Advanced Audio Compression for Multimedia

Lars Rüdiger[#], Ana Maria Tomé, João Manuel Rodrigues
[#]Fachhochschule Kiel, Germany

***Resumo* – Este artigo descreve a implementação de um sistema de codificação que integra aspectos psico-acústicos do ouvido humano para conseguir compressão dos dados áudio mantendo a qualidade do sinal áudio após a reconstrução.**

Abstract – This paper reports the implementation of a coding system that includes the psychoacoustic aspects of the human ear to achieve compression of audio data while keeping the quality of the audio signal after the reconstruction.

I. INTRODUCTION[*]

During the enormous progress in multimedia it became more and more important to find convenient algorithms for the compression of audio signals of high quality. Those are needed for transmission as well as for storage of multimedia sequences. In recent years, it became common to design audio coding systems according to the psychoacoustic properties of the human hearing system. Traditional distortion measures like the signal-to-noise ratio (SNR) or the mean square error (MSE) are not adequate measures for the perceived quality of an audio signal, because auditory perception depends largely on the frequency contents of the signal. Experiments during recent years show that the same amount of noise injected to different bands causes more or less audible effects. This leads to the analysis and encoding of spectral components instead of the time domain coding. One of the most used coding systems that takes advantage of these psychoacoustical properties is the ISO/MPEG audio standard [1].

The aim of this project is to implement a coding and decoding system in the programming language C that uses a psychoacoustical model to estimate the introduceable amount of quantization noise. The coder should be able to take audio information from files in wave format and write the quantized samples to an output file. The decoder side should of course be able to decode this file and write the data to a WAVE file in order to reproduce the audio data. The whole coding and decoding system should not leave a disturbing noise, so that the perceived quality after decoding stays almost the same in spite of bit rate reduction.

The structure of the coder and decoder

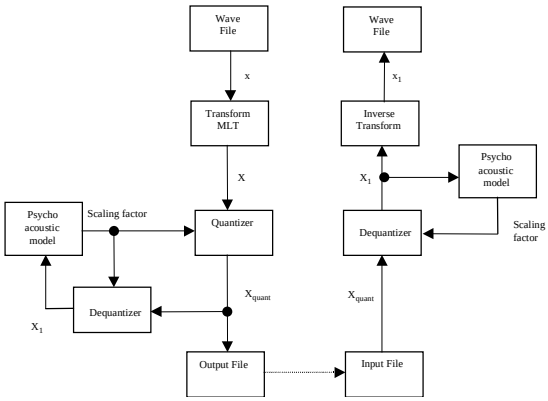

Fig. 1: Structure of the coder/decoder

Figure 1 shows the block diagram of the complete coder and decoder structure, where x is the input stream of PCM samples that is read in blocks of M samples from a file in the WAVE format. Other information besides the samples that is needed for the process like, for instance, the sampling rate is taken from the WAVE file header.

The basic idea of this coding system is to transmit the signal split in its frequencies. With the modulated lapped transform (MLT) the time signal is transformed into the frequency domain. Since an audio signal is a non-stationary random signal, the spectrum varies in time, so it should be computed in short periods of M samples. Those spectrum fragments are the so-called short-time spectrum of the signal and their time and frequency resolution depend on the dimension of the transform (M) as well as on the sampling rate (fs).

Prior to storage, the frequency domain samples are quantized so that only a limited number of symbols are used to represent the whole range of possible values. This quantization is non-uniform with a quasi-logarithmic compression characteristic.

In order to reach high quality compression, it is necessary to control the amount of noise that is introduced in the signal by the quantization. In the quantizer this is done by a scaling factor which changes the scale of the compression curve according to the psychoacoustical properties of the ear and the current state of the signal. The computation of the scaling factor is done by the psychoacoustical model, which estimates the scaling factor for the next block from the current signal power distribution in the frequency domain.

[*] Work supported by the Erasmus scholarship program.

