Hindawi Publishing Corporation
EURASIP Journal on Audio, Speech, and Music Processing
Volume 2007, Article ID 51563, 14 pages
doi:10.1155/2007/51563

Research Article

Perceptual Coding of Audio Signals Using Adaptive Time-Frequency Transform

Karthikeyan Umapathy and Sridhar Krishnan

Department of Electrical and Computer Engineering, Ryerson University, 350 Victoria Street, Toronto, ON, Canada M5B 2K3

Received 22 January 2006; Revised 10 November 2006; Accepted 5 July 2007

Recommended by Douglas S. Brungart

Wide band digital audio signals have a very high data-rate associated with them due to their complex nature and demand for high-quality reproduction. Although recent technological advancements have significantly reduced the cost of bandwidth and miniaturized storage facilities, the rapid increase in the volume of digital audio content constantly compels the need for better compression algorithms. Over the years various perceptually lossless compression techniques have been introduced, and transform-based compression techniques have made a significant impact in recent years. In this paper, we propose one such transform-based compression technique, where the joint time-frequency (TF) properties of the nonstationary nature of the audio signals were exploited in creating a compact energy representation of the signal in fewer coefficients. The decomposition coefficients were processed and perceptually filtered to retain only the relevant coefficients. Perceptual filtering (psychoacoustics) was applied in a novel way by analyzing and performing TF specific psychoacoustics experiments. An added advantage of the proposed technique is that, due to its signal adaptive nature, it does not need predetermined segmentation of audio signals for processing. Eight stereo audio signal samples of different varieties were used in the study. Subjective (mean opinion score—MOS) listening tests were performed and the subjective difference grades (SDG) were used to compare the performance of the proposed coder with MP3, AAC, and HE-AAC encoders. Compression ratios in the range of 8 to 40 were achieved by the proposed technique with subjective difference grades (SDG) ranging from –0.53 to –2.27.

1. INTRODUCTION

The proposed audio coding technique falls under the transform coder category. The usual methodology of a transform-based coding technique involves the following steps: (i) transforming the audio signal into frequency domain coefficients, (ii) processing the coefficients using psychoacoustic models and computing the audio masking thresholds, (iii) controlling the quantizer resolution using the masking thresholds, (iv) applying intelligent bit allocation schemes, and (v) enhancing the compression ratio with further lossless compression schemes. A comprehensive review of many existing audio coding techniques can be found in the works of Painter and Spanias [1]. The proposed technique nearly follows the above general transform coder methodology however, unlike the existing techniques, the major part of the compression was achieved by exploiting the joint time-frequency (TF) properties of the audio signals. Hence, the main focus of this work would be in demonstrating the benefits of using an adaptive time-frequency transformation (ATFT) for coding the audio signals (i.e., improvement and novelty in step (i)) and developing a psychoacoustic model (i.e., improvement and novelty in step (ii)) adapted to TF functions.

The block diagram of the proposed technique is shown in Figure 1. The ATFT used in this work was based on the matching pursuit algorithm [2]. The Matching pursuit algorithm is a general framework where any given signal can be modeled/decomposed into a collection of iteratively selected, best matching signal functions from a redundant dictionary. The basis functions chosen to form the redundant dictionary determine the nature of the modeling/decomposition. When the redundant dictionary is formed using TF functions, the matching pursuit yields an ATFT [2]. The ATFT approach provides higher TF resolution than the existing TF techniques such as wavelets and wavelet packets [2]. This high-resolution sparse decomposition enables us to achieve a compact representation of the audio signal in the transform domain itself. Also, due to the adaptive nature of the ATFT, there was no need for signal segmentation.

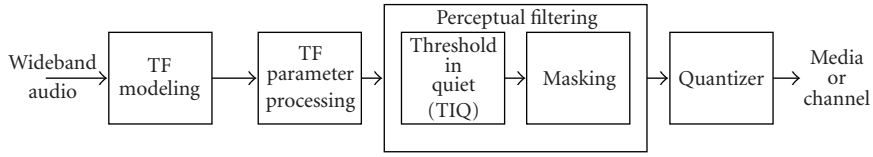

Figure 1: Block diagram of the ATFT audio coder.

Psychoacoustics was applied in a novel way [3, 4] on the TF decomposition parameters to achieve further compression. In most of the existing audio coding techniques, the fundamental decomposition components or building blocks are in the frequency domain with corresponding energy associated with them. This makes it much easier for them to adapt the conventional, well-modeled psychoacoustics techniques into their encoding schemes. In few existing techniques [5, 6] based on sinusoidal modeling using matching pursuits, psychoacoustics was applied either by scaling the dictionary elements or by defining a psychoacoustic adaptive norm in the signal space. As the modeling was done using a dictionary of sinusoids and segment-by-segment basis approach [7, 8], these techniques do not qualify as a true adaptive time-frequency transformation. Also, due to the fact that sinusoids were used in the modeling process, it was easier to incorporate the existing psychoacoustics models into these techniques. On the other hand, in ATFT, the signal was modeled using TF functions which have a definite time and frequency resolution (i.e., each individual TF function is time limited and band limited), hence the existing psychoacoustics models need to be adapted to apply on the TF functions.

The audio coding research is very dynamic and fast changing. There are a variety of applications (offline, IP streaming, embedding in video, etc.) and situations (network traffic, multicast, conferencing, etc.) for which many specific compression techniques were introduced. A universal comparison of the proposed technique with all audio coding techniques would be out of the scope of this paper. The objective of this paper is to demonstrate the application of ATFT for coding audio signals with some modifications to the conventional blocks of transform-based coders. Hence we restrict our comparison only with the two commonly known audio codecs MP3 and MPEG-4 AAC/HE-AAC [9–12]. These comparisons merely assess the performance of the proposed technique in terms of compression ratio achieved under similar conditions against the mean opinion scores (MOS) [13].

Eight reference wideband audio signals (ACDC, DEFLE, ENYA, HARP, HARPSICHORD, PIANO, TUBULARBELL, VISIT) of different categories were used for our analysis. Each was a stereo signal of 20-second duration extracted from CD quality digital audio sampled at 44.1 kHz. The ACDC and DEFLE were rapidly varying rock-like audio signals, ENYA and VISIT were signals with voice and humming components, PIANO and HARP were slowly varying classical-like signals, HARPSICHORD and TUBULARBELL were fast varying stringed instrumental audio signals. The ACDC, DEFLE, ENYA, and VISIT are polyphonic sounds with many sound sources.

The paper is organized as follows: Section 2 covers the ATFT algorithm, Section 3 describes the implementation of psychoacoustics, Sections 4 and 5 cover quantization, compression ratios and reconstruction process, Section 6 explains the quality assessment of the proposed coder, Section 7 covers results and discussion, and Section 8 summarizes the conclusions.

2. ATFT ALGORITHM

Audio signals are highly nonstationary in nature and the best way to analyze them is to use a joint TF approach. TF transformations can be performed either decomposing a signal into a set of scaled, modulated, and translated versions of a TF basis function or by computing the bilinear energy distributions (Cohen's class) [14, 15]. TF distributions are nonparametric and mainly used for visualisation purposes. For the application in hand, the automatic choice would be a parametric decomposition approach. There are variety of TF decomposition techniques with different TF resolution properties. Some examples in the increasing order of TF resolution superiority are short-time Fourier transform (STFT), wavelets, wavelet packets, pursuit-based algorithms [14]. As explained in Section 1, the proposed ATFT technique was based on the matching pursuit algorithm with time-frequency dictionaries. ATFT has excellent TF resolution properties (better than wavelets and wavelet packets) and due to its adaptive nature (handling nonstationarity), there is no need for signal segmentations. Flexible signal representations can be achieved as accurate as possible depending upon the characteristics of the TF dictionary.

In the ATFT algorithm, any signal $x(t)$ is decomposed into a linear combination of TF functions $g_{\gamma_n}(t)$ selected from a redundant dictionary of TF functions [2]. In this context, redundant dictionary means that the dictionary is overcomplete and contains much more than the minimum required basis functions, that is, a collection of nonorthogonal basis functions, that is, much larger than the minimum required basis functions to span the given signal space. Using ATFT, we can model any given signal $x(t)$ as

$$x(t) = \sum_{n=0}^{\infty} a_n g_{\gamma_n}(t), \tag{1}$$

where

$$g_{\gamma_n}(t) = \frac{1}{\sqrt{s_n}} g\left(\frac{t - p_n}{s_n}\right) \exp\{j(2\pi f_n t + \phi_n)\} \quad (2)$$

and a_n are the expansion coefficients.

