Adaptive Karhunen-Loeve Transform for Enhanced Multichannel Audio Coding

Dai Yang, Hongmei Ai, Chris Kyriakakis and C.-C. Jay Kuo

Integrated Media Systems Center and Department of Electrical Engineering-Systems
University of Southern California, Los Angeles, CA 90089-2564, USA

ABSTRACT

A modified MPEG Advanced Audio Coding (AAC) scheme based on the Karhunen-Loeve transform (KLT) to remove inter-channel redundancy, which is called the MAACKL method, has been proposed in our previous work. However, a straightforward coding of elements of the KLT matrix generates about 240 bits per matrix for typical 5 channel audio contents. Such an overhead is too expensive so that it prevents MAACKL from updating KLT dynamically in a short period of time. In this research, we study the de-correlation efficiency of adaptive KLT as well as an efficient way to encode elements of the KLT matrix via vector quantization. The effect due to different quantization accuracy and adaptation period is examined carefully. It is demonstrated that with the smallest possible number of bits per matrix and a moderately long KLT adaptation time, the MAACKL algorithm can still generate a very good coding performance.

Keywords: Karhunen-Loeve Transform (KLT), Advanced Audio Coding (AAC), Mask-to-Noise Ratio (MNR), Scalar Quantization (SQ), Vector Quantization (VQ)

1. INTRODUCTION

Since audio CD appeared in 1982 it has been widely accepted as a way to deliver high-quality digital music. One of the drawbacks of the CD format, however, is that it allows for only two channels. Multichannel audio can provide listeners with a more compelling experience and is becoming more and more widely used by the music industry. New delivery formats such as DVD-Audio and Multichannel SACD allow up to six channels of program material. As a result, an efficient coding scheme is needed for storage and transmission of multichannel audio. This subject has generated a lot of research activity recently [1–5].

Based on today's most distinguished multichannel audio coding system, a Modified Advanced Audio Coding with Karhunen-Loeve Transform (MAACKL) method was proposed to perceptually losslessly compress a multichannel audio source in Yang *et al.*'s work [6,7]. This method utilizes the Karhunen-Loeve Transform (KLT) in the pre-processing stage for the powerful multichannel audio compression tool, i.e. MPEG Advanced Audio Coding (AAC), to remove inter-channel redundancy and further improve the coding performance. However, as described in these papers, each element of the covariance matrix, from which the KLT matrix is derived, is scalar quantized to 16 bits. This results in 240 bits overhead for each KL transform matrix for typical 5 channel audio contents. Since the bit budget is the most precious resource in the coding technique, every effort must be made to minimize the overhead due to the additional pre-processing stage while maintaining a similar high-quality coding performance. Moreover, the original MAACKL algorithm did not fully explore the KLT temporal adaptation effect.

In this research, we investigate the KLT de-correlation efficiency versus the quantization accuracy and the temporal KLT adaptive period. Extensive experiments on the quantization of the covariance matrix by using scalar and vector quantizers have been performed. Based on these results, the following two interesting points are concluded.

- Coarser quantization can dramatically degrade the de-correlation capability in terms of the normalized covariance matrix of de-correlated signals. However, the degradation of decoded multichannel audio quality is not as obvious.
- Shorter temporal adaptation of KLT will not significantly improve the de-correlation efficiency while considerably increase the overhead. Thus, a moderately long adaptation time is a good choice.

E-mail contact information: daiyang@usc.edu for Dai Yang; ahm@costard.usc.edu for Dr. Hongmei Ai; ckyriak@imsc.usc.edu for Prof. Chris Kyriakakis; and cckuo@sipi.usc.edu for Prof. C.-C. Jay Kuo

It is shown in this work that, with vector quantization, we can reduce the overhead from more than 200 bits to less than 3 bits per KL transform while maintaining comparable coding performance. Even with scalar quantization, a much lower overhead bit rate can still generate decoded audio with comparable quality. Our experimental results indicate that although a coarser quantization of the covariance matrix gives a poorer de-correlation effect, reduction of bits in the overhead is able to compensate this degradation to result in a similar coding performance in terms of the objective MNR measurement.

The rest of this paper is organized as follows: in section 2 we explain how inter-channel de-correlation is performed over multichannel audio sources. In section 3 we introduce vector quantization and its application to the MAACKL algorithm. In sections 4 and section 5 we explore how the quantization method and the temporal adaptive scheme affect the KLT de-correlation efficiency and the coding performance by applying scalar and vector quantizers to encode the KLT matrix with a range of different bit rates. In section 6 we examine computational complexity issues. Some experimental results are presented in section 7. Finally concluding remarks are given in section 8.

2. INTER-CHANNEL DE-CORRELATION

Removing the inter-channel correlation of multi-channel audio has been demonstrated to result in bandwidth reduction in our previous work, which is achieved via KLT in the pre-processing stage [6,7]. KLT is a signal dependent transform. It is adopted since it is theoretically the optimal method to de-correlate signals across channels. Figure 1 illustrates how KLT is performed on multichannel audio signals, in which columns of the KL transform matrix is composed of eigenvectors calculated from the cross-covariance matrix associated with the original multichannel audio signals.

