# Perceptually-Driven Scalable MDCT Enhancement of Compressed Audio Based on Statistical Conversion

Demetrios Cantzos
Dept. of Automation Engineering,
Technological Education Institute
of Piraeus (TEI Piraeus)
Athens, Greece
cantzos@teipir.gr

Athanasios Mouchtaris
Institute of Computer Science,
Foundation for Research and
Technology (FORTH-ICS) and
Dept. of Computer Science,
University of Crete (UOC),
Heraclion, Crete, Greece
mouchtar@ics.forth.gr

Chris Kyriakakis
Ming Hsieh Dept. of
Electrical Engineering,
Viterbi School of Engineering,
University of Southern
California (USC),
Los Angeles, CA, USA
ckyriak@usc.edu

***Abstract*—Many state-of-the-art audio codecs operating in a transform domain provide scalability as a core function by allowing to selectively subtract bits –usually according to a nonperceptual criterion- from the full bitrate data stream. This work presents a different, or even reverse, scalability approach in which a scalable codec can selectively add perceptually significant bits to a low bitrate data stream. The scalable enhancement algorithm presented here operates in the Modified Discrete Cosine Transform domain, which is popular among perceptual audio transform encoders, but its extension on other domains is straightforward. By exploiting the information of an existing low bitrate base layer, the algorithm adds perceptually significant data to the data stream according to a psychoacoustic model, and improves the audio quality at a fraction of the bitrate that would normally be required for the encoding or transmission of the whole audio piece of the same quality. Applications of this can be found in packet retransmission schemes of compressed audio networks and in remote audio enhancement.**

***Audio coding; scalability; MDCT; enhancement; conversion; psychoacoustic model***

## I. INTRODUCTION

Scalability has become an extremely desirable feature as modern audio codecs are being deployed in internet transmission and multimedia broadcasting scenarios. The need for scalability arises naturally in these scenarios, which require variable bitrate delivery of audio, at little or no quality loss as compared to nonscalable codecs. Typically, a scalable codec incorporates an enhancement layer that lies on top of the core codec, or base layer. This enhancement layer is generally the difference between the uncompressed signal and the signal produced by the base layer, i.e. the signal compressed by the core codec. The decoder then subtracts enough bits from the enhancement layer until a target bitrate is reached. According to a well-known scalability method, the MPEG-4 Scalable Lossless Coding (SLS) standard [1], the enhancement layer is produced in the Modified Discrete Cosine Domain (MDCT), subsequently entropy encoded and then algorithms such as Bit Slice Arithmetic Coding (BSAC) [2] and Context –Based Arithmetic Coding (BSAC) [3] assume the bitrate reduction task. The main drawback of this standard is that the subtraction of bits from the enhancement layer is not psychoacoustically-driven, but instead the least significant bits are subtracted in a bit-plane slicing fashion. This point has also been stressed in [4], in which the SLS method for AAC is further improved by additionally considering psychoacoustic information from the base layer.

In our work, scalability is studied in a statistical conversion context wherein the enhancement layer essentially contains a conversion function between the base layer, or *source*, and the full bitrate, uncompressed signal, or *target.* The algorithm works in conjunction with a psychoacoustic model to convert the base layer to a higher quality audio piece at a fine grain, incremental step. Specifically, the derivation of the enhancement layer is guided by a psychoacoustic criterion and the bitrate of that layer is finely tuned according to a bit allocation scheme, described later. The statistical conversion itself is based on a Linear Predictive Coding (LPC) scheme of MDCT coefficients. The MDCT domain is selected in order to ensure compatibility with modern transform codecs, although the parametric nature of LPC allows the whole algorithm to be applied directly even on PCM data.

In contrast to the SLS and other scalability methods, where the starting point for subtracting bits is the full or high bitrate data stream, our approach implements scalability by adding (instead of subtracting) bits to the low bitrate, base layer (instead of the high bitrate). This approach has a definite advantage in cases in which the base layer or source is already at hand or has been transmitted to the decoder and thus only the enhancement layer needs to be conveyed. Such a case would be the request for retransmission of a higher quality version of a previously transmitted packet in an audio network wherein only the enhancement layer would be sent to the receiver node. The same holds for remote enhancement scenarios in which a higher audio quality piece is requested, conditional on the fact that the base layer is already present at the receiving end.