The scale factor s_n, also called as octave parameter, is used to control the width of the window function, and the parameter p_n controls the temporal placement. The parameters f_n and ϕ_n are the frequency and phase of the exponential function, respectively. The index γ_n represents a particular combination of the TF decomposition parameters (s_n, p_n, f_n, and ϕ_n). The signal $x(t)$ is projected over a redundant dictionary of TF functions with all possible combinations of scaling, translations, and modulations. The dictionary of TF functions can either suitably be modified or selected based on the application in hand. When $x(t)$ is real and discrete, like the audio signals in the proposed technique, we use a dictionary of real and discrete TF functions. Due to the redundant or overcomplete nature of the dictionary it gives extreme flexibility to choose the best fit for the local signal structures (local optimisation) [2]. This extreme flexibility enables to model a signal as accurate as possible with the minimum number of TF functions providing a compact approximation of the signal.

In our technique, we used the Gabor dictionary (Gaussian functions) which has the best TF localization properties [15]. At each iteration, the best correlated TF function was selected from the Gabor dictionary. The remaining signal called the residue was further decomposed in the same way at each iteration subdividing them into TF functions. After M iterations, signal $x(t)$ could be expressed as

$$x(t) = \sum_{n=0}^{M-1} \langle R^n x, g_{\gamma_n} \rangle g_{\gamma_n}(t) + R^M x(t), \quad (3)$$

where the first part of (3) is the decomposed TF functions until M iterations, and the second part is the residue which will be decomposed in the subsequent iterations. This process was repeated till all the energy of the signal was decomposed. At each iteration, some portion of the signal energy was modeled with an optimal TF resolution in the TF plane. Over iterations, it can be observed that the captured energy increases and the residue energy falls. Based on the signal content, the value of M could be very high for a complete decomposition (i.e., residue energy = 0). Examples of Gaussian TF functions with different scale and modulation parameters are shown in Figure 2. The order of computational complexity for one iteration of the ATFT algorithm is given by $O(N \log N)$ where N is the length of the signal samples. The time complexity of the ATFT algorithm increases with the increase in the number of iterations required to model a signal, which in turn depends on the nature of the signal. Compared to this, the computational complexity of MDCT (in MP3 and AAC) is only $O(N \log N)$ (same as FFT).

Any signal could be expressed as a combination of coherent and noncoherent signal structures. Here the term "coherent signal structures" means those signal structures that have a definite TF localisation (or) exhibit high correlation

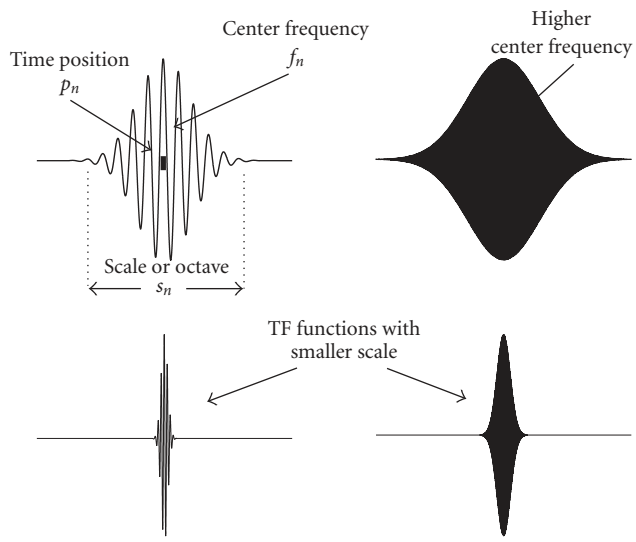

Figure 2: Gaussian TF function with different scale and modulation parameters.

with the TF dictionary elements. In general, the ATFT algorithm models the coherent signal structures well within the first few 100 iterations, which in most cases contribute to > 90% of the signal energy. On the other hand, the noncoherent noise like structures cannot be easily modeled since they do not have a definite TF localisation or correlation with dictionary elements. Hence, these noncoherent structures are broken down by the ATFT into smaller components to search for coherent structures. This process repeats until the whole residue information is diluted across the whole TF dictionary [2]. From a compression point of view, it would be desirable to keep the number of iterations ($M \lll N$) as low as possible and at the same time sufficient enough to model the signal without introducing perceptual distortions. Considering this requirement, an adaptive limit has to be set for controlling the number of iterations. The energy capture rate (signal energy capture rate per iteration) could be used to achieve this. By monitoring the cumulative energy capture over iterations we could set a limit to stop the decomposition when a particular amount of signal energy was captured. The minimum number of iterations required to model a signal without introducing perceptual distortions depends on the signal composition and the length of the signal.

In theory, due to the adaptive nature of the ATFT decomposition, it is not necessary to segment the signals. However, due to the computational resource limitations (Pentium III, 933 MHZ with 1 GB RAM), we decomposed the signals in 5-seconds durations. The larger the duration decomposed, the more efficient is the ATFT modeling. This is because if the signal is not sufficiently long, we cannot efficiently utilize longer TF functions (highest possible scale) to approximate the signal. As the longer TF functions cover larger signal segments and also capture more signal energy in the initial iterations, they help to reduce the total number of TF functions required to model a signal. Each TF function has a definite time and frequency localization, which means all the

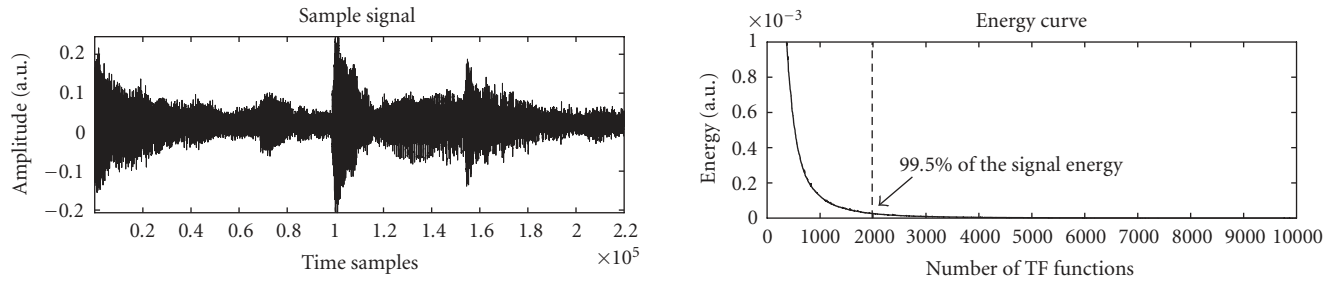

Figure 3: Energy cutoff of a sample signal (au:arbitrary units).

information about the occurrences of each of the TF functions in time and frequency of the signal is available. This flexibility helps us later in our processing to group the TF functions corresponding to any short time segments of the signal for computing the psychoacoustic thresholds. In other words, the complete length of the audio signal can be first decomposed into TF functions and later the TF functions corresponding to any short time segment of the signal can be grouped together. In comparison, most of the DCT- and MDCT-based existing techniques have to segment the signals into time frames and process them sequentially. This is needed to account for the nonstationarity associated with the audio signals and also to maintain a low-signal delay in encoding and decoding.

In the proposed technique for a signal duration of 5-second, the limit was set to be the number of iterations needed to capture 99.5% of the signal energy or to a maximum of 10 000 iterations. For a signal with less noncoherent structures, 99.5% of signal energy could be modeled with a lower number of TF functions than a signal with more noncoherent structures. In most cases, a 99.5% of energy capture nearly characterizes the audio signal completely. The upper limit of the iterations is fixed to 10 000 iterations to reduce the computational load. Figure 3 demonstrates the number of TF functions needed for a sample signal. In the figure, the right panel (b) shows the energy capture curve for the sample signal in the left panel (a) with number of TF functions in the X-axis and the normalized energy in the Y-axis. On average, it was observed that 6000 TF functions are needed to represent a signal of 5-second-duration sampled at 44.1 kHz. Using the above procedure, all eight (ACDC, DEFLE, ENYA, HARP, HARPSICHORD, PIANO, TUBULARBELL, VISIT) reference wideband audio signals were decomposed into their respective number of TF functions.

3. IMPLEMENTATION OF PSYCHOACOUSTICS

In this work, psychoacoustics was applied in a novel way on the TF functions obtained by decomposition. In the conventional method, the signal is segmented into short time segments and transformed into frequency domain coefficients. These individual frequency components are used to compute the psychoacoustic masking thresholds and accordingly their quantization resolutions are controlled. In contrast, in our approach we computed the psychoacoustic masking properties of individual TF functions and used them to decide whether a TF function with certain energy was perceptually relevant or not based on its time occurrence with other TF functions. TF functions are the basic components of the proposed technique and each TF function has a certain time and frequency support in the TF plane. So their psychoacoustical properties have to be studied by taking them as a whole to arrive at a suitable psychoacoustical model.

