High-Fidelity Multichannel Audio Coding With Karhunen-Loève Transform

Dai Yang, *Member, IEEE*, Hongmei Ai, *Member, IEEE*, Chris Kyriakakis, *Member, IEEE*, and C.-C. Jay Kuo, *Fellow, IEEE*

***Abstract*—A new quality-scalable high-fidelity multichannel audio compression algorithm based on MPEG-2 Advanced Audio Coding (AAC) is presented in this research. The Karhunen-Loève Transform (KLT) is applied to multichannel audio signals in the pre-processing stage to remove inter-channel redundancy. Then, signals in de-correlated channels are compressed by a modified AAC main profile encoder. Finally, a channel transmission control mechanism is used to re-organize the bitstream so that the multichannel audio bitstream has a quality scalable property when it is transmitted over a heterogeneous network. Experimental results show that, compared with AAC, the proposed algorithm achieves a better performance while maintaining a similar computational complexity at the regular bit rate of 64 kbit/sec/ch. When the bitstream is transmitted to narrow-band end users at a lower bit rate, packets of some channels can be dropped, and slightly degraded yet full-channel audio can still be reconstructed in a reasonable fashion without any additional computational cost.**

***Index Terms*—Advanced audio coding (AAC), Karhunen-Loève transform (KLT), MPEG, multichannel audio, quality scalable audio.**

I. Introduction

EVER since the beginning of the twentieth century, the art of sound coding, transmission, recording, mixing, and reproduction has been constantly evolving. Starting from the monophonic technology, technologies on multichannel audio have been gradually extended to include stereophonic, quadraphonic, 5.1 channels, and more. Compared with traditional mono or stereo audio, multichannel audio provides end users with a more compelling experience and becomes more and more appealing to music producers. As a result, an efficient coding scheme is needed for multichannel audio's storage and transmission, and this subject has attracted a lot of attention recently.

Manuscript received February 16, 2001; revised February 28, 2003. This work was supported by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, under Cooperative Agreement EEC-9529152. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Bryan George.

D. Yang is with the NTT Cyber Space Laboratory, Tokyo, Japan (e-mail: daiyang@alumni.usc.edu).

H. Ai is with Pharos Science & Applications, Inc., Torrance, CA 90501 USA (e-mail: hongmeiai@hotmail.com).

C. Kyriakakis and C.-C. J. Kuo are with the Integrated Media Systems Center and Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: ckyriak@imsc.usc.edu; cckuo@sipi.usc.edu).

Digital Object Identifier 10.1109/TSA.2003.814375

Among several existing multichannel audio compression algorithms, Dolby AC-3 and MPEG Advanced Audio Coding (AAC) are the two most prevalent perceptual digital audio coding systems. Dolby AC-3 is the third generation of digital audio compression systems from Dolby Laboratories, and has been adopted as the audio standard for High Definition Television (HDTV) systems. It is capable of providing transparent audio quality at 384 kbit/sec for 5.1 channels [1]. AAC is currently the most powerful multichannel audio coding algorithm in the MPEG family. It can support up to 48 audio channels and provide perceptually lossless audio at 320 kbit/sec for 5.1 channels [2]. In general, these low bit rate multichannel audio compression algorithms not only utilize transform coding to remove statistical redundancy within each channel, but also take advantage of the human auditory system to hide lossy coding distortions.

Despite the success of AC-3 and AAC, not much effort has been made in reducing inter-channel redundancy inherent in multichannel audio. The only technique used in AC-3 and AAC to eliminate redundancy across channels is called "Joint Coding", which consists of Intensity/Coupling and Mid/Side(M/S) stereo coding. Coupling is adopted based on the psychoacoustic evidence that, at high frequencies (above approximately 2 kHz), the human auditory system localizes sound primarily based on envelopes of critical-band-filtered signals that reach human ears, rather than signals themselves [3], [4]. M/S stereo coding is only applied to lower frequency coefficients of Channel-Pair-Elements (CPEs). Instead of direct coding of original signals in the left and right channels, it encodes the sum and the difference of signals in two symmetric channels [5], [6].

Our experimental results show that high correlation is very likely to be present between every pair of channels besides CPE in all frequency regions, especially for those multichannel audio signals that are captured and recorded in a real space [7]. Since neither AAC nor AC-3 exploits this property to reduce redundancy, none of them can efficiently compress this kind of multichannel audio content. On the other hand, if the input multichannel audio signals presented to the encoder module have little correlation between channels, the same bit rate encoding would result in higher reconstructed audio quality. Therefore, a better compression performance can be achieved if inter-channel redundancy can be effectively removed via a certain kind of transform together with redundancy removal techniques available in the existing multichannel audio coding algorithms. One possibility to reduce

the cross-channel redundancy is to use inter-channel prediction [8] to improve the coding performance. However, a recent study [9] argues that this kind of technique is not applicable to perceptual audio coding. In this paper, we present a new algorithm called MAACKLT, which stands for Modified AAC with Karhunen-Loève Transform (KLT). In MAACKLT, a temporal-adaptive KLT is applied in the pre-processing stage to remove inter-channel redundancy. Then, de-correlated signals in the KL transformed channels, called eigen-channels, are compressed by a modified AAC main profile encoder module. Finally, a prioritized eigen-channel transmission policy is enforced to achieve quality scalability.

As the world is evolving into the information era, media compression for a pure storage purpose is far less than enough. The design of a multichannel audio codec which takes the network transmission condition into account is also important. When a multichannel audio bitstream is transmitted through a heterogeneous network to multiple end users, a quality-scalable bitstream would be much more desirable than the nonscalable one. The quality scalability of a multichannel audio bitstream makes it possible that the entire multichannel sound can be played at various degrees of quality for end users with different receiving bandwidths. To be more precise, when a single quality-scalable bitstream is streamed to multiple users over the Internet via multicast, some lower priority packets can be dropped, and a certain portion of the bitstream can be transmitted successfully to reconstruct different quality multichannel sound according to different users' requirement or their available bandwidth. This is called the multicast streaming [10]. With nonscalable bitstreams, the server has to send different users different unicast bitstreams. This is certainly a waste of resources. Not being considered for audio delivery over heterogenous networks, the bitstream generated by most existing multichannel audio compression algorithms, such as AC-3 or AAC, is not scalable by nature [11]. In this work, we show that the proposed MAACKLT algorithm provides a coarse-grain scalable audio solution. That is, even if packets of some eigen-channels are dropped completely, a slightly degraded yet full-channel audio can still be reconstructed in a reasonable fashion without any additional computational cost.

To summarize, we focus on two issues in this research. First, the proposed MAACKLT algorithm exploits inter-channel correlation existing in audio data to achieve a better coding gain. Second, it provides a quality-scalable multichannel audio bitstream which can be adaptive to networks of time-varying bandwidth. The rest of this paper is organized as follows. Section II summarizes the inter-channel de-correlation scheme and its efficiency. Section III discusses the temporal adaptive approach. Section IV describes the eigen-channel coding method and its selective transmission policy. Section V demonstrates the audio concealment strategy at the decoder end when the bitstream is partially received. The system overview of the complete MAACKLT compression algorithm is provided in Section VI. The computational complexity of MAACKLT is compared with that of MPEG AAC in Section VII. Experimental results are shown in Section VIII. Finally, concluding remarks are given in Section IX.

II. Inter-Channel Redundancy Removal

A. Karhunen-Loève Transform

For a given time instance, removing inter-channel redundancy would result in a significant bandwidth reduction. This can be done via an orthogonal transform $MV = U$, where V and U denote the vector whose n elements are samples in original and transformed channels, respectively. Among several commonly used transforms, including the Discrete Cosine Transform (DCT), the Fourier Transform (FT), and the Karhunen-Loève Transform (KLT), the signal-dependent KLT is adopted in the pre-processing stage because it is theoretically optimal in de-correlating signals across channels. If M is the KLT matrix, the transformed channels are called the eigen-channels. Fig. 1 illustrates how KLT is performed on multichannel audio signals, where the columns of the KL transform matrix is composed by eigenvectors calculated from the covariance matrix C_V associated with original multichannel audio signals V.

Suppose that an input audio signal has n channels, then the covariance of KL transformed signals is

$$\begin{aligned} E[\bar{U}\bar{U}^T] &= E[(M\bar{V})(M\bar{V})^T] = ME[\bar{V}\bar{V}^T]M^T \\ &= MC_VM^T = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \end{aligned} \tag{1}$$

where $\bar{X}$ $(X = U, V)$ represents mean removed signal of X and $\lambda_1, \lambda_2, \ldots, \lambda_n$ are eigenvalues of C_V. Thus, the transform produces statistically de-correlated channels in the sense of having a diagonal covariance matrix for transformed signals. Another property of KLT, which can be used in the reconstruction of audio of original channels, is that the inverse transform matrix of M is equal to its transpose. Since C_V is real and symmetric, the matrix formed by normalized eigenvectors is orthonormal. Therefore, we have $V = M^TU$ in reconstruction. From KL expansion theory [12], we know that selecting eigenvectors associated with the largest eigenvalues can minimize the error between original and reconstructed channels. This error will go to zero if all eigenvectors are used. KLT is thus optimum in the least-square-error sense.

B. Evidence for Inter-Channel De-Correlation

Multichannel audio sources can be roughly classified into three categories. Those belonging to class I are mostly used in broadcasting, where signals in one channel may be completely different from the other. Either broadcasting programs are different from channel to channel, or the same program is broadcast but in different languages. Samples of audio sources in class I normally contain relatively independent signals in each channel and present little correlation among channels. Therefore, this type of audio sources will not fall into the scope of high-quality multichannel audio compression discussed here.

