AUDIO COMPRESSION USING WAVELETS IN MATLAB

A Dissertation submitted in partial fulfillment of the requirements for

the award of degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATIONS ENGINEERING

Submitted By

Madupathi Gayathri (17251A0446)

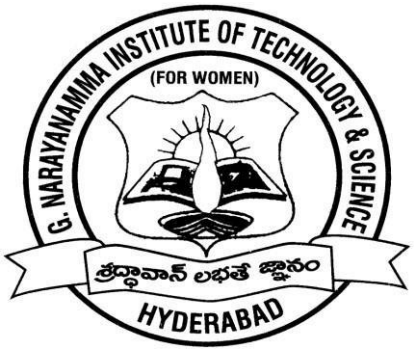
Chinmayee GVP (17251A0436)

Ramavath Bindu Madhavi (17251A0418)

SuraagaPradeepthi. komsani (17251A0426)

Under the esteemed guidance of

Mr.P.Satyanarayana Goud, Assistant Professor ,ECE, GNITS (FOR WOMEN).



DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

G. NARAYANAMMA INSTITUTE OF TECHNOLOGY & SCIENCE

(For Women)

AN ISO 9001-2015 CERTIFIED INSTITUTION

(Accredited by NBA& NAAC)

Affiliated to JNTU, Hyderabad

2020-2021

DEPT. OF ELECTRONICS & COMMUNICATION ENGG CERTIFICATE

THIS IS TO CERTIFY THAT THE MINI PROJECT ENTITLED

AUDIOCOMPRESSION USING WAVELETS IN MATLAB

IS THE BONAFIDE WORK OF

Madupathi Gayathri (17251A0446)

Chinmayee GVP (17251A0436)

Ramavath Bindu Madhavi (17251A0418)

Suraagapradeepthi.komsani (17251A0426)

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF DEGREE OF BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING DURING THE YEAR 2020-2021

Dr.B.VenkateshuluMr.	P.Satyanarayana Goud
PROF &HOD,ECE	Assistant Professor

ACKNOWLEDGEMENT

This project would not have been possible without the support of many people. We would like to acknowledge the contribution of the following individuals to the development of our project.

We thank our internal guide **Mr.P.Satyanarayana Goud** , Asst. Professor, ECE department, GNITS for her continued assistance and priceless suggestions concerning our project.

We wish to express our deep gratitude to our project coordinator **Dr.G.Srivalli**, Associate Professor, ECE department, GNITS for her encouragement during the project.

We would like to extend our heartful gratitude to **Dr.B.Venkateshulu**, Head of Department, ECE department, GNITS for his unstinting support in all our endeavors.

We owe our thanks **to Dr.K.Ramesh Reddy**, Principal, GNITS for permitting us to do the project.

We have been fortunate enough to have highly experienced staff that provides ceaseless encouragement and helped us directly or indirectly in the successful completion of our project.

And finally, we thank our parents who have been great sources of mental strength, always having support love and friends who have made the completion of this project possible.

Madupathi Gayathri (17251A0446)

Chinmayee GVP (17251A0436)

Ramavath Bindu Madhavi (17251A0418)

SuraagaPradeepthi.komsani (17251A0426)

TABLE OF CONTENTS

ABSTRACT

Audio frequencies range from 20Hz, but these frequencies are not heard in the same way. Frequencies below 20Hz are exceedingly difficult to hear, while those not much more than 20Hz, cannot be heard by most people. We often need to process these audio signals for various applications. We use internet for various purposes including entertainment. Audio is common in all these applications. If an audio file size is large, it takes more space to store. To reduce this problem audio compression is necessary.

Audio Compression is one of the basic technologies of the modern telecommunication age. Compression is the technique to convert high input data stream into smaller size. Audio coding gives us the digital form of audio with as few bits as possible and maintains the quality. the reduction in bit rates conserve bandwidth. It frees up space substantially. During compression, an audio signal is taken and analyzed using MATLAB for frequency and amplitude. Haar and Daubechies algorithms are applied on the speech signal and audio is compressed. Comparison of performance metrics such as PSNR, MSE and compression ratio shows the effective performance of audio encoding methods.

The compressive technique used in this project is better than other earlier coding techniques like μ-law coding, code excited linear predictive coding. The merits of the compression technique are reduction in storage space, bandwidth, transmission power and energy. Speech compression plays a prominent role in speech signal processing such as satellite communications, internet communications, transmission of biomedical signals and other applications.

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The purpose of this section 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.

1.1 Auditoryproperties

1.1.1 Nonlinear frequency response of thehear

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 extreme's frequencies (i.e. close to 20 Hz and20kHz).After many cochlear studies, scientists have found that the frequency range from 20Hz to 20 kHz can be broken up into critical bandwidths, which are non-uniform, nonlinear, and dependent on the level of the incoming sound. Signals within one critical bandwidth arehard to separate for a human observer. A detailed description of this behavior is describedin the Bark scale and Fletcher curves.

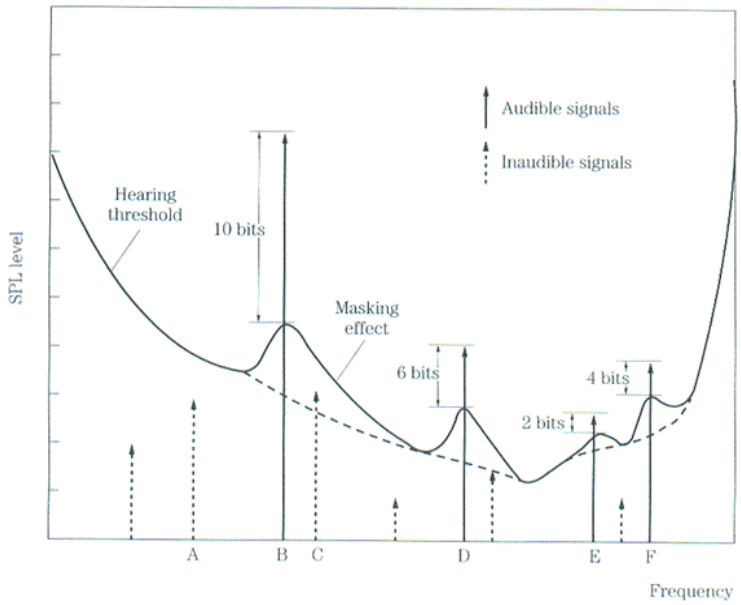


Figure 1.- An example that shows how the auditory properties can be used to compress an digital audio signal.