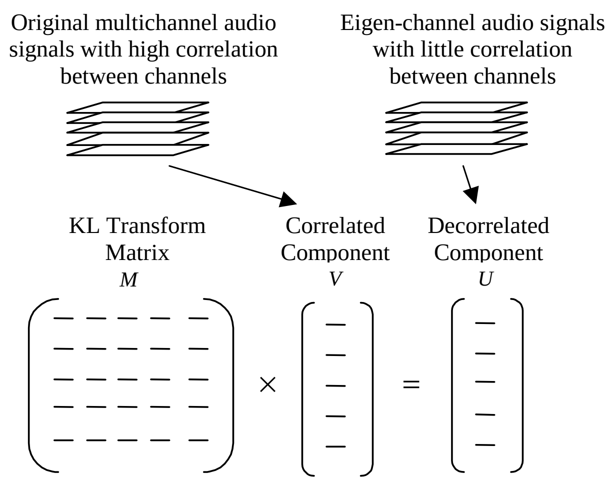


Figure 1. Inter-channel de-correlation via KLT.

If the input audio has n channels, then we form a $n \times n$ KL transform matrix M composing of n eigenvectors of the cross-covariance matrix associated with these n channels. Let $V(i)$ denote the vector whose n elements are the i^{th} sample value in channel $1, 2, \ldots, n$, i.e.

$$V(i) = [x_1, x_2, \ldots, x_n]^T,$$

where $i = 1, 2, \ldots, k$, x_j is the i^{th} sample value in channel j $(1 \leq j \leq n)$, k represents the number of samples in each channel, and $[*]^T$ represents the transpose of $[*]$. The mean vector μ_V and the covariance matrix C_V are defined as

$$\mu_V = E[V] = \frac{\sum_{i=1}^{k} V(i)}{k},$$

$$C_V = E[(V - \mu_V)(V - \mu_V)^T] = \frac{\sum_{i=1}^{k} [V(i) - \mu(i)][V(i) - \mu(i)]^T}{k}.$$

The KL transform matrix M is $M = [m_1, m_2, \ldots, m_n]^T$, where m_1, m_2, …, m_n are eigenvectors of C_V. The covariance of KL transformed signals is

$$\begin{aligned} E[(U-\mu_U)(U-\mu_U)^T] &= E[(MV - M\mu_V)(MV - M\mu_V)^T] \\ &= ME[(V-\mu_V)(V-\mu_V)^T]M^T \\ &= MC_VM^T \\ &= \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}, \end{aligned}$$

where λ_1, λ_2, …, λ_n are eigenvalues of C_V. Thus, the transform produces statistically de-correlated channels in the sense of having a diagonal covariance matrix.

3. VECTOR QUANTIZATION

The MAACKL algorithm described in our previous work [6,7] dealt only with scalar quantization of the covariance matrix. If the input audio material is short or if the KLT matrix is updated more frequently, the overhead that results from transmitting the covariance matrix will increase significantly, which will degrade the coding performance of the MAACKL algorithm to a certain extent. To alleviate this problem, we have to resort to a look-up-table (LUT) approach. Here, a stored table of pre-calculated covariance matrices is searched to find the one that approximates the estimated covariance matrix of the current block of the input audio. This approach yields a substantial savings in the overhead bit rate since only pointers to the table, instead of the entire covariance matrix itself, will be transmitted to the receiver.

Vector quantization (VQ) [8–10] provides an excellent choice to implement the LUT idea. By vector quantization, we identify a set of possible vectors both at the encoder and the decoder side. They are called the codebook. The VQ encoder pairs up each source vector with the closest matching vector (i.e. ”codeword”) in the codebook, thus ”quantizing” it. The actual encoding is then simply a process of sequentially listing the identity of codewords that match most closely with vectors making up the original data. The decoder has a codebook identical to the encoder, and decoding is a trivial matter of piecing together the vectors whose identity have been specified. Vector quantizers in this work consider the entire set of non-redundant elements of each covariance matrix as an entity, or a vector. The identified codebook should be general enough to include the characteristics of different types of multichannel audio sources. Since VQ allows for direct minimization of the quantization distortion, it results in smaller quantization distortion than scalar quantizers (SQ) at the same bit rate. In other words, VQ demands a smaller number of bits for source data coding while keeping the quantization error similar to that achieved with scalar quantizer.

Four different five-channel audio pieces (each containing center (C), left (L), right (R), left surround (Ls) and right surround (Rs) channels), are used to generate more than 80,000 covariance matrices. Each covariance matrix is treated as one training vector $\mathbf{X}$, which is composed of fifteen non-redundant elements of the covariance matrix as shown below.

$$\mathbf{X} = \begin{matrix} x_1 & & & & \\ x_2 & x_3 & & & \\ x_4 & x_5 & x_6 & & \\ x_7 & x_8 & x_9 & x_{10} & \\ x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \end{matrix},$$

where $x_1, x_2, \cdots, x_{15}$ are elements in the lower triangular part of the covariance matrix. During the codebook generation procedure, the Generalized Lloyd Algorithm (GLA) was run on the training sequence by using the simple square error distortion measurement, i.e.

$$d(\mathbf{X}, \mathbf{Q}(\mathbf{X})) = \sum_{\mathbf{i=1}}^{\mathbf{15}} [\mathbf{X} - \mathbf{Q}(\mathbf{X})]^{\mathbf{2}},$$

where $Q(\mathbf{X})$ represents the quantized value of $\mathbf{X}$. The same distortion measurement is used with the full searching method during the encoding procedure.