The second class of multichannel audio sources can be found in most film soundtracks, which are typically in the format of 5.1 channels. Most of this kind of program material has a symmetry property among CPEs and presents high correlation in CPEs, but

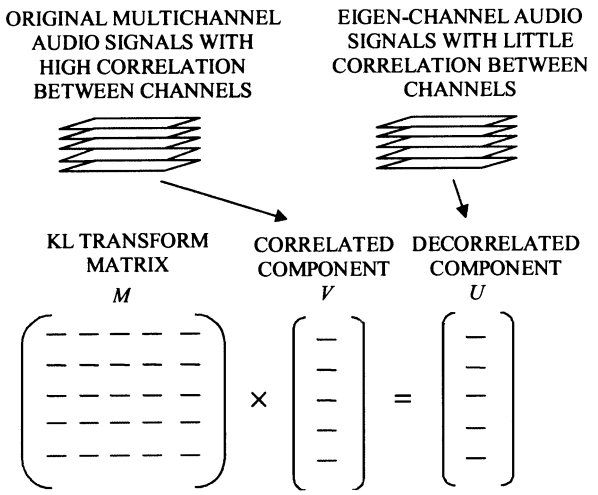

Fig. 1. Inter-channel de-correlation via KLT.

little correlation across CPEs and SCEs (Single Channel Elements). Almost all existing multichannel audio compression algorithms such as AAC and Dolby AC-3 are mainly designed to encode audio material that belongs to this category. Fig. 2 shows the normalized covariance matrix generated from one sample audio of class II, where the normalized covariance matrix is derived from the cross-covariance matrix by multiplying each coefficient with the reciprocal of the square root of the product of their individual variance. Since the magnitude of nondiagonal elements in a normalized covariance matrix provides a convenient and useful measure for the degree of inter-channel redundancy, it is used as a correlation metric throughout the paper.

A third emerging class of multichannel audio sources consists of material recorded in a real space with multiple microphones that capture acoustical characteristics of that space. Audio of class III is becoming more prevalent with the introduction of consumer media such as DVD-Audio. This type of audio signals has considerably larger redundancy inherent among channels especially adjacent channels as graphically shown in Fig. 3, which corresponds to the normalized covariance matrix derived from a test sequence named "Messiah." As shown in the figure, a large degree of correlation is present between not only CPEs (e.g., left/right channel pair and left-surround/right-surround channel pair) but also SCE (e.g., the center channel) and any other channels.

The work presented in this research will focus on improving the compression performance for multichannel audio sources that belong to classes II and III. It will be demonstrated that the proposed MAACKLT algorithm not only achieves good results for class III audio sources, but also improves the coding performance to a certain extent for class II audio sources compared with original AAC.

Two test data sets are used to illustrate the de-correlation effect of KLT. One is a class III 10-channel audio piece called "Messiah"[1] . It is a piece of classical music recorded live in a concert hall. Another one is a class II 5-channel audio piece called "Herre,"[2] which is a piece of pop music and was used in MPEG-2 AAC standard (ISO/IEC 13 818-7) conformance work. These test sequences are chosen because they contain a diverse range of frequency components played by several different instruments so that they are very challenging for inter-channel de-correlation and subsequent coding experiments. In addition, they provide good samples for result comparison between original AAC and the proposed MAACKLT algorithm.

Figs. 4 and 5 show absolute values of elements in the lower triangular part of the normalized cross-covariance matrix after KLT for 5-channel set "Herre" and 10-channel set "Messiah." These figures clearly indicate that KLT method achieves a high degree of de-correlation. Note that the nondiagonal elements are not exactly zeros because we are dealing with an approximation of KLT during calculation. We predict that by removing redundancy in the input audio with KLT, a much better coding performance can be achieved when encoding each channel independently, which will be verified in later sections.

C. Energy Compaction Effect

The KLT pre-processing approach not only significantly de-correlates the input multichannel audio signals but also considerably compacts the signal energy into the first several eigen-channels. Fig. 6(a) and (b) show how energy is accumulated with an increased number of channels for original audio channels and de-correlated eigen-channels. As clearly shown in these two figures, energy accumulates much faster in the case of eigen-channels than original channels, which provides a strong evidence of data compaction of KLT. It implies that, when transmitting data of a fixed number of channels with the proposed MAACKLT algorithm, more information content will be received at the decoder side, and better quality of reconstructed multichannel audio can be achieved.

Another convenient way to measure the amount of data compaction can be obtained via eigenvalues of the cross-covariance matrix associated with the KL transformed data. In fact, these eigenvalues are nothing else but variances of eigen-channels, and the variance of a set of signals reflects its degree of jitter, or the information content. Fig. 7(a) and (b) are plots of variances of eigen-channels associated with the "Messiah" test set consisting of 10 and 5 channels, respectively. As shown in figures, the variance drops dramatically with the order of eigen-channels. The steeper the variance drop is, the more efficient the energy compaction is achieved. These experimental results also show that the energy compaction efficiency increases with the number of input channels. The area under the variance curve reflects the amount of information to be encoded. As illustrated from these two figures, this particular area is substantially much smaller for the 10-channel set than that of the 5-channel set. As the number of input channels decreases, the final compression performance of MAACKLT tends to be more influenced by the coding power of the AAC main profile encoder.

[1]The 10 channels include Center (C), Left (L), Right (R), Left Wide (Lw), Right Wide (Rw), Left High (Lh), Right High (Rh), Left Surround (Ls), Right Surround (Rs) and Back Surround (Bs). They were obtained by mixing signals from 16 microphones placed in various locations in a concert hall.

[2]The 5 channels include C, L, R, Ls, and Rs.

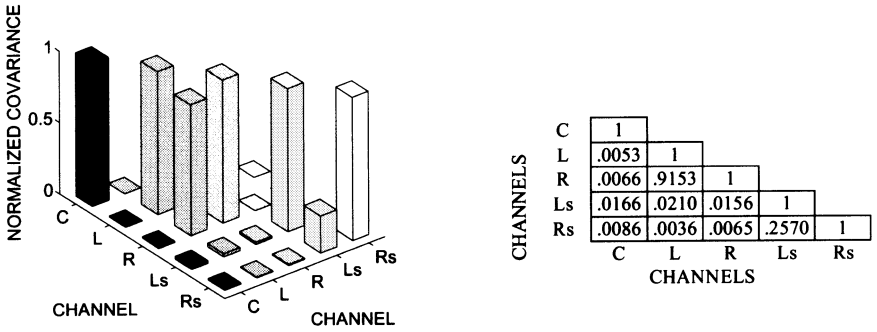

Fig. 2. Absolute values of elements in the lower triangular normalized covariance matrix for 5-channel "Herre."

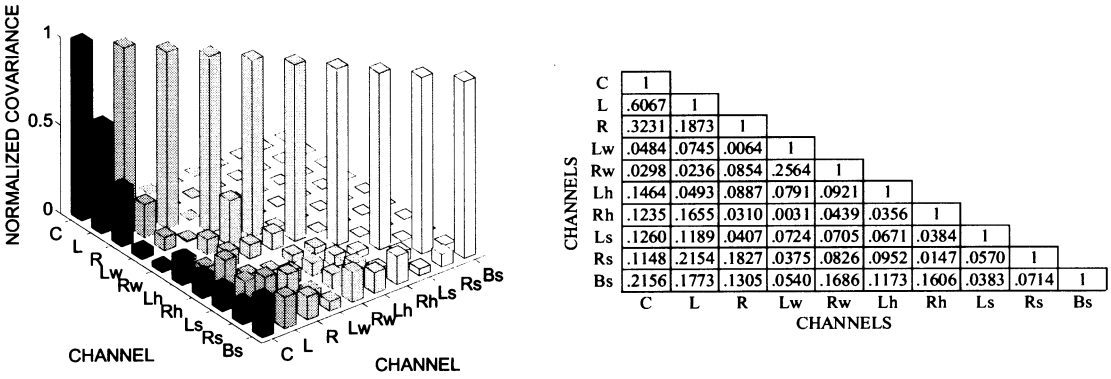

Fig. 3. Absolute values of elements in the lower triangular normalized covariance matrix for 10-channel "Messiah."

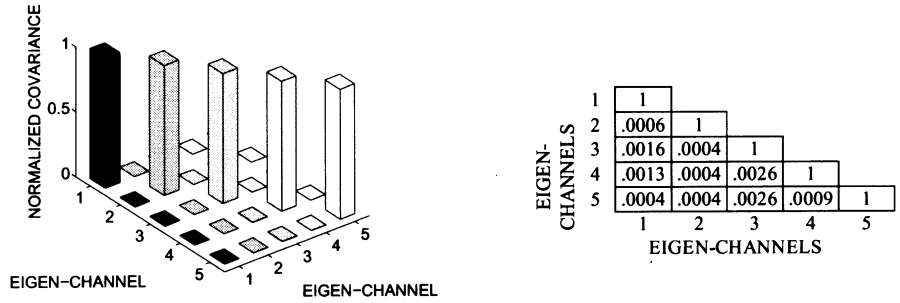

Fig. 4. Absolute values of elements in the lower triangular normalized covariance matrix after KLT for 5-channel "Herre."

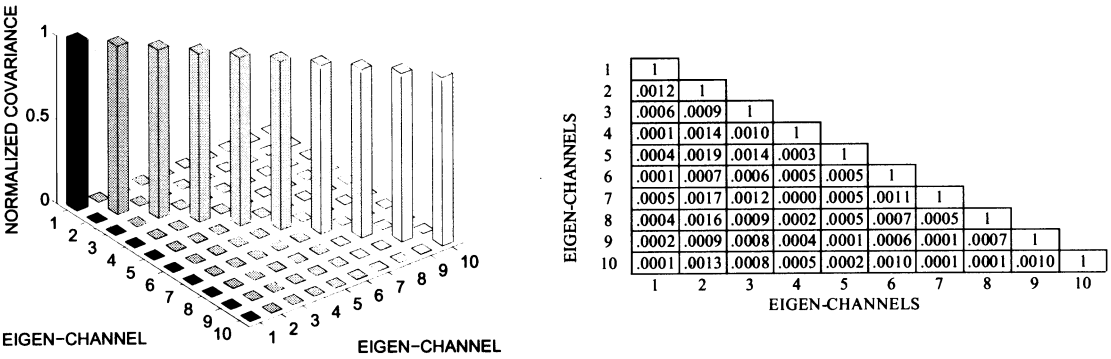

Fig. 5. Absolute values of elements in the lower triangular normalized covariance matrix after KLT for 10-channel "Messiah."

D. Frequency-Domain Versus Time-Domain KLT

In all previous discussion, we considered only the case of applying KLT to time-domain signals across channels. However, it is also possible to apply the inter-channel de-correlation procedure after time-domain signals are transformed into the frequency-domain via MDCT (Modified Discrete Cosine Transform) in the AAC encoder.