3.1. Threshold-in-quiet (TiQ)

TiQ is the minimum audible threshold below which we do not perceive a signal component. TF functions form fundamental building blocks of the proposed coder and they can take all possible combinations of time duration and frequency. However in the ATFT algorithm implementation, they could take any time width between 2^2 samples (90 microseconds) to 2^{14} samples (0.4 second) in steps with any frequency between 0 and 22 050 Hz (max frequency). The time support of a frequency component also plays an important role in the hearing process. From our experiments we observed that longer duration TF functions were heard much better even with lower energy levels than the shorter duration TF functions. Hence, out of all the possible durations of the TF functions, the highest possible time duration of 16 384 samples corresponding to the octave 14 (the term octave is from the implementation nomenclature, i.e., the scale factor doubles in each step) was the most sensitive TF function for different combinations of frequencies. This forms the worst case TF function in our modeling for which our ears are more sensitive. So it is obvious that this TF function has to be used to obtain the worst case threshold in quiet (TiQ) curve for our model. The curve obtained in this way will hold good for all other TF functions with all possible combinations of time-widths and center frequencies. Figure 4 demonstrates the different modulated versions of the TF function with maximum time-width (octave 14).

3.2. Experimental setup

Experiments were performed with 5 listeners to arrive at the TiQ curve for the above-mentioned TF function with maximum time width. The experimental setup consisted

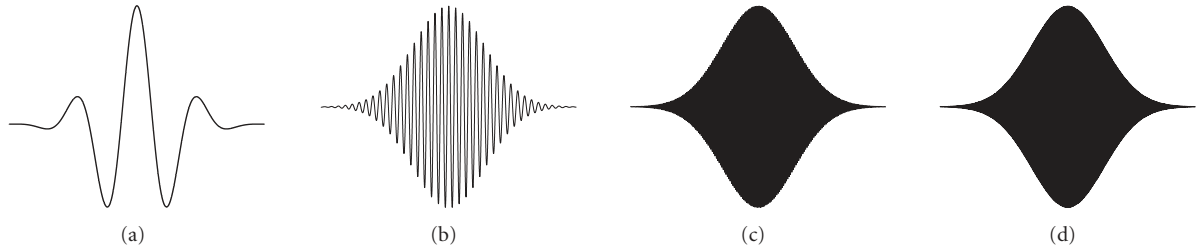

Figure 4: TF function with time width of 16 384 samples modulated at different center frequencies.

of a Windows 2000 PC (Intel Pentium III 933 MHz), creative sound blaster PCI card, high-quality head phones (Sennheiser HD490), and Matlab software package.

The TF functions (duration 0.4 seconds) with different center frequencies were played to each of the listeners. It should be noted that the "frequency" here means the center frequency of the TF function and not the absolute frequency as used in regular psychoacoustics experiments. In general, each of the TF functions will have a center frequency and a frequency spread based on the time width they can take. For this experiment as we are using only the TF function with the longest width (duration 0.4 second), the frequency spread is fixed. For each frequency setting the amplitude of the TF function was reduced in steps until the listener could no longer hear the TF function anymore. Once this point is reached, the amplitude of the TF function is increased and played back to the listener to confirm the correct point of minimum audibility. This is repeated for the following values of center frequencies: 10 Hz, 100 Hz, 500 Hz, 1 kHz, 2 kHz, 4 kHz, 6 kHz, 8 kHz, 10 kHz, 12 kHz, 16 kHz, and 20 kHz. The minimum audible amplitude level for each frequency setting was recorded. The values obtained from 5 listeners were averaged to obtain the absolute threshold of audibility for TF functions.

To reduce the computational complexity, the frequency range is divided into three bands of low frequency (500 Hz and below), sensitive frequencies (500 Hz to 15 kHz), and high frequencies (15 kHz and above). The experimental values were averaged to get uniform thresholding for the low- and high-frequency bands. In the middle or sensitive band, the lowest averaged experimental value was selected as threshold of audibility throughout the band. Figure 5 illustrates the averaged TiQ curve superimposed on the actual TiQ curve. The TF functions are grouped into the above-mentioned three frequency groups. Amplitude values of the TF functions are calculated from their energy and octave values. These amplitude values are checked with the TiQ average values. The TF functions whose amplitude values fall below the averaged TiQ values were discarded.

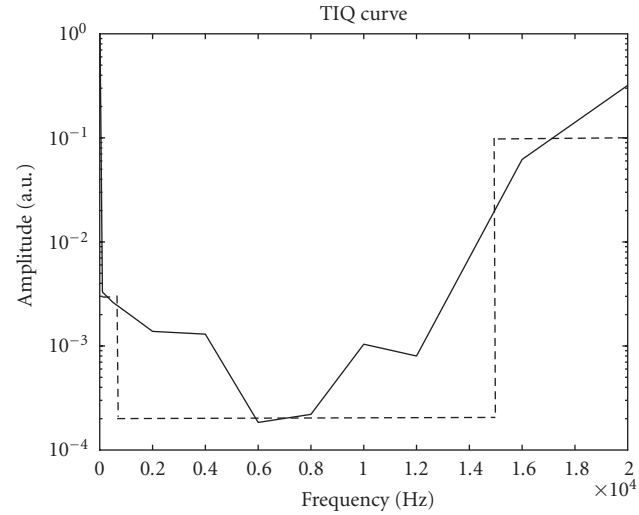

Figure 5: Average thresholding applied to TiQ curve. Solid line denotes the actual TiQ curve and dashed line denotes the applied threshold. (au:arbitrary units).

3.3. Audio masking applied to TF functions

Similar to TiQ, the existing masking techniques cannot be used directly on the proposed coder for the same reasons explained earlier. So masking experiments were conducted to arrive at masking thresholds for TF functions with different time-widths with a similar experimental setup as described in Section 3.2. The possible time duration of TF functions varies between 2^2 to 2^{14} in steps of powers of 2, each of the time width TF function was examined for its masking properties. Each of this different duration TF functions, can occur at any point in time with frequencies between 20 Hz to 20 kHz. Out of the possible durations of the TF functions the shorter durations (2^2 to 2^7) are transient-like structures which have larger bandwidths but little time support. Removing these TF functions in the process of masking will introduce more tonal artifacts in the reconstructed signal. This happens because the complex frequency pattern of the signal is disturbed to some extent. Hence, these functions were preserved and not used for masking purposes.

The remaining TF functions with time widths (2^8 to 2^{14}) were used for the masking experiments. TF functions with each of these time widths (durations from 256 to 16 384 samples) were tested for their masking capabilities with other time-width TF functions at various energies and frequencies. The TF functions were first grouped into equivalents of 400 time samples (10 milliseconds). This is possible as each of the TF functions has the precise information about its time occurrence. Once they were grouped into time slots equivalent

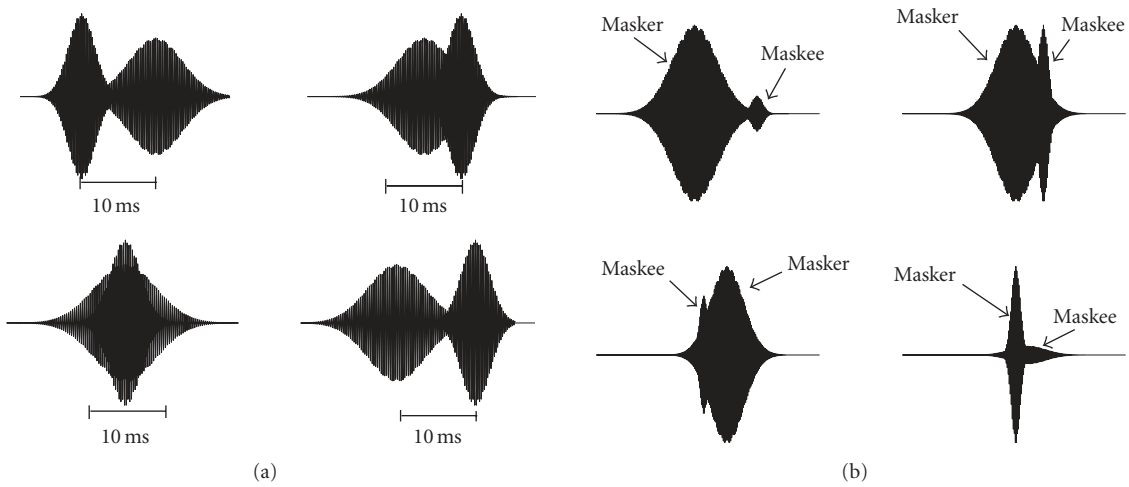

Figure 6: (a) Illustration of few possible time occurrences of two TF functions as masker and maskee, (b) possible masking conditions that can occur within the 10 milliseconds time slot.

to 10 milliseconds, the TF functions falling in each time slot were divided into 25 critical bands based on their center frequencies. In each critical band, the TF function with highest energy was located. Relative energy difference of this TF function with the remaining TF functions in the same critical band was computed. Using a lookup table, each of the remaining TF functions was verified if it would be masked by the relative energy difference with the TF function having the highest energy. The experimental procedure for computing the lookup table of masking thresholds will be explained in subsequent paragraphs. The TF functions which fall below the masking threshold defined by the lookup tables will be discarded.