This paper is organized as follows. In Section II, the statistical conversion process is described along with a sorting transformation process. In Section III, the two

978-0-7695-4589-9/11 $26.00 © 2011 IEEE
DOI 10.1109/ISM.2011.16

psychoacoustic models employed by our algorithm are described. The bit allocation scheme, stemming from the psychoacoustic model, is also explained. Section IV provides the implementation details and the comparative results on the scalable enhancement algorithm, while Section V concludes this work.

## II. STATISTICAL CONVERSION

### A. Pre-Processing

The statistical conversion process is based on our previous work in [5]. Our starting point is a pair of source and target signals, taken from the same audio piece, with the source being a low bitrate signal produced by a standard perceptual codec (i.e. the base layer), and the target being the uncompressed version (Fig. 1). Note that the target signal is only available at the encoding side, not at the decoding. At the encoder side, the source and target signals are transformed by the MDCT filterbank on a frame-by-frame basis. Each frame is pre-windowed with a Kaiser-Bessel window and adjacent MDCT spectral coefficients of each source or target frame are grouped into 32 subbands, similarly to a standard codec's approach, although other subband configurations are possible. After the MDCT filterbank, an LPC analysis is applied on the MDCT coefficients of each of the 32 subbands to extract the Line Spectral Frequency (LSF) feature vectors and their corresponding residual vectors.

The LPC analysis of each MDCT subband group is performed on each frame separately, and not across all MDCT frames, leading to interframe independence and thus enabling us to accurately map a signal subband segment to the LSF or residual vectors and vice versa. This is important during the perceptually-driven conversion process as we will be able to modify the LSF or residual vectors of a particular MDCT frame without influencing adjacent frames due to the LPC analysis window overlap. Consequently, a minimum of two LSF and two residual vectors per subband and per frame (i.e. a minimum of two overlapping LPC subframes) has to be produced to ensure perfect MDCT frame recovery during the inverse operation (LPC synthesis) at the decoder.

### B. Conversion Function

After the LSF and residual vectors extraction from the source and target signals, the conversion function needs to be computed. Statistical conversion is based on the assumption that the source LSF or residual vectors of each subband are generated by a random process whose samples follow a diagonal Gaussian mixture model (GMM) pdf, given as

$$G(\mathbf{x}) = \sum_{k=1}^{K} p(C_k) \prod_{j=1}^{q} g(x^{(j)}; \mu_k^{(j)}, \sigma_k^{(j)}), \qquad (1)$$

where $C_k$ denotes the cluster (component) $k$, $K$ is the number of clusters and $p(C_k)$ denotes the prior probability that the source vector $\mathbf{x}$ belongs to cluster $k$. The source vector is $q$ dimensional and the $j$th coefficient is denoted by $x^{(j)}$. The mean and variance of each GMM cluster for coordinate $j$ are

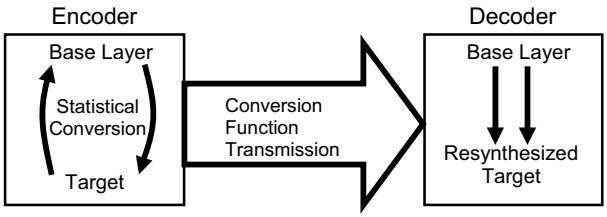


Figure 1. Overview of the enhancement algorithm

noted as, $\mu_k^{(j)}$, and $\sigma_k^{(j)}$, respectively. The vector coefficients are considered to be independent and thus the vector pdf is the product of the $q$ coefficient pdf's. The complete model parameters ($\mu_k^{(j)}$, $\sigma_k^{(j)}$, $p(C_k)$) for the LSF and residual vector GMMs can be estimated by an ML estimation algorithm as the one in [5], using LSF and residual vectors originating from pink noise, and are permanently stored.