One frame of the audio signal from the center channel of "Herre" in the frequency-domain and in the time-domain are

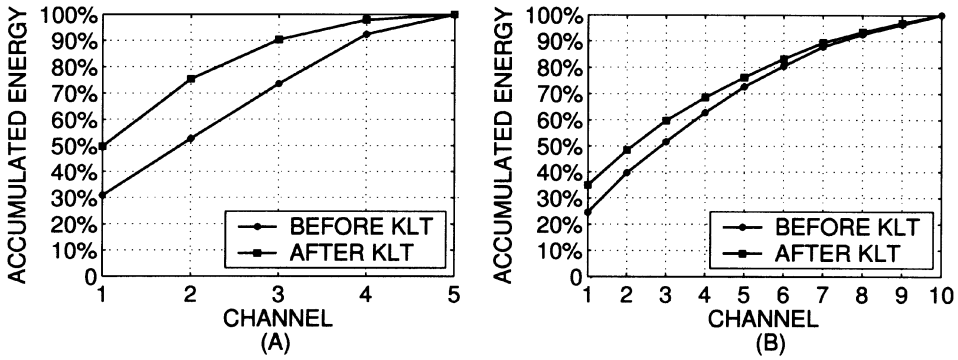

Fig. 6. Comparison of accumulated energy distribution for (a) 5-channel "Herre" and (b) 10-channel "Messiah."

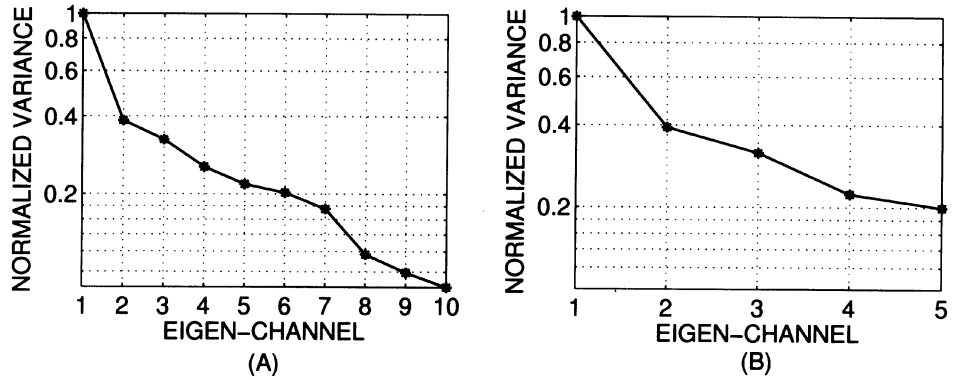

Fig. 7. Normalized variances for (a) 10-channel "Messiah" and (b) 5-channel "Messiah," where the vertical axis is plotted in the log scale.

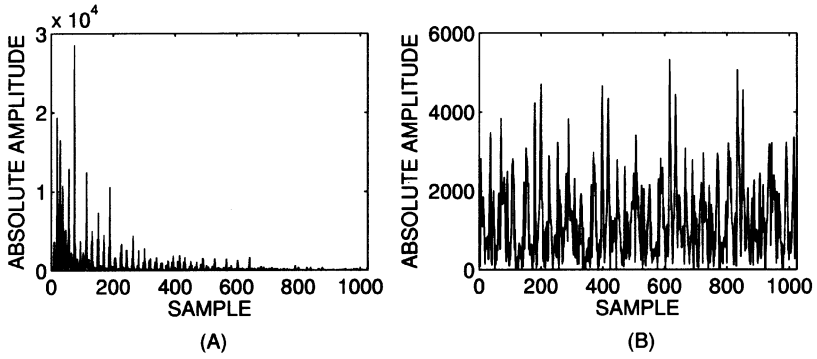

Fig. 8. (a) Frequency-domain and (b) time-domain representations of the center channel from "Herre."

shown in Fig. 8(a) and (b), respectively. The energy compaction property can be clearly seen from the simple comparison between the time-domain and the frequency-domain plots. Generally speaking, applying KLT to frequency-domain signals achieve a better performance than directly applying KLT to time-domain signals. In addition, a certain degree of delay and reverberant sound copies may exist in time-domain signals among different channels, which is especially true for class III multichannel audio sources. The delay and reverberation effects affect the time-domain KLTs de-correlation capability, however, they may not have that much impact on frequency-domain signals. Figs. 9 and 10 show absolute values of off-diagonal nonredundant elements for normalized covariance matrices generated from frequency- and time-domain KL transforms with test audio "Herre" and "Messiah," respectively. Clearly, the frequency-domain KLT has a much better inter-channel de-correlation capability than that of the time-domain KLT. This implies that applying KLT to frequency-domain signals should lead to a better coding performance, which will be verified by experimental results shown in Section VIII. Any result discussed hereafter will focus on frequency-domain KLT method unless otherwise mentioned.

III. Temporal-Adaptive KLT

A multichannel audio program may comprise different periods, each of which has its unique spectral signature. For example, a piece of music may begin with a piano prelude followed by a chorus. In order to achieve the highest information compactness, the de-correlation transform matrix should be adaptive to the characteristics of different periods. In this section, we present a temporal-adaptive KLT approach, in which

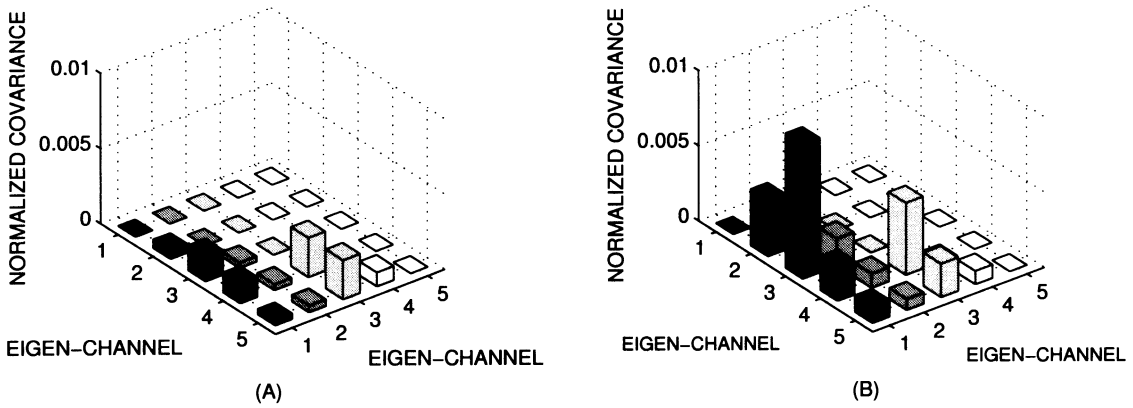

Fig. 9. Absolute values of off-diagonal elements for the normalized covariance matrix after (a) frequency-domain and (b) time-domain KL transforms with test audio "Herre."

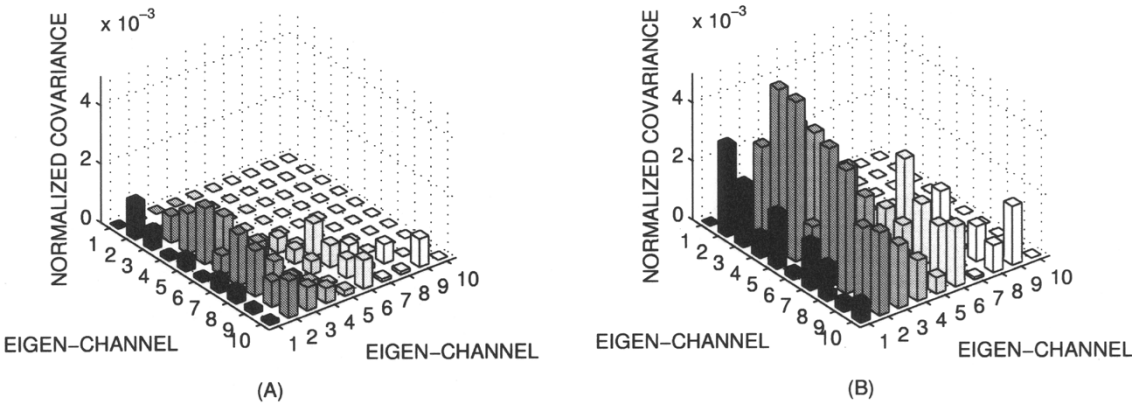

Fig. 10. Absolute values of off-diagonal elements for the normalized covariance matrix after (a) frequency-domain and (b) time-domain KL transforms with test audio "Messiah."

the covariance matrix (and, consequently, the corresponding KL transform matrix) is updated from time to time. Each adaptation period is called a block.

Fig. 11 shows the variance of each eigen-channel of one nonadaptive and two temporal-adaptive approaches for test set "Messiah." Compared with the nonadaptive method, the adaptive method achieves a smaller variance for each eigen-channel. Furthermore, the shorter the adaptation period, the higher inter-channel de-correlation is achieved. The only drawback of the temporal-adaptive approach over the nonadaptive approach goes to the overhead bits, which have to be transmitted to the decoder so that the multichannel audio can be reconstructed to its original physical channels. Due to the increase of the block number, the shorter the adaptation period is, the larger the overhead bit rate is. The trade-off between this block size and the overhead bit rate will be discussed below.

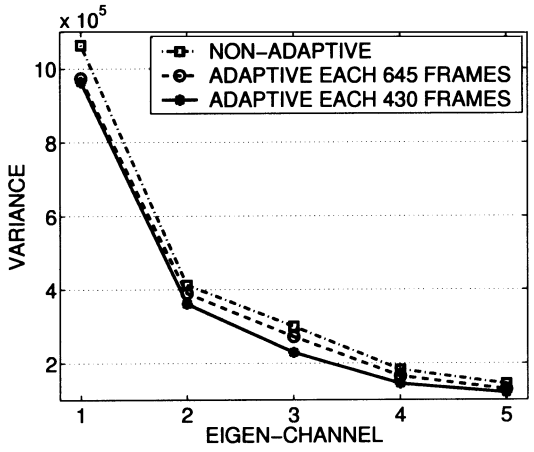

Fig. 11. De-correlation efficiency of temporal-adaptive KLT.