1.1.1 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 temporalmasking.

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 sinusoidalmasker, 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 10ms and the offset attenuation lasting approximately 50ms.

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.

1.2 Audiocompression

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 loss compression.

1.2.1 Losslesscompression

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 commander (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 μ-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 compressionwithout obtaining significant compression ratio.

1.2.2 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

usetransforms 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.

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 blockboundaries.

1.2.3 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 implementationissues.Itisinterestingforourreporttomentionsomebriefcomments 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 audiospectrum.

(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 tonalityindices.

(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 Huffmancoding.

All three layers of MPEG-1 use a polyphonefirebrand for signal decomposition into 32 equal width sub bands. This is a computational 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 quantizationerror.
- Adjacent filter banks overlap, then a single tone can affect two filterbanks.

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 fix 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 MP3at equivalent or slightly lowerbitrate

1.3 Speechcompression

Speech signals has unique properties that differ from a general audio/music signal. 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 interpolationcoders.

1.4 Evaluating compressedaudio

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 sincethen.

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.

2. DISCRETE WAVELET TRANSFORM APPROACH

This chaptersummarizes the approach to audio compression using discrete wavelet transform as described above “Low Bit Rate Transparent Audio Compression using Adapted Wavelets”, by D. Sinha and A. Tewfik.

This section 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 48-66kb/s.

2.1 Generalpicture

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 audiosignals.
- Design a psychoacoustic model to perform perceptual coding and adapt it to the waveletrepresentation.
- 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 thatrepresentation
- Transmit or store the steam of data. Decode andreconstruct.
- Evaluate the quality of the compressedsignal.

2.2 Wavelet representation for audio signals:

DWT signal representation is mostly used technique 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 (46ms at44.1 kHz). The two ends of each frame are weighted by the square root of a Hanning window of size 128 to avoid border distortions.

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 subbandsofthewaveletdecomposition.Thisallowsseveralchoicesforabasis.Thisfilter 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.

2.3 Psychoacousticmode

2.3.1 Simplified maskingmodel

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 Powerin the reproduced audio signal as long as the power spectrum is less than the masked noise power at each frequency.

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 bynoise.
- The model is based on psychoacoustic studies for the masking of a single tone like signal (quantization error could happen if it contains severalcomponents).
- Masking is assumed to be additive (a power law rule of addition should be used instead).

2.3.2 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 x 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 longas

$e_i^2 r_{ii} \leq N$, for $i=1,\ldots\ldots,N$, ……*(1)*

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

$e' R_D e \leq N$ ……(2)

which is equivalent to

$e_q' QW' R_D WQ' e_q \leq N$,……(3)

where $e_q$is the N x 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 therectangle.

2.4Reducing 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,efficients 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.

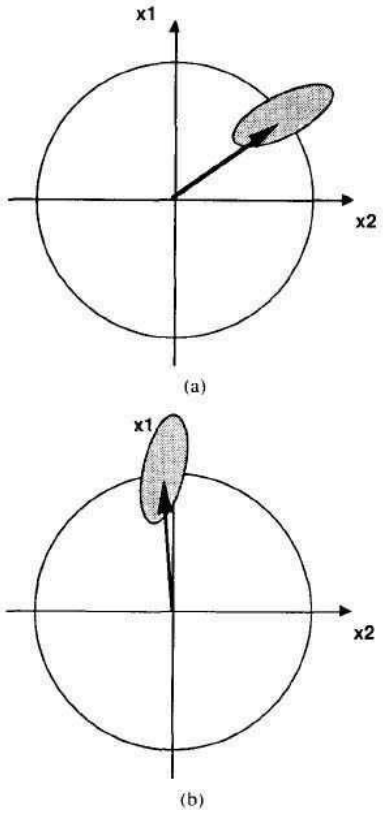
The Figure Nº2 explains how this technique works. It shows a signal vector representation by a particular choice of a basis. The radius of the sphere shown is equal of the norm of the time domain signal, and the error ellipse corresponds to the perceptual semi norm calculated by the psychoacoustic model. 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 byrotation of the vector representation of the signal via a unitary wavelettransformation.

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.

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

$$R(\theta) = \sum R_k(\theta) k=1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ldots(4)$$

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



The best wavelet is then identified by minimizing the following over all vectors:

$$R_{min}(\theta) = 1/2 \sum_{k} \log_2 \frac{(x^q(\theta))^2 W_{kk}(\theta)}{C} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

where Wkk comes from the matrix W and C is a arbitrary constant.

Thus, this wavelet based encoding method essentially involves an optimization over all wavelets of a gives 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 large number of vanishingmoments.
- 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 large number of vanishingmoments.

2.5 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 audioframes.

This dictionary work 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.

