Hindawi Publishing Corporation
EURASIP Journal on Audio, Speech, and Music Processing
Volume 2008, Article ID 462830, 15 pages
doi:10.1155/2008/462830

Research Article

Quality Enhancement of Compressed Audio Based on Statistical Conversion

Demetrios Cantzos,[1, 2] Athanasios Mouchtaris,[3, 4] and Chris Kyriakakis[1, 2]

[1] *Integrated Media Systems Center (IMSC), University of Southern California, Los Angeles, CA 90089, USA*
[2] *Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA*
[3] *Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH-ICS), Heraklion, Crete 70013, Greece*
[4] *Department of Computer Science, University of Crete, Heraklion, Crete 71409, Greece*

Correspondence should be addressed to Demetrios Cantzos, cantzos@usc.edu

Received 9 October 2007; Revised 17 February 2008; Accepted 20 May 2008

Recommended by Q.-J. Fu

Most audio compression formats are based on the idea of low bit rate transparent encoding. As these types of audio signals are starting to migrate from portable players with inexpensive headphones to higher quality home audio systems, it is becoming evident that higher bit rates may be required to maintain transparency. We propose a novel method that enhances low bit rate encoded audio segments by applying multiband audio resynthesis methods in a postprocessing stage. Our algorithm employs the highly flexible Generalized Gaussian mixture model which offers a more accurate representation of audio features than the Gaussian mixture model. A novel residual conversion technique is applied which proves to significantly improve the enhancement performance without excessive overhead. In addition, both cepstral and residual errors are dramatically decreased by a feature-alignment scheme that employs a sorting transformation. Some improvements regarding the quantization step are also described that enable us to further reduce the algorithm overhead. Signal enhancement examples are presented and the results show that the overhead size incurred by the algorithm is a fraction of the uncompressed signal size. Our results show that the resulting audio quality is comparable to that of a standard perceptual codec operating at approximately the same bit rate.

1. INTRODUCTION

Audio compression formats such as MPEG 1 Layer III (MP3, [1]) may create audible coding artifacts when they encode audio at very low bit rates. Content compressed at higher bit rates will have sufficient sound quality, but can become prohibitively large to transmit or store. As compressed audio moves into the dominant position among source content that is played over high fidelity consumer and home theater systems, there is an increased need for enhancing low bit rate audio data without imposing excessive storage or transmission requirements. In this work, we propose a novel method that improves the quality of compressed audio and builds on our previous work in audio resynthesis [2]. In audio resynthesis, one channel of a multichannel audio segment (target signal) can be recreated from another channel (source signal) of the same audio segment using a linear function determined by a small set of parameters.

Audio resynthesis can be easily applied to compressed audio enhancement [3] by viewing a low bit rate compressed segment as the *source signal* and its uncompressed version as the *target signal*. We define two terms in order to further describe our algorithm, namely, the *transmitter* and the *receiver*. The transmitter has access to both the source and target signals. The receiver has access to the source signal only and needs to acquire a quality enhanced signal. As depicted in Figure 1, the transmitter derives the small parameters set (linear function) and sends it to the receiver. The derivation of the small parameters set is based on a statistical conversion between the source and target signals. The receiver, in turn, applies this parameters set on the source signal to create the enhanced signal. Naturally, the enhanced signal should be of better audio quality than the source (i.e., compressed) signal.

This approach is similar to the MPEG-4 Scalable Lossless Coding (SLS) [4] methodology according to which a compressed signal is enhanced by transmitting additional data to the receiver. The difference though between our method and the MPEG-4 SLS method is that the latter uses a residual coding approach. In residual coding, a residual signal between the compressed and uncompressed signals is generated and subsequently entropy-coded. In MPEG-4 SLS, the residual signal lies in the integer modified discrete cosine transform (IntMDCT) [5] domain and the preferred transform codec for encoding IntMDCT data is the advanced audio coding (AAC) [6] scheme. Our method does not employ the residual coding approach but instead it converts a compressed signal into its uncompressed version through a statistical conversion function applied strictly on the compressed signal. Therefore, no residual or difference signal between the compressed and uncompressed signals is generated.

A key characteristic of the MPEG-4 SLS standard is the scalability property which enables the coding and transmission of the enhancement data at variable bitrates, thus allowing for variable audio quality improvement. This scalability is achieved by creating additional encoding stages for the residual signal. In each stage, a subset of the residual signal is encoded through bit-plane coding schemes based on the Bit-Plane Golomb Code (BPGC) [7]. The MPEG-4 SLS method and previous methods on scalable encoding [8–10] generally follow a cascaded residual coding approach, in which the residual signal of one encoding stage is the input to another encoding stage. Since we do not employ this residual coding approach, we have devised an alternative scalability scheme that works in conjunction with our conversion-based method and allows for variable bit rate during transmission of the enhancement parameters.

In comparison, even though the MPEG-4 residual coding technique with bit-plane coding is very efficient in terms of transmission rate and audio quality delivered, it enhances signals created by transform codecs only (and particularly by the AAC) as it operates directly on a transform domain (e.g., IntMDCT). Therefore, it cannot be readily applied for enhancement of audio signals of arbitrary compression formats. On the other hand, our conversion-based enhancement method works as a general postprocessing stage of any codec and can be applied directly on pulse code modulation (PCM) data. This means that the type of codec that generated the compressed signal is not important. In essence, the audio enhancement algorithm presented in this work has the advantage of interoperability across various compression formats.

The most direct application of a scheme like this is the case of internet audio transmission in which the transmitter side has access to the uncompressed target signal—and consequently to the compressed source signal—while the receiver end has access only to the compressed source signal which it wishes to enhance. The advantage of our method in this scenario is that the transmitter exploits the availability of the source signal at the receiver to reduce the size of the transmitted parameters set. To clarify this more, let us assume that the receiver has a 32 kbps MP3 file and wishes to acquire an enhanced file with quality similar to the 64 kbps MP3 file. Normally, the receiver would request the whole 64 kbps MP3 file and the fact that the receiver already possesses the corresponding 32 kbps MP3 file would not affect the transmission size. We show later that our algorithm can actually enhance a lower bit rate MP3 signal to a signal with quality corresponding to a higher MP3 bit rate by transmitting parameters of smaller size than the size of the higher bit rate MP3 signal. As mentioned previously, in contrast to the MPEG-4 method, the compression format of the source signal can be arbitrary allowing for the enhancement of signals created by any internet audio compression format.

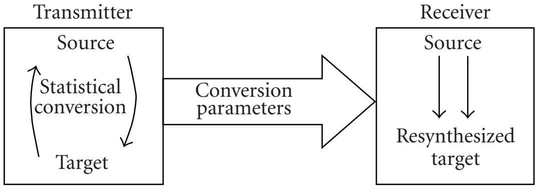


Figure 1: The audio resynthesis scheme. The transmitter has access to both source and target signals and derives the conversion parameters which are sent to the receiver. The receiver resynthesizes the target signal using the source signal and the conversion parameters.

A second scenario, which does not necessarily involve transmission over a medium, is to derive the statistical conversion parameters using the compressed (source) and uncompressed (target) versions of the signal and then discard the uncompressed signal while storing the small conversion parameters set and the source. In this scenario, our algorithm operates similarly to a codec and the fact that the receiver (i.e., where the source file and conversion parameters are stored) already contains the source file does not provide any advantages over regular compression schemes. Nevertheless, as shown later, our algorithm's performance is comparable to the MP3 scheme at equal total bit rates and this is despite the fact that we have not used a psychoacoustic model. This fact alone suggests that our method's performance can be further improved. In addition, this scheme can operate on any compression format of the source signal.

The method in this paper is different compared to recent enhancement methods such as spectral band replication (SBR) [11] and other bandwidth expansion techniques. It basically performs enhancement on all subbands (low and high frequencies) that are degraded and not just on the high frequencies. However, our algorithm could be modified to work in a similar philosophy as SBR by applying statistical conversion between the low- and high-frequency subbands of the same signal in order to resynthesize *strictly* the high subbands with less information. Normally, in this case the low subbands of the signal are already of high-audio quality and would not be replaced but instead they would be used in conjunction with a conversion function to replace only the high-frequency subbands. Furthermore, our current enhancement method can be easily combined with structured audio compression methods [12] because statistical conversion is applied between relatively long signal segments which may exhibit repetition patterns. In such scheme, the conversion function would be replicated at the receiver. The aforementioned capabilities of the enhancement algorithm support our claim that there is great potential for further improvement.

