Wavelets" by
Deepen Sinha %and Ahmed H. Tewfik published in IEEE Trans. ASSP, Vol.
41, No. 12, %December 1993.
%
%NOTES: The file must be in the same folder where this file is located.
%If you want to try this scheme with other audio file, please change
%the name of the %variable "file". Avoid using long audio files or long
%silences at the beginning of the file for computational constrains.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;clc;
file='Coltrane.wav';
wavelet='dB10';
level=5;
frame_size=2048;
psychoacoustic='on '; %if it is off it uses 8 bits/frame as default
wavelet_compression = 'on ';
heavy_compression='off';
compander='on ';
quantization = 'on ';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%          ENCODER          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[x,Fs,bits] = wavread(file);
xlen=length(x);
t=0:1/Fs:(length(x)-1)/Fs;

%decomposition using N equal frames
step=frame_size;
N=ceil(xlen/step);
%computational variables
Cchunks=0;
Lchunks=0;
Csize=0;
PERF0mean=0;
PERFL2mean=0;
n_avg=0;

```

```

n_max=0;
n_0=0;
n_vector=[];
for i=1:1:N
    if (i==N)
        frame=x([(step*(i-1)+1):length(x)]);
    else
        frame=x([(step*(i-1)+1):step*i]);
    end
%wavelet decomposition of the frame
    [C,L] = wavedec(frame,level,wavelet);
%wavelet compression scheme
    if wavelet_compression=='on '
        [thr,sorh,keepapp] = ddencomp('cmp','wv',frame);
        if heavy_compression == 'on '
            thr=thr*10^6;
        end
        [XC,CXC,LXC,PERF0,PERFL2] = wdencomp('gbl',C, L, wavelet,
level,thr,sorh,keepapp);
        C=CXC;
        L=LXC;
        PERF0mean=PERF0mean + PERF0;
        PERFL2mean=PERFL2mean+PERFL2;
    end

%Psychoacoustic model
    if psychoacoustic=='on '
        P=10.*log10((abs(fft(frame,length(frame))).^2);
        Ptm=zeros(1,length(P));
        %Inspect spectrum and find tones maskers
        for k=1:1:length(P)
            if ((k<=1) | (k>=250))
                bool = 0;
            elseif ((P(k)<P(k-1)) | (P(k)<P(k+1))),
                bool = 0;
            elseif ((k>2) & (k<63)),
                bool = ((P(k)>(P(k-2)+7)) & (P(k)>(P(k+2)+7)));
            elseif ((k>=63) & (k<127)),
                bool = ((P(k)>(P(k-2)+7)) & (P(k)>(P(k+2)+7)) & (P(k)>(P(k-
3)+7)) & (P(k)>(P(k+3)+7)));
            elseif ((k>=127) & (k<=256)),
                bool = ((P(k)>(P(k-2)+7)) & (P(k)>(P(k+2)+7)) & (P(k)>(P(k-
3)+7)) & (P(k)>(P(k+3)+7)) & (P(k)>(P(k-4)+7)) & (P(k)>(P(k+4)+7)) &
(P(k)>(P(k-5)+7)) & (P(k)>(P(k+5)+7)) & (P(k)>(P(k-6)+7)) &
(P(k)>(P(k+6)+7)));
            else
                bool = 0;
            end
        end
        if bool==1
            Ptm(k)=10*log10(10.^(0.1.*(P(k1)))+10.^(0.1.*(P(k)))+10.^(0.1.*P(k+1)));
        end
    end
    sum_energy=0;%sum energy of the tone maskers
    for k=1:1:length(Ptm)
        sum_energy=10.^(0.1.*(Ptm(k)))+sum_energy;
    end
end

```

```

E=10*log10(sum_energy/(length(Ptm)));
SNR=max(P)-E;
n=ceil(SNR/6.02);%number of bits required for quantization
if n<=3%to avoid distortion by error of my psychoacoustic model.
    n=4;
    n_0=n_0+1;
end
if n>n_max
    n_max=n;
end
n_avg=n+n_avg;
n_vector=[n_vector n];
end

%Compander(compressor)
if compander=='on '
    Mu=255;
    C = compand(C,Mu,max(C), 'mu/compressor');
end

%Quantization
if quantization=='on '
    if psychoacoustic=='off'
        n=8;%default number of bits for each frame - sounds better but
uses more bits
    end
    partition = [min(C):(max(C)-min(C))/2^n:max(C)];
    codebook = [min(C):(max(C)-min(C))/2^n:max(C)];
    [index,quant,distor] = quantiz(C,partition,codebook);
%find and correct offset
    offset=0;
    for j=1:1:N
        if C(j)==0
            offset=-quant(j);
            break;
        end
    end
    quant=quant+offset;
    C=quant;
end

%Put together all the chunks
Cchunks=[Cchunks C]; %NOTE: if an error appears in this line just
modify the transpose of C
Lchunks=[Lchunks L'];
Csize=[Csize length(C)];
Encoder = round((i/N)*100) %indicator of progress
end
Cchunks=Cchunks(2:length(Cchunks));
Csize=[Csize(2) Csize(N+1)];
Lsize=length(L);
Lchunks=[Lchunks(2:Lsize+1) Lchunks((N-1)*Lsize+1:length(Lchunks))];
PERF0mean=PERF0mean/N %indicator
PERFL2mean=PERFL2mean/N %indicator
n_avg=n_avg/N%indicator
n_max%indicator
end_of_encoder='done'%indicator of progress

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%In this part the signal is stored with the new format
%or transmitted by frames
%This new format uses this parameters:
%header: N, Lsize, Csize.
%body: Lchunks (small), Cchunks(smaller signal because now it is
%quantized with less bit and coded)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%          DECODER          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%reconstruction using N equal frames of length step (except the last
one)
xdchunks=0;
for i=1:1:N
    if i==N
        Cframe=Cchunks([(Csize(1)*(i-1))+1]:Csize(2)+(Csize(1)*(i-
1))]);
        %Compander (expander)
        if compander=='on '
            if max(Cframe)==0
            else
                Cframe = compand(Cframe,Mu,max(Cframe), 'mu/expander');
            end
        end
        xd = waverec(Cframe,Lchunks(Lsize+2:length(Lchunks)),wavelet);
    else
        Cframe=Cchunks([(Csize(1)*(i-1))+1]:Csize(1)*i]);
        %Compander (expander)
        if compander=='on '
            if max(Cframe)==0
            else
                Cframe = compand(Cframe,Mu,max(Cframe), 'mu/expander');
            end
        end
        end
        xd = waverec(Cframe,Lchunks(1:Lsize),wavelet);
    end

    xdchunks=[xdchunks xd];
    Decoder = round((i/N)*100) %indicator of progress
end
xdchunks=xdchunks(2:length(xdchunks));
distorsion = sum((xdchunks-x').^2)/length(x)
end_of_decoder='done'

%creating audio files with compressed schemes
wavwrite(xdchunks,Fs,bits,'output.wav') %this does not represnet the
real compression achieved. It is only to hear the results
end_of_writing_file='done'%indicator of progress

```

```
figure(1);clf;
subplot(2,1,1)
plot(t,x);
ylim([-1 1]);
title('Original audio signal');
xlabel('Time in [seg]','FontSize',8);
subplot(2,1,2)
plot(t,xdchunks,'r');
ylim([-1 1]);
title('Compressed audio signal');
xlabel('Time in [seg]','FontSize',8);
```