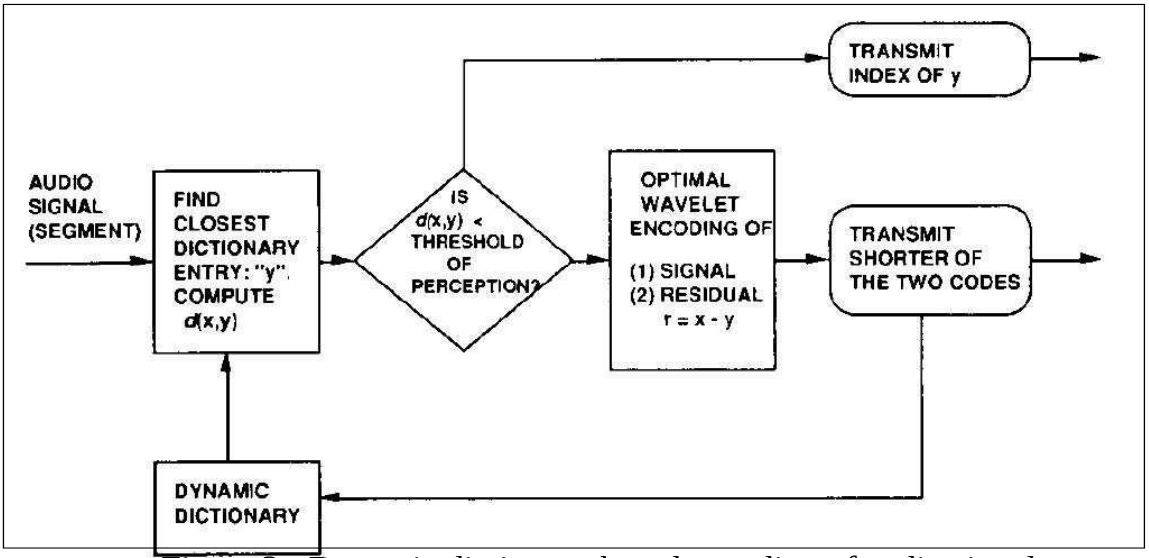


Figure 3.- Dynamic dictionary-based encoding of audio signals

2.6 ImplementationIssues

- The technique described in this paper requires a long coding delay. Decoding on the other hand can be accomplished in realtime.
- 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 number 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) leads the best result with the current design of thealgorithm.
- Side information requires just a few bits perframe.
- Wavelet transform coefficients may still contain redundancies which can be exploited using an entropy coding method.

2.7 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 followingtables.

Table 1.- Subjective listening test results: Transparency test

Music Sample	Average probability of original music preferredover WTCencode Music	Sample Size [Nº of people]	Comments
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

Music Sample	Average probability of original music preferredover WTC encode music	Sample Size [Nº of people]	Comments
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.

3. WAVELET PACKET BASED APPROACH

This chaptersummarizes the approach to audio compression using wavelet packet "High Quality Audio Compression using an Adaptive Wavelet Packet Decomposition and Psychoacoustic Modeling", by Pramila Srinivasan and Leah H. Jamieson.

This section 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.

3.1 General Picture

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:1.Design a sub band structure for wavelet representation of audio signals. It also determines the computational complexity of the algorithm for each frame;

2.Design a scheme for efficient bit allocation, which depends on the temporal resolution of thedecomposition.

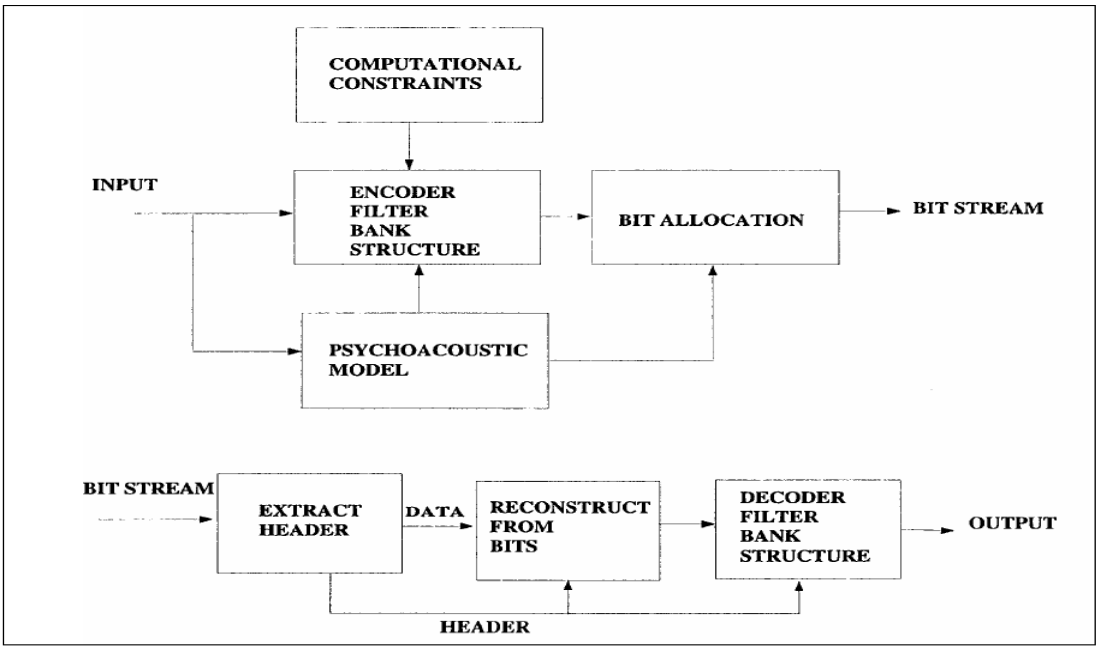


Figure 4.- Block diagram of the described encoder/decoder

3.2 Psychoacoustic Model

3.2.1 Sub band 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.

3.2.2 Masking constraint in 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 onesubband.

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)