In every case, the purpose of the algorithm is to convert a low quality compressed piece to a higher quality piece with the least amount of conversion parameters. The derived conversion parameters would either be transmitted or stored, depending on the application. Optimally, the resynthesized signal should be identical to the target signal while the source signal size and the conversion parameters size added together would be much smaller than the target signal size.

Early work on audio resynthesis [2] has been based on widespread voice conversion algorithms [13–15] and the terminology used here (i.e., source, target, conversion parameters) is often found in voice conversion schemes. The basic assumption made by these algorithms is that the spectral parameters are of Gaussian nature and hence are modeled by a Gaussian mixture. This greatly facilitates the Maximum Likelihood (ML) parameters estimation since the popular expectation-maximization (EM) algorithm [16] can be applied. A new approach on modeling the cepstral coefficients was introduced in [17] by employing the Generalized Gaussian mixture model. This model is very flexible and incorporates a large number of distributions, including the Gaussian. The advantage of using the generalized Gaussian distribution [18] over the Gaussian is that the former is a more general function, and as a result fewer mixtures might be used for estimating the probability density function of the actual data.

A novel technique related to residual processing [19–21] is also implemented. In many cases of low bit rate compressed sources, reconstruction in the cepstral domain is not adequate for distortion-free enhanced audio. For this reason, we employ a method for residual conversion by applying statistical conversion on the residual vectors as well. In [3, 17], we proved the importance of cepstral smoothing and model overfitting. As we show later, we have improved on this step by using a more efficient, sorting transformation of the data that allows us to use fewer mixture components and thus reduce the conversion overhead. It is based on our previous work in multichannel audio [22] and it is modified here to operate on compressed audio enhancement. This technique, which is also extended on residual conversion, leads to a more accurate data conversion and it significantly improves the audio quality in the enhanced music piece. As an additional advantage, the sorting transformation greatly facilitates the quantization efficiency of the final conversion parameters. Through the implementation of a new special inverse transformation, the amount of information required to invert the sorting transformation is reduced. In addition, a new mechanism for varying the transmission rate of this information is developed, enabling scalable audio enhancement. The appropriate use of the discrete cosine transform (DCT) [23] and singular value decomposition (SVD) [24] are also described. The whole algorithm is applied and tested on the enhancement of 10 mono 32-kbps and 10 mono 64-kbps MP3 files. At this point, we mention that to our knowledge, we are the first group that attempts to address audio enhancement by applying statistical conversion methods.

The remainder of this paper is organized as follows. In Section 2, we describe the core component of the algorithm which is the statistical conversion of features. In Section 3, the residual conversion method is described, as an extension of the statistical conversion algorithm of Section 2. In Section 4, we present the methods on reducing the conversion parameters size and specifically the sorting transformation method and its inverse. The specific implementation details are presented in Section 5. In Section 6, the audio enhancement results are presented and evaluated in order to demonstrate the improvement of the resynthesized signal. In Section 7, concluding remarks on the algorithm are made.

2. CEPSTRAL STATISTICAL CONVERSION

The approach followed is based on previous statistical conversion algorithms related to speech synthesis [13–15]. Usually, these algorithms treat only the spectral features, while as we discuss in this and following sections, our algorithm converts both spectral features and residual data. In our work, the short term spectral features used are the LPC cepstral vectors [25]. The LPC analysis is carried out in overlapping frames through a sliding window and hence each frame is modeled as an AR filter excited by a residual. We extract the LPC cepstral vectors of the target (which is unknown at the receiving end) and source signals. Our goal is to modify the cepstral and residual vectors of the source signal so that they become close in the least squares sense to the target cepstral and residual vectors of the same music piece. This is accomplished by deriving a mapping function that will convert each of the source cepstral/residual vectors to the target cepstral/residual vector of the same time frame (the source and target signals are time-aligned). The function is assumed linear and will be fully determined by a small set of parameters.

In order to implement the conversion function, we assume that the source cepstral vectors are realizations of a random process characterized by a probability density function (pdf). The estimation of the pdf parameters is referred to as system *training* (i.e., not including the extraction of the conversion parameters). The audio segment used during training (*training set*) is chosen so that it is capable of modeling a large and diverse number of audio pieces. In this paper, we call source and target signals the particular signals on which we apply the conversion scheme and derive the conversion function. The data used for the training set is generic and has no association with the source and target signals so that the system training does not depend on the particular signal under enhancement and does not have to be repeated each time a new source signal is processed. Once derived, the training parameters (i.e., the pdf) are stored permanently in both the transmitter and receiver sides since they will be part of the conversion function.

2.1. Generalized GMM and cepstral modeling

In the majority of current statistical conversion algorithms, a common assumption is that the spectral features are of Gaussian nature and hence the Gaussian mixture model is employed. The Gaussian mixture model has been treated in numerous other applications and an algorithm to estimate its parameters, (EM) is readily available. A more general model is adopted in this work that better models the (non-Gaussian) cepstral vector properties, which includes the Gaussian mixture as a subcase, and is called the generalized Gaussian mixture. Its component pdf, the Generalized Gaussian pdf, is more flexible and adapts to virtually any unimodal distribution. Its analytical form for a random variable z is

$$g(z;\mu,\sigma,\alpha) = \frac{\alpha\beta}{2\sigma\Gamma(1/\alpha)}\exp\left[-\left|\beta\frac{(z-\mu)}{\sigma}\right|^{\alpha}\right], \qquad (1)$$

where μ is the mean, σ is the variance, α is the shape parameter, $\Gamma(\cdot)$ is the Gamma function, and β is a dependent parameter

$$\beta = \left[\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}\right]^{1/2}. \qquad (2)$$

If $\alpha = 2.0$, we have the Gaussian pdf; and if $\alpha = 1.0$, we have the Laplace pdf. When $\alpha \gg 1$, the distribution tends to the uniform pdf and when $\alpha < 1$, the distribution becomes impulsive.

We consider the training cepstral vectors (and the testing source vectors) to be generated by a mixture with component pdf as described in (1). The mixture formulation of the generalized Gaussian case (with diagonal covariance matrices) is

$$G(\mathbf{x}) = \sum_{k=1}^{K} p(C_k)\prod_{j=1}^{q} g\left(x^{(j)};\mu_k^{(j)},\sigma_k^{(j)},\alpha_k^{(j)}\right), \qquad (3)$$

where C_k denotes the cluster (component) k, K is the number of clusters, and $p(C_k)$ denotes the prior probability of cluster k. The cepstral vector is q dimensional where q is the cepstral order and the jth coefficient or coordinate is denoted by $x^{(j)}$. The vector coefficients are considered to be independent and thus the joint pdf is the product of the q coefficient pdf's. This diagonal formulation is favorable since it decreases the computational complexity during implementation.

2.2. Mixture parameter estimation and clustering

The inclusion of a third independent parameter compared to the Gaussian case (the shape parameter α), incurs additional complexity when it comes to maximum likelihood (ML) estimation of the pdf parameters. This problem becomes more evident in a mixture pdf, where the number of parameters to be estimated increases, and consequently the computational complexity and the time needed for convergence increase as well.

In this work, we follow a different approach than the one used in the conventional mixture estimation methods, by clustering the vectors and focusing on each cluster separately. This divides the parameters estimation task into K simpler tasks. In order to perform this decomposition, we employ fuzzy clustering techniques through the c-means algorithm [26], and cluster the training vectors into K groups. The c-means is known to avoid local minima better than the k-means and it also provides a "fuzziness" option that regulates the occurrence of outliers.