A distinctive feature in this coding system is that the scaling factor is computed backwards [5]. That means the power is computed from the dequantized signal (X1). The advantage of this is the reproducibility of the factor on the decoder side without transmitting any side information for the inverse quantizer. The quantized samples (Xquant) are written to a formatted output file together with a small header containing a few coding parameters.

The decoder takes the samples and dequantizes them. This results in exactly the same signal (X1) that was used in the coder in order to find the signal power distribution. The same is done in the dequantizer by using the same psychoacoustic model as in the coder.

After the dequantization, the samples are inverse transformed into the time domain by using the inverse MLT. The resulting signal (x1) is the reconstructed signal, which is written to a WAVE file by using some pieces of information from the header.

II. THEORETICAL FUNDAMENTALS

A. Block Transforms

To get an idea of what the MLT is doing and how it can be implemented the starting point is the Discrete Fourier Transform (DFT). The traditional and most common form of the DFT is:

$$X[k]=\sqrt{\frac{1}{M}}\sum_{n=0}^{M-1}x[n]e^{-jnk\frac{2\pi}{M}} \quad k=0,1,\ldots\hat{M} \qquad (1)$$

Where $\hat{M}=M-1$. $X[k]$ can also be interpreted as the result of a simple matrix multiplication, which can easily be implemented in MATLAB, for instance.

$$X=A^T\cdot x \qquad (2)$$

Where X is the transform of x and A is the transformation matrix. The superscript T denotes the conjugated transposition. Or, equivalently,

$$\begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{\hat{M}} \end{bmatrix}=\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0\hat{M}} \\ a_{10} & a_{11} & \cdots & a_{1\hat{M}} \\ \vdots & \vdots & \vdots & \vdots \\ a_{\hat{M}0} & a_{\hat{M}1} & \cdots & a_{\hat{M}\hat{M}} \end{bmatrix}^T\cdot\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{\hat{M}} \end{bmatrix} \qquad (3)$$

So the DFT can also be defined by the coefficients of the transform matrix A:

$$a_{nk}=\sqrt{\frac{1}{M}}\cdot\exp(j\frac{2\pi kn}{M}) \quad n,k=0,1,\cdots,\hat{M} \qquad (4)$$

The scaling factor $\sqrt{1/M}$ is chosen to keep the orthogonality of A. The inverse transform is given by the inverse of the transform matrix:

$$x=\left[A^T\right]^{-1}\cdot X \qquad (5)$$

Since the matrix A is orthogonal, which means that $A^T=A^{-1}$, the inverse transform is quite easy, because the computation of the inverse matrix turns to a simple transposition. In that case the inverse transform is given by:

$$x=A\cdot X \qquad (6)$$

From a number of M *real* input samples, the DFT produces an equal number of *complex* output samples. That means the computed spectrum contains information about the magnitude as well as the phase of each existing frequency component. The *M* frequency samples are in the interval of 0 to fs, but the spectrum has conjugate symmetry around fs/2. This means that the last M/2 samples have the same amplitude and opposite phase as the first and so they do not contain new information.

Instead of directly using the definitions, there is a more efficient family of algorithms to compute the transform; the so-called Fast Fourier transforms (FFT). These algorithms take advantage of the discrete Fourier transform properties to reduce the number of multiplications to be executed. The Fourier transform is one of the most important transforms in signal processing, so that FFT algorithms are available in signal processing libraries for a number of different platforms.

A second and special kind of block transform is the Discrete Cosine Transform (DCT). The main difference between the DFT and the DCT is that the DCT is a real transform. For *M* real input samples it produces *M* different real frequency coefficients in the interval of [0...fs/2]. Recall that the DFT returns just *M*/2 different spectral samples in the same interval, because of the symmetry around fs/2. The following Figure 2 shows the absolute value of the DFT and the DCT of a 32-sample rect pulse.