As shown in Figure 6(a) within the 10 milliseconds duration the location of masker and maskee TF functions can occur anywhere. The worst case situation would be when the masker TF function occurs at the beginning of the time slot, and the maskee TF function occurs at the end of the time slot or vice versa. So all of our testing was done for this worst case scenario by placing the masker TF function and the maskee TF function at the maximum distance of 10 milliseconds.

Based on the duration of masker and maskee TF functions, one of the following could occur as depicted in Figure 6(b).

(1) Masker and maskee are apart in time within the 10 milliseconds, in which case they do not occur simultaneously. In this situation masking is achieved due to temporal masking effects where a strong occurring masker masks preceding and following weak signals in time domain.
(2) Masker duration is large enough that the maskee duration falls within the masker (two scenarios shown in Figure 6(b)) even after a 400 samples shift. In this case, simultaneous masking occurs.
(3) Masker duration is shorter than the maskee duration. In this case, both simultaneous and temporal maskings are achieved. The simultaneous masking occurs during the duration of the masker when the maskee is also present. Temporal masking occurs before and after the duration of the masker.

Four sets of experiments were conducted with masker TF function (normalized in amplitude) taking center frequency of 150 Hz, 1 kHz, 4.8 kHz, and 10.5 kHz (critical band center frequencies) and the maskee TF function taking center frequency of 200 Hz, 1.1 kHz, 5.3 kHz, and 12 kHz (corresponding critical band upper limits), respectively. As the masking thresholds depend also on the frequency separation of masker and maskee, maximum separation from the critical band center frequency was taken for our experiments for maskee TF functions. TF functions of each time width were used as maskers to measure their masking capabilities on the remaining of each time width TF functions for all the above 4 different frequency sets. Both (masker and maskee TF functions) were placed apart with 10 millisecond duration and played to the listeners. Each time the amplitude of the maskee TF function was reduced till the listener perceived only the masker TF function, or in other words, until there was no difference observed between the masker TF function played individually or played together with the maskee TF function. At this point, the masker TF function's energy was sufficient to mask the maskee TF function. The difference in their energies is calculated in dB and used as the masking threshold for the particular time-width maskee TF function when occurring simultaneously with that particular time-width masker TF function. Once all the measurements were finished, each time-width TF function was analyzed as a maskee against all the remaining time-width TF functions as masker. An average energy difference was computed for each time-width TF function below which they will be masked by any other time-width TF functions. Five different listeners participated in the test and their average masking curves for each time-width of TF functions were computed. Figure 7 shows the different masking curves obtained for different durations of TF functions. The X-axis represents the different time-width

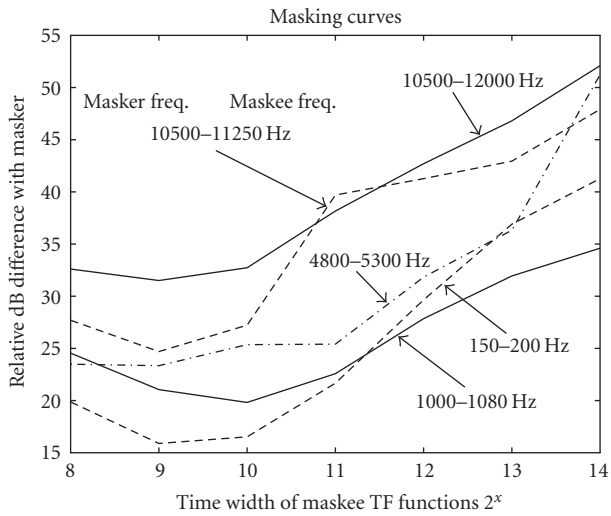

Figure 7: Masking curves for different time width of TF functions.

TF functions and the Y-axis represents the relative energy difference with the masker in dB.

The masking curve obtained for critical band center frequency 10.5 kHz deviates from the remaining curves considerably. This is due to the fact that the frequency separation between the masker and the maskee becomes very high at this band. This is because we use for all our experiments the upper limit of the critical band as the maskee frequency to simulate the worst case scenario. To demonstrate this frequency separation dependence on masking performance, a second masking curve was obtained for the critical band with a center frequency of 10.5 kHz for masker but this time the frequency separation between masker and maskee was reduced by half. The curve dropped down explaining the increase in masking performance, that is, when the frequency separation between the masker and maskee was reduced, the average relative dB difference required for masking also reduces.

From these curves it could be observed that the masking curves of critical bands with center frequencies 150 Hz, 1 kHz, and 4.8 kHz remain almost the same. Hence, the masking curve obtained for 1 kHz was used as the lookup table for the first 20 critical bands. The remaining 5 critical bands use the masking curve obtained for the critical band with a centre frequency of 10.5 kHz (with 12 kHz upper limit) as the lookup table. These lookup tables were used to verify if a TF function will be masked by the relative dB difference of it with the TF function having highest energy within the same critical band.

The flow chart in Figure 8 gives an overview of the masking implementation used in the proposed coder.

4. QUANTIZATION

Most of the existing transform-based coders rely on controlling the quantizer resolution based on psychoacoustic thresholds to achieve compression. Unlike this, the proposed technique achieves a major part of the compression in the transformation itself followed by perceptual filtering. That is,

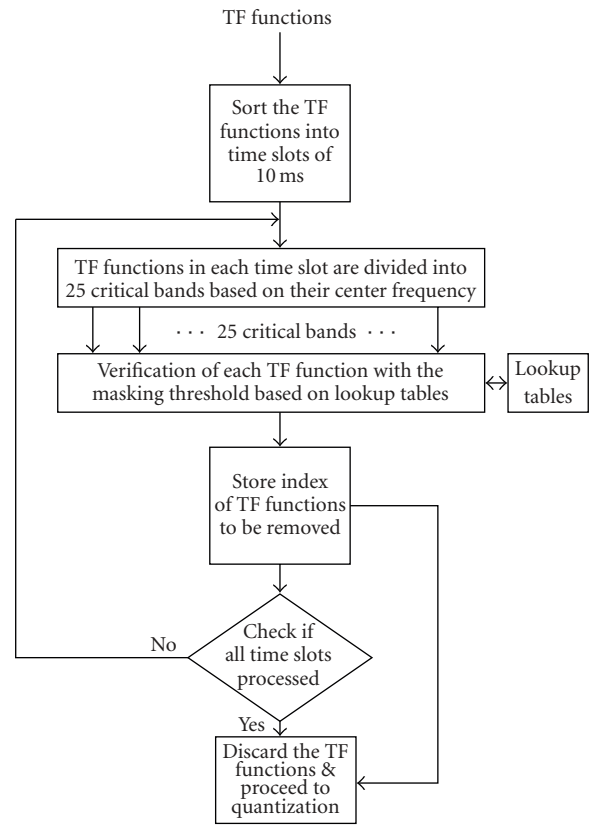

Figure 8: Flow chart of the masking procedure.

when the number of iterations M needed to model a signal is very low compared to the length of the signal, we just need $M \times L$ bits. Where L is the number of bits needed to quantize the 5 TF parameters that represent a TF function. Hence, we limit our research work to scalar quantizers as the focus of the research mainly lies on the TF transformation block and the psychoacoustics block rather than the usual subblocks of the data-compression application.

As explained earlier, each of the five parameters energy (a_n), centre frequency (f_n), time position (p_n), octave (s_n), and phase (ϕ_n) are needed to represent a TF function and thereby the signal itself. These five parameters were to be quantized in such a way that the quantization error introduced was imperceptible while, at the same time, obtaining good compression. Each of the five parameters has different characteristics and dynamic range. After careful analysis of them, the following bit allocations were made. In arriving at the final bit allocations informal MOS tests were conducted to compare the quality of the 8 audio samples before and after quantization stage.