Since the inverse KLT has to be performed at the decoder side, the information of the transform matrix should be included in the coded bitstream. As mentioned before, the inverse KLT matrix is the transpose of the forward KLT matrix, which is composed by eigenvectors of the cross-covariance matrix. To reduce the overhead bit rate, elements of the covariance matrix are included in the bitstream instead of those of the KLT matrix since the covariance matrix is real and symmetric and we only have to send the lower (or higher) triangular part that contains nonredundant elements. As a result, the decoder also has to calculate eigenvectors of the covariance matrix before the inverse KLT can be performed.

Only one covariance matrix has to be coded for the nontemporal-adaptive approach. However, for the temporal-adaptive approach, every covariance matrix must be coded for each block. Assume that n channels are selected for simultaneous inter-channel de-correlation, and the adaptation period is K seconds, i.e., each block contains K seconds of audio. The size of the covariance matrix is $n \times n$, and the number of nonredun-

dant elements is $n \times (n+1)/2$. In order to reduce the overhead bit rate, the floating-point covariance matrix is quantized to 16 bits per element. Therefore, the total bit requirement for each covariance matrix is $8n \times (n+1)$ bits, and the overhead bit rate r_{overhead} is

$$r_{\text{overhead}} = \frac{8n \times (n+1)}{nK} = \frac{8(n+1)}{K} \tag{2}$$

in bit per second per channel (bit/sec/ch). The above equation suggests that the overhead bit rate increases approximately linearly with the number of channels. The overhead bit rate is, however, inversely proportional to the adaptation time (or the block size).

Fig. 12 illustrates the overhead bit rate for different channel numbers and block sizes. The optimal adaptation time is around 10 seconds, since shorter adaptation time dramatically increases the overhead bit rate. Extensive experimental results [13] suggest that, when shorter adaptation time is adopted, the improvement of de-correlation efficiency is not sufficient to compensate for coding performance degradation due to the excessive overhead bit rate.

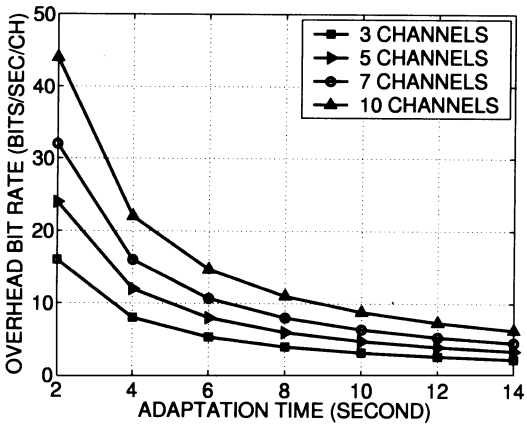

Fig. 12. Overhead bit rate versus the number of channels and the adaptation period.

IV. Eigen-Channel Coding and Transmission

A. Eigen-Channel Coding

The main profile of the AAC encoder is modified to compress audio signals in de-correlated eigen-channels. The detailed encoder block diagram is given in Fig. 13, where the shaded parts represent coder blocks that are different from the original AAC algorithm.

The major difference between Fig. 13 and the original AAC encoder block diagram is the KLT block added after the filter bank. When the original input signals are transformed into frequency domain, the cross-channel KLT are performed to generate the de-correlated eigen-channel signals. Masking thresholds are then calculated based on the KL transformed signals in the perceptual model. The KLT-related overhead information is sent into the bitstream afterwards.

The original AAC is typically used to compress class II audio sources. Its M/S stereo coding block is specifically used for symmetric CPEs. It encodes the mean and difference of CPEs instead of two independent SCEs, which reduces redundancy existing in symmetric channel pairs. In the proposed algorithm, since inter-channel de-correlation has been performed in an earlier stage and audio signals after KLT are from independent eigen-channels with little correlation between any channel pairs, the M/S coding block is no longer needed. Thus, the M/S coding block of the AAC main profile encoder is disabled.

The AAC encoder module originally assigns an equal amount of bits to each input channel. However, since signals into the iteration loops are no longer the original multichannel audio in the new system, the optimality of the same strategy has to be investigated. Experimental results indicate that the compression performance will be strongly influenced by the bit assignment scheme for de-correlated eigen-channels.

According to the bit allocation theory [14], the optimal bit assignment for identically distributed normalized random variables under the high rate approximations while without nonnegativity or integer constraints on the bit allocations is

$$b_i = \bar{b} + \frac{1}{2}\log_2 \frac{\sigma_i^2}{\rho^2} \tag{3}$$

where $\bar{b} = B/k$ is the average number of bits per parameter, k is the number of parameters, and $\rho^2 = \left(\prod_{i=1}^{k} \sigma_i^2\right)^{1/k}$ is the geometric mean of the variances of the random variables. A normalized random variable of X is

$$\bar{X} = \frac{X - E(X)}{std(X)} \tag{4}$$

where $E(X)$ and $std(X)$ represent the mean and the standard deviation of X, respectively. It is verified by experimental data that the normalized probability density functions of signals in eigen-channels are almost identical. They are given in Figs. 14 and 15. This optimal bit allocation method is adopted for rate/distortion control processing when encoding eigen-channel signals.

B. Eigen-Channel Transmission

Fig. 6(a) and (b) show that the signal energy accumulates faster in eigen-channel form than original multichannel form. This implies that, with a proper channel transmission and recovery strategy, transmitting the same number of eigen-channels and of original multichannels, the eigen-channel approach should result in a higher quality reconstructed audio since more energy is transmitted.

It is desirable to re-organize the bitstream so that bits of more important channels can be received at the decoder side first for audio decoding. This should result in the best audio quality given a fixed amount of received bits. When this re-organized audio bitstream is transmitted over a heterogeneous network, for those users with a limited bandwidth, the network can drop packets belonging to less important channels.

The first instinct about the metric of channel importance would be the energy of the audio signal in each channel. However, this metric does not work well in general. For example, for some multichannel audio sources, especially those

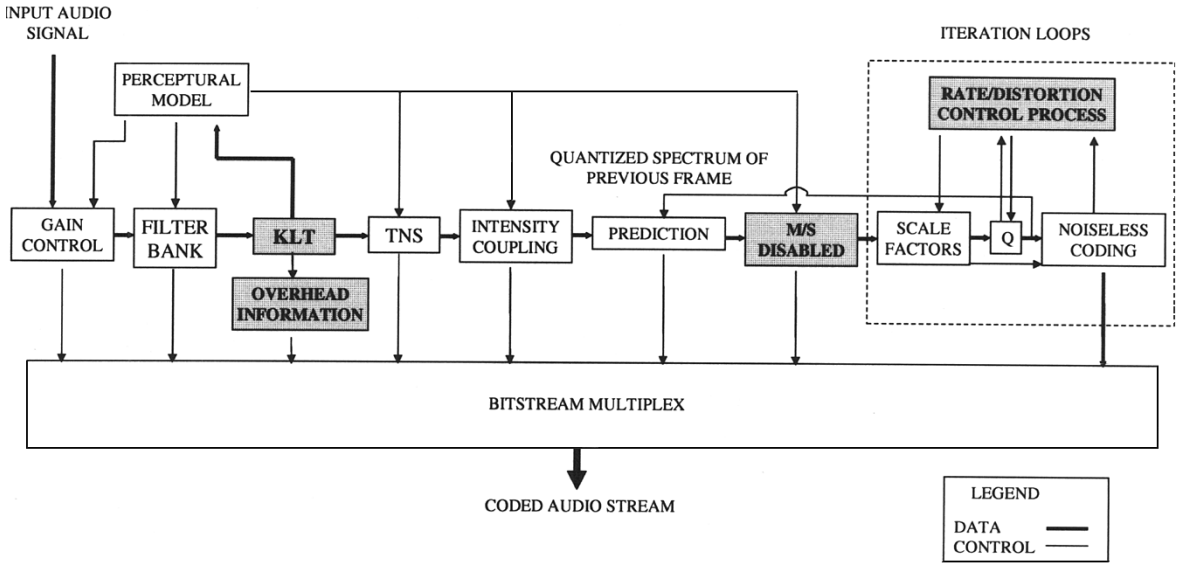

Fig. 13. Modified AAC encoder block diagram.

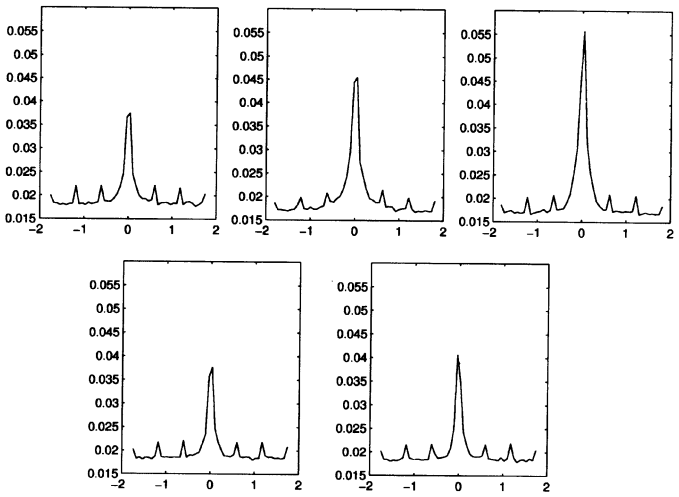

Fig. 14. Empirical probability density functions of normalized signals in 5 eigen-channels generated from test audio "Herre", where x axes represent the value of normalized random variable and y axes represent the corresponding probability.

belonging to class II, since they are re-produced in a music studio artificially, the side channel which normally does not contain the main melody may even have larger energy than the center channel. Based on our experience with multichannel audio, loss or significant distortion of the main melody in the center channel would be much more annoying than loss of melodies in side channels. In other words, the location of channels also plays an important role. Therefore, for a regular 5.1 channel configuration, the order of channel importance from the largest to the least should be

1) Center channel;
2) L/R channel pair;
3) Ls/Rs channel pair;
4) Low frequency channel.

Between channel pairs, their importance can be determined by their energy values. This rule is adopted in experiments below.