The next step is to perform ML estimation on each cluster. The estimation is now straightforward because the mean for each component is known (it is the cluster center). We also compute $p(C_k)$ as the number of vectors that belong to cluster k divided by the total number of vectors. The ML estimator for the shape parameter $\alpha_k^{(j)}$ of cluster k and coordinate j is given by [27]

$$\begin{aligned}&\frac{\psi\left(1/\alpha_k^{(j)}+1\right)+\log\left(\alpha_k^{(j)}\right)}{\alpha_k^{(j)^2}}\\&\quad+\frac{1}{\alpha_k^{(j)^2}}\log\left(\frac{1}{n_k}\sum_{t:\mathbf{x_t}\in C_k}\left|x_t^{(j)}-\mu_k^{(j)}\right|^{\alpha_k^{(j)}}\right)\\&\quad-\frac{\sum_{t:\mathbf{x_t}\in C_k}\left|x_t^{(j)}-\mu_k^{(j)}\right|^{\alpha_k^{(j)}}\log\left(\left|x_t^{(j)}-\mu_k^{(j)}\right|\right)}{\alpha_k^{(j)}\sum_{t:\mathbf{x_t}\in C_k}\left|x_t^{(j)}-\mu_k^{(j)}\right|^{\alpha_k^{(j)}}}=0,\end{aligned} \qquad (4)$$

where n_k is the number of vectors that belong to class C_k and $\psi(\cdot)$ is a function given by

$$\psi(\tau) = -0.5777 + \int_0^1 \left(1-t^{\tau-1}\right)(1-t)^{-1}dt. \qquad (5)$$

The expression in (4) is solved by iterative methods. The variance parameter $\sigma_k^{(j)}$ of the kth cluster and jth coordinate is then estimated as follows [27]:

$$\sigma_k^{(j)} = \left[\frac{\alpha_k^{(j)}\beta^{\alpha_k^{(j)}}\sum_{t:\mathbf{x_t}\in C_k}\left|x_t^{(j)}-\mu_k^{(j)}\right|^{\alpha_k^{(j)}}}{n_k}\right]^{1/\alpha_k^{(j)}}. \qquad (6)$$

Note that the zeroth cepstral coefficients (energy coefficients) are neglected because they introduce strong bias during parameter estimation. The frame energy information (relative to the other frames) is given by the LPC gain factors which are transmitted as side information.

2.3. Conversion function and conversion parameters set

The conversion function $F(\cdot)$ acts on the source vector sequence $[\mathbf{x_1},\ldots,\mathbf{x_n}]$ and produces a vector sequence close in the least squares sense to the target sequence $[\mathbf{y_1},\ldots,\mathbf{y_n}]$. Since we have selected a diagonal implementation, this function will act on the individual vector components and minimize the error

$$E = \sum_{t=1}^{n}\sum_{j=1}^{q}\left|y_t^{(j)}-F\left(x_t^{(j)}\right)\right|^2, \qquad (7)$$

as in [13]. To address this problem, we consider F as piecewise linear, that is,

$$F(x_t^{(j)}) = \sum_{k=1}^{K} P(C_k|\mathbf{x}_t)\left[v_k^{(j)} + \frac{u_k^{(j)}}{\sigma_k^{(j)}}(x_t^{(j)} - \mu_k^{(j)})\right] \quad (8)$$

for $t = 1, \dots, n$ and $j = 1, \dots, q$. The conditional probability that a given vector belongs to cluster k, $P(C_k|\mathbf{x}_t)$, is given by

$$P(C_k \mid \mathbf{x}_t) = \frac{p(C_k)\prod_{j=1}^{q} g(x_t^{(j)}; \mu_k^{(j)}, \sigma_k^{(j)}, \alpha_k^{(j)})}{G(\mathbf{x}_t)}. \quad (9)$$

The unknown parameters set $[\mathbf{v}, \mathbf{u}]$ can be found by minimizing (7) which reduces to solving a typical set of q independent least-square equations [13], and hence the linear conversion function F is fully determined. Notice that in the nondiagonal case and when F is in vector form [13], the unknown parameter corresponding to $u_k^{(j)}$ is actually a nondiagonal square matrix with q^2 elements. In the diagonal case, this matrix is diagonal with q elements $u_k^{(j)}$ on the diagonal. Therefore the diagonal formulation is also preferred over the full covariance method because of the significantly smaller size of the unknown parameters.

We call the set $[\mathbf{v}, \mathbf{u}]$ the *conversion parameters set*. These are the only parameters that have to be transmitted for audio enhancement since they are dependent on the particular source and target signals. The remaining parameters of (8) are part of the mixture model and—as mentioned previously—they are already stored at the receiver.

3. RESIDUAL MODELING AND CONVERSION

In practice, accurate cepstral reconstruction is not sufficient for acoustically undistorted enhancement of MP3 compressed segments. Especially in the case of a very low bit rate source (e.g., 32 kbps MP3), many audible artifacts are present in the source signals compared to the target signals. Instruments that are inaudible in the source signal will usually appear in the enhanced signal as distortions since the LPC coefficients alone fail to represent them. In such cases, the signal differences lie mainly in the residuals and therefore some residual processing is essential for better enhancement results.

We adopt the assumption that the residual vectors are correlated with their corresponding cepstral vectors [28] and thus share similar statistical properties. Therefore, we can apply the statistical conversion method described in the previous sections to the residual vectors also. The probabilistic model used here is the same used for cepstral conversion (i.e., it is derived from the training cepstral vectors). However, the dimensionality of the residual vectors is much higher than that of the training cepstral vectors, and therefore we have to divide them into subvectors of dimensionality equal to or lower than that of the training cepstral vectors. For instance, in the case of 30 training cepstral and 52 residual coordinates (i.e., coefficients), we would divide the residual vectors into two subvector sets of 30 and 22 coordinates each and apply statistical conversion in each subvector set separately. The 30 coordinates set would use the complete training cepstral model while the 22 coordinates set would use a truncated cepstral training model (i.e., without the last 8 training cepstral coordinates). The validity of this technique is proven in the results section.

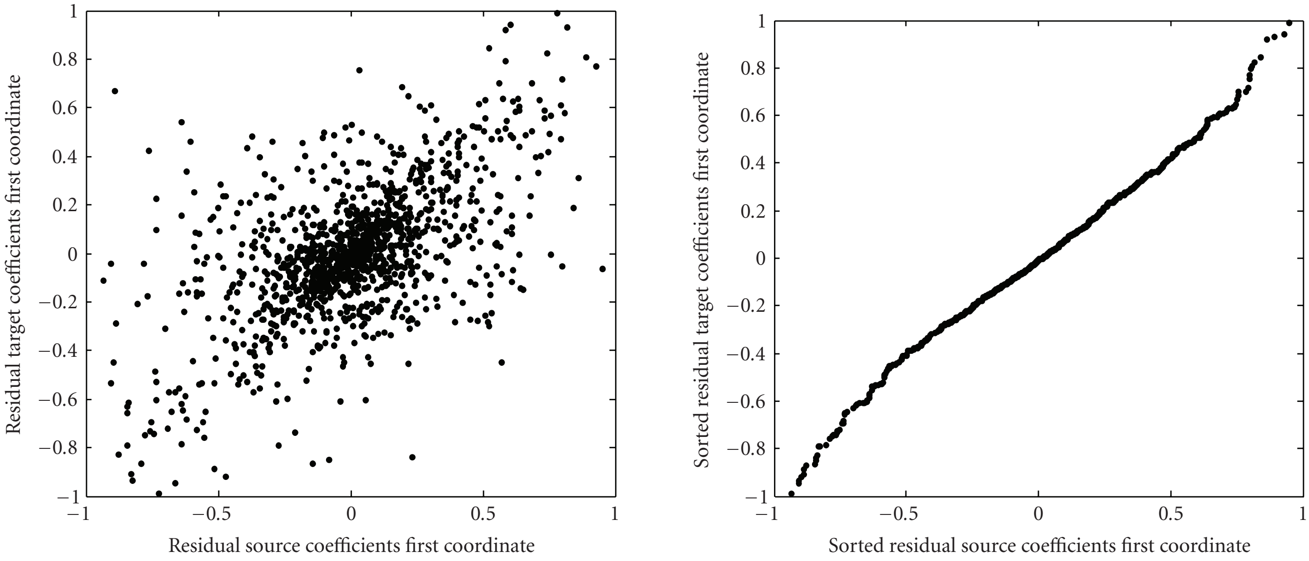
Clearly, we do not expect a residual reconstruction with accuracy as high as that of cepstral reconstruction because the residuals are, in essence, noise signals. We have not derived a training set or a probabilistic model specifically for the residual vectors since the extremely high residual vector dimensionality would make this impractical. Furthermore, we would have to design a global mixture pdf that could efficiently model any set of testing residual vectors even though these are highly diverse and contain the fine details of the signal. However, in our experiments, we found that using the mixture pdf derived from the training cepstral vectors results in converted residuals that are much closer to the target residuals (than the source residuals). In addition, the residual reconstruction accuracy attained is high enough to provide distortion-free audio signals. Nevertheless, the next section describes a novel technique to improve even more both cepstral and residual conversion performance.

4. CONVERSION OVERHEAD REDUCTION

Accurate cepstral and residual conversion is a challenging task. A new technique, based on [22], is introduced here that can significantly increase cepstral and residual conversion accuracy, while reducing the conversion parameters size. We sort the source and target vector coefficients (cepstral or residual) along each coordinate in ascending order as shown in Figure 2. The motivation behind the sorting transformation is found in the form of the conversion function. The conversion function is a (piecewise) linear estimator that estimates the target features from the source features. Its optimal performance is achieved when the true relation between the source and target features is linear along each coordinate. Clearly, if the source and target features formed straight lines along each coordinate, the dependence among the two would be linear and the conversion function would need only one mixture group (a simple linear estimator), leading to zero reconstruction errors.