In total, 54 bits are needed to represent each TF function without introducing significant perceptual quantization noise in the reconstructed signal. The final form of data for M TF functions will contain the following:

(1) energy parameter (log companded) = $M * 12$ bits;
(2) time position parameter = $M * 15$ bits;
(3) center frequency parameter = $M * 13$ bits;

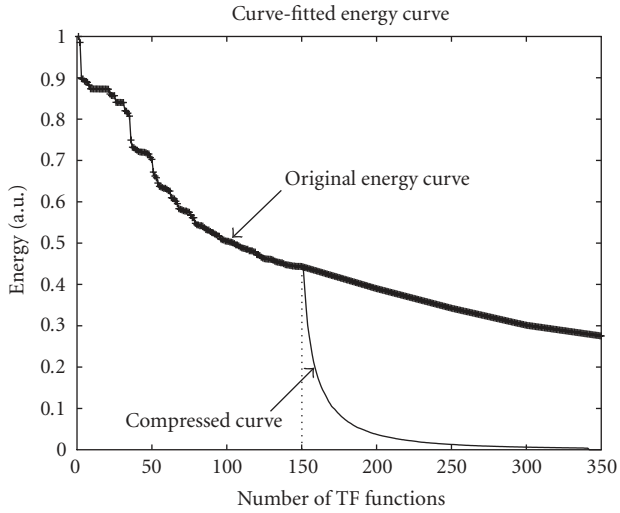

Figure 9: Log companded original and curve-fitted energy curve for a sample signal (au:arbitrary units).

(4) phase parameter = $M * 10$ bits;

(5) octave parameter = $M * 4$ bits.

The sum of all the above ($= 54 * M$ bits) will be the total number of bits transmitted or stored representing an audio segment of duration 5 seconds. The energy parameter after log companding was observed to be a very smooth curve as shown in Figure 9. Fitting a curve to the energy parameter further reduces the bitrate. Nearly 90% of the energy is present in the first few 100 TF functions and hence they are not used for curve fitting. The remaining number of TF functions is divided into equal lengths of 50 points on the curve. Only the values corresponding to these 50 points need to be sent with the first few original 100 values. The distance between these 50 points can be treated as linear comparing the spread of total number of TF functions. In the reconstruction stage, these 50 points can be interpolated linearly to the original number of points. The error introduced in this procedure was very small due to the smooth slope of the curve. Moreover, this error was introduced only in the 10% energy of the signal which was not perceived. To better explain the benefit of the proposed curve fitting approach in reducing the bitrate, let us take an example of transmitting 5000 TF functions. To transmit the energy parameter for 5000 TF functions (without applying curve fitting) will require $5000 * 12$ bits $= 60\,000$ bits. With curve fitting, say we preserve the energy parameter for the first 150 TF functions and thereafter select the energy parameter from every 50th TF function in the remaining 4850 TF functions. This will result in $[150 + (4850/50 = 97)] = 247$ values of the energy parameter requiring only $247 * 12 = 2964$ bits for transmission. We see a massive reduction in bits due to curve fitting. Figure 9 demonstrates the original curve superimposed with the fitted curve. Every kth point in the compressed curve corresponds to actually the $(3 + k) * 50$th point in the original curve. A correlation value of 1 was achieved between the original curve and the interpolated reconstructed curve.

With just a simple scalar quantizer and curve fitting of the energy parameter, the proposed coder achieves high compression ratios. Although a scalar quantizer was used to reduce the computational complexity of the proposed coder, sophisticated vector quantization techniques can be easily incorporated to further increase the coding efficiency. The 5 parameters of the TF function can be treated as one vector and accordingly quantized using predefined codebooks. Once the vector is quantized, only the index of the codebook needs to be transmitted for each set of TF parameters resulting in a large reduction of the total number of bits. However, designing the codebooks would be challenging as the dynamic ranges of the 5 TF parameters are drastically different. Apart from reducing the number of total bits, the quantization stage can also be utilized to control the bitrates suitable for constant bitrate (CBR) applications.

5. COMPRESSION RATIOS

Compression ratios achieved by the proposed coder were computed for the eight sample signals as described below.

(1) As explained earlier, the total number of bits needed to represent each TF function is 54.

(2) The energy parameter is curve fitted and only the first 150 points in addition to the curve-fitted point need to be coded.

(3) So the total number of bits needed for M iterations for a 5 second duration of the signal is $TB_1 = (M * 42) + ((150 + C) * 12)$, where C is the number of curve-fitted points, and M is the number of perceptually important functions.

(4) The total number of bits needed for a CD quality 16 bit PCM technique for a 5 second duration of the signal sampled at 44 100 Hz is $TB_2 = 44\,100 * 5 * 16 = 3\,528\,000$.

(5) The compression ratio can be expressed as the ratio of the number of bits needed by the proposed coder to the number of bits needed by the CD quality 16 bit PCM technique for the same length of the signal, that is,

$$\text{Compression ratio} = \frac{TB_2}{TB_1}. \tag{4}$$

(6) The overall compression ratio for a signal was then calculated by averaging all the 5 seconds duration segments of the signal for both the channels.

The proposed coder is based on an adaptive signal transformation technique, that is, the content of the signal and the dictionary of basis functions used to model the signal play an important role in determining how compact a signal can be represented (compressed). Hence, variable bitrate (VBR) is the best way to present the performance benefit of using an adaptive decomposition approach. The inherent variability introduced in the number of TF functions required to model a signal and thereby the compression is one of the highlights of using ATFT. Although VBR would be more appropriate to present the performance benefit of the proposed coder, CBR mode has its own advantages when used with applications

that demand network transmissions over constant bitrate channels with limited delays. The proposed coder can also be used in CBR mode by fixing the number of TF functions used for representing signal segments, however due to the signal adaptive nature of the proposed coder, this would compromise the quality at instances where signal segments demand a higher number of TF functions for perceptually lossless reproduction. Hence, we choose to present the results of the proposed coder using only the VBR mode.

We compare the proposed coder with two existing popular and state-of-the-art audio coders viz MP3 (MPEG 1 layer 3) and MPEG-4 AAC/HE-AAC. Advanced audio coding (AAC) is the current industrial standard which was initially developed for multichannel surround signals (MPEG-2 AAC [16]). The transformation technique used is the modified discrete cosine transform (MDCT). Compared to mp3 which uses a polyphase filter bank and an MDCT, new coding tools were introduced to enhance the performance. The core of MPEG-4 AAC is basically the MPEG-2 AAC but with added tools to incorporate additional coding enhancements and MPEG-4 features so that a broad range of applications are covered. There are many application specific profiles that can be chosen to adaptively configure the MPEG-4 audio for the user needs. It is claimed that at 128 kbps the MPEG-4 AAC is indistinguishable from the original audio signal [17]. As there are ample studies in the literature [9, 11, 12, 16, 18, 19] available for both MP3 and MPEG-2/4 AAC, more details about these techniques are not provided in this paper.

As the proposed coder is of VBR type, in our first comparison we compare the proposed coder with both the MP3 and MPEG-4 AAC coders in VBR mode. All eight sample signals were MP3 coded using the Lame MP3 encoder (version 1.2, Engine 3.88 Alpha 8) in VBR mode [20, 21]. For the MPEG-4 AAC, we used the AAC encoder developed by PysTel research (currently ahead software). As there are many profiles possible in AAC, we choose the following suitable profile for our comparison-VBR high quality with main long-term prediction (LTP) [10]. All eight signals were MPEG-4 AAC encoded. The average bitrates for each signal for both MP3 and MPEG-4 AAC was found using the Winamp decoder [22]. These average bitrates were used to calculate the compression ratio as described below.

(1) Bitrate for a CD quality 16 bit PCM technique for 1-second stereo signal is given by $TB_3 = 2 * 44\,100 * 16$.

(2) The average bitrate/s achieved by (MP3 or MPEG-4 AAC) in VBR mode = TB_4.

(3) Compression ratio achieved by (MP3 or MPEG-4 AAC) = TB_3/TB_4.