LSF or residual vectors conversion between the various source and target subband segments is implemented through a linear conversion function. The conversion function, $F(\cdot)$, acts on the source LSF/residual vector sequence $[\mathbf{x_1},...,\mathbf{x_n}]$ and produces a reconstructed vector sequence close in the least squares sense to the target LSF/residual vector sequence $[\mathbf{y_1},...,\mathbf{y_n}]$. Since we have selected a diagonal implementation, this function will act on the individual vector components and minimize the error

$$E = \sum_{t=1}^{n} \sum_{j=1}^{q} | \, y_t^{(j)} - F(x_t^{(j)}) \, |^2 , \qquad (2)$$

as in [6]. To address this task we consider the function $F$ as piecewise linear i.e.

$$F(x_t^{(j)}) = \sum_{k=1}^{K} P(C_k \mid \mathbf{x}_t)[v_k^{(j)} + \frac{u_k^{(j)}}{\sigma_k^{(j)}}(x_t^{(j)} - \mu_k^{(j)})] \qquad (3)$$

for $t$ =1,..,$n$ and $j$ =1,..,$q$. The conditional probability that a given vector belongs to cluster $k$, $P(C_k|\boldsymbol{x}_t)$, is given by Bayes' Rule. The unknown parameters set $[\boldsymbol{v},\boldsymbol{u}]$ can be found by minimizing (2) which reduces to solving a typical set of $q$ independent least-squares equations [6] and hence the linear conversion function $F$ is fully determined. We call $[\boldsymbol{v},\boldsymbol{u}]$ the *conversion parameters set*. These parameters are to be transmitted to the receiver or decoder in order to reconstruct the target LSF/residual data because these are the only parameters that are dependent on the particular source and target LSF/residual data. Note that a diagonal implementation is favored because the computation of the conversion parameters set is faster and, most importantly, the size of the parameters set itself is much smaller [6]. The remaining parameters of equation (3) are part of the LSF or residual mixture model and they are precomputed (once) and permanently stored during the mixture pdf estimation.

### C. Sorting Transformation

A technique, based on our previous work [5], is adopted here that can significantly increase LSF and residual conversion accuracy, while reducing the conversion parameters size. We sort the source and target vector coefficients (LSF or residual) along each coordinate in ascending order. The motivation behind the sorting

TABLE I

| Source Indices | Target Indices |
|---|---|
| 1 | 1 |
| 4 | 4 |
| 3 | 3 |
| 6 | 5 |
| 5 | 2 |
| 9 | 7 |
| 8 | 6 |
| 2 | 8 |
| 7 | 9 |

Information that is available at the encoder. The decoder has access to the left column.

TABLE II

| Target Indices | Actual Transmission |
|---|---|
| 1 | 0 |
| 4 | 0 |
| 3 | 0 |
| 5 | 3 |
| 2 | 0 |
| 7 | 3 |
| 6 | 2 |
| 8 | 0 |
| 9 | 0 |

Original target positions and sorting information that is actually transmitted to the decoder.

transformation is found in the form of the conversion function. The conversion function is a (piecewise) linear estimator that estimates the target data from the source data. Its optimal performance is achieved when the true relation between the source and target data is linear along each coordinate. Therefore, this sorting technique allows us to reduce the number of mixture classes because the estimation is easier and consequently the number of conversion parameters is also reduced. Details of this scheme can be found in [5].

In order for the decoder to be able to use the (sorted) reconstructed LSF or residual coefficients and create the final subband signal, the original order of the reconstructed coefficients has to be known. The source data are available at the decoder and thus the original order of the source LSF and residual coefficients is known. Our focus is on deducing the original order of the target data and use that info to reorder the reconstructed data at the decoder end. We term this information, the *sorting information* and it is transmitted along with the conversion parameters. These two sets combined form the *transmitted parameters* and they can be considered as the equivalent of the enhancement layer.

The straightforward solution would be to transmit the original order of the target LSF and residual data as side information, along with the conversion parameters. At the decoder, the coefficients would be reconstructed one by one and a side index would determine where to place the particular LSF/residual coefficient. This scheme would require transmission of $n \cdot \log_2 n$ bits of information where $n$+1 is the number of elements being sorted (assuming $n$ is a power of two). Instead of directly transmitting the sorting indices of the target data to the decoder, we can derive a sequence of minimum insertions and shifts that will take us from the source sorting indices to the target sorting indices. The reasoning behind this is that the source and target data have not identical but similar original position configurations and thus the target original positions could be inferred from the source original positions with fewer than $n \cdot \log_2 n$ bits of information.