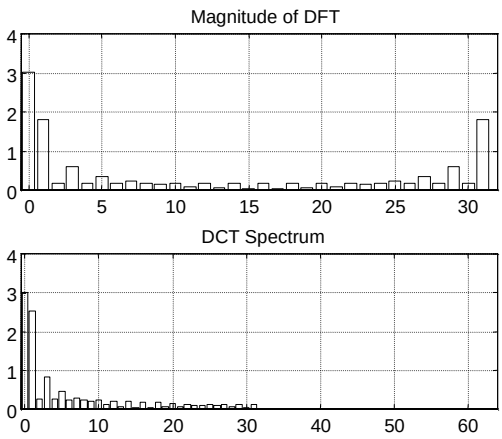

Fig. 2: DFT versus DCT of a rect pulse.

This means that the spectral resolution of the DCT is twice as high as that of the DFT, but the spectrum contains no phase interpretation. This can lead to problems. If, for instance, the input signal has a strong component with a 90 degrees phase shift to a DCT basis function the corresponding DCT coefficient will be zero, even if that spectral component is a significant part of the input signal.

Hence, the transform of just one block can easily lead to mistakes in the interpretation of the spectrum. Only an average of some blocks leads to a good interpretation of the DCT output.

There are, in fact, several types of DCT. We are particularly interested in the DCT Type-IV (DCT-IV) which is used as a building block in the fast MLT algorithm described below.

1. Fast Algorithm for the DCT-IV

A fast algorithm for the DCT-IV, which takes advantage of the FFT properties, is described in [4]. The advantage of this implementation is the usage of an *M*/2-point complex FFT for each block of *M* real input samples.

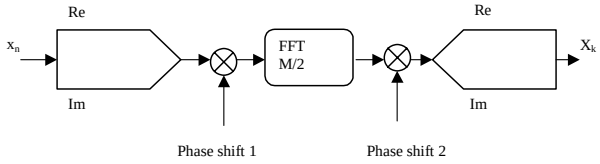

Fig. 3: DCT-IV

The first step is to interpret the sequence of real input samples as a vector of *M*/2 complex values:

$$Z_n = x_{2n} + jx_{M-1-2n} \quad n = 0,1,\cdots,M/2-1 \qquad (7)$$

Now the complex values have to be phase shifted (Phase shift 1) according to:

$$Z_n \leftarrow Z_n \cdot \exp\left(-j(n+0.25)\frac{\pi}{M}\right) \quad n = 0,1,\cdots,\frac{M}{2}-1 \qquad (8)$$

Z is now passed to the *M*/2-point complex FFT

$$Z \leftarrow \sqrt{\frac{2}{M}} \cdot FFT(Z) \qquad (9)$$

The factor of $\sqrt{2/M}$ is chosen for normalization. The result is again a complex structure, which has to be shifted in phase (Phase shift 2).

$$Z_n \leftarrow Z_n \cdot \exp\left(\frac{-j(n\pi)}{M}\right) \qquad (10)$$

Finally the result is converted back into a number of *M* real samples, which are the result of the DCT-IV.

$$\begin{cases} X_{2n} \leftarrow \mathrm{Re}(Z_n) \\ X_{M-1-2n} \leftarrow \mathrm{Im}(Z_n) \end{cases} \quad n = 0,1,\cdots,M/2-1 \qquad (11)$$

It is important to mention that the DCT-IV transform is its own inverse.

B. Lapped Transforms

If a signal is transformed block by block, no error will occur after the inverse transform. However, if quantization of the frequency domain samples is done, quantization errors will occur after the inverse transform and these will be most noticeable near the edges of each block. These disturbances are the so-called “blocking effects”. Those happen because of the independent processing of each block: the encoded last samples of one block will not match the first samples of the following and will result in audible transitions.

To reduce these blocking effects the input blocks can be overlapped, for instance by 50%. The spectrum is in fact less disturbed, but the number of output samples will be twice as high as the number of input samples. This problem led to the development of lapped transforms. The modulated lapped transform, for instance, returns the same number of output samples as of input samples, even though the windows are overlapped by 50%.