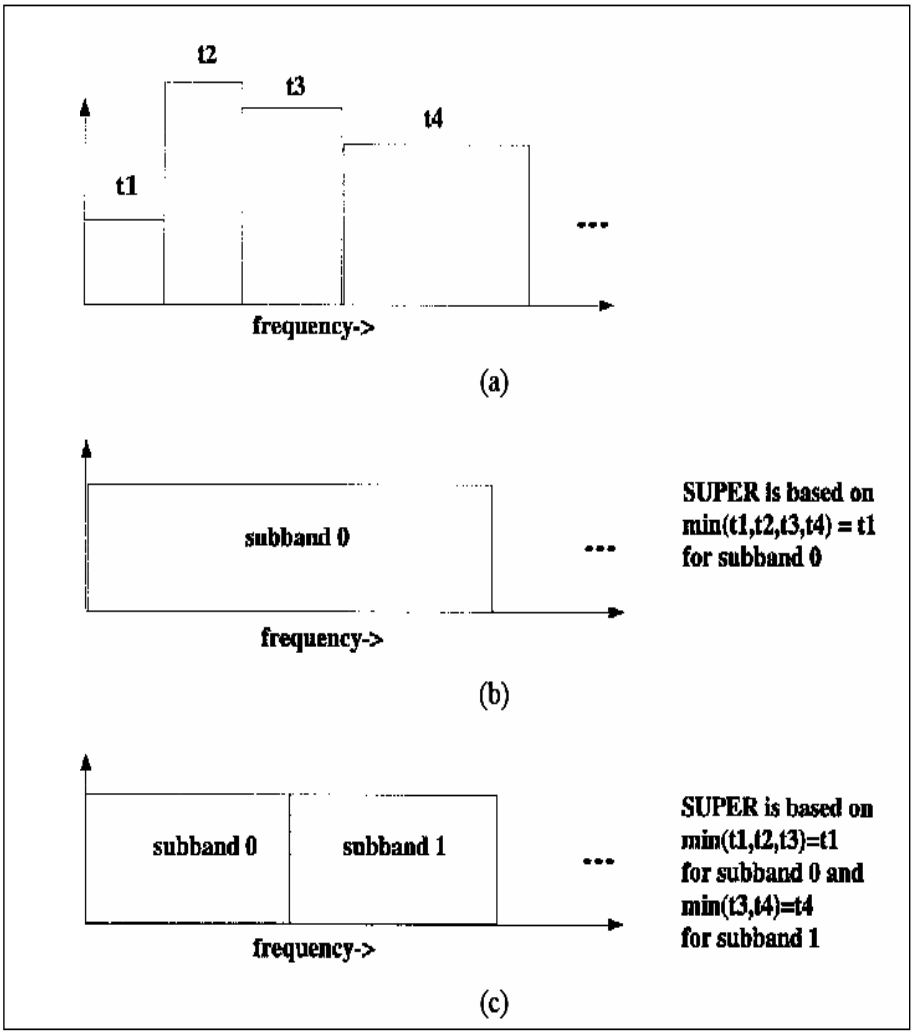


Figure5.-CalculationofSUPERbasedonthesubbandstructure.(a)Thresholdfrothe criticalbands,(b)Possiblesubband(c)Subbandafterdecomposition

3.3 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. 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.

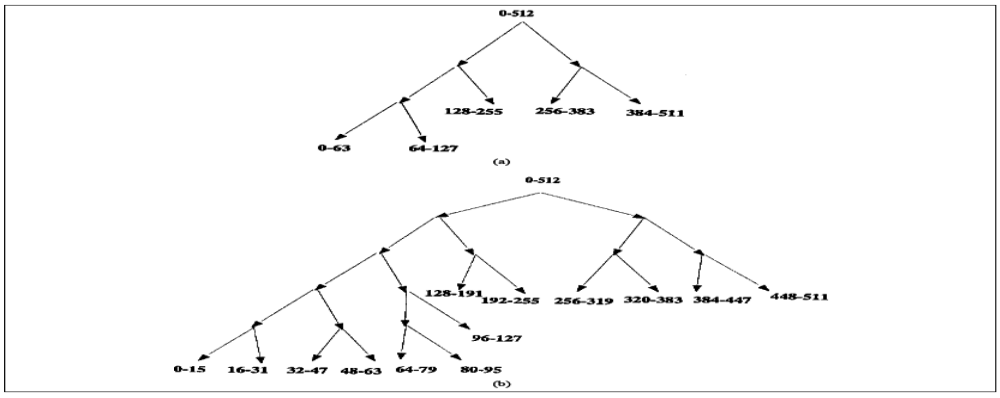


Figure6.-Exampleofadaptationofthetreefora(a)low-complexityand(b)high-complexity decoder for the example in the previous figure

3.4 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 thisreport.

3.5 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-treealgorithm.
- After the zero-tree algorithm, a lossless compression technique is used. No details are mentioned about this finalcompression.
- All results presented in this paper use filter banks that implement the spline-based biorthogonal wavelet transform (order 5 was considered theoptimum).
- 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 theencoder.

3.6 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.

Finally, the authors claim thatperceptually 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 internetapplications.

Table 3.- Subjective listening test results.

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

4. LITERATURE SURVEY

This session 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 multitrackrecording.

As observed in approach, 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 audiblesometimes.

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 signal

5. SOME COMMENTS ON THE PAPERS

Even though these two approaches 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 the approaches are completely different. It is possible to compare a few ideas observed in the summaries:

- First approach uses a discrete wavelet decomposition of the audio signal and second approach uses a wavelet packet approach. Therefore, second approach is based in the frequency domain behavior and first one performs most of the steps in timedomain.
- The psychoacoustic model defined in approach-1 is simpler (does not consider the behavior of previous frames) than the one in approach-2, and it is designed in time domain instead of frequency domain, to be congruent with the previouspoint.
- Exploiting the previous idea, tree structured filter banks yield better descriptions of the psychoacousticmodel.
- The efficiency of the bit allocation in approach-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, approach-2 uses a known algorithm (zero- tree algorithm) to perform the bitallocation.
- It is notable that approach-1 presents more insights when describing the approaches and considerations of the algorithm. On the other hand, approach-2 uses most of its ideas from other authors, it only describes some minordetails.
- The final evaluation is best presented in approach-1, because it considers more test subjects and shows more details anddiscussion.