As an example of this similarity, in Table I we give the original positions of 9 sorted residual coefficients (across the first coordinate) of a randomly chosen source-target dual set. This information is sufficient to recover the original order of the source and target data and in this example it is 3 bits/coefficient for 8 uniformly encoded coefficients (for the 9th coefficient it is trivial to find its original place). In terms of transmission, only the target indices column has to be conveyed to the decoder since the source signal is available. The steps of an algorithm that allows us to transmit less information to the decoder without explicitly sending every index of the target column are:

1. The encoder checks if the source and target indices of the current row are the same. If yes, then a zero is transmitted. If no, then proceed to the next step.
2. The encoder looks in the target index of the current row and finds the position (new row) of that index in the source column. The distance between the current row and the new row is transmitted. All values in the current row of the source column up to the new row are circularly shifted by one position towards the end of the column, so that the value of the new row replaces the value of the current row.
3. Repeat steps 1 and 2 until all rows of the target column have been traversed and the source column has been converted to the target column.

After the algorithm is completed, the source column has been converted to the target column and the only information that has to be transmitted is the second column of Table II. This lossless operation will enable us to send fewer bits at the decoder, especially after we perform entropy coding.

### *D. Bitrate control*

The sorting information, as described in the previous section, accounts for more than 85% of the transmitted parameters set, or the enhancement layer. Therefore, it is essential to be able to accurately control the size of the sorting information in order to control the total bitrate of the enhancement layer. It should be noted that bitrate control entails loss of conversion accuracy, i.e. in order to further reduce the sorting information, the reconstruction accuracy of the LSF or residual vectors is ultimately compromised. Naturally, this should happen in a fine step manner, so that fine grain scalability is achieved. A variant of our work in [5] is used in which prior to being sorted, the source vectors $X$ and target vectors $Y$ that participate in the derivation of the conversion function are modified according to

$$X' = \left| X \right| \tag{4}$$

$$Y' = Y \cdot \mathrm{sign}(X) + c \cdot \left| X \right| \tag{5}$$

where $c$ is called the *multiplier* and takes values from a predefined set of positive integers including zero (available to both the encoder and the decoder), and sign(·) is the sign function which outputs +1 if the sign is positive and -1 if the sign is negative. The role of the multiplier is to increase the similarity between $X'$ and $Y'$ in terms of their sorting position indices so that, after sorting, the sorting information according to Section II-C will be less. The higher the multiplier is, the lower the size of the sorting information becomes -at the cost of decreased conversion accuracy- since $X$ will dominate over $Y$ in (5).

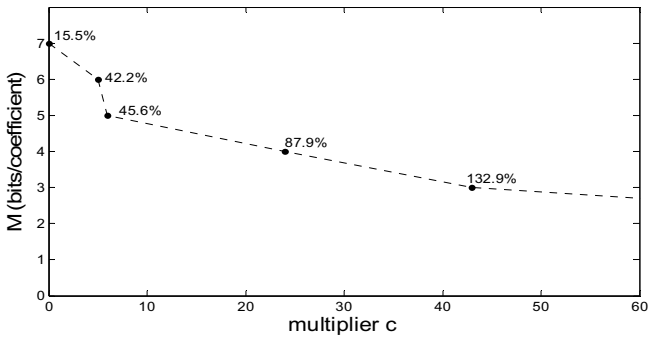


Figure 2. An example of the relation between the multiplier, $c$, and the size of the sorting information, $M$, in bits/coefficient for the residual vector conversion of subband 1 (20 Hz - 690 Hz) of a random music piece. The percentages on the plot are the converted vectors relative errors as compared to the original source-target vector errors.

The role of the sign function is auxiliary and it only benefits residual conversion and not LSF conversion, since the LSFs are always positive. One can observe that the product $Y \cdot \text{sign}(X)$ is usually positive because the residual vectors $X$ and $Y$ have similar sign (especially in the low subbands), as they are taken from the compressed and uncompressed version of the same audio piece. Therefore, it is expected that the individual $Y'$ residual vector coefficients of (5) will be, in most cases, positive. Inserting positive $X'$ and $Y'$ vector coefficients in the conversion process (after they are sorted) increases the conversion accuracy at no extra transmission overhead since $X$ and $\text{sign}(X)$ are available at the decoder side. Note that the inversion of (5) to derive $Y$ at the decoder is straightforward as long as $c$ is known.