After KLT, eigen-channels are no longer the original physical channels, and sounds in different physical channels are mixed in every eigen-channel. Thus, spatial dependency of eigen-channels is less trivial. We observe from experiments that although it is true that one eigen-channel may contain sounds from more than one original physical channel, there still exists a close correspondence between eigen-channels and physical channels. To be more precise, audio of eigen-channel 1 would sound similarly to that of the center channel, audio of eigen-channels 2 and 3 would sound similarly to that of the L/R channel pair etc. Therefore, if eigen-channel 1 is lost in transmission, we would end up with a very distorted center channel. Moreover, it happens that, sometimes, eigen-channel 1 may not be the channel with a very large energy and could be easily discarded if the channel energy is adopted as the metric of channel importance. Thus, the channel importance of eigen-channels should be similar to that of physical channels. That is, eigen-channel 1 corresponding to the center channel, eigen-channel 2 and 3 corresponding to the L/R channel pair, eigen-channel 4 and 5 corresponding to the Ls/Rs channel pair. Within each channel pair, the importance is still determined by their energy values.

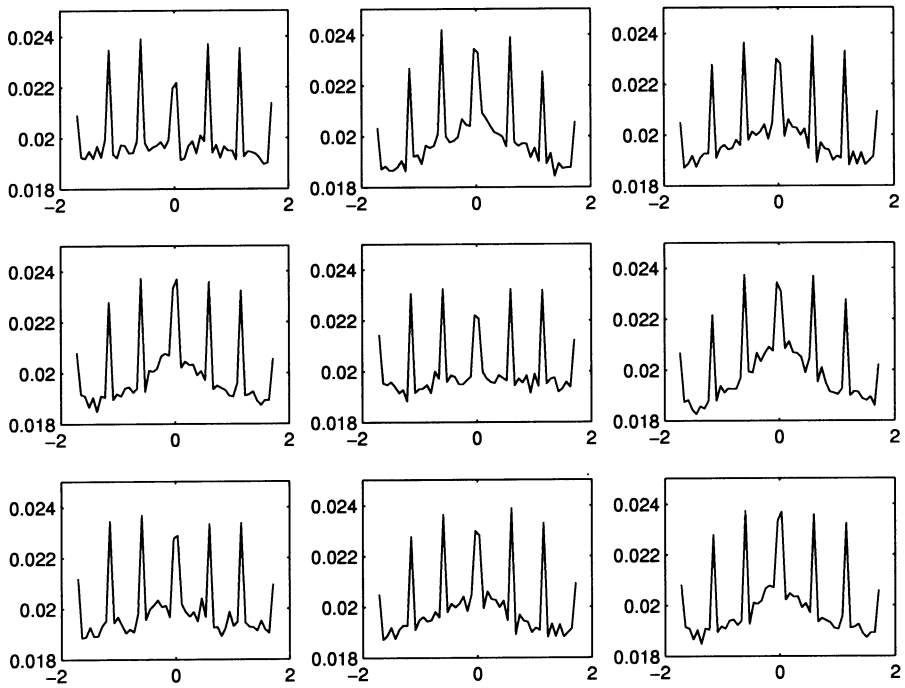

Fig. 15. Empirical probability density functions of normalized signals in the first 9 eigen-channels generated from test audio "Messiah", where x axes represent the value of normalized random variable and y axes represent the corresponding probability.

V. Audio Concealment for Channel-Scalable Decoding

Consider the scenario that an AAC-coded multichannel bitstream is transmitted in a heterogeneous network such as the Internet. For end-users who do not have enough bandwidth to receive full channel audio, some packets have to be dropped. In this section, we consider the bitstream of each channel as one minimum unit for audio reconstruction. When the bandwidth is not sufficient, we may drop bitstreams of a certain number of channels to reduce the bit rate. It is called channel-scalable decoding, which has an analogy in MPEG video coding, i.e., dropping B frames while keeping only I and P frames.

For an AAC channel pair, the M/S stereo coding block will replace low frequency coefficients in symmetric channels to be their sum and difference at the encoder, i.e.,

$$spec_l[i] \leftarrow \frac{(spec_l[i] + spec_r[i])}{2} \tag{5}$$

$$spec_r[i] \leftarrow \frac{(spec_l[i] - spec_r[i])}{2} \tag{6}$$

where $spec_l[i]$ and $spec_r[i]$ are the ith frequency-domain coefficient in the left and right channels of the channel pair, respectively.

The intensity coupling coding block will replace high frequency coefficients of the left channel with a value proportional to the envelope of the sound signal in the symmetric channel, and set the value of right channel high frequency coefficients to zero, i.e.,

$$spec_l[i] \leftarrow (spec_l[i] + spec_r[i]) \times \sqrt{\frac{E_l[SFB]}{E_s[SFB]}}, \tag{7}$$

$$spec_r[i] \leftarrow 0 \tag{8}$$

where $E_l[SFB]$, $E_r[SFB]$ and $E_s[SFB]$ are, respectively, energy values of the left channel, the right channel and the sum of left and right channels of the scale factor band that sample i belongs to. Values of $E_l[SFB]/E_r[SFB]$ are included in the coded bitstream as scaling factors.

At the decoder end, the low frequency coefficients of the left and right channel are reconstructed via

$$spec_l[i] \leftarrow spec_l[i] + spec_r[i] \tag{9}$$

$$spec_r[i] \leftarrow spec_l[i] - spec_r[i]. \tag{10}$$

For high frequency coefficients, audio signals in the left channel will remain the same as they are received from the bitstream, while those in the right channel will be reconstructed via

$$spec_r[i] = f(\text{scale}) \times spec_l[i] \tag{11}$$

where $f(\text{scale})$ is a function of the scaling factor.

When packets of one channel of a channel pair are dropped, we drop frequency coefficients of the right channel while keeping all other side information including scaling factors. Therefore, what we receive at the decoder side are just coefficients in the left channel. For low frequency coefficients, they correspond to the mean value of the original frequency coefficient in the left and right channels. For high frequency coefficients, they correspond to the energy envelope of the symmetric channel. That is, we have

$$spec_l[i] \rightarrow \frac{(spec_l[i] + spec_r[i])}{2} \tag{12}$$

$$spec_r[i] \rightarrow 0 \tag{13}$$

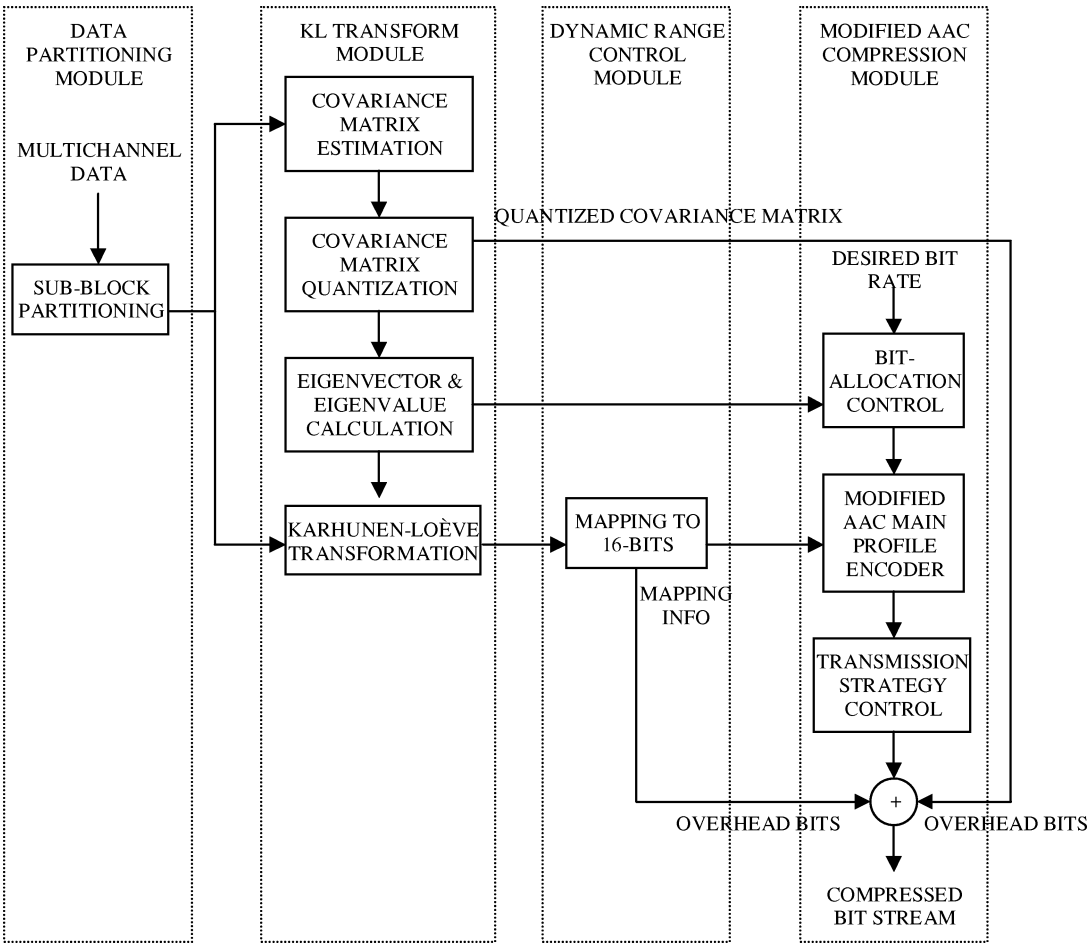

Fig. 16. Block diagram of the proposed MAACKLT encoder.

for the low frequency part and

$$spec_l[i] \rightarrow (spec_l[i] + spec_r[i]) \times \sqrt{\frac{E_l[SFB]}{E_s[SFB]}} \quad (14)$$

$$spec_r[i] \rightarrow 0 \quad (15)$$

for the high frequency part.

Note that since scaling factors are contained in the received bitstream, reconstruction of high frequency coefficients in the right channel will remain the same as the original AAC when data of all channels are received. Therefore, only low frequency coefficients in the right channel need to be recovered. The strategy used to reconstruct these coefficients is just to let values of right channel coefficients equal to values of received left channel coefficients. This is nothing else but the mean value of coefficients in the original channel pair, i.e.,

$$spec_r[i] = spec_l[i] \rightarrow \frac{(spec_l[i] + spec_r[i])}{2}. \quad (16)$$

Audio concealment for the proposed eigen-channel coding scheme is relatively simple. All coefficients in dropped channels will be set to 0, then a regular decoding process is performed to reconstruct full multichannel audio. For the situation where packets of two or more channels are dropped, the reconstructed dropped channel may have a much smaller energy than other channels after inverse KLT. In order to get better reconstructed audio quality, an energy boost up process can be enforced so that the signal in each channel will have a similar amount of energy.