By sorting the source and target features along each coordinate we can achieve a relation between the source and target features that is very close to linear, as shown in Figure 2(b) for the first coordinate. The relation without sorting between the source and target features along the first coordinate is plotted in Figure 2(a) and it is clear that the number of mixture classes required to model is large. 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. In Section 6, we show that this method significantly reduces the conversion errors when compared to our previous method [3] in which no sorting is applied and the number of mixture groups is large.

In order for the receiver to be able to use the resynthesized cepstral or residual coefficients and create the final signal, the original order of the resynthesized coefficients has



(a)

(b)

Figure 2: Residual target coefficients against residual source coefficients for the first coordinate before sorting (a) and after sorting (b). The data on the right can be easily manipulated by a piecewise linear estimator.

to be known.The reason for this is that once the receiver resynthesizes the cepstral and residual data, they would be sorted in ascending order since the source and target features were also sorted during the derivation of the conversion function at the transmitter. The source signal is available at the receiver side, and thus the original order of the source cepstral and residual coefficients is known. Our focus is on deducing the original order of the target features and using that info to reorder the resynthesized features at the receiver end. We call this information the *sorting information* and it is transmitted along the conversion parameters. These two sets combined form the *transmitted parameters*.

The straightforward solution would be to transmit the original order of the target cepstral and residual data as side information along with the conversion parameters. At the receiver, the coefficients would be resynthesized one by one and a side index would determine where to place the particular cepstral/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). This sorting information together with the conversion parameters set is still smaller when compared to the conversion parameters set acquired from conversion without sorting and with more mixture classes. Nevertheless, there is an even more efficient way—in terms of bits transmission—to transmit the desired sorting information. Instead of directly transmitting the sorting indices of the target features to the receiver, 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 features 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 1 we give the original positions of the first 10 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. What we are actually missing is the target column (and this is what we would normally transmit) since the source column is known to the receiver. An algorithm that allows us to transmit less information to the receiver without explicitly sending every index of the target column is the following.

(1) The transmitter checks if the source and target indices of the particular row are the same. If yes, then a zero is transmitted. If no, then proceed to the next step.

(2) The transmitter looks in the target index of the current row and finds the position (row) of that index in the source column. The distance between the current row and the new row is transmitted. The value in the new row of the source column is inserted in the current row of the source column and all values in that row and higher are shifted by one position towards the end of the column.

(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 2. Note that in this example some source indices cannot be illustrated because they are further down the source column (but still near row 10). This lossless operation will enable us to send fewer bits at the receiver, especially after we perform lossless coding such as run-length, Huffman, and

Table 1: Original positions of sorted source and target features for a random set.

Source indices	Target indices
126	126
74	74
43	43
19	19
93	99
90	93
100	45
99	55
54	90
67	100

Information that is available at the transmitter. The receiver has access only to the left column. The values in each row are the original positions of the sorted source and target features.

Table 2: Original positions of sorted target features for a random set and actual transmission.

Target indices	Actual transmission
126	0
74	0
43	0
19	0
99	3
93	0
45	7
55	5
90	0
100	0

Original target positions and the information that is actually transmitted. Values after row 10 are not shown but the actual number of rows is 256.

Lempel Ziv. More results on the sorting information size are given in Section 6.

It is expected that the linear system comprising of (7), (8), and (9) is ill-conditioned, especially since the system matrix is large. A well-known approach to deal with ill-conditioned systems is based on the pseudoinverse instead of the exact inverse matrix, which can be computed based on SVD. We apply this approach for calculating the inverse of the correlation matrices that are encountered during the conversion parameters derivation [13]. The conversion parameters that are created using SVD are more robust to quantization errors, even though they do not always yield an exact solution for the linear system.

As the final step of our method, we transform the derived conversion parameters set using the DCT and the transformed parameters are then finally quantized. This step has demonstrated a noticeable decrease in cepstral and residual reconstruction errors and has allowed us to quantize the conversion parameters with 14 bits or less without audible artifacts.

5. IMPLEMENTATION

The algorithm described previously was implemented and tested on 10 mono audio pieces of 6.4 seconds duration each. These audio pieces comprise of four generic music pieces and six single instrument pieces from the EBU-SQAM [29] testing database. The generic music pieces are one Rock piece with a male singing voice, one Jazz piece, one Electronic piece with very high energy at high frequencies, and one Classical music symphony piece. The six SQAM pieces are taken from a flute, a violin, a piano, a double bass, drums, and a harp instrument. In all 10 cases, we attempt to enhance a 6.4-second source segment which is MP3 encoded (with LAME) at a 32 kbps constant bit rate or a 64 kbps constant bit rate. The testing target segments are the corresponding uncompressed (WAV) versions of the same audio segments. The source and target pieces in all examples are time-aligned and since the algorithm is applied in a post-processing stage, the MP3 sources are also converted to a WAV format (PCM data).

5.1. *Critical band analysis*

The first stage of the algorithm implementation is to apply a subband analysis on the source and target signals as well as on the training set. A popular method for subband analysis uses wavelet filters which achieve perfect reconstruction [30] but other perfect reconstruction filterbanks can be used instead. A suitable wavelet candidate is the Daubechies [31] wavelet filter of order 40, which achieves a very efficient subband separation.

For the source and target signals, several different wavelet tree structures were tested (e.g., equidistant subbands) but the most successful structure proved to be one that emulates the critical bands of the human hearing system as in [32]. It assigns increased frequency resolution to the range 20 Hz–5.5 kHz in which human hearing is most sensitive. In addition, the large number of subbands selected allows us, as we show later, to take advantage of the interband redundancy and also to process accurately the subbands that are the most significant (i.e., the ones that are more degraded or carry the perceptually important parts of the signal). The actual wavelet filterbank is shown in Figure 3 and is applied to both source (compressed) and target (uncompressed) signals leading to 17 source and target subbands. Note that each time a signal is wavelet-filtered, the resulting two signals are decimated by a factor of two (i.e., critically sampled). The training set is also separated in subbands but for reasons explained in the next subsection, the subband tree is different than that of the source and target signals.

5.2. *The training set*

An important part of the algorithm is to derive a generalized Gaussian mixture pdf that does not have to adjust to the particular testing music piece. This probabilistic model should be global in the sense that it will include the statistical properties of all possible audio pieces and both transmitting and receiving ends will have access to it (e.g., prestored in

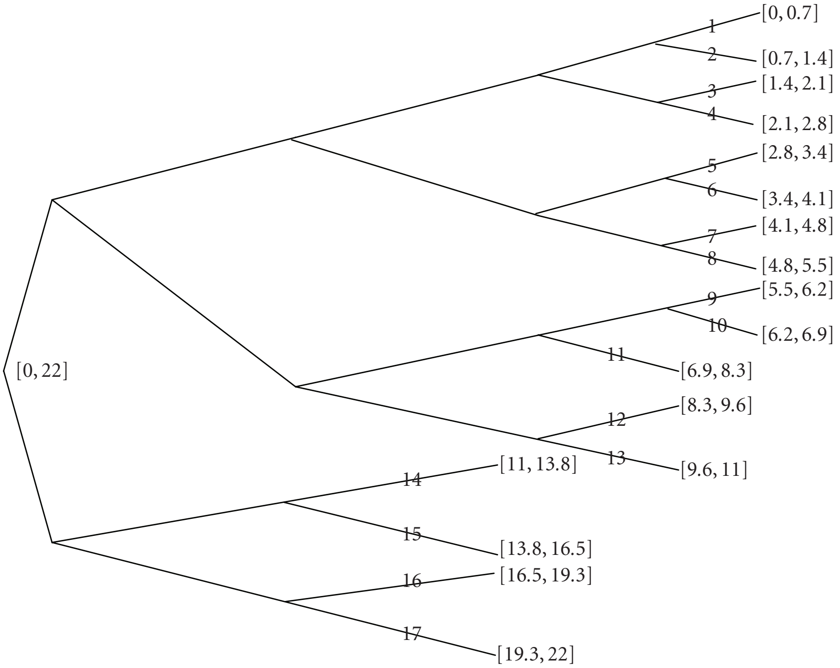


Figure 3: Wavelet tree structure used for subband analysis of the source and target signals. Numbers in brackets indicate the frequency region in kHz in each subband. Numbers on leafs indicate the subband index from 1 to 17.

both sides). This does not mean that the mixture pdf will accurately model any of the particular testing music pieces but rather capture the main (subband-specific) statistical properties of the testing source cepstral vectors. In essence, we ensure that the conversion function acquires appropriate mixture model parameters (i.e., cluster means and variances) so that the conversion parameters derivation is not ill-conditioned.