The 2nd, 4th, and 6th columns of Table 1 show the compression ratio (CR) achieved by the MP3, MPEG-4 AAC, and the proposed ATFT coders for the set of 8 sample audio files. It is evident from the table that the proposed coder has better compression ratios than MP3. When comparing with MPEG-4 AAC, 5 out of 8 signals are either comparable or have better compression ratios than the MPEG-4 AAC. It is noteworthy to mention that for slow music (classical type), the ATFT coder provides 3 to 4 times better comparison than MPEG-4 AAC or MP3. The compression ratio alone cannot be used to evaluate an audio coder. The compressed audio signals has to undergo a subjective evaluation to compare the quality achieved with respect to the original signal. The combination of the subjective rating and the compression ratio will provide a true evaluation of the coder performance. A second comparison was also performed by comparing the HE-AAC profile of the MPEG-4 audio at the same bitrates to that was achieved by the ATFT coder in the VBR mode. More details on the HE-AAC profile of the MPEG-4 audio will be discussed in the subsequent sections. A subjective evaluation was performed as will be explained in Section 6.

Before performing the subjective evaluation, the signal has to be reconstructed. The reconstruction process is a straight forward process of linearly adding all the TF functions with their corresponding five TF parameters. In order to do that, first the TF parameters modified for reducing the bitrates have to be expanded back to their original forms. The log-compressed energy curve was log expanded after recovering back all the curve points using interpolation on the equally placed 50 length points. The energy curve was multiplied with the normalization factor to bring the energy parameter as it was during the decomposition of the signal. The restored parameters (energy, time-position, centre frequency, phase, and octave) were fed to the ATFT algorithm to reconstruct the signal. The reconstructed signal was then smoothed using a third order Savitzky-Golay [23] filter and saved in a playable format.

Figure 10 demonstrates a sample signal (/"HARP"/) and its reconstructed version and the corresponding spectrograms. It can be clearly observed from the reconstructed signal spectrogram compared with the original signal spectrogram, how accurately the ATFT technique has filtered out the irrelevant components from the signal (evident from Table 1-(/"HARP"/)-high compression ratio vs. acceptable quality). The accuracy in adaptive filtering of the irrelevant components is made possible by the TF resolution provided by the ATFT algorithm.

6. QUALITY ASSESSMENT OF THE PROPOSED CODER

6.1. Subjective evaluation of ATFT coder

Subjective evaluation of audio quality is needed to assess the audio codec performance. We use the subjective evaluation method recommended by ITU-R standards (BS. 1116). It is called a "double blind triple stimulus with hidden reference" [1, 13]. In this method, listeners are provided with three stimuli A, B, and C for each sample under test. A is the reference/original signal, B and C are assigned to either of the reference/original signal or the compressed signal under test. Basically the reference signal is hidden in either B or C and the other choice is assigned to the compressed (or impaired) signal. The choice of reference or compressed signal for B and C is completely randomized. For each sample audio signal, listeners listen to all three (A, B, C) stimuli, and compare A with B and A with C. After each comparison of A with B, and A with C, they grade the quality of the B and C

Table 1: Compression ratio (CR) and subjective difference grades (SDG). MP3-moving picture experts group I layer 3, AAC-MPEG-4 AAC, moving picture experts group 4 advanced audio coding-VBR main LTP profile, ATFT:adaptive time-frequency transform.

Samples	MP3		AAC		ATFT	
—	CR	SDG	CR	SDG	CR	SDG
ACDC	7.5	0.067	9.3	−0.067	8.4	−0.93
DEFLE	7.7	−0.2	9.5	−0.067	8.3	−1.73
ENYA	9	0	9.6	−0.133	20.6	−0.8
HARP	11	−0.067	9.4	−0.067	36.3	−1
HARPSICHORD	8.5	−0.067	10.2	0.33	9.3	−0.73
PIANO	13.6	0.067	9.6	−0.2	40	−0.8
TUBULARBELL	8.3	0	10.1	0.067	10.5	−0.53
VISIT	8.4	−0.067	11.5	0	11.6	−2.27
Average	9.3	−0.03	9.9	−0.02	18.3	−1.1

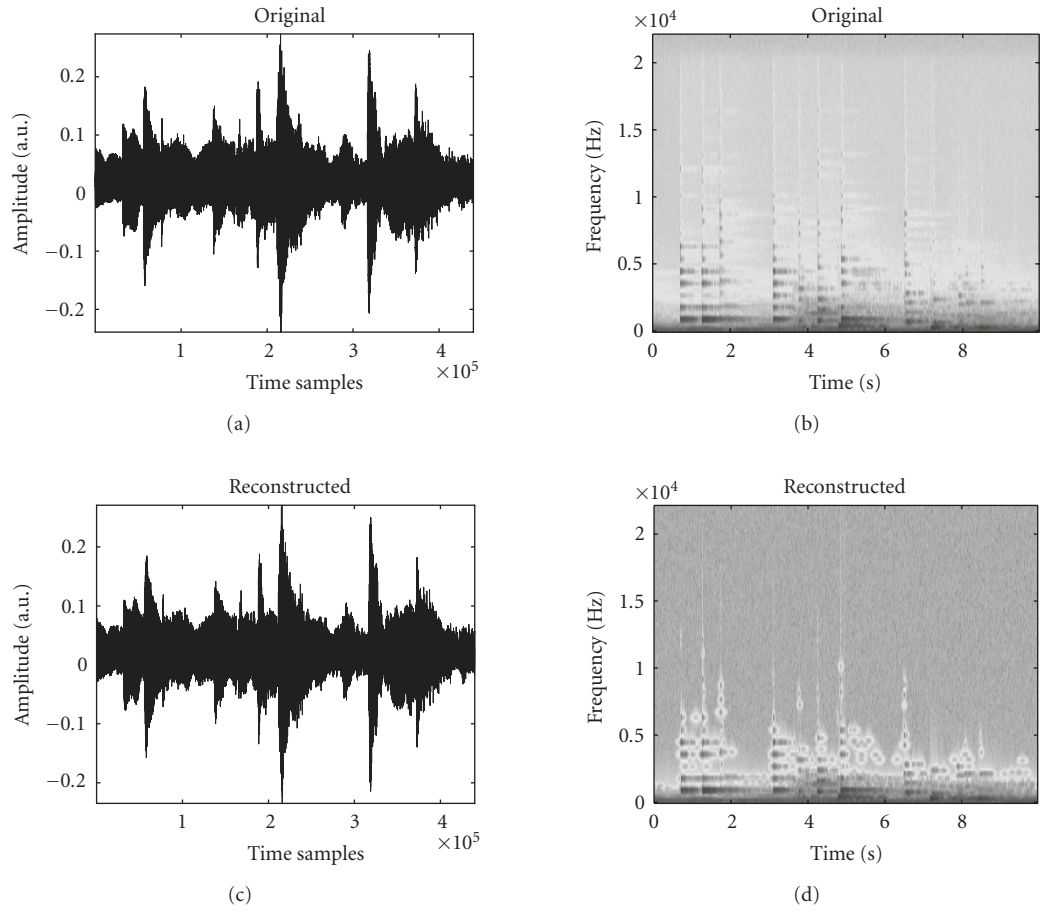

Figure 10: Example of a sample original (/"HARP"/) and the reconstructed signal with their respective spectrograms. X-axes for the original and reconstructed signal are in time samples, and X-axes for the spectrogram of the original and the reconstructed signal are in equivalent time in seconds. Note that the sampling frequency = 44.1 kHz (au:arbitrary units).

signals with respect to A in 5 levels from 1 to 5. The levels 1 to 5 corresponds to (1) unsatisfactory (or) very annoying, (2) poor (or) annoying, (3) fair (or) slightly annoying, (4) good (or) perceptible but not annoying, and (5) excellent (or) imperceptible [1, 13]. A subjective difference grade (SDG) [1] is computed by subtracting the absolute score assigned to the hidden reference from the absolute score assigned to the compressed signal. It is given by

$$\text{SDG} = \text{Grade}_{\{\text{compressed}\}} - \text{Grade}_{\{\text{reference}\}}. \quad (5)$$

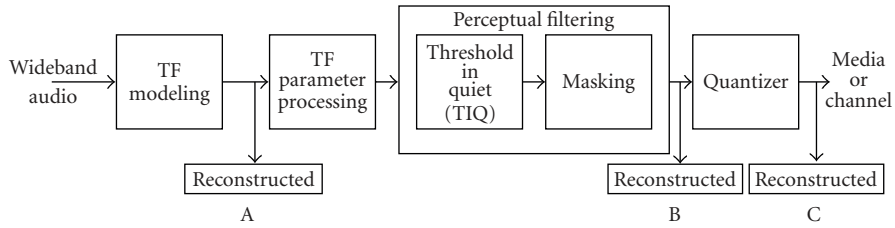

Figure 11: Block diagram explaining MOS choices A, B, and C for the subjective evaluation of perceptual filtering and quantization stages.