To illustrate that the proposed algorithm MAACKLT has a better quality-degradation property than AAC (via a proper audio concealment process described in this section), we perform experiments with lossy channels where packets are dropped in a coded bitstream in Section VIII.

VI. Compression System Overview

The block diagram of the proposed compression system is illustrated in Fig. 16. It consists of four modules: (1) data partitioning, (2) Karhunen-Loève transform, (3) dynamic range control, and (4) the modified AAC main profile encoder. In the data partitioning module, audio signals in each channel are partitioned into sets of nonoverlapping intervals, i.e., blocks. Each block contains K frames, where K is a pre-defined value. Then, data in each block are sequentially fed into the KLT module to perform inter-channel de-correlation. In the KLT module, multichannel block data are de-correlated to produce a set of statistically independent eigen-channels. The KLT matrix consists of eigenvectors of the cross-covariance matrix associated with the multichannel block set. The covariance matrix is first estimated and then quantized into 16 bits per element. The quantized covariance coefficients will be sent to the bitstream as the overhead. Note that the KLT de-correlation module will add a certain amount of delay in the encoder

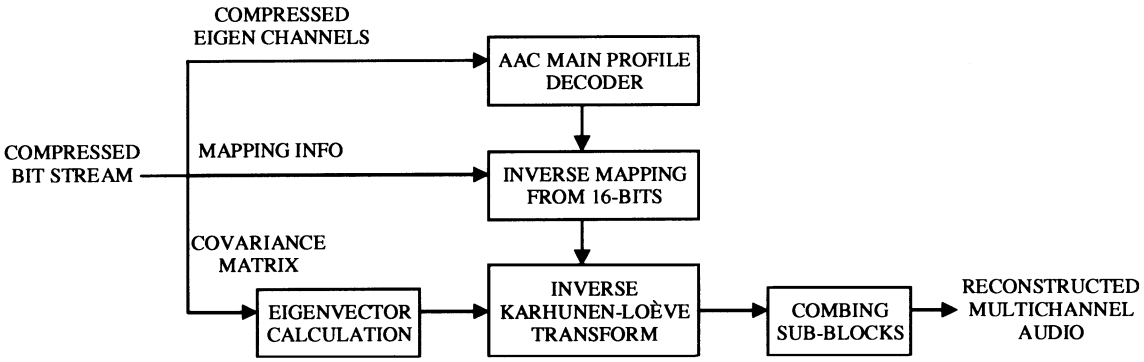

Fig. 17. Block diagram of the proposed MAACKLT decoder.

TABLE I
COMPARISON OF COMPUTATIONAL COMPLEXITY BETWEEN MAACKLT AND AAC

	Encoding					Decoding				
Time used (seconds)	MAACKLT		AAC	Extra		MAACKLT		AAC	Extra	
				Time	Percent				Time	Percent
Messiah	1-sec AP	344.28		26.15	8.2%	1-sec AP	16.92		4.62	37.6%
Messiah	5-sec AP	340.43		22.30	7.0%	5-sec AP	16.04		3.74	30.4%
Messiah	10-sec AP	339.44		21.31	6.7%	10-sec AP	15.60		3.30	26.8%
Messiah	NonA	337.62	318.13	19.49	6.1%	NonA	14.66	12.30	2.36	19.2%
Herre	NonA	112.15	101.23	10.92	10.8%	NonA	2.75	2.42	0.33	13.6%

and the decoder. For the nontemporal-adaptive method, the delay can be as long as the length of the input audio. For the temporal-adaptive method, this delay can be reduced to that of the block length.

As shown in Fig. 1, eigen-channels are generated by multiplication of the KLT matrix and the block data set. Therefore, after the transform, the sample value in eigen-channels may have a larger dynamic range than that of original channels. To avoid any possible data overflow in the later compression module, data in eigen-channels are rescaled in the dynamic range control module so that the sample value input to the modified AAC encoder module does not exceed the dynamic range of that in regular 16-bit PCM audio files. This rescaling information will also be sent to the bitstream as the overhead.

Signals in de-correlated eigen-channels are compressed in the next module by a modified AAC main profile encoder. The AAC main profile encoder is modified in our algorithm so that it is more suitable in compressing the audio signal in eigen-channels. To enable channel-scalability, a transmission strategy control block is adopted in this module right before the compressed bitstream is formed.

The block diagram of the decoder is shown in Fig. 17. The mapping information and the covariance matrix together with the coded information for eigen-channels are extracted from the received bitstream. If data of some eigen-channels are lost due to the network condition, the eigen-channel concealment block will be enabled. Then, signal values in eigen-channels will be reconstructed by the AAC main profile decoder. The mapping information is used to restore from a 16-bit dynamic range of the decoded eigen-channel back to its original range. The inverse KLT matrix can be calculated from the extracted covariance matrix via transposing its eigenvectors. Then, inverse KLT is performed to generate the reconstructed multichannel block set. These block sets are finally combined together to produce the reconstructed multichannel audio signals.

VII. COMPLEXITY ANALYSIS

Compared with the original AAC compression algorithm, the additional computational complexity required by the MAACKLT algorithm mainly comes from the KLT pre-processing module, which includes generation of the cross-covariance matrix, calculation of its eigenvalues and eigenvectors, and matrix multiplication required by KLT.

Table I illustrates the running time of MAACKLT and AAC for both the encoder and the decoder at a typical bit rate of 64 kbit/sec/ch, where "n-SEC AP" means the MAACKLT algorithm with a temporal-adaptation period of n seconds while "NONA" means a nonadaptive MAACKLT algorithm. The input test audio signals are 20-second 10-channel "Messiah" and 8-second 5-channel "Herre." The system used to generate the above result is a Pentium III 600 PC with 128M RAM.

These results indicate that the coding time for MAACKLT is still dominated by the AAC compression and de-compression part. When the optimal 10-second temporal-adaptation period is used for test audio "Messiah", the additional KLT computational time is less than 7% of the total encoding time at the encoder side while the MAACKLT algorithm only takes about 26.8% longer than that of the original AAC at the decoder side. The MAACKLT algorithm with a shorter adaptation period will take a little bit more time in encoding and decoding since more KL transform matrices are needed to be generated. Note also that we have not made any attempt to optimize our experimental

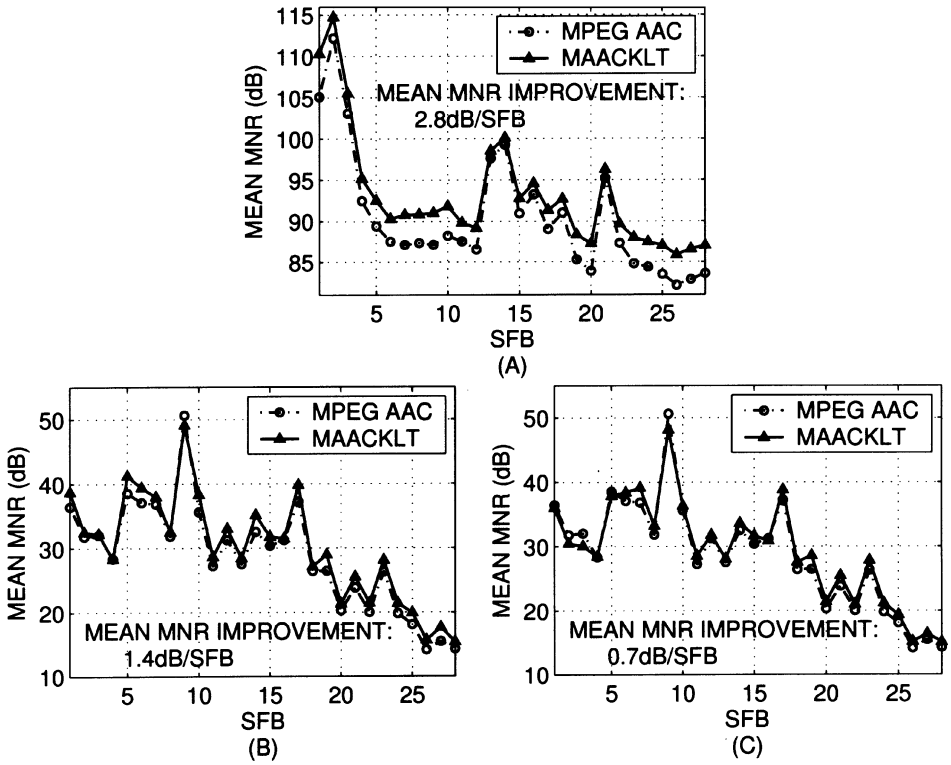

Fig. 18. The MNR comparison for (a) 10-channel "Herbie" using frequency-domain KLT, (b) 5-channel "Herre" using frequency-domain KLT, and (c) 5-channel "Herre" using time-domain KLT.

codes. A much lower amount of encoding/decoding time of MAACKLT is expected if the source code for the KLT pre-processing part is carefully re-written to optimize the performance.

In channel-scalable decoding, when packets belonging to less important channels are dropped during transmission in the heterogeneous network, the audio concealment part adds a negligible amount of additional complexity in the MAACKLT decoder. The decoding time remains about the same as that of regular bit rate decoding at 64 kbit/sec/ch when all packets are received at the decoder side.

VIII. Experimental Results

A. Multichannel Audio Coding

The proposed MAACKLT algorithm has been implemented and tested under the PC Windows environment. We supplemented an inter-channel redundancy removal block and a channel transmission control block to the basic source code structure of MPEG-2 AAC [15]. The proposed algorithm is conveniently parameterized to accommodate various input parameters, such as the number of audio channels, the desired bit rate, the window size of temporal adaptation, etc.