In order to fully control the bitrate, we need a method to directly relate the multiplier $c$, for each subband and for either LSF or residual conversion, to the number of bits of the sorting information. A straightforward way to adjust the size of the sorting information to exactly $M$ bits/coefficient by tuning $c$ is as follows:

1. At the encoding side, set $c = 0$. After sorting $X'$ and $Y'$ along each coordinate, the encoder derives the sorting information and computes its maximum value, $m$, for all coefficients and coordinates. If $m \leq 2^M$, then store $c$ and stop. Else, proceed to step 2.
2. At the encoding side, set $c = c + 1$. After sorting $X'$ and $Y'$ along each coordinate, the encoder derives the sorting information and computes its maximum value, $m$, for all coefficients and coordinates. If $m \leq 2^M$, then store $c$ and stop. Else, go to the beginning of step 2.

An example of the above algorithm is illustrated in Fig. 2 for the residual vectors conversion of a random music piece. Notice that as $c$ increases, $M$ decreases, along with the conversion accuracy. In addition, for $M = 3$ bits/coefficient or less, the conversion process creates errors greater than the original source-target vector distance and obviously $M$ should not take such low values. The psychoacoustic model described next, helps to adaptively determine the appropriate values for $M$ and maintain conversion accuracy at acceptable levels.

## III. Psychoacoustic Model and Bit Allocation

A method for controlling the bitrate of the enhancement layer (conversion function) has been described in Section II-D. A problem remains, though, in determining which vectors (and of which subbands) will be converted and with how many bits of sorting information. The standard approach is to minimize a psychoacoustic distortion metric, such as the Noise-to-Mask ratio (NMR) [7]. Two psychoacoustic models, one analytical based on the NMR, and one simpler based on masking threshold curves, are described next.

### A. Psychoacoustic Model Based on NMR

An accurate method to take advantage of the available conversion function bitrate is to minimize the NMR by considering the effect of each of the 32 signal subbands separately. The difficulty in this task is that, by default, the NMR is calculated over a bark scale while our analysis runs across a linear frequency scale of 32 equidistant subbands. Nevertheless, by studying each subband separately, bit allocation becomes more efficient and yields considerable bitrate savings. The steps described below lead to a NMR variation matrix that is sensitive to the particular subband of a particular MDCT frame:

1. Starting with $T$ MDCT frames, calculate the initial $T$ NMR values between the source and target frames, and store them into a $T$ x 1 vector, $\text{NMRmat}_0$.Set $i = 1$, $t = 1$.
2. For MDCT frame $t$ and subband $i$ of the source signal, replace the source LSF and residual vectors with the corresponding target vectors. Calculate the NMR for that frame and subtract $\text{NMRmat}_0[t]$ from it. Store that value to a 32 x $T$ matrix NMRmat as the $[i,t]$ entry. Set $i = i + 1$. If $i \leq 32$, go to step 2. Else, go to step 3.
3. Set $t = t + 1$, $i = 1$. If $t \leq T$, go to step 2. Else, terminate.

With the above method, we derive a 32 x $T$ matrix of the NMR decrease that each subband (out of a total of 32 subbands) of each MDCT frame (out of a total of $T$ frames) incurs, as compared to the original NMR between the source and target MDCT frames. By picking the matrix indices for which the NMR decrease values are the largest, we know for which frames and subbands the conversion process would be most beneficial for the reduction of the total NMR of the reconstructed signal. Hence, for these indices we allocate more bits/coefficient for the sorting information, through the algorithm of Section II-D. For the indices with the smallest decrease, conversion does not even take place. This concept is similar to the water-filling bit allocation method of MPEG1-Layer 3 (MP3) [8]. While this model is quite accurate and saves a significant amount of bits during bit allocation, it is computationally intensive as for each frame 32 +1 NMR values have to be calculated. A faster, but less accurate, bit allocation method is described next.