6. MATLAB SIMULATIONS

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, it is more convenient to design this implementation based on the features described in the discrete wavelet transform approach.

A Graphical User Interface is designed in this project to display the output effectively to the user. A user interface (UI) is a graphical display in one or more windows containing controls, called components, that enable a user to perform interactive tasks. The user does not have to create a script or type commands at the command line to accomplish the tasks. UI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders—just to name a few. UIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other UIs, and display data as tables or as plots.

The UI contains these components:

• An axes component

• A pop-up menu listing three data sets that correspond to MATLAB functions: peaks, membrane, and sinc

• A static text component to label the pop-up menu

• Three buttons that provide different kinds of plots: surface, mesh, and contour When you click a push button, the axes component displays the selected data set using the specified type of 3-D plot.Steps to create simple UI using MATLAB GUIDE are

1.Open a New UI in the GUIDE Layout Editor

- Start GUIDE by typing guide at the MATLAB prompt.
- In the GUIDE Quick Start dialog box, select the Blank GUI (Default) template, and then click OK.
- Display the names of the UI components in the component palette:

  a) Select File > Preferences > GUIDE.

  b) Select Show names in component palette.

  c) Click OK.

2. Set the Window Size in GUIDE

Set the size of the UI window by resizing the grid area in the Layout Editor. Click the lower-right corner and drag it until the canvas is approximately 3 inches high and 4 inches wide. If necessary, make the canvas larger.

3. Layout the Simple GUIDE UI

Add, align, and label the components in the UI.

- Add the two push buttons to the UI. Select the push button tool from the component palette at the left side of the Layout Editor and drag it into the layout area. Create two buttons, positioning them approximately as shown in the following figure.
- Add the remaining components to the UI.
    - A static text area
    - A pop-up menu
    - An axes Arrange the components as shown in the output picture. Resize the axes component to approximately 2-by-2 inches

Label the push buttons

Each of the two push buttons specifies a plot type: surf, mesh, and contour. This topic shows you how to label the buttons with those options.

- Select View > Property Inspector.
- In the layout area, click the top push button.
- In the Property Inspector, select the String property, and then replace the existing value with the word Surf.
- Click outside the String field. The push button label changes to our desired name. Click each of the remaining push buttons in turn and repeat steps 3 and 4.

4.Modify the Static Text

In order to change the static text to read Select Data.

- In the layout area, click the static text.
- In the Property Inspector, click the button next to String. In the String dialog box that displays, replace the existing text with the phrase Select Data.
- Click OK.

5. Save the UI Layout

When you save a layout, GUIDE creates two files, a FIG-file and a code file. The FIG-file, with extension.fig, is a binary file that contains a description of the layout. The code file, with extension .m, contains MATLAB functions that control the UI behavior.

- Save and run your program by selecting Tools > Run.
- GUIDE displays a dialog box displaying: “Activating will save changes to your figure file and MATLAB code. Do you wish to continue? Click Yes.
- GUIDE opens a Save As dialog box in your current folder and prompts you for a FIG-file name.
- Browse to any folder for which you have write privileges, and then enter the file name Audio compression for the FIG-file. GUIDE saves both the FIG-file and the code file using this name.
- If the folder in which you save the files is not on the MATLAB path, GUIDE opens a dialog to allows you to change the current folder.
- GUIDE saves the files Audio compression.fig and Audio compression.m, and then runs the program. It also opens the code file in your default editor.

  The UI opens in a new window. Notice that the UI lacks the standard menu bar and toolbar that MATLAB figure windows display. You can add your own menus and toolbar buttons with GUIDE, but by default a GUIDE UI includes none of these components.

  When you run Audio compression, you can select a data set in the pop-up menu and click the push buttons, but nothing happens. This is because the code file contains no statements to service the pop-up menu and the buttons.

  To run a program created with GUIDE without opening GUIDE, execute its code file by typing its name. Audio compression You can also use the run command with the code file, for example, run Audio compression.

6.1 Main features of the implementation

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

(a) Signal division and processing using smallframes

(b) Discrete wavelet decomposition of eachframe

(c) Compression in the waveletdomain

(d) A psychoacousticmodel

(e) Non linear quantization over the wavelet coefficient using the psychoacoustic model