We have tested the coding performance of the proposed MAACKLT algorithm with three 10-channel sets of audio data "Messiah", "Band"[3] and "Herbie"[4] and one 5-channel set audio data "Herre" at a typical rate of 64 kbit/sec/ch. "Messiah" and "Band" are class III audio files, while "Herbie" and "Herre" are class II audio files. Fig. 18(a) and (b) show the mean Mask-to-Noise-Ratio (MNR) comparison between the original AAC[5] and the MAACKLT scheme for the 10-channel set "Herbie" and the 5-channel set "Herre," respectively. The mean MNR values in these figures are calculated via

$$\text{mean MNR}_{\text{SFB}} = \frac{\sum_{\text{channel}} \text{MNR}_{\text{channel,SFB}}}{\text{number of channels}} \tag{17}$$

where SFB represent the "scale factor band." The mean MNR improvement shown in these figures are calculated via

$$\text{mean MNR improvement} = \frac{\sum_{\text{SFB}} \left(\text{mean MNR}_{\text{SFB}}^{\text{MAACKLT}} - \text{mean MNR}_{\text{SFB}}^{\text{AAC}}\right)}{\text{number of SFB}}. \tag{18}$$

Experimental results shown in Fig. 18(a) and (b) are generated by using the frequency-domain nonadaptive KLT method. These plots clearly indicate that MAACKLT outperforms AAC in the objective MNR measurement for most scale factor bands and achieves mean MNR improvement of more than 1 dB for both test audio. It implies that, compared with AAC, MAACKLT can achieve a higher compression ratio while maintaining similar indistinguishable audio quality. It is worthwhile to mention that no software optimization has been performed for any codec used in this section and all coder blocks adopted from AAC have not been modified to improve the performance of our codec.

Fig. 18(c) shows the mean MNR comparison between AAC and MAACKLT with the time-domain KLT method using 5-channel set "Herre." Compared with the result shown in

[3]"Band" is a rock band music lively recorded in a football field.

[4]"Herbie" is a piece of music played by an orchestra.

[5]All audio files generated by AAC in this section are processed by the AAC main profile codec.

Fig. 18(b), we confirm that frequency-domain KLT achieves a better coding performance than time-domain KLT.

The experimental result for the temporal-adaptive approach for 10-channel set "Messiah" is shown in Fig. 19. This result verifies that a shorter adaptive period de-correlates the multichannel signal better but sacrifices the coding performance by adding the overhead in the bitstream. On the other hand, if the covariance matrix is not updated frequently enough, inter-channel redundancy cannot be removed to the largest extent. As shown in the figure, to compromise these two constraints, the optimal adaptation period for "Messiah" is around 10 s.

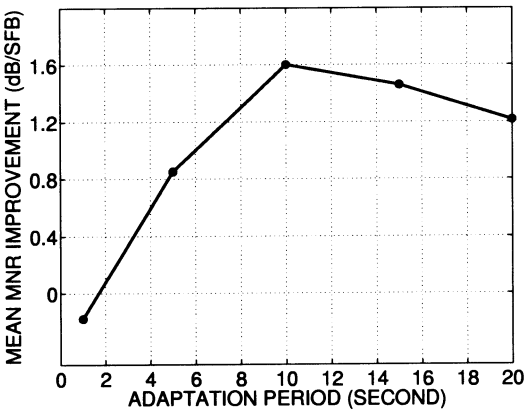

Fig. 19. Mean MNR improvement over AAC for temporal-adaptive KLT applied to the coding of 10-channel "Messiah," where the overhead information is included in the overall bit rate calculation.

B. Audio Concealment With Channel-Scalable Coding

As described in Section V, when packets of one channel from a channel pair are lost, we can conceal the missing channel at the decoder side. Experimental results show that the quality of the recovered channel pair with the AAC bitstream is much worse than that of the MAACKLT bitstream when it is transmitted under the same network condition.

Take the test audio "Herre" as an example. If one signal of the L/R channel pair is lost, the reconstructed R channel using the AAC bitstream has obvious distortion and discontinuity in several places while the reconstructed R channel by using the MAACKLT bitstream has little distortion and is much smoother. If one signal of the Ls/Rs channel pair is lost, the reconstructed Rs channel using the AAC bitstream has larger noise in the first one to two seconds in comparison with that of MAACKLT. The corresponding MNR values are compared in Fig. 20(a) and (b) when AAC and MAACKLT are used, missing channels are concealed when packets of one channel from L/R and Ls/Rs channel pairs are lost. We see clearly that MAACKLT achieves better MNR values than AAC for about 2 dB per scale factor band for both cases.

For a typical 5.1 channel configuration, when packets of more than two channels are dropped, which implies that at least one channel pair's information is lost, some lost channel can no longer be concealed from the received AAC bitstream. In contrast, the MAACKLT bitstream can still be concealed to obtain a full 5.1 channel audio with poorer quality. Although the recovered channel pairs do not sound exactly the same as the original ones, a reconstructed full multichannel audio would give the listener a much better acoustical effect than a three- or mono-channel audio.

Take the 5-channel "Messiah," which includes C, L, R, Ls and Rs channels, as an example. At the worst case, when packets of four channels are dropped and only data of the most important channel are received at the decoder side, the MAACKLT algorithm can still recover 5-channel audio. Compared with the original sound, the recovered Ls and Rs channels lost most of the reverberant sound effect. Since eigen-channel 1 does not contain much reverberant sound, the MAACKLT decoder can hardly recover these reverberant sound effects in the Ls and Rs channels.

Similar experiments were also performed by using test audio "Herre." However, the advantage of MAACKLT over AAC is not as obvious as that of test audio "Messiah." The reason can be easily found out from the original covariance matrix as shown in Fig. 2. It indicates that little correlation exists between SCE and CPE for class II test audio such as "Herre." Thus, once one CPE is lost, little information can be recovered from other CPEs or SCEs.

C. Subjective Listening Test

In order to further confirm the advantage of the proposed algorithm, a formal subjective listening test according to ITU recommendations [16]–[18] was conducted in an audio lab to compare the coding performance of the proposed MAACKLT algorithm and that of the MPEG AAC main profile codec. At the bit rate of 64 kbit/sec/ch, the reconstructed sound clips are supposed to have perceptual quality similar to that of the original ones. This implies that the difference between MAACKLT and AAC would be so small that nonprofessionals can hardly hear it. Thus, instead of inviting a large number of nonexpert listeners, four well-trained professionals, who have no knowledge of either of two algorithms, participated in the listening test [18]. During the test, for each test sound clip, subjects listened to three versions of the same sound clip, i.e., the original one followed by two processed ones (one by MAACKLT and one by AAC in random order), subjects were allowed to listen to these files as many times as possible until they were comfortable to give scores to the two processed sound files for each test material.

The five-grade impairment scale given in Recommendation ITU-R BS. 1284 [17] was adopted in the grading procedure and utilized for final data analysis. Four multichannel audio materials, i.e., "Messiah", "Band", "Herbie" and "Herre", are all used in this subjective listening test. According to ITU-R BS. 1116-1 [16], audio files selected for the listening test are of short durations (10 to 20 seconds long), so all test files coded by MAACKLT are generated by nonadaptive frequency-domain KLT method.

Fig. 21 shows the listening test results, where bars represent the score given to each test material coded at 64 kbit/sec/ch. The dark shaded area on the top of each bar represents the 95% confidence interval, where the middle line shows the mean value and the other two lines at the boundary of the dark shaded area represent the upper and lower confidence limits [19]. Fig. 21

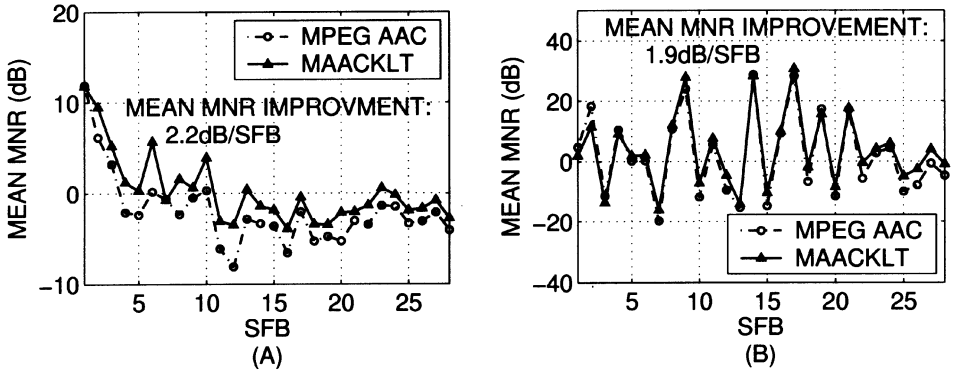

Fig. 20. MNR comparison for 5-channel "Herre" when packets of one channel from the (a) L/R and (b) Ls/Rs channel pairs are lost.

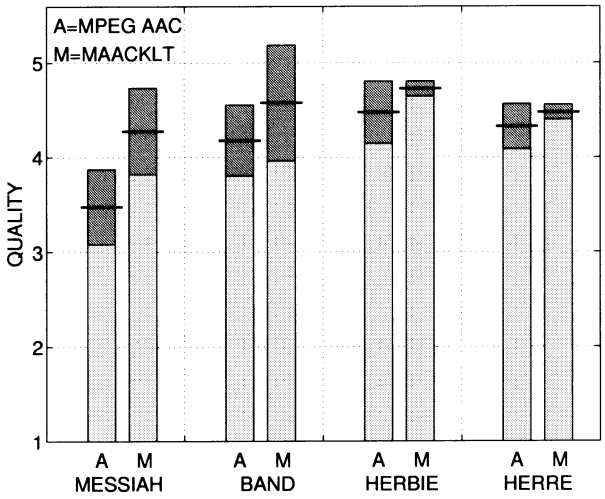

Fig. 21. Subjective listening test results.

indicates that the proposed MAACKLT algorithms outperforms MPEG AAC in all four test pieces, and indicates statistically signficant improvements for the "Messiah" and "Band" pieces.

Besides the 95% confidence interval comparison, it is possible to analyze the obtained listening test results with the sign-test [20], [21] as shown below. Table II lists listening test results in terms of signs $+$, 0 and $-$, which represent cases where the score given to the sound file processed by MAACKLT is higher than, the same as or worse than that of AAC, respectively. It is assumed that the score given by each listener to each sound file is independent of each other so that these test results can be viewed as 16 independent experiments. The total number S of sign $+$ can be viewed as a random variable having a binomial distribution with parameters n and p, where n is the number of experiments and p is the probability of getting sign $+$. Let us consider the following null hypothesis (H_0) and its counter hypothesis (H_1):

H_0: The proposed MAACKLT algorithm is no better than AAC.

H_1: The proposed MAACKLT algorithm is better than AAC.