### B. Psychoacoustic Model Based on Masking Curves

A simpler psychoacoustic model that forms the basis for bit allocation can be derived from the masking threshold curves. For each of the $T$ MDCT frames of the source signal, we calculate the global masking threshold curve (MTH)

across the full frequency spectrum (up to 22 kHz) according to the MP3 standard [8]. Depending on the value of MTH, averaged over a particular subband, the bits/coefficient of the sorting information are assigned in a water-filling fashion, i.e., the lower the MTH is, the higher the number of bits/coefficient of sorting information for that subband and MDCT frame should be to mask audible distortion. For the highest MTH values, and when there are no available bits for allocation, conversion does not even take place. Note that this method could be considered as being similar to Perceptual LPC [9], where the LPC vectors are calculated according to a weighted $l_2$ norm, with the weight being the reciprocal of the MTH. However, that psychoacoustic scheme takes into consideration the LPC vectors only, not the residual vectors which are important in our scenario.

## IV. Implementation and Results

Three MP3 (LAME version 3.98.3) music pieces are tested by our algorithm, one rock music piece, one classical music piece and one jazz music piece of 5.3 s duration, each. The MDCT filterbank outputs frames of 3872 coefficients. The LPC window length is constant at 46ms with a hop size of 42ms, while the LPC order is 5. Note that we have not implemented variable window analysis or other coding tools such as Temporal Noise Shaping (TNS) [10] used by standard enhancement methods and nonscalable codecs and this incurs some performance loss. The performance of the algorithm is evaluated by the ITU-R BS.1387 PEAQ test, basic model [11], which emulates a subjective listening test. Its output is the Objective Difference Grade (ODG) value which ranges from -4 ("very annoying") to 0 ("imperceptible"). The PEAQ test results have been shown to be highly correlated with the Subjective Difference Grades (SDG) from a subjective listening test, especially at medium to high bitrates. A secondary output of the PEAQ test is the NMR.

At the given signal duration, 120 LSF or residual vectors across each subband are created. This means that the maximum number of bits/coefficient of the sorting information cannot exceed $\log_2(120)$ or 7 bits/coefficient. After experimenting with bit allocation, we concluded that three modes are sufficient for our enhancement algorithm. In mode A, LSF and residual conversion occurs with 6 bits/coefficient of sorting information. In mode B, LSF and residual conversion occurs with 6 bits/coefficient and 4 bits/coefficient of sorting information, respectively. In both modes the target LPC gains are encoded uniformly with 8 bits. In mode C, no conversion occurs and no target LPC gains are encoded. The conversion parameters set of (3) is uniformly encoded with 12 bits and, as mentioned before, it accounts for very little of the total parameters size (<15%). Naturally, mode A is suited for the most perceptually significant parts of the signal, mode B for the less perceptually significant parts and mode C for the least perceptually significant parts. The user can simply select percentages for the three modes, e.g. 10% of the signal for mode A, 20% for mode B and 70% for mode C. Alternatively, if a target bitrate is to be met, the algorithm has predefined values stored for the three percentages.