Several candidate training sets were processed to produce a mixture pdf among which were the multichannel training set of [2] (1 minute of an orchestra recording), a white noise training set, a Brownian noise training set, and a pink noise training set. Pink noise proved to be the most suitable training set and led to smaller cepstral reconstruction errors (up to 5% less in all subbands compared to the other sets) during enhancement of the 4 generic music pieces. The power spectrum of pink noise is proportional to $1/f$, where f is the frequency. An approximation to pink noise can be created by starting from the discrete Fourier transform (DFT) magnitude (taken as proportional to $1/f^{1/2}$), adding uniformly distributed random phase and applying the inverse DFT (real part).

In order to reduce the training model size and allow for the data diversity needed in the case of many mixture components ML estimation, we divide the training data set into 4 large equidistant subbands (instead of 17 subbands) covering the frequency range 20 Hz–22 kHz. Each training subband consists of 12 000 cepstral vectors of cepstral order 30. ML parameters estimation, as described in Section 2, is performed on each training subband separately. The training procedure is shown in the flow diagram of Figure 4.

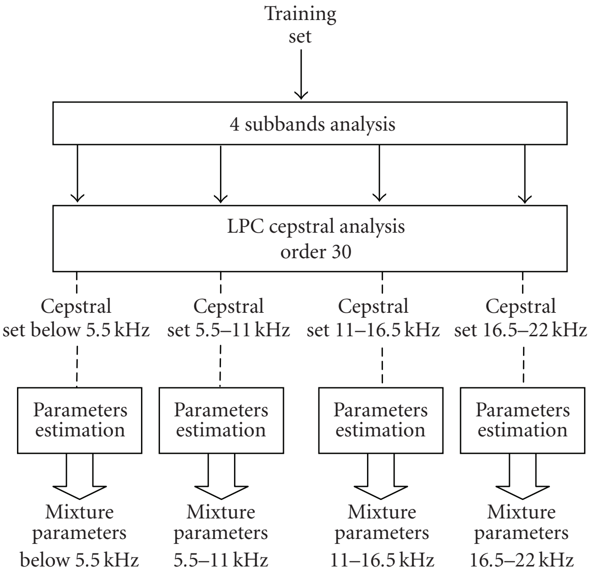


Figure 4: The training scheme for the pink noise set. The training set is separated into 4 equidistant subbands and for each one the LPC cepstral vectors are extracted. Mixture parameters estimation is performed on each cepstral set and the mixture parameters of each training subband are derived.

In Figure 5, the validity of the estimation algorithm, as described in Section 2, is illustrated. Even though a generalized Gaussian mixture model of 40 groups is recommended (as determined by the MDL [33] and AIC [34, 35] criteria),

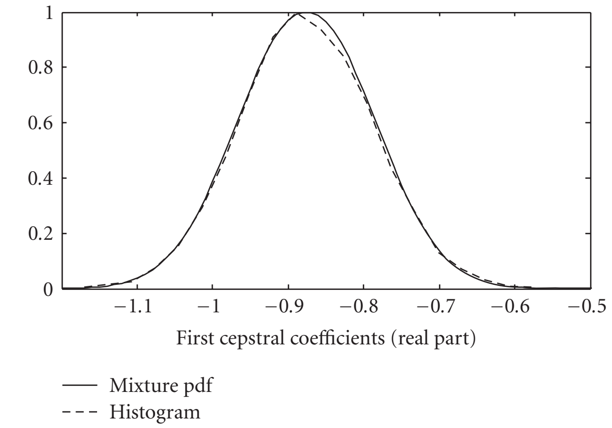


Figure 5: Fitting of mixture pdf (4 groups) to the normalized histogram of the first cepstral coefficients of the band 20 Hz–5.5 kHz of the pink noise training set.

we decrease this number to 4 for all 4 training subbands. The fitting of the mixture pdf to the histogram is still accurate.

5.3. *Signal enhancement*

The signal enhancement procedure that takes place at the transmitter side for a source (compressed) and target (uncompressed) signal is shown in Figure 6. The source and target signals are separated into 17 subbands as mentioned before. The resulting subband signals are LPC analyzed and the LPC cepstral, and residual vectors are extracted. The cepstral order for subbands 1–13 is 8 while for subbands 13–17 it is 15, accounting for the larger frequency bandwidth of the high subbands. Generally, not all subband signals and not all vectors within each subband require cepstral or residual conversion as explained later and this is the role of the subband and vector selection that occurs right after the LPC analysis. The selected source and target vectors (cepstral or residual) of the selected subband are then sent for statistical conversion. However, the statistical conversion process requires the mixture model parameters and each selected subband acquires these model parameters from one of the 4 larger training subbands that it is part of. This is shown in Figure 6 as the role of the training subband switch. Statistical conversion can now be performed for the selected subband i and the corresponding conversion parameters and sorting information are derived as described in Sections 2 and 3. These are finally transmitted to the receiver as the transmitted parameters. During cepstral conversion, the cepstral order of the training model is truncated appropriately for each source/target subband to adjust to the lower cepstral order of the particular source and target cepstral vectors (8 or 15). The reason for this is that the source cepstral vectors are assumed to be generated by the mixture pdf derived during training and the dimensionality of the training and particular source cepstral vectors should be the same.

At the receiver side, the compressed source signal is separated into 17 subbands and LPC analyzed in the same way as at the transmitter side. The transmitted parameters are applied oneach source subband signal that was selected for statistical conversion during the parameters extraction procedure of Figure 6. Specifically, the conversion parameters are used to convert each of the selected sorted source subband vectors (cepstral or residual) to the resynthesized ones and the sorting information is used to rearrange them to their correct order. After the resynthesized subband vectors are created, we perform LPC synthesis on each resynthesized subband to produce the time domain subband signals. These signals along with the source subband signals that were not converted are combined through wavelets (i.e., inverse of the 17 subbands separation) and the final time domain resynthesized signal is created.

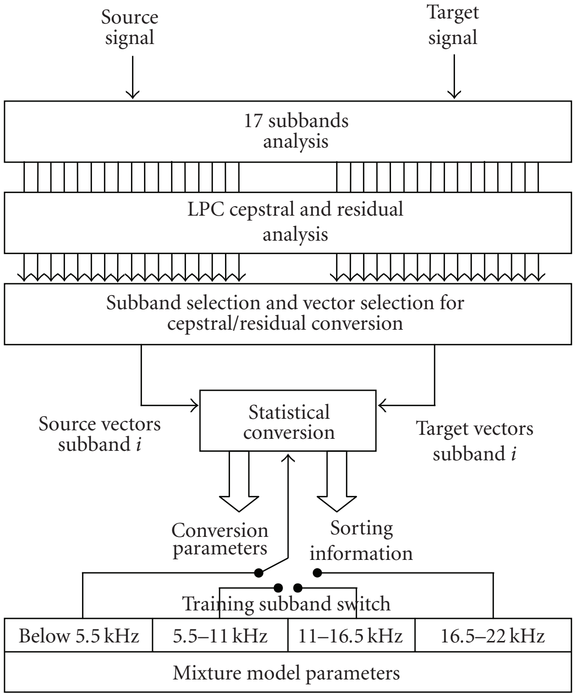


Figure 6: The transmitted parameters extraction procedure that takes place at the transmitter side. The source and target signals are both separated into 17 subbands and the LPC cepstral and residual vectors are extracted. The subband selector selects which subbands and which vectors within each subband require cepstral/residual conversion. At the final stage, each of the 17 analysis subbands will be classified to one of the four frequency intervals used during training. Under this classification, the conversion parameters are extracted for the source and target signals, using one of the four trained pdfs.

5.3.1. *Phase redundancy*

In the low subbands particularly, it has been observed that the signs of the source and target vectors are mostly the same. To take advantage of this observation, we take the absolute value of the target vectors before the derivation of the conversion function. The absolute value of the target vectors can be estimated more accurately compared to the raw target data. To recover the lost sign of the resynthesized vectors at the receiver, a 1-bit sequence is included along with the sorting information. For every source-target coefficient pair, a 0 is sent if the signs are equal, otherwise a 1 is sent. For subbands below 5.5 kHz, the proportion of 1's in each sign sequence is between 1% and 10% while in the higher subbands it tends towards the expected 50% value. This means that in the high frequencies there is little phase redundancy.