(f) Signalreconstruction

(g) Main output: Audiofile

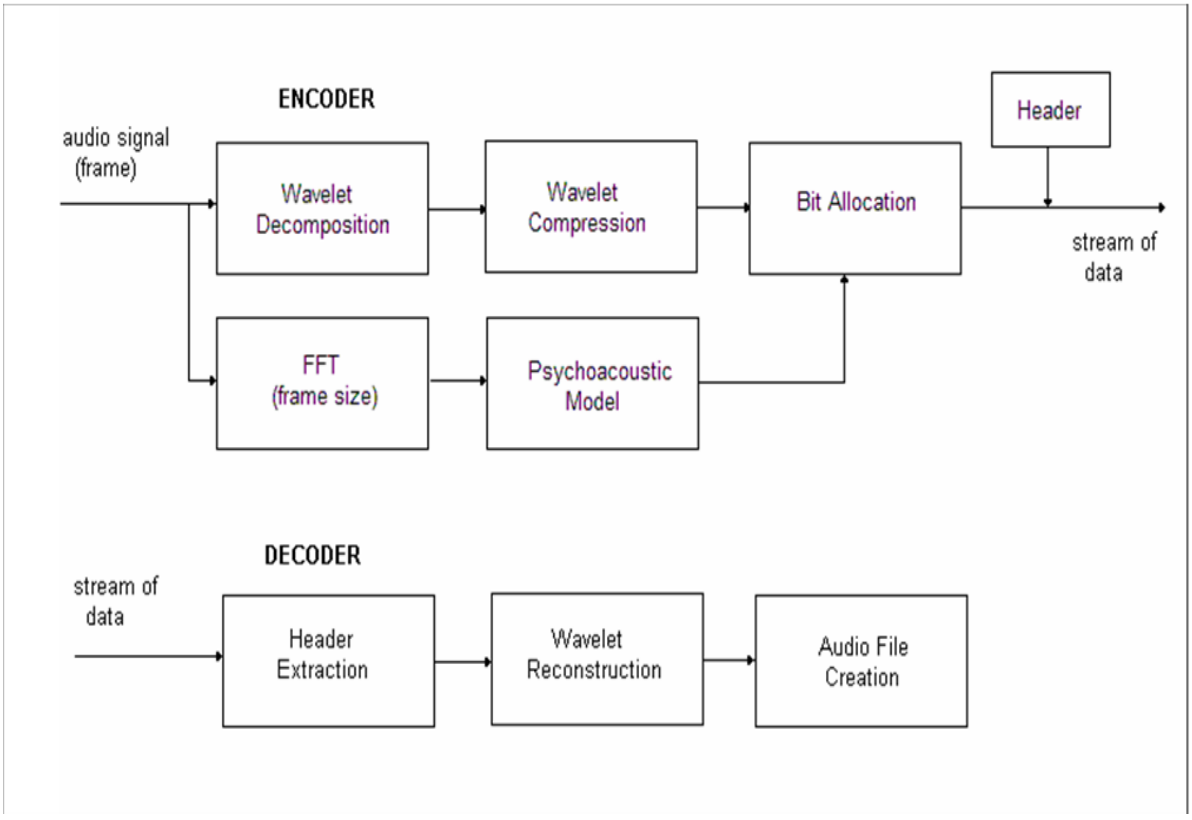


Fig:7 Block diagram of Matlab implementation

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. Using the recommendations of that paper, the best results were observed when using a Daubechieswavelet with 10 vanishing moments (dB10 in MATLAB) and 5 levels of decomposition. These choices will overcome the lack of an optimal basissearch.

(b) Non overlapping frames areincluded

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 perframe.

(c)The psychoacoustic model issimplified

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 DWT 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.

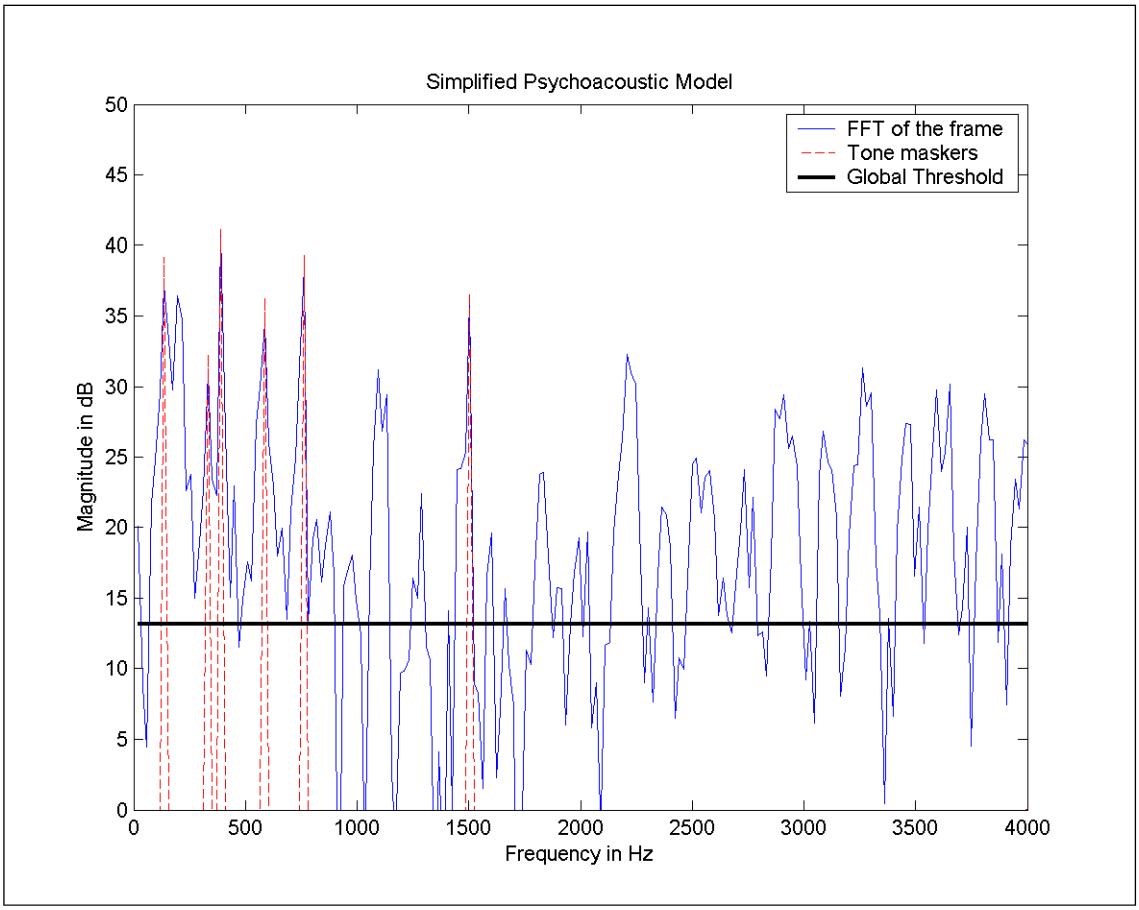


Figure 8.- Tone masker detection in a frame. Matlab implementation

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 wavwritecommand was used to create the audio files (.wav). The compression ratio for each case is calculated using other variables of the simulation.

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 do 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 worse 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 , the quality is considerable better. For auditory test please refer to the CD attached with thisreport.