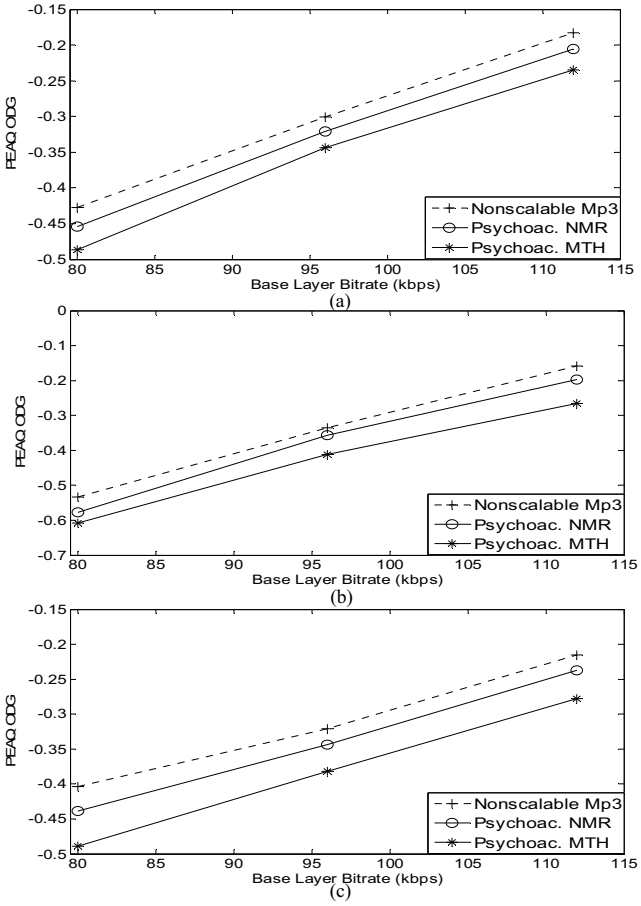


Figure 3. PEAQ test ODG scores for various bitrates of the base layer with an enhancement layer at a constant bitrate of 16 kbps for: (a) the rock piece, (b) the classical piece and (c) the jazz piece. For comparison, the performance of nonscalable MP3 is also plotted at the equivalent bitrates. The enhancemnt algorithm is realized with two psychoacoustic models, i.e. the NMR based (Psychoac. NMR) and the MTH based (Psychoac. MTH).

In our first testing scenario, we keep the enhancement layer's bitrate low compared to that of the base layer. This is beneficial in cases when the decoder requests retransmission of a higher quality version of the already received base layer, or when the enhancement layer becomes corrupted during transmission through an audio network. Hence, in our first testing scenario, we keep the enhancement layer at a constant small bitrate of 16 kbps while the source or base layer MP3 bitrate ranges from 80 kbps to 112 kbps, in steps of 16 kbps. The PEAQ test ODG values are shown in Fig. 3, with a comparison to nonscalable MP3 of the same total bitrate. It is clear that the enhancement algorithm under the NMR based psychoacoustic model of Section III-A outperforms the enhancement algorithm under the MTH based psychoacoustic model of Section III-B and it is very close to the performance of nonscalable MP3.

In the second scenario, we test the scalability of the enhancement algorithm by maintaining a low base layer bitrate of 80 kbps and by increasing the enhancement layer bitrate, at a 16 kbps step, in order to directly compare with

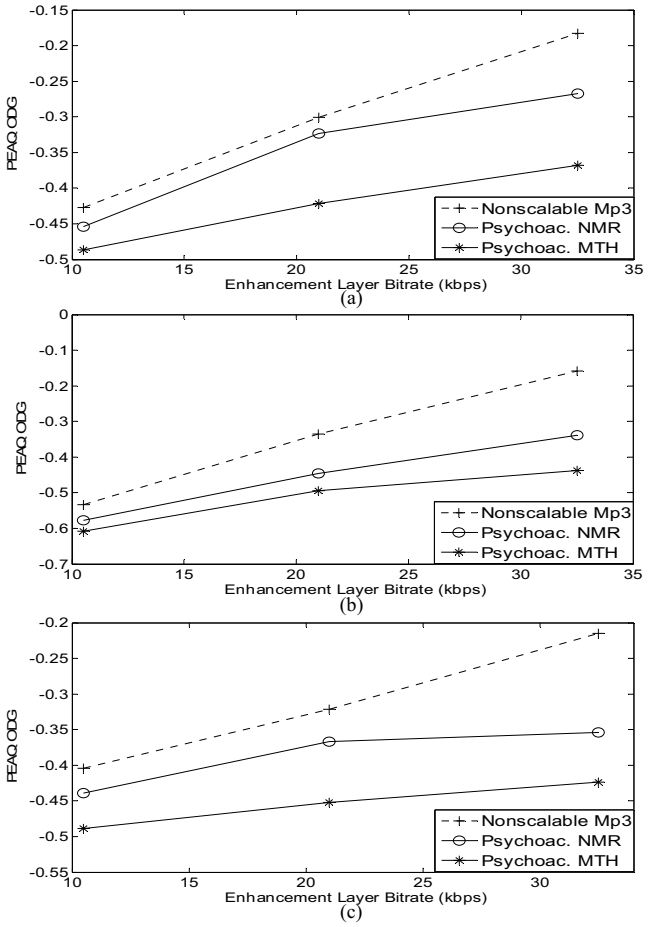


Figure 4. PEAQ test ODG scores for various bitrates of the enhancement layer with a base layer at a constant bitrate of 80 kbps for: (a) the rock piece, (b) the classical piece and (c) the jazz piece. For comparison, the performance of nonscalable MP3 is also plotted at the equivalent bitrates. The enhancemnt algorithm is realized with two psychoacoustic models, i.e. the NMR based (Psychoac. NMR) and the MTH based (Psychoac. MTH).