5.3.2. *Intraband redundancy*

Further redundancies can be found in the time domain for each subband signal pair that goes through cepstral or residual conversion. The differences between the samples of the source and target subband signals do not carry the same significance in terms of resulting audio quality and in some cases the source and target subband samples of a particular time frame are almost identical. For instance, the first subband of Figure 3 for a 32-kbps bit rate MP3 signal is usually not severely degraded and many source-target sample pairs can be neglected during conversion. Another example is the silence regions that occur in a speech signal.

For the subbands that preserve their energy in the source signal (and thus are not severely degraded), we adopt a threshold rule based solely on the source signal information. According to this rule, the source subband samples that have absolute value below a certain threshold are neglected from conversion. The rationale behind this is that source subband samples with relatively smaller amplitude either correspond to less audible parts of the subband target signal or they have been suppressed as the particular codec that carried the compression of the source classified them to be perceptually insignificant. The advantage of this method is that the receiver knows which samples have been discarded, as long as the threshold for each processed subband is transmitted, which is side information of negligible size. The disadvantage of this method is that other samples that are perceptually irrelevant are not detected. This is a more general problem of our algorithm since, as mentioned before, it does not use a psychoacoustic model.

The samples that pass the threshold test will now form new source and target subband signals on which cepstral and residual extraction is applied for the derivation of the particular conversion function. Nevertheless, the residual vectors incur significant conversion overhead compared to cepstral conversion and thus a second, less strict, threshold rule can now be applied on the residual vectors only to further reduce the residual conversion overhead. The most significant source-target residual vector pairs are selected by locating the pairs that yield a high quadratic vector distance between them. Consequently, little side information has to be transmitted to the receiver which indicates which residual vectors in each subband are selected for conversion.

In the case that a source-target sample pair is determined to be insignificant, the source sample of that pair is used directly for signal resynthesis at the receiver, bypassing the conversion process. However, note that in the case where a source subband has lost most of its energy, and it is perceptually important, all of its samples pass through cepstral and residual conversion and no threshold rules are applied. The LPC target gains for the selected subbands are transmitted as side information since they are crucial in recovering the lost energy of the source subbands.

5.3.3. *Interband redundancy*

Naturally, not all subbands are expected to be severely degraded by the compression process and not all of them are perceptually important. A 32-kbps MP3 signal will usually sustain moderate distortion in the frequency range 20 Hz–5.5 kHz and some of the corresponding subbands can be completely neglected from residual or even cepstral conversion. On the other hand, the higher subbands are the most distorted, mainly because they are less perceptible to the human ear. The 7 highest subbands, as seen from Figure 3, are large and therefore require longer cepstral and residual vectors during LPC analysis compared to the first 10 subbands. These subbands, if selected for cepstral and residual conversion, will add considerable transmission overhead. A simple method to determine which high-frequency subbands of the source signal to process is to compare them with the subband energies of a signal that has been compressed with the same codec as the source file but at a higher bit rate such that its audio quality is roughly comparable to the expected quality of the enhanced signal. This should give us an insight into which subbands are perceptually important with the use of the codec's own psychoacoustic model.

The selected high subbands require only an approximate reconstruction such that the overall envelope of the desired subband signal is preserved. The human ear is more sensitive to the low subbands but even for these we determined that the cepstral and residual conversion, as described in the previous sections, is extremely accurate at the cost of high overhead size. For this reason, we apply a more subband-adaptive technique by increasing the degree of sorting similarity between the source vectors X and target vectors Y according to the perceptual significance of the subband they belong to. The straightforward way to achieve this is to add the source vectors to the target vectors multiple times creating a modified target set Y' which, combined with the phase redundancy observation, is shown below:

$$Y' = |Y| + cX. \qquad (10)$$

After sorting the modified target set Y', the original positions of its coefficients will be more similar to the original positions of the sorted source set X depending on how many times the source set was added to the target set (i.e., the constant c). We call constant c *the multiplier*. This modification is easily reversible at the receiver because the source set is always available and the multiplier can be transmitted as side information. The resulting sorting information size, as derived in Section 4, will be now less than $n \cdot \log_2 n$ bits and can be adjusted through the multiplier depending on the degree of enhancement desired, enabling scalable overhead transmission. As a rule of thumb, we increase the multiplier as we move to higher subbands so that we progressively decrease the reconstruction accuracy and the sorting information size.

6. RESULTS

The results of audio enhancement for the music signals are analyzed and evaluated in this section. The effectiveness of our method is shown through objective and subjective evaluation techniques. Cepstral and residual conversion accuracy is measured by taking the Euclidean quadratic distances and determining how close the reconstructed cepstral/residual vectors to the corresponding target cepstral/residual vectors are. Note that each of the source subband signals is time-aligned with the target subbands signals and the same holds for the resynthesized subband signals. Thus there is a one-to-one correspondence among the cepstral or residual vectors of the source, target, and resynthesized subband sets.

Throughout the analysis, the LPC frame length is 37.5 milliseconds and the LPC frame slide is 35 milliseconds for all 17 subbands, thus effectively eliminating the redundancy that is attributed to the overlapping frames. The number of the generalized Gaussian mixture classes is set to 4 for all music signals because the sorting transformation allows us to use few classes, as mentioned in Section 4. For the same reason, the conversion parameters are uniformly quantized with only 12 bits. Since all audio signals pass through a similar but extensive analysis, as described by our algorithm, we present the specific cepstral and residual conversion details for the Rock piece only and only for the case of 32 kbps MP3 enhancement. These results are similar for the remaining audio pieces. Objective and subjective quality test results are presented in Section 6.2 along with the corresponding transmission sizes.

6.1. *Cepstral and residual conversion results*

The cepstral and residual conversion results along each 182 vector subband for the Rock music piece are shown in Tables 3 and 4. The subbands that have been selected for cepstral or residual conversion are also shown. Notice that subbands 11–13 are fully processed because they have significant energy loss. Each entry of the tables is the average quadratic distance between either the source and target cepstral/residual vectors or between the resynthesized and target cepstral/residual vectors. The third column shows the number of vectors selected for conversion in the particular subband while the fourth column shows the multiplier as described by (10). The fifth column shows the average number of bits/coefficient for transmission of the sorting information when the full algorithm is applied. The sorting information has passed through additional lossless compression (e.g., run-length coding), as described in Section 4. The sixth column shows the average quadratic distance between the cepstral or residual vectors of the source and target subband signals. The seventh column shows the average quadratic distance between the cepstral or residuals vectors of the resynthesized and target subband signals when the full algorithm is applied (i.e., conversion with sorting and DCT) while the eighth and ninth columns show the same distance when sorting is not applied and when DCT and sorting are not applied, respectively.

Note that in the two scenarios without sorting, no sorting information is derived along with the conversion parameters (and the multiplier is 0) and therefore the total size of the transmitted parameters is smaller than that of the full algorithm scenario. Thus, in order to have roughly equal size of transmitted parameters for all three scenarios, we modify the two scenarios in which no sorting is applied by increasing the mixture classes from 4 to 20 while keeping the same number of quantization bits for the conversion parameters. This should mean a more accurate linear estimator but, as the results of Tables 3 and 4 show, our 4 classes mixture estimator is more accurate.

It is clear from both tables (seventh column) that when statistical conversion is applied along with sorting and the DCT, the reconstructed vectors are much closer to the target vectors (than the source vectors are). If no sorting is used, the cepstral errors increase more than 50% as the eighth column of Tables 3 and 4 shows, while if in addition no DCT is used these errors grow even higher as the last column of Tables 3 and 4 shows. This proves the necessity of the sorting transformation and the DCT for achieving low reconstruction errors. It is especially noticeable in the residual results where the errors are multiple orders of magnitude higher than the ones of the case in which sorting and DCT are applied. As expected, when the full algorithm is applied and the multiplier increases across the subbands, the average number of bits of the sorting information decreases.

6.2. *Transmission overhead and quality evaluation tests*

The size of the transmitted parameters for the 10 enhanced music signals in the case of 32-kbps MP3 enhancement is shown in Table 5. This size, if added to the source size, is close to the size of the corresponding constant bit rate 64-kbps MP3 file. Therefore, the audio quality of the enhanced file is expected to be at least comparable to the quality of the 64-kbps MP3 scheme. Similar transmission sizes are produced in the case of 64-kbps MP3 enhancement and, in that scenario, the enhanced signal should be at least comparable to the 96 kbps MP3 signal. We show this by performing two perceptual quality evaluation tests.