During this project a fast algorithm of the MLT is used which was introduced by [4]. This MLT uses a special block transform (DCT-IV) and a special way of windowing and overlapping.

1. Fast Algorithm for the MLT

An algorithm for a modulated lapped transform with low computational complexity is one using the DCT-IV described in [4].

To achieve perfect reconstruction, a window function is needed. This windowing is implemented by a so-called butterfly structure. Each butterfly implements a multiplication by a matrix:

$$\begin{bmatrix} x'_i \\ x'_{M-1-i} \end{bmatrix} = \begin{bmatrix} -\cos(\Theta_i) & \sin(\Theta_i) \\ \sin(\Theta_i) & \cos(\Theta_i) \end{bmatrix} \begin{bmatrix} x_i \\ x_{M-1-i} \end{bmatrix} \qquad (12)$$

For the butterfly structure, M/2 of those butterflies are needed. Hence, M/2 butterfly angles are also needed. They are computed, as required, in the half-sine window:

$$\Theta_i = (M-1-i+0.5)\left(\frac{\pi}{2M}\right) \qquad (13)$$

Where *i* = 0, 1,...., *M*/2-1. The block diagrams in Figure 4 and Figure 5 show the structure of the MLT and the inverse MLT completely. Besides the blocks of the decimator, the butterfly and the DCT-IV, the samples have to be swapped. While swapping the samples, M/2 of the samples are stored for the next block and the samples from the previous block have to be inserted. This implements the overlapping in the lapped transform.

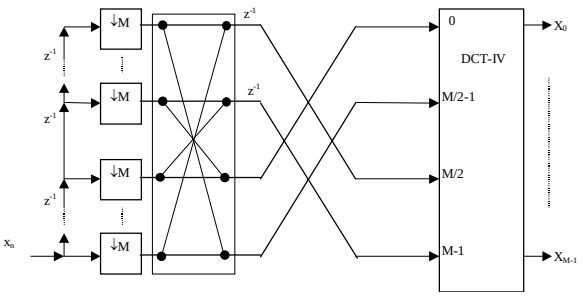

Fig. 4: The MLT

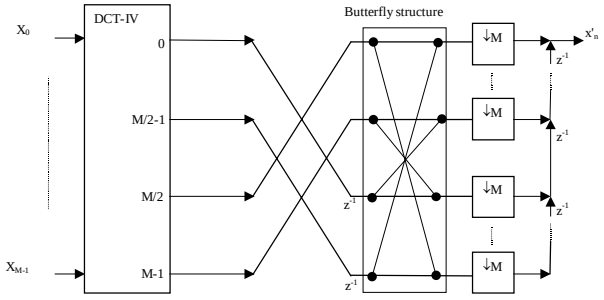

Fig. 5: The IMLT

C. The Psychoacoustical Model

During the development of audio coders, some properties of the human auditory system were found and advantage of those was taken to hide introduced quantization noise. It was discovered that quite some information was coded even if it is not perceptible by the human ear. This led to a model of human hearing, which uses so-called *critical bands* to analyze the wide band audio signals. The aim of this psychoacoustical model is to split the signal into these critical bands in order to find the amount of noise that can be introduced by quantization and coding in each of those bands, and so to adapt the coding process to the human hearing properties. Other important aspects of the psychoacoustical model are the so-called *masking effect* and the *absolute threshold* of hearing. These items of psychoacoustics will be discussed below and their use in coding will be described as well.

1. The Absolute Threshold

The absolute threshold denotes the threshold of human perceptibility of pure tones. Tones below this threshold, shown in Figure 6, are not perceptible; therefore it is not necessary to encode them. Instead of that, all sound signals under or around this threshold are suppressed or coarsely quantized.