Table 2: Average SDG for PFO and QO (PFO:perceptually filtered output, QO:quantization output).

Samples	PFO	QO
ACDC	−0.8	−0.4
DEFLEP	−0.6	−0.6
ENYA	−0.8	−1
HARP	−0.8	−0.6
HARPSICHORD	0	−0.6
PIANO	−0.2	−0.8
TUBULARBELL	−0.6	−1.2
VISIT	−0.2	−0.4
Average	−0.5	−0.7

Accordingly, the scale of SDG will range from (−4 to 0) with the following interpretation, (−4) unsatisfactory (or) very annoying, (−3) poor (or) annoying, (−2) fair (or) slightly annoying, (−1) good (or) perceptible but not annoying, and (0) excellent (or) imperceptible. Fifteen listeners (randomly selected) participated in the MOS studies and evaluated all the 3 audio coders (MP3, AAC, and ATFT in VBR mode). The average SDG was computed for each of the audio sample. The 3rd, 5th, and 7th columns of Table 1 show the SDGs obtained for MP3, AAC, and ATFT coders, respectively. MP3 and AAC SDGs fall very close to the Imperceptible (0) region, whereas the proposed ATFT SDGs are spread out between −0.53 to −2.27.

A second listening test was performed using the HE-AAC v1/v2 (also known as MPEG-4 AAC v2, AAC PLUS v2) encoder [12]. The HE-AAC encoder is an enhanced high-efficiency version of the AAC with improved audio quality. The HE-AAC v1 encoder comprises of the basic AAC and spectral band replication (SBR) technologies whereas the v2 encoder comprises of AAC, SBR, and parametric stereo (PS) coding technologies. The HE-AAC v2 encoder is rated as the best audio codec at low bitrates. In the second test, SDG were computed for the 8 audio samples by encoding them using HE-AAC v1/v2 coder at the same bitrates/compression ratios as that of the ATFT. Six listeners participated in the second MOS study and the obtained SDG are shown in Table 3. Detailed discussion of the compression ratio versus subjective evaluation scores is given in Section 7.

6.2. Subjective evaluation of perceptual filtering and quantization stages

In order to evaluate the performance of the developed perceptual model and the quantization stage, another listening test was conducted with 5 listeners. The procedure as described in Section 6.1 was repeated but the choices A, B, and C were assigned as shown in Figure 11. The output of the TF decomposition (TF modeling stage) forms the input to the perceptual filtering module, hence the reference A was assigned to the reconstructed signal at the output of the TF modeling stage. Similarly, choice B was assigned to the reconstructed signal at the output of the perceptual filtering module and C to the reconstructed signal at the output of the quantization stage. Listeners were asked to rate the choices B (perceptual filtering output) and C (quantization stage output) with the reference A (TF modeling output) on a scale of 1 to 5 as explained in Section 6.1.

The results were averaged for the five listeners and given in Table 2. From Table 2, it can be observed on an average, SDGs of −0.5 and −0.7 were achieved for the perceptual filtering stage and the quantization stage, respectively. The SDG scores indicate that the novel perceptual filtering technique proposed is performing exceedingly well with the eight sample signals and the noise introduced in the process of quantization affects the output quality minimally. Interestingly, it can be noted from Table 2 that the signals ACDC, DEFLEP, and VISIT have better SDG scores when the reference is the reconstructed output signal of the TF modeling block than when the reference is the original signal. This gives a valid clue that the low SDG scores achieved by these signals (as seen in Table 1) are not due to the perceptual filtering or the quantization stages but may be due to TF modeling with symmetrical type Gaussian functions.

7. RESULTS AND DISCUSSION

7.1. Performance comparison in VBR mode

The compression ratios (CR) and the SDG for all three coders (MP3, AAC, and ATFT) are shown in Table 1. All the coders were tested in the VBR mode. For the proposed technique, VBR was the best way to present the performance benefit of using an adaptive decomposition approach. In ATFT, the type of the signal and the characteristics of the TF functions (type of dictionary) control the number of transformation parameters required to approximate the signal and thereby the compression ratio. The inherent variability introduced in the number of TF functions required to model a signal is one of the highlights of using ATFT. Hence, we choose to present comparison of the coders in the VBR mode.

The results show that the MP3 and AAC coders perform well with excellent SDG scores (Imperceptible) at a

Table 3: SDG comparison of MPEG-4 AAC PLUS coder for the same compression ratio/bitrate achieved by the proposed ATFT coder. Modes: 1. HE-high efficiency AAC v1 (AAC and spectral band replication (SBR)) and 2. HEv2-high efficiency AAC v2 (AAC, SBR and parametric stereo (PS) coding).

Samples	HE-AAC v1/2				ATFT
—	MODE	Kbps	CR	SDG	SDG
ACDC	HE	168	8.4	0.17	−0.93
DEFLE	HE	170	8.3	0.17	−1.73
ENYA	HEv2	68	20.6	−0.17	−0.8
HARP	HEv2	38	36.3	0	−1
HARPSICHORD	HE	151	9.3	−0.33	−0.73
PIANO	HEv2	35	40	0.17	−0.8
TUBULARBELL	HE	134	10.5	0	−0.53
VISIT	HE	121	11.6	0.33	−2.27
Average	—	111	18.3	0.04	−1.1

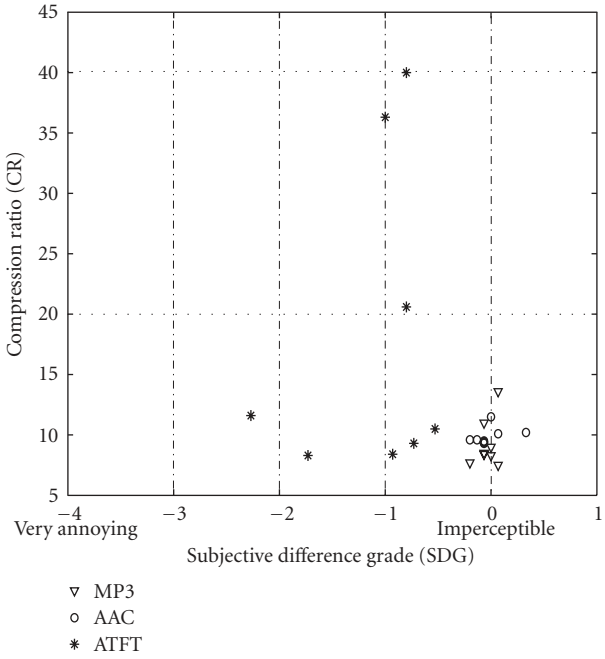

Figure 12: Subjective difference grade (SDG) versus compression ratios (CR).

compression ratio around 10. The proposed coder does not perform well with all of the eight samples. Out of the 8 samples, 6 samples have an SDG between −0.53 to −1 (imperceptible-perceptible but not annoying) and 2 samples have SDG below −1. Out of the 6 samples with SDGs between (−0.53 and −1), 3 samples (ENYA, HARP, and PIANO) have compression ratios 2 to 4 times higher than MP3 and AAC and 3 samples (ACDC, HARPSICHORD, and TUBULARBELL) have comparable compression ratios with moderate SDGs.

Figure 12 shows the comparison of all three coders by plotting the samples with their SDGs in X-axis and compression ratios in the Y-axis. If we can virtually divide this plot in segments of SDGs (horizontally) and the compression ratios (vertically), then the ideal desirable coder performance should be in the right top corner of the plot (high compression ratios and excellent SDG scores). This is followed next by the right bottom corner (low compression ratios and excellent SDG scores) and so on as we move from right to left in the plot. Here, the terms "low" and "high" compression ratios are used in a relative sense based on the compression ratios achieved by all the 3 coders in this study. From the plot it can be seen the MP3 and AAC coders occupy the right bottom corner, whereas the samples from ATFT coder are spread over. As mentioned earlier, 3 out the 8 samples of the ATFT coder occupy the right top corner however only with moderate SDGs that are much less than the MP3 and the AAC. Three out of the remaining 5 samples of the ATFT coder occupy the right bottom corner, however again with only moderate SDGs that are less than MP3 and AAC. The remaining 2 samples perform the worst occupying the left bottom corner.