The first one is the ITU-R BS.1387 perceptual evaluation of audio quality (PEAQ) test, basic model [36]. This objective quality test measures the perceptual difference between the original signal (uncompressed in our case) and the processed one. It simulates the responses of human listeners to a real listening test by modeling the auditory system. The 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 (SDGs) from a subjective listening test [37]. Tables 6 and 7 show the ODG scores for all 10 music examples for the cases of 32-kbps and 64-kbps MP3 enhancement, respectively. In the case of 32-kbps MP3 enhancement, it is clear that the quality of the enhanced file is much higher than that of the 32-kbps

Table 3: Cepstral conversion results for the rock music signal.

Subband index	Cepstral order	Selected vectors per subband	Multiplier	Sorting information size (bits/coeff.)	Distance between source and target	Distance between resynthesized and target	Distance between resynthesized and target without sorting	Distance between resynthesized and target without sorting, without DCT
2	8	168	8	2.61	0.0182	0.0053	0.0137	0.0158
3	8	162	8	2.61	0.0259	0.0079	0.0194	0.0240
4	8	163	8	2.61	0.0222	0.0063	0.0153	0.0168
5	8	153	8	2.61	0.1255	0.0126	0.0455	0.0536
6	8	163	8	2.61	0.0733	0.0118	0.0333	0.0419
7	8	170	8	2.61	0.0252	0.0073	0.0190	0.0214
8	8	166	8	2.61	0.0439	0.0086	0.0253	0.0278
9	8	70	8	2.61	0.2489	0.0179	0.1955	1.1005
10	8	66	8	2.61	0.2208	0.0189	0.1367	1.3197
11	8	182	16	2.53	0.7676	0.0274	0.0973	0.1177
12	8	182	16	2.53	0.2695	0.0091	0.0433	0.0464
13	8	182	16	2.53	1.3958	0.0110	0.0905	0.0931
15	15	182	30	2.39	0.1161	0.0096	0.0391	0.0581
16	15	182	30	2.39	0.6692	0.0220	0.1137	0.1433
17	15	182	30	2.39	0.3000	0.0149	0.0458	0.0666

Table 4: Residual conversion results for the rock music signal.

Subband index	Vector length	Selected vectors per subband	Multiplier	Sorting information size (bits/coeff.)	Distance between source and target	Distance between resynthesized and target	Distance between resynthesized and target without sorting	Distance between resynthesized and target without sorting, without DCT
9	52	70	4	2.78	1.8670	0.0540	0.4346	0.6155
10	52	66	4	2.78	1.9462	0.0523	0.4696	1.1201
11	103	182	60	1.04	1.9266	0.2238	0.7906	1.1925
12	103	182	60	1.04	0.7779	0.2032	0.4832	0.6792
13	103	182	100	1.04	1.2707	0.2414	0.8083	1.1748

Cepstral and residual conversion results in terms of the average quadratic distance between time-aligned vectors over each subband. The fourth column shows the multiplier along each subband and the fifth column shows the resulting number of bits/coefficient of the sorting information. The sixth column shows the initial distances between source and target vectors. The seventh column shows the distance between the resynthesized and target vectors when the full algorithm is applied. The two last columns show the results when the algorithm is not applied correctly.

Table 5: Size of transmitted parameters for 32-kbps MP3 enhancement.

	Rock	Symphony	Electronic	Jazz	Flute	Violin	Bass	Piano	Harp	Drums
Source signal size (kB)	26	26	26	26	26	26	26	26	26	26
Target signal size (kB)	550	550	550	550	550	550	550	550	550	550
64-kbps MP3 size (kB)	52	52	52	52	52	52	52	52	52	52
Cepstral conversion parameters size (kB)	1.7	1.5	1.5	1.5	1.5	1.9	1.3	1.5	1.5	1.5
Cepstral sorting info size (kB)	6.5	7.1	5.6	7.7	7.1	8.7	6.8	7.5	5.6	7.7
Residual conversion parameters size (kB)	5.0	5.0	5.0	5.0	5.0	2.5	5.0	5.0	5.0	5.0
Residual sorting info size (kB)	16.6	16.7	17.7	16.6	17.3	17.4	17.9	16.8	18.7	16.1
Miscellaneous parameters size (kB)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Total transmitted parameters size (kB)	30.8	31.3	31.8	31.8	31.9	31.5	32.0	31.8	31.8	31.3

Size of transmitted parameters and size of source and target signals. The total size of the transmitted parameters is much smaller than the target signal size and together with the source signal size it is close to the size of the 64-kbps MP3 version of the signal.

Table 6: PEAQ test scores for 32-kbps MP3 enhancement.

	Rock	Symphony	Electronic	Jazz	Flute	Violin	Bass	Piano	Harp	Drums
32 kbps MP3	−3.5	−3.6	−3.4	−3.6	−3.6	−3.6	−3.4	−2.3	−2.4	−3.4
64 kbps MP3	−1.3	−1.8	−1.2	−1.5	−2.1	−1.5	−1.9	−1.2	−1.8	−1.6
Enhanced	−1.2	−1.6	−1.4	−1.7	−2.0	−1.6	−2.0	−1.4	−1.8	−1.4

ODG scores for the 32-kbps MP3 enhancement scenario. The range is from −4 (“very annoying”) to 0 (“imperceptible”). The reference is always the uncompressed file.

Table 7: PEAQ test scores for 64-kbps MP3 enhancement.

	Rock	Symphony	Electronic	Jazz	Flute	Violin	Bass	Piano	Harp	Drums
64 kbps MP3	−1.3	−1.8	−1.2	−1.5	−2.1	−1.5	−1.9	−1.2	−1.8	−1.6
96 kbps MP3	−0.2	−0.2	−0.2	−0.2	−0.5	−0.3	−0.5	−0.5	−0.6	−0.2
Enhanced	−0.4	−0.3	−0.4	−0.4	−0.8	−0.5	−0.7	−0.6	−0.5	−0.4

ODG scores for the 64-kbps MP3 enhancement scenario. The range is from −4 (“very annoying”) to 0 (“imperceptible”). The reference is always the uncompressed file.

MP3 source and it is similar to the audio quality of the 64-kbps MP3 scheme. Similarly, for the case of 64-kbps MP3 enhancement the enhanced file’s quality is much better than that of the 64-kbps MP3 signal and it is comparable to that of the 96-kbps MP3 signal.

The PEAQ test has not been thoroughly validated at low bit rates (e.g., 32 kbps) and in order to complement the PEAQ results, we also conducted a subjective listening test based on the ITU-R BS.1116 recommendation [38] using 10 listeners. We simulated the scenario of 32-kbps MP3 enhancement only since the relative audio impairment in this case is audible enough to nonexpert listeners. In addition, our preliminary tests showed that without the use of expert listeners, it is quite difficult to distinguish among low bit rate audio samples and especially grade them in a consistent scale. Thus we decided to use a simplified version of this test as in [39]. The listener is presented with the reference (uncompressed) signal and with two processed signals A and B. One of these two signals is processed by our algorithm while the other one by a benchmark codec. The benchmark codec produces either a 32-kbps MP3 file or a 64-kbps MP3 file (using the LAME encoder). Thus for each of the 10 music cases the listener is presented with two pairs of files and in each pair the benchmark codec signal is the 64-kbps or 32-kbps MP3 file (the two pairs are in random order). The listener is asked to select the file, A or B, which sounds closer to the reference. The listeners can also answer that the two files sound the same if they cannot detect a difference between them. Each subject listened through headphones and could repeat each individual sequence as desired. They could also switch in the middle of one sequence to the other, instantly picking up the other sequence at the same point in the playback. A few training samples were also provided to familiarize the listeners with the coding artifacts.

The results of this test are shown in Table 8. The enhanced file is of higher audio quality than the 32-kbps MP3 source file since, in direct comparison, all listeners preferred the enhanced file. The results also suggest that our algorithm produces similar quality audio to the 64-kbps MP3 scheme since there was no significant preference between the enhanced signal and the 64-kbps MP3 signal. Note, however, that in a scenario related to enhancing a file remotely (e.g., over the internet as described in Section 1), the required amount of information to create a signal with quality similar to the 64-kbps MP3 scheme is around 30 kB. On the other hand, as seen in Table 5, the amount of information required to transmit the whole 64-kbps MP3 file is 52 kB. This means that in such cases our algorithm would be more efficient than sending the 64-kbps file instead.