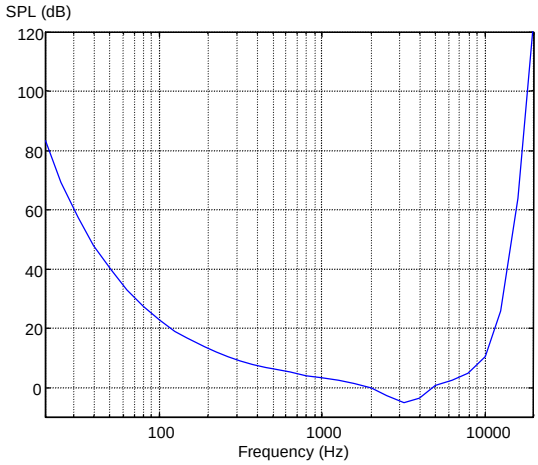

Fig. 6: The absolute threshold

2. Critical bands

Through various experiments it was found that the human ear separates signals into its frequencies by several processes for further processing. The ear uses a kind of special scale for that, which is revealed by the so-called critical bands which were measured by hearing experiments. Those kinds of tests done with many different volunteers gave the idea for a model of how the human ear works.

One such experiment was summarized in [3] as follows:

- A tone was presented at a fixed level, well above the absolute threshold.
- A narrow band noise source was injected into the sine wave, with the center of the noise band at the tone frequency.
- The energy of the noise was adjusted until it reached the Just Noticeable Difference (JND), i.e.: the level at which the noise was perceptible.
- The noise source was then adjusted to a slightly wider bandwidth by moving either the lower or upper cut-off a small distance away from the tone.
- The level test was repeated.

The result of these tests was that the JND power of the noise remained almost constant as long as the width remained within a certain frequency band around each tone. Those frequency bands are called the *critical bands* and their width grows nonlinearly with the center frequency. To reflect this nonlinearity a new frequency scale was devised in which an interval of one unit would always represent a critical band. This unit is called the *Bark* and the scale is known as the Bark scale [7].

3. The Masking Effect

When a strong signal is present, the audibility of a weaker signal or noise is reduced or even eliminated. This hiding of noise under other sound events is the so-called *Masking Effect*. Masking depends on the time domain as well as the frequency domain proximity of the signal and noise. Specifically, a short time before and after a sound burst, other events with lower intensity are not perceptible. Similarly, strong frequency components preferentially mask nearby noise bands. These effects effectively raise the perceptibility threshold above the absolute threshold of hearing. The aim of the psychoacoustical model in the coder is to compute this global masking threshold.

4. Computation of the Global Masking Threshold

As discovered by psychoacoustical research, the amount of maskable noise power depends foremost on the power distribution of the signal in the Bark domain. Hence, the first step to compute the masking threshold is to find out the power of the signal in each Bark band. Then, time averaging of the power in each band is computed by a first-order IIR low-pass filter that simultaneously models the forward time masking effect.

To compute the frequency masking effect on a certain band, it is necessary to consider the influence of all the other bands and this is modeled by a convolution of the Bark spectrum with an empirically-derived *spreading function* given by (see e.g. [7, p.257]):

$$10\cdot\log_{10}[S(\Delta i)] = 15.81 + 7.5(\Delta i + 0.474) - 17.5\left[1 + (\Delta i + 0.474)^2\right]^{\frac{1}{2}} \quad (14)$$

Where the distance between two critical bands in Bark is denoted as Δi. To find the actual threshold $B_m(i)$ in a given Bark band in consideration of the influence of all the other bands, the $S(\Delta i)$ must first be computed for every $\Delta i=i-j$, then it must be multiplied with the band power $B(j)$ and at last, it has to be summed up. This operation can be expressed by a matrix multiplication, as shown below:

$$\begin{bmatrix} B_m(1) \\ B_m(2) \\ \vdots \\ B_m(25) \end{bmatrix} = \begin{bmatrix} S(0) & S(-1) & S(-2) & \cdots & S(-24) \\ S(1) & S(0) & S(-1) & \cdots & S(-23) \\ \vdots & \vdots & \vdots & & \vdots \\ S(24) & S(23) & S(22) & \cdots & S(0) \end{bmatrix} \cdot \begin{bmatrix} B(1) \\ B(2) \\ \vdots \\ B(25) \end{bmatrix} \quad (15)$$