We analyzed the poorly performing ATFT-coded signals DEFLE and VISIT. DEFLE is a rapidly varying rock-like signal with minimal voice components and VISIT is a signal with dominant voice components. We observed that the symmetrical and smooth Gaussian dictionary used in this study does not model the transients well, which are the main features of all rapidly varying signals like DEFLE. This inefficient modeling of transients by the symmetrical Gaussian TF functions resulted in the poor SDG for the DEFLE. A more appropriate dictionary would be a damped sinusoids dictionary [24] which can better model the transient like decaying structures in audio signals. However, a single dictionary alone may not be sufficient to model all types of signal structures. The second signal VISIT has significant amount(s) of voice components. Even though, the main voice components are modeled well by the ATFT, the noise like hissing and shrilling sounds (noncoherent structures) could not be modeled within the decomposition limit of 10 000 iterations. These hissing and shrilling sounds actually add to the pleasantness of the music. Any distortion in them is easily perceived which could have reduced the SDG of the signal to the

lowest of the group −2.27. The poor performances with the two audio sample cases could be addressed by using a hybrid dictionary of TF functions and residue coding the noncoherent structures separately. However, this would increase the computational complexity of the coder and reduce the compression ratios.

7.2. Performance comparison with same bitrates (high-efficiency AAC)

In the second performance comparison of the ATFT coder, we choose to test the high-efficiency profile of the MPEG-4 AAC v2 at the same bitrates as that of the ATFT coder. As per [12], the HE-AAC v2 improves the coding gain of the AAC by 4 times and outperforms most of the existing coders in audio quality especially at low bitrates. All the 8 samples were encoded using the HE-AAC v1/v2 encoder at the same bitrates as that of the VBR bitrates of the ATFT. The choice for the v1 or the v2 encoder was determined by the target bitrates (below 70 kbps v2 was used). From the SDGs obtained as shown in Table 3, it is very evident that the HE-AAC profiles of the MPEG-4 audio codec outperforms the proposed ATFT coder for the same bitrates. However, it would not be fair to compare the HE-AAC directly with the ATFT, since ATFT is a basic coder without the additional enhancements achieved by SBR or any form of parametric coding.

Although we did not include standalone sinusoidal coders in our comparison, the MPEG-4 HE-AAC v2 includes the parametric stereo coding based on the transient-sinusoid-noise (TSN) model and is derived from the MPEG-4 audio sinusoidal coding (also abbreviated as SSC) [25]. The TSN model though in existence for quite some time for audio and speech coding, received much attention recently with its inclusion in the MPEG-4 audio standard for low-bitrate applications. The formal verification tests of the MPEG-4 SSC indicate that the SSC coder performs either comparable to or better than MPEG-4 AAC even at lower bitrates than MPEG-4 AAC [25]. Another recent well-known family of sinusoidal codec (the SiCAS codec and its variants) is from the research group of Heusdens et al. and Philips Research Laboratories [26]. It is shown that the psychoacoustics model used in this family of codecs is better than the MPEG I Layer I-II psychoacoustics model [27]. The subjective listening tests indicate that this family of codec performs equal to or better than MPEG-4 at low bitrates (16 Kbps (HILN), 24 Kbps (SSC), and 32 Kbps (AAC)) [26]. Various improvements have been proposed for this codec family over the years with an incremental performance gain [28, 29]. Interestingly, the improvements proposed in using amplitude modulated sinusoids over constant amplitude sinusoids indicate the migration of these approaches towards completely adaptive signal decompositions such as the one proposed in the ATFT coder [29].

8. CONCLUSIONS

This paper presented a novel ATFT coding technique for wideband audio signals. The proposed approach demonstrated the application of adaptive time-frequency transform for audio coding and the development of a novel psychoacoustics model adapted to TF functions. The compression strategy was changed from the conventional way of controlling quantizer resolution to achieving majority of the compression in the transformation itself. Eight stereo sample signals were used in the study. Listening tests were conducted and the performance comparison of the proposed coder with MP3 and AAC coders was presented. From the preliminary results, although the proposed coder achieves high compression ratios, its SDG scores are well below the MP3 and AAC family of coders. The proposed coder however performs moderately well for slowly varying classical-like signals with acceptable SDGs. The proposed coder is not as refined as the state-of-the-art commercial coders, which to some extent explains its poor performance. Future work involves testing the proposed coder with a hybrid dictionary of TF functions and including professional refinements.

REFERENCES

[1] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 451–515, 2000.

[2] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[3] K. Umapathy and S. Krishnan, "Joint time-frequency coding of audio signals," in *Proceedings of the 5th WSES/IEEE Multiconference on Circuits, Systems, Communications, and Computers (CSCC '01)*, pp. 32–36, Crete, Greece, July 2001.

[4] K. Umapathy and S. Krishnan, "Low bit-rate coding of wideband audio signals," in *Proceedings of IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA '01)*, pp. 101–105, Rhodes, Greece, July 2001.

[5] R. Heusdens, R. Vafin, and W. B. Kleijn, "Sinusoidal modeling using psychoacoustic-adaptive matching pursuits," *IEEE Signal Processing Letters*, vol. 9, no. 8, pp. 262–265, 2002.

[6] T. S. Verma and T. H. Y. Meng, "Sinusoidal modeling using frame-based perceptually weighted matching pursuits," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '99)*, vol. 2, pp. 981–984, Phoenix, Ariz, USA, March 1999.

[7] R. Heusdens, J. Jensen, P. Korten, and R. Vafin, "Rate-distortion optimal high-resolution differential quantisation for sinusoidal coding of audio and speech," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (ASPAA '05)*, pp. 243–246, New Paltz, NY, USA, October 2005.

[8] R. Heusdens and J. Jensen, "Jointly optimal time segmentation, component selection and quantization for sinusoidal coding of audio and speech," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '05)*, vol. 3, pp. 193–196, Philadelphia, Pa, USA, March 2005.

[9] I. JTC1/SC29/WG11, "Overview of the MPEG-4 standard," in International Organization for Standardisation, March 2002.

[10] J. Herre and B. Grill, "Overview of MPEG-4 audio and its applications in mobile communications," in *Proceedings of the 5th International Conference on Signal Processing (ICSP '00)*, vol. 1, pp. 11–20, Beijing, China, August 2000.

[11] J. Herre, K. Brandenburg, et al., "Second generation ISO/MPEG audio layer-3 coding," in *The 98th Audio Engineering Society Convention (AES '95)*, Paris, France, February 1995.

[12] S. Meltzer and G. Moser, "MPEG-4 HE-AAC v2—audio coding for today's digital media world," Tech. Rep., EBU Technical Review, Geneva, Switzerland, January 2006.
[13] T. Ryden, "Using listening tests to assess audio codecs," in *Collected Papers on Digital Audio Bit Rate Reduction*, N. Gilchrist and C. Grewin, Eds., pp. 115–125, Audio Engineering Society, New York, NY, USA, 1996.
[14] S. Mallat, *A wavelet Tour of Signal Processing*, Academic Press, San Diego, Calif, USA, 1998.
[15] L. Cohen, "Time-frequency distributions: a review," *Proceedings of the IEEE*, vol. 77, no. 7, pp. 941–981, 1989.
[16] K. Brandenburg and M. Bosi, "MPEG-2 advanced audio coding: overview and applications," in *The 103rd Audio Engineering Society Convention*, p. 4641, Ney York, NY, USA, August 1997.
[17] http://www.apple.com/MPEG4/aac/.
[18] E. Eberlein and H. Popp, "Layer-3, a flexible coding standard," in *The 94th Audio Engineering Society Convention*, p. 3493, Berlin, Germany, March 1993.
[19] http://www.iis.fraunhofer.de/bf/amm/.
[20] http://lame.sourceforge.net/index.php.
[21] http://www.mp3dev.org/.
[22] http://www.winamp.com/.
[23] S. J. Orfanidis, *Introduction to Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1996.
[24] M. M. Goodwin, *Adaptive Signal Models: Theory, Algorithms and Audio Applications*, Kluwer Academic Publishers, Boston, Mass, USA, 1998.
[25] ISO/IEC JTC 1/SC 29/WG 11N6675, "Report on the verification tests of MPEG-4 parametric coding for high quality audio," in International Organization for Standardisation, July 2004.
[26] R. Heusdens, J. Jensen, W. B. Kleijn, et al., "Bit-rate scalable intraframe sinusoidal audio coding based on rate-distortion optimization," *Journal of the Audio Engineering Society*, vol. 54, no. 3, pp. 167–188, 2006.
[27] S. van de Par, A. Kohlrausch, R. Heusdens, J. Jensen, and S. H. Jensen, "A perceptual model for sinusoidal audio coding based on spectral integration," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 9, pp. 1292–1304, 2005.
[28] P. Korten, J. Jensen, and R. Heusdens, "High-resolution spherical quantization of sinusoidal parameters," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 966–981, 2007.
[29] M. G. Christensen and S. van de Par, "Efficient parametric coding of transients," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1340–1351, 2006.