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.

Music Style / Instrument	Comments
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

Table 4.- Subjective listening test results.

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

	Current scheme		With the lossless compression**	
Average Number ofbits usedper Sampl e	Bit Rate	Compression Ratio	Bit Rate	Compression Ratio
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 fewtests

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. MATLAB CODE & OUTPUTS

%%%

%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.

%%%

```
functionvarargout = audcomp(varargin)
%AUDCOMP MATLAB code file for audcomp.fig
%      AUDCOMP, by itself, creates a new AUDCOMP or raises the existing
%      singleton*.
%
%      H = AUDCOMP returns the handle to a new AUDCOMP or the handle to
%      the existing singleton*.
%
%      AUDCOMP('Property','Value',...) creates a new AUDCOMP using the
%      given property value pairs. Unrecognized properties are passed via
%      varargin to audcomp_OpeningFcn.  This calling syntax produces a
%      warning when there is an existing singleton*.
%      AUDCOMP('CALLBACK') and AUDCOMP('CALLBACK',hObject,...) call the
%      local function named CALLBACK in AUDCOMP.M with the given input
%      arguments.
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help audcomp
% Last Modified by GUIDE v2.5 23-Jan-2021 15:49:50
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @audcomp_OpeningFcn, ...
'gui_OutputFcn',  @audcomp_OutputFcn, ...
'gui_LayoutFcn',  [], ...
'gui_Callback',   []);
ifnargin&&ischar(varargin{1})
```

```
gui_State.gui_Callback = str2func(varargin{1});
end

ifnargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before audcomp is made visible.
functionaudcomp_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from the
%            command line (see VARARGIN)
% Choose default command line output for audcomp
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes audcomp wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
functionvarargout = audcomp_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
globalfile_name;
%guidata(hObject,handles)
file_name=uigetfile({'*.wav'},'Select an Audio File');
fileinfo = dir(file_name);
SIZE = fileinfo.bytes;
Size = SIZE/1024;
[x,Fs] = audioread(file_name);
xlen=length(x);
t=0:1/Fs:(length(x)-1)/Fs;
```

```
set(handles.text1,'string',Size);
%plot(t,x);
axes(handles.axes1) % Select the proper axes
plot(t,x)
set(handles.axes1,'XMinorTick','on')
grid on
function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double


% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in edit1.
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
globalfile_name;
%filename=char(filename)
if(~ischar(file_name))
errordlg('Please select Audio first');
else
[x,Fs] =audioread(file_name);
xlen=length(x);
t=0:1/Fs:(length(x)-1)/Fs;
wavelet='haar';
level=5;
frame_size=2048;
psychoacoustic='on '; %if it is off it uses 8 bits/frame as default
wavelet_compression = 'on ';
```