Before the computation of the global masking threshold, the offset between the signal level and the masking threshold has to be calculated. This offset grows with the Bark frequency and depends on the tonal or noisy character of the signal.

$$\frac{O(i)}{dB} = \Phi(14.5 + i) + (1 - \Phi)5.5 \quad (16)$$

The tonality index Φ represents the type of the signal. It can be between Φ=1 for a tone-like signal masking noise and Φ=0 for noise-like signals masking a tone. Φ set to 1 is the worst case (larger offset), and so this should result in the best quality, which means that less noise is injected even if the computed tonality index would allow more. The factor Φ can be computed from the Spectral Flatness Measure (SFM). The SFM is defined as the ratio of the geometric to the arithmetic mean of the power spectrum values [7, p.256].

After the computation of the spreading function and the offset, the global masking threshold can be computed. This is the threshold of noise that can be introduced into the critical band without being audible. The global masking threshold is given by [7, p.258]:

$$T_m(i) = 10^{\log_{10}(B_m(i)) - O(i)/10} \quad (17)$$

Actually, as described in [2], a deconvolution should be made to find the masking threshold. However, this process is unstable and often results in zero thresholds, negative threshold energy, etc. Instead of that, Johnston suggests a renormalization, which is done by the multiplication of T_m by the inverse of the energy gain that results when a uniform energy of 1 is input in each band. This renormalized threshold is called T'_m.

It is important to mention that this threshold denotes the total noise power that can be introduced into a critical band, but the single frequency components in that band have to be individually quantized. Therefore, the noise power that can be injected into each frequency line inside a Bark band must be estimated as T'_m divided by the number of frequency lines in that critical band.

Finally, any values that are below the absolute hearing threshold are set to that absolute threshold, resulting in the final estimate of the global masking threshold (e_m^2).

Figure 7 shows the global masking threshold around a sinusoid with a frequency of 4 kHz. This result is labeled as the threshold of the noise. The spectrum shown is computed by a 128-band MLT and a sampling frequency of 64 kHz, which means that the signal peak has the index 16.

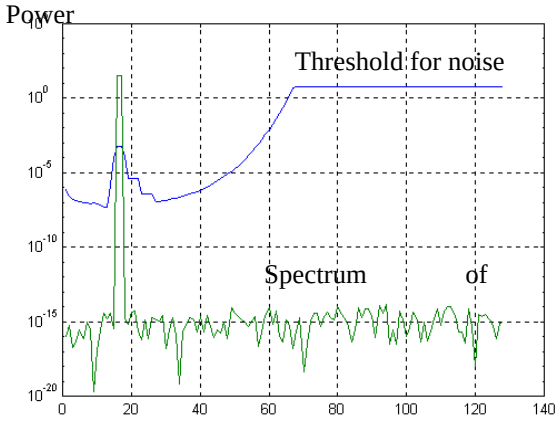

Fig. 7: Masking threshold of a pure tone

D. The Quantizer

The samples of the frequency domain produced by the MLT must be quantized in order to assign them to a limited set of symbols that can later be encoded in several ways before storage in a digital medium. Quantization causes errors, which are also known as quantization noise. It is known that if a uniform quantizer with a step of Δ is used, then the error e_m will be uniformly distributed in the interval $\pm\Delta/2$, and its Root Mean Square (RMS) value is [6, p. 81]:

$$e_{m_{RMS}} = \frac{\Delta}{\sqrt{12}} \quad (18)$$

From this, the noise power is available by simple squaring. Since the amount of introduceable noise is given by the global masking threshold (e_m^2) of the psychoacoustical model, the needed quantizer step size is found by solving (18) with respect to Δ:

$$\Delta = \sqrt{12 \cdot e_m^2} \quad (19)$$

When doing this, the quantizer is dynamically adapted according to perceptibility constraints, and so the introduced noise should not be audible at all.