Table 8: Subjective listening test scores.

	Preference ratio of enhanced file over 64-kbps file	
		95% confidence interval
Rock	0.40	± 0.27
Electronic	0.50	± 0.25
Symphony	0.47	± 0.26
Jazz	0.46	± 0.27
Flute	0.53	± 0.25
Violin	0.44	± 0.23
Bass	0.46	± 0.27
Piano	0.47	± 0.24
Harp	0.54	± 0.27
Drums	0.56	± 0.23

Preference ratios (normalized to a maximum value of 1) of the enhanced file over the 64-kbps MP3 file along with the 95% confidence intervals, for the case of 32-kbps MP3 enhancement. The enhanced file is always preferred over the 32 kbps MP3 file for all signals.

7. CONCLUSIONS

We have presented a new method on quality enhancement of compressed audio based on statistical features conversion. A basic challenge of this scheme was to enhance an audio piece while producing small overhead size. As the music examples showed, the algorithm presented has similar performance to the MP3 scheme under comparison. Moreover, in a scenario in which the enhanced file is created by requesting the transmitted parameters remotely, it is clear that the amount of information required for enhancement is smaller compared to that of the MP3 scheme. Unlike the enhancement approach of MPEG-4, our algorithm does not require an embedded codec and thus can enhance signals created by any compression scheme.

Future work could further optimize the algorithm performance by including psychoacoustic information. A promising alternative is to apply features extraction in the frequency domain or a perceptual domain in a way similar to MFCC [25] and perceptual LPC (PLP) [40] schemes. Our method can also be modified to work as in the SBR philosophy by applying statistical conversion between the low- and high-frequency subbands of a signal in order to resynthesize the high subbands only. A direct comparison with SBR codecs such as MP3 Pro would then be feasible. In addition, the statistical framework of our algorithm can provide us with the option of enhancing a signal without any prior information of the uncompressed signal. Preliminary results indicate cases in which, instead of the specific target uncompressed signal, we can use a predefined set of similar signals to derive the conversion parameters. This unique feature of the algorithm will be demonstrated in future publications as the synthesis problem.

ACKNOWLEDGMENTS

This work was supported in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center under Cooperative Agreement no. EEC-9529152. Statements and opinions expressed do not necessarily reflect the position or policy of the National Science Foundation and no official endorsement should be inferred. The research of the second author was cofunded by the Greek General Secretariat for Research and Technology (GSRT) and the European Social Fund, Program ΕΠΑΝ Code 05NON-EU-1, and by the EU-funded program “ASPIRE”, a Marie Curie Transfer of Knowledge (TOK) grant.

REFERENCES

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

[2] A. Mouchtaris, S. S. Narayanan, and C. Kyriakakis, “Multiresolution spectral conversion for multichannel audio resynthesis,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME ’02)*, vol. 2, pp. 273–276, Lausanne, Switzerland, August 2002.

[3] D. Cantzos and C. Kyriakakis, “Quality enhancement of low bit rate MPEG1-layer 3 audio based on audio resynthesis,” in *Proceedings of the 119th Audio Engineering Society Convention (AES ’05)*, New York, NY, USA, October 2005, preprint 6569.

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

[5] R. Geiger, T. Sporer, and J. Koller, “Audio coding based on integer transforms,” in *Proceedings of the 111th Audio Engineering Society Convention (AES ’01)*, New York, NY, USA, September 2001, preprint 5471.

[6] *Information technology—Coding of Audiovisual Objects, Part 3: Audio*, ISO/IEC 14496-3, 1999.

[7] R. Yu, C. C. Ko, S. Rahardja, and X. Lin, “Bit-plane Golomb coding for sources with Laplacian distributions,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’03)*, vol. 4, pp. 277–280, Hong Kong, April 2003.

[8] K. Brandenburg and B. Grill, “First ideas on scalable audio coding,” in *Proceedings of the 97th Audio Engineering Society Convention (AES ’94)*, San Francisco, Calif, USA, November 1994, preprint 3924.

[9] B. Grill and K. Brandenburg, “A two or three-stage bit rate scalable audio coding system,” in *Proceedings of the 99th Audio Engineering Society Convention (AES ’95)*, New York, NY, USA, October 1995, preprint 4132.

[10] B. Grill, “A bit rate scalable perceptual coder for MPEG-4 audio,” in *Proceedings of the 103rd Audio Engineering Society Convention (AES ’97)*, New York, NY, USA, September 1997, preprint 4620.

[11] M. Dietz, L. Liljeryd, K. Kjörling, and O. Kunz, “Spectral Band Replication, a novel approach in audio coding,” in *Proceedings of the 112th Audio Engineering Society Convention (AES ’02)*, Munich, Germany, May 2002.

[12] E. D. Scheirer, “Structured audio, Kolmogorov complexity, and generalized audio coding,” *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 8, pp. 914–931, 2001.

[13] Y. Stylianou, O. Cappé, and E. Moulines, “Continuous probabilistic transform for voice conversion,” *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, 1998.

[14] A. Kain and M. W. Macon, “Spectral voice conversion for text-to-speech synthesis,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’98)*, vol. 1, pp. 285–288, Seattle, Wash, USA, May 1998.

[15] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, “Voice conversion through vector quantization,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’88)*, vol. 1, pp. 655–658, New York, NY, USA, April 1988.

[16] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society. Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[17] D. Cantzos, A. Mouchtaris, and C. Kyriakakis, “Multichannel audio resynthesis based on a generalized Gaussian mixture model and cepstral smoothing,” in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (ASPAA ’05)*, pp. 215–218, New Paltz, NY, USA, October 2005.

[18] J. H. Miller and J. B. Thomas, “Detectors for discrete-time signals in non-Gaussian noise,” *IEEE Transactions on Information Theory*, vol. 18, no. 2, pp. 241–250, 1972.

[19] D. Sündermann, A. Bonafonte, H. Ney, and H. Höge, “A study on residual prediction techniques for voice conversion,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’05)*, vol. 1, pp. 13–16, Philadelphia, Pa, USA, March 2005.

[20] M. Goodwin, “Residual modeling in music analysis-synthesis,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’96)*, vol. 2, pp. 1005–1008, Atlanta, Ga, USA, May 1996.

[21] A. Kain and M. W. Macon, “Design and evaluation of a voice conversion algorithm based on spectral envelope mapping

and residual prediction," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '01)*, vol. 2, pp. 813–816, Salt Lake, Utah, USA, May 2001.

[22] D. Cantzos, A. Mouchtaris, and C. Kyriakakis, "Enhanced multichannel audio resynthesis through residual processing and features alignment," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '07)*, pp. 1267–1270, Beijing, China, July 2007.

[23] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, Boston, Mass, USA, 1990.

[24] G. Strang, *Introduction to Linear Algebra*, Wellesley-Cambridge Press, Wellesley, Mass, USA, 2nd edition, 1998.

[25] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

[26] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, NY, USA, 1981.

[27] F. Mueller, "Distribution shape of two-dimensional DCT coefficients of natural images," *Electronics Letters*, vol. 29, no. 22, pp. 1935–1936, 1993.

[28] B. Gillett and S. King, "Transforming voice quality," in *Proceedings of the 8th European Conference on Speech Communication and Technology (EuroSpeech '03)*, pp. 1713–1716, Geneva, Switzerland, September 2003.

[29] "Sound quality assessment material recordings for subjective tests," Technical Review 3253-E, European Broadcasting Union (EBU), Geneva, Switzerland, April 1988.

[30] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, Mass, USA, 1996.

[31] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure & Applied Mathematics*, vol. 41, no. 7, pp. 909–996, 1998.

[32] D. Sinha and A. H. Tewfik, "Low bit rate transparent audio compression using adapted wavelets," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3463–3479, 1993.

[33] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[34] H. Akaike, "A new look at the statistical model selection," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[35] S. L. Sclove, "Application of the conditional population-mixture model to image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 4, pp. 428–433, 1983.

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

[37] W. C. Treurniet and G. A. Soulodre, "Evaluation of the ITU-R objective audio quality measurement method," *Journal of the Audio Engineering Society*, vol. 48, no. 3, pp. 164–173, 2000.

[38] ITU-R Recommendation BS.1116, "Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems," International Telecommunications Union, Geneva, Switzerland, 1997.

[39] R. Yu and C. C. Ko, "A warped linear-prediction-based subband audio coding algorithm," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 1, pp. 1–8, 2002.

[40] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.