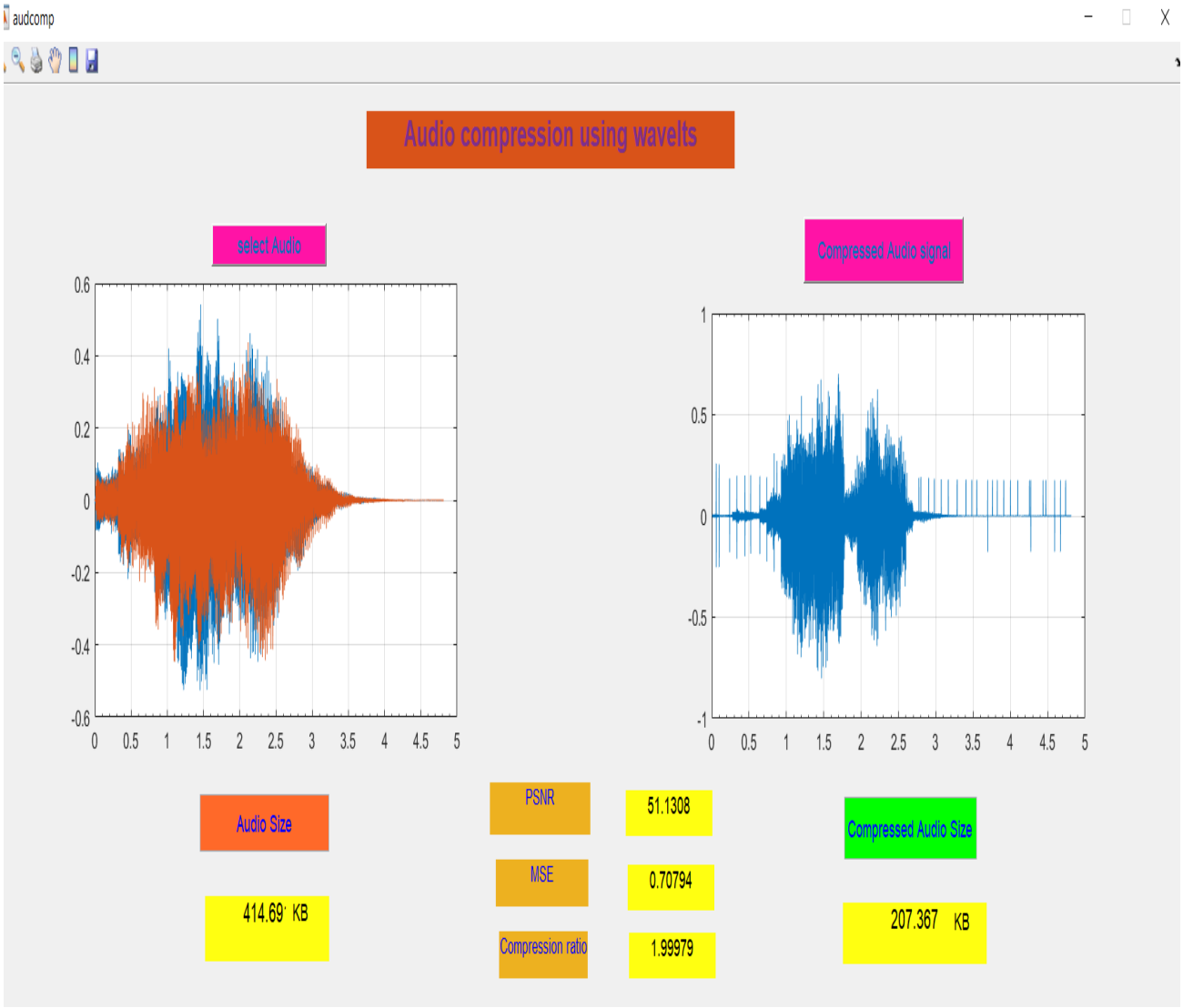
```
heavy_compression='off';
compander='on ';
quantization ='on ';
% ENCODER
step=frame_size;
N=ceil(xlen/step);
%computational variables
Cchunks=0;
Lchunks=0;
Csize=0;
PERF0mean=0;
PERFL2mean=0;
n_avg=0;
n_max=0;
n_0=0;
n_vector=[];
fori=1:1:N
if (i==N);
frame=x([(step*(i-1)+1):length(x)])
else
frame=x([(step*(i-1)+1):step*i]);
end
%wavelet decomposition of the frame
[C,L] = wavedec(frame,level,wavelet);
%wavelet compression scheme
ifwavelet_compression=='on '
[thr,sorh,keepapp] = ddencmp('cmp','wv',frame);
ifheavy_compression == 'on '
thr=thr*10^6;
end
[XC,CXC,LXC,PERF0,PERFL2] = wdencmp('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
if bool==1
Ptm(k)=10*log10(10.^(0.1.*(P(k-1)))+10.^(0.1.*(P(k)))+10.^(0.1.*P(k+1)));
end
end
sum_energy=0;
for k=1:1:length(Ptm)
sum_energy=10.^(0.1.*(Ptm(k)))+sum_energy;
end
E=10*log10(sum_energy/(length(Ptm)));
SNR=max(P)-E;
n=ceil(SNR/6.02);
if n<=3
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)
ifcompander=='on '
Mu=255;
C = compand(C,Mu,max(C),'mu/compressor');
end
%Quantization
if quantization=='on '
if psychoacoustic=='off'
n=8;
end
partition = [min(C):((max(C)-min(C))/2^n):max(C)];
codebook = [1 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];
Lchunks=[Lchunks L];
Csize=[Csize length(C)];
Encoder = round((i/N)*100); %indicator of progess
end
Cchunks=Cchunks(2:length(Cchunks));
%wavwrite(Cchunks,Fs,bits,'output1.wav')
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';
xdchunks=0;
fori=1:1:N;
ifi==N;
Cframe=Cchunks([((Csize(1)*(i-1))+1):Csize(2)+(Csize(1)*(i-1))]);
%Compander (expander)
ifcompander=='on '
ifmax(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)
ifcompander=='on '
ifmax(Cframe)==0
else
Cframe = compand(Cframe,Mu,max(Cframe),'mu/expander');
end
end
xd = waverec(Cframe,Lchunks(1:Lsize),wavelet);
end
xdchunks=[xdchunksxd];
Decoder = round((i/N)*100); %indicator of progess
end
xdchunks=xdchunks(2:length(xdchunks));
%distorsion = sum((xdchunks-x').^2)/length(x)
end_of_decoder='done';
%creating audio files with compressed schemes
audiowrite('output1.wav',xdchunks,Fs);
```

```
end_of_writing_file='done';%indicator of progess;
[x,Fs] = audioread('output1.wav');
fileinfo = dir('output1.wav');
SIZE = fileinfo.bytes;
Size = SIZE/1024;
set(handles.text3,'string',Size)
xlen=length(x);
t=0:1/Fs:(length(x)-1)/Fs;
axes(handles.axes2) % Select the proper axes
plot(t,xdchunks)
set(handles.axes2,'XMinorTick','on')
grid on

[y1,fs1]=audioread(file_name);
[y2,fs2]=audioread('output1.wav');
[c1x,c1y]=size(y1);
[c2x,c2y]=size(y1);
if c1x ~= c2x
disp('dimeonsions do not agree');
else
 R=c1x;
 C=c1y;
  err = (sum(y1(2)-y2).^2)/(R*C);
 MSE=sqrt(err);
 MAXVAL=255;
  PSNR = 20*log10(MAXVAL/MSE);
  MSE= num2str(MSE);
if(MSE > 0)
  PSNR= num2str(PSNR);
else
PSNR = 99;
end
fileinfo = dir(file_name);
SIZE = fileinfo.bytes;
Size = SIZE/1024;
fileinfo1 = dir('output1.wav');
SIZE1 = fileinfo1.bytes;
Size1 = SIZE1/1024;

CompressionRatio = Size/Size1;

  set(handles.text9,'string',PSNR)
  set(handles.text10,'string',MSE)
  set(handles.text11,'string',CompressionRatio)

end
end
```

OUTPUT:


audcomp
Audio compression using wavelts
select Audio
Compressed Audio signal
Audio Size
414.69 KB
PSNR
51.1308
MSE
0.70794
Compression ratio
1.99979
Compressed Audio Size
207.367 KB

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.
- 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” thesignal.
- Further upgrades can be considered to the MATLAB implementation to obtain betterresults.

9.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, December1993.

[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, April1998.

[3] J.I. Agbinya, “Discrete Wavelet Transform Techniques in Speech Processing”, IEEE Tencon Digital Signal Processing Applications Proceedings, IEEE, New York, NY, 1996, pp514-519.

[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. ISBN0-12-466606-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] 2http://perso.wanadoo.fr/polyvalens/clemens/wavelets/wavelets.html,April 24, 2005.

[13] http://www.mp3developments.com/article4.php, April