Recall, however, that instead of a uniform quantizer, we have employed a quasi-logarithmic quantizer to achieve a large dynamic range without an excessive number of levels.

Nevertheless, the noise characteristics of this quantizer were studied [5] with realistic input amplitude distributions and it was found that equations 1 and 2 still apply over the expected operating range of the quantizer.

II. RESULTS

To evaluate the performance of any coding system, the main criteria are the quality of the reconstructed signal and the compression factor. The only reasonable way of measuring the quality of a perceptual audio coder is to subject reconstructed signals to comparative listening tests against the original versions. In this project only very informal preliminary tests were conducted with a few volunteers. The results were quite satisfactory: the quantization noise was not that disturbing and not perceptible at all in some of the test files.

Since no entropy coding was integrated into this system, we evaluated the achievable compression by finding the entropy of the symbols output by the quantizers. Additionally, we decided to use an external program, the Winzip application, to compress the output data of the quantizers. By comparing the sizes of the original and the resulting file, an effective compression factor was computed. Table 1 contains the compression results for various test files. It shows the size of the original WAVE file, the size of the Winzip-compressed quantizer output file (ZIP file), and the estimated entropy of the quantizer output. The values in parenthesis represent the percentages of information reduction in each case.

Original WAVE File (16 bit/sample)	Size of the WAVE file (bytes)	Size of the ZIP file (bytes)	Entropy (bits/sample)
Suzanne.wav	856.108	225.889 (-74%)	2,95 (-82 %)
Violin.wav	851.298	215.587 (-75%)	1,60 (-90%)
Castanets.wav	595.132	154.562 (-74%)	1,64 (-90%)
Harpsichord.wav	464.940	150.565 (-68%)	3,60 (-78%)

Table 1. Compression results.

By directly compressing the original files with Winzip, we found that less than 20% of reduction is achieved. These results show that the implemented coding system significantly reduces the amount of information to store, with almost no loss of signal quality.

III. CONCLUSIONS

During this project a coding system for audio signals was developed, which uses a psychoacoustical model to control the amount of introduced quantization noise. The coding system can be used to compress WAVE files, which are a very common format in multimedia applications. This work shows the principles used in the implementation of the system, which includes the concept of lapped transform, the development of a psychoacoustical model and the quantization of the audio data. Of course coders like MPEG are much more advanced but this project highlights the important points in the different modules that are needed to design an audio compression system. This project started from scratch and so it is possible to improve the quality and the compression factor by future enhancements in the single modules.

The Diploma thesis [8] that describes this work is available in the library of the Universidade de Aveiro. The C-language program source code for the coder and decoder and some audio test files are included in the accompanying CD-ROM.

REFERENCES

[1] Noll, Peter: "MPEG Digital Audio Coding". *IEEE Signal Processing Magazine*, Vol. 14 No. 5, September 1997.

[2] Johnston, James D.: "Transform Coding of Audio Signals Using Perceptual Noise Criteria". *IEEE Journal on Selected Areas in Communications*, Vol. 6 No. 2, February 1988.

[3] Johnston, James D. and Brandenburg, Karlheinz: "Wideband Coding Perceptual Considerations for Speech and Music". In Furui, S. and Sondhi, M. M., editors, *Advances in Speech Signal Processing*, chap. 4. Marcel Dekker, Inc., New York, 1991.

[4] Malvar, Henrique S.: *Signal Processing with Lapped Transforms*. Artech House, Boston, 1992.

[5] Rodrigues, João Manuel: "Compressão Digital de Sinais Áudio Aplicando Critérios Perceptuais e Adaptação para Trás". Master's thesis, Universidade de Aveiro, November 1995.

[6] Stearns, Samuel D. and Hush, Don R.: *Digital Verarbeitung analoger Signale*. München/ Wien 1994.

[7] Zölzer, Udo: *Digital Audio Signal Processing*. John Wiley, Chichester, 1997.

[8] Rüdiger, Lars: "Advanced Audio Compression for Multimedia". Diploma thesis, Universidade de Aveiro, 1998.