the first scenario. The ODG values are shown in Fig. 4, with a comparison to nonscalable MP3 whose scores are exactly the same as in Fig. 3 and are plotted for comparison. The enhancement algorithm total bitrates of Fig. 3 and 4 are the same at the three x-axis points and they are equal to 96, 112 and 128 kbps. Fig. 4 shows that the performance of the algorithm is slightly worse as compared to the first scenario of Fig. 3. This suggests that our algorithm works better when the enhancement layer bitrate is low compared to the base layer bitrate. As the enhancement layer increases, the algorithm relies less on the base layer and cannot exploit as much base layer information.

Generally, it seems that nonscalable MP3 is slightly better in terms of perceptual distortion at equal bitrates, but this is expected. According to the study in [4], a scalable codec can only asymptotically reach the performance of a nonscalable codec in a rate/distortion framework, and thus we do not expect to exceed the performance of nonscalable MP3, either.

## V. Conclusion

A scalable, psychoacoustically-driven enhancement algorithm was presented, based on statistical conversion. It operates by adding enhancement data on the base layer, rather than subtracting data from a lossless enhancement layer. As the algorithm relies on the base layer information, it works best when the enhancement layer is small compared to the base layer, and when the base layer is already available at the decoder, e.g., in packet retransmission or remote enhancement applications. The MDCT domain is the preferred domain of operation but due to the parametric nature of LPC analysis, any other desirable domain would suffice, even the time domain (PCM data). Finally, it was shown that the performance of the algorithm is close to nonscalable MP3, while improvements can be made by incorporating coding techniques such as variable window length analysis and TNS.

## Acknowledgment

This work has been funded in part by the European Community's Seventh Framework Programme under grant agreement No. 230709 (PEOPLE-IAPP "AVID-MODE" grant).

## References

[1] Scalable Lossless Coding (SLS), ISO/IEC 14496-3:2005/Amd 3, 2006.

[2] R. Yu, C. C. Ko, S. Rahardja, and X. Lin, "Bit-plane Golomb coding for sources with Laplacian distributions," Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 03), Hong Kong, April 2003, pp. 277-280.

[3] R. Yu, X. Lin, S. Rahardja, C. C. Ko, H. Huang, "Improving Coding Efficiency for MPEG-4 Audio Scalable Lossless Coding," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 05), Philadelphia, PA, USA, May 2005, pp. 169-172.

[4] A. Aggarwal, S.L. Regunathan, K. Rose, "Efficient bit-rate scalability for weighted squared error optimization in audio coding," IEEE Transactions on Speech and Audio Processing, vol. 14, no. 4, July 2006, pp. 1313-1327.

[5] D. Cantzos, A. Mouchtaris, and C. Kyriakakis, "Quality Enhancement of Compressed Audio Based on Statistical Conversion," EURASIP Journal on Audio, Speech, and Music Processing, vol. 2008, Article ID 462830.

[6] Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," IEEE Transactions on Speech and Audio Processing, vol. 6, no. 2, March 1998, pp. 131-142.

[7] K. Brandenburg, "Evaluation of quality for audio encoding at low bitrates," Proc. 82nd AES Convention, London, UK, March 1987, preprint 2433.

[8] P. Noll, MPEG Digital Audio Coding Standards, CRC Press, New York, NY, USA, 2000.

[9] R.C. Hendriks, R. Heusdens, J. Jensen, "Perceptual linear predictive noise modelling for sinusoid-plus-noise audio coding," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 04), Montreal, Canada, May 2004, pp. 189-192.

[10] J. Herre, J.D. Johnston, "Enhancing the Performance of Perceptual Audio Coders by Using Temporal Noise Shaping (TNS)," Proc. 101st AES Convention, Los Angeles, CA, USA, Nov. 1996, preprint 4384.

[11] ITU-R Recommendation BS.1387, "Methods for objective measurements of perceptual audio quality," International Telecommunications Union, Geneva, Switzerland, 1999.