TABLE II
PERFORMANCE COMPARISON OF AAC AND MAACKLT

	listener #1	listener #2	listener #3	listener #4
Messiah	+	+	+	+
Band	+	-	+	+
Herbie	+	0	+	+
Herre	+	+	0	+

Under the null hypothesis, the probability that a listener gives a higher score to the sound file processed by MAACKLT should be smaller than or equal to 0.5 (i.e., $p \leq 0.5$).

The Strict Sign Test : In the strict sign test, all experimental results are taken into account, i.e., $n = 16$. Because $p \leq 0.5$ (under H_0) and

$$\begin{aligned}(1-p)^{16-i}p^i &\leq \left[\frac{(16-i)(1-p)+ip}{16}\right]^{16} \\ &= \left[\frac{16-i+(2i-16)p}{16}\right]^{16} \\ &\leq \left[\frac{\frac{16-i+(2i-16)}{2}}{16}\right]^{16} \\ &= 0.5^{16} (\text{for } \forall\, i \geq 8)\end{aligned}$$

where the first inequality is based on the fact that the arithmetic mean is greater than or equal to the geometric mean. The probability of getting a result as shown in Table II or getting a result which is even more favorable to our proposed algorithm under H_0 is

$$\begin{aligned}P[S \geq 13] &= \sum_{13 \leq i \leq 16} \binom{16}{i} (1-p)^{16-i}p^i \\ &\leq \sum_{13 \leq i \leq 16} \binom{16}{i} 0.5^{16} = \frac{697}{65\,536} \approx 0.01.\end{aligned}$$

The above inequality indicates that, under null hypothesis H_0, the probability of getting a listening test result that is so favorable (as given in Table II) or even more favorable to the proposed algorithm would be as small as 1%. Thus, it is concluded that the null hypothesis H_0 is rejected in favor of H_1 at the level of 0.01.

The Loose Sign Test: In the loose sign test, the experiment that has sign 0 in Table II is not counted. Then, we have total of 14 effective experimental result, i.e., $n = 14$. The probability of getting a result as shown in Table II or getting a result which is even more favorable to our proposed algorithm under the null hypothesis H_0 is

$$P[S \geq 13] = \sum_{13 \leq i \leq 14} \binom{14}{i} (1-p)^{14-i} p^i \leq \sum_{13 \leq i \leq 14} \binom{14}{i} 0.5^{14} = \frac{15}{16\,384} < 0.001.$$

This means that, if the null hypothesis is true, the probability of getting a listening test result so favorable or even more favorable to the proposed algorithm would be as small as 0.1%. In other words, the null hypothesis H_0 is rejected in favor of H_1 at the level of 0.001.

Based on the strict sign test or the loose sign test shown above, we reject the null hypothesis with a comfortable degree of confidence and conclude that the proposed algorithm MAACKLT has a better performance than that of AAC.

IX. Conclusion

We presented a new channel-scalable high-fidelity multichannel audio compression scheme called MAACKLT based on the existing MPEG-2 AAC codec. This algorithm explores the inter- and intra-channel correlation in the input audio signal and allows channel-scalable decoding. The compression technique utilizes KLT in the pre-processing stage to remove the inter-channel redundancy, then compresses the resulting relatively independent eigen-channel signals with a modified AAC main profile encoder module, and finally uses a prioritized transmission policy to achieve quality scalability. The novelty of this technique lies in its unique and desirable capability to adaptively vary the characteristics of the inter-channel de-correlation transform as a function of the covariance of a certain period of music and its ability to reconstruct different quality audio with single bitstream. It achieves a good coding performance especially for the input audio source whose channel number goes beyond 5.1. In addition, it outperforms AAC according to both objective (MNR measurement) and subjective (listening) tests at the typical low bit rate of 64 kbit/sec/ch while maintaining a similar computational complexity for both encoder and decoder modules. Moreover, compared with AAC, the channel-scalable property of MAACKLT enables users to conceal full multichannel audio of reasonable quality without any additional cost.

References

[1] *Digital Audio Compression Standard (AC-3)*, aTSC Document A/52.

[2] K. Brandenburg and M. Bosi, "ISO/IEC MPEG-2 advanced audio coding: Overview and applications," in AES 103rd Conv., New York, Sept. 1997, ser. AES preprint 4641.

[3] M. Davis, "The AC-3 multichannel coder," in AES 95th Conv., New York, Oct. 1993, ser. AES preprint 3774.

[4] C. Todd, G. Davidson, M. Davis, L. Fielder, B. Link, and S. Vernon, "AC-3: Flexible perceptual coding for audio transmission and storage," in AES 96th Conv., Amsterdam, The Netherlands, February 1994, ser. AES preprint 3796.

[5] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, M. Dietz, J. Herre, G. Davidson, and Y. Oikawa, "ISO/IEC MPEG-2 advanced audio coding," in AES 101st Conv., Los Angeles, CA, Nov. 1996, ser. AES preprint 4382.

[6] J. Johnson and A. Ferreira, "Sum-difference stereo transform coding," in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing*, 1992, pp. 569–571.

[7] D. Yang, H. Ai, C. Kyriakakis, and C.-C. Kuo, "An inter-channel redundancy removal approach for high-quality multichannel audio compression," in *AES 109th Conv.*, Los Angeles, CA, September 2000, ser. AES preprint 5238.

[8] H. Fuchs, "Improving joint stereo audio coding by adaptive inter-channel prediction," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1993, pp. 39–42.

[9] S. Kuo and J. D. Johnston, "A study of why cross channel prediction is not applicable to perceptual audio coding," *IEEE Signal Processing Lett.*, vol. 8, Sept. 2001.

[10] D. Wu, Y. T. Hou, and Y. Zhang, "Transporting real-time video over the internet: Challenges and approaches," *Proc. IEEE*, vol. 88, pp. 1855–1877, Dec. 2000.

[11] D. Yang, H. Ai, C. Kyriakakis, and C.-C. Kuo, "An exploration of Karhunen-Loève transform for multichannel audio coding," *Proc. SPIE*, vol. 4207, pp. 89–100, 2000.

[12] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[13] D. Yang, H. Ai, C. Kyriakakis, and C.-C. Kuo, "Adaptive Karhunen-Loève transform for enhanced multichannel audio coding," *Proc. SPIE*, vol. 4475, pp. 43–54, 2001.

[14] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1991.

[15] "Information Technology—Generic Coding of Moving Pictures and Associated Audio Information—Part 7 Advanced Audio Coding,", ISO/IEC 13 818-7, 1997.

[16] "Methods for the Subjective Assessment of Small Impairments in Audio Systems Including Multichannel Sound Systems,", R. I.-R. B. 1116-1.

[17] "Methods for the Subjective Assessment of Sound Quality—General Requirements,", R. I.-R. B. 1284.

[18] "Pre-Selection Methods for the Subjective Assessment of Small Impairments in Audio Systems,", R. I.-R. B. 1285.

[19] R. A. D Jr and W. R. Harvey, *Experimental Design ANOVA, and Regression*. New York: Harper & Row, 1987.

[20] X. Chen, *Advanced Mathematical Statistics*. He Fei, China: Press of Univ. Science and Technology of China, 1999.

[21] R. A. Lohnson and G. Bhattacharyya, *Statistics: Principles and Methods*, 4th ed. New York: Wiley, 2001.

[22] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*. Berlin, Germany: Springer-Verlag, 1990.

[23] J. Saghri, A. Tescher, and J. Reagan, "Practical transform coding of multispectral imagery," *IEEE Signal Processing Mag.*, vol. 12, pp. 32–43, Jan. 1995.

[24] J. Lee, "Optimized quadtree for Karhunen-Loève transform in multispectral image coding," *IEEE Trans. Image Processing*, vol. 8, pp. 453–461, Apr. 1999.

Dai Yang (M'03) received the B.S. degree in electronics from Peking University, Beijing, China, in 1997 and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1999 and 2002, respectively.

She is currently a Postdoctoral Researcher in NTT Cyber Space Laboratories in Tokyo, Japan. Her research interests are in the areas of digital signal and image processing, audio, speech, video, graphics coding, and their network/wireless applications.

Hongmei Ai (S'95–M'97) received the B.S. and M.S. and Ph.D. degrees in electronic engineering at Tsinghua University, Beijing, China, in 1991 and in 1996, respectively.

She was Assistant Professor (1996–1998) and Associate Professor (1998–1999) in the Department of Electronic Engineering at Tsinghua University. She was a Visiting Scholar in the Department of Electrical Engineering-Systems at the University of Southern California, Los Angeles, from 1999 to 2002. She is a Principal Software Engineer at Pharos Science & Applications, Inc., Torrance, CA. Her research interests focus on signal and information processing and communications, including data compression, video and audio processing, and wireless communications.

Chris Kyriakakis (M'97) is an Associate Professor in the Electrical Engineering—Systems Department at the University of Southern California (USC), Los Angeles. He heads the Immersive Audio Laboratory and his research is focused on multichannel audio acquisition, synthesis, rendering, room equalization, streaming, and compression. He is also the Research Area Director for Sensory Interfaces in the Integrated Media Systems Center, an NSF ERC at USC.

C.-C. Jay Kuo (S'83–M'86–SM'92–F'99) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1980 and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1985 and 1987, respectively, all in electrical engineering.

He was Computational and Applied Mathematics (CAM) Research Assistant Professor in the Department of Mathematics at the University of California, Los Angeles, from October 1987 to December 1988. Since January 1989, he has been with the Department of Electrical Engineering—Systems and the Signal and Image Processing Institute at the University of Southern California, where he currently has a joint appointment as Professor of electrical engineering and mathematics. His research interests are in the areas of digital signal and image processing, audio and video coding, wavelet theory and applications, multimedia technologies and large-scale scientific computing. He has authored around 500 technical publications in international conferences and journals, and graduated more than 50 Ph.D. students. He is Editor-in-Chief for the *Journal of Visual Communication and Image Representation* and Editor for the *Journal of Information Science and Engineering*.

Dr. Kuo is a member of SIAM, ACM, and a Fellow of SPIE. He is Associate Editor for IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING. He served as Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING in 1995–1998 and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 1995–1997. He received the National Science Foundation Young Investigator Award (NYI) and Presidential Faculty Fellow (PFF) Award in 1992 and 1993, respectively.