4. EFFICIENCY OF KLT DE-CORRELATION

The magnitudes of non-diagonal elements in a normalized covariance matrix provide a convenient metric to measure the degree of inter-channel correlation. 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, i.e.

$$C_N(i,j) = \frac{C(i,j)}{\sqrt{C(i,i) \times C(j,j)}},$$

where $C_N(i,j)$ and $C(i,j)$ are elements of the normalized covariance matrix and the cross-covariance matrix in row i and column j, respectively.

Tables 1 and Table 2 show the absolute values of non-redundant elements (i.e. elements in only the lower or the upper triangle) of the normalized covariance matrix calculated from original signals and KLT de-correlated signals respectively, where no quantization is performed during the KLT de-correlation. From these tables, we can easily see that KLT reduces the inter-channel correlation from around the order of 10^{-1} to the order of 10^{-4}.

Table 1. Absolute values of non-redundant elements of the normalized covariance matrix calculated from original signals.

1				
5.36928147e-1	1			
3.26056331e-1	1.02651220e-1	1		
1.17594877e-1	8.56662289e-1	5.12340667e-3	1	
7.46899187e-2	1.33213668e-1	1.15962389e-1	6.55651089e-2	1

Table 2. Absolute values of non-redundant elements of the normalized covariance matrix calculated from KLT de-correlated signals.

1				
1.67971275e-4	1			
2.15059591e-4	1.01530173e-3	1		
4.19255484e-4	4.03864289e-4	2.56863610e-4	1	
3.07486032e-4	4.23535476e-4	3.48484672e-4	5.20389082e-5	1

Table 3. Absolute values of non-redundant elements of the normalized covariance matrix calculated from scalar quantized KLT de-correlated signals.

1				
1.67971369e-4	1			
2.15059518e-4	1.01530166e-3	1		
4.19255341e-4	4.03863772e-4	2.56863464e-4	1	
3.07486076e-4	4.23536876e-4	3.48484820e-4	5.20396538e-5	1

In order to investigate the de-correlation efficiency affected by various quantization schemes, a sequence of experiments, including SQ and VQ with a different number of bits per element/vector, was performed. Table 3 shows the absolute values of non-redundant elements of the normalized covariance matrix calculated from KLT de-correlated signals, where each element of the covariance matrix is scalar quantized into 32 bits. Compared with Table 2, values in Table 3 are almost identical to those in Table 2 with less than 0.0001% distortion per element. This suggests that, with 32 bits per element scalar quantizer, we can almost faithfully reproduce the covariance matrix with negligible quantization error.

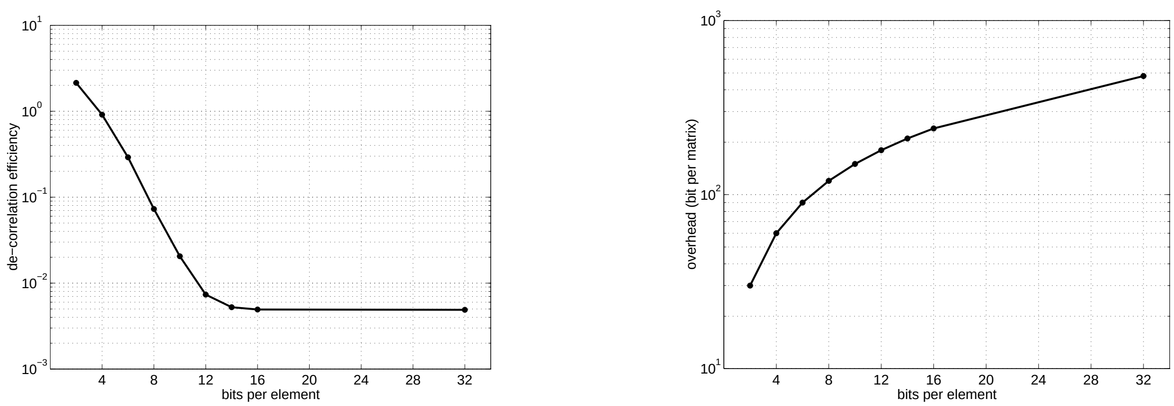
Figures 2 and Figure 3 illustrate how de-correlation efficiency and the corresponding overhead changes with SQ and VQ, respectively. It is observed that simple Mean Square Error (MSE) measurement is not a good choice when evaluating the de-correlation efficiency. A better measure is the average distortion D, which is the summation of magnitudes of lower triangular elements of the normalized covariance matrix, i.e.

$$D = \sum_{i=2}^{N} \sum_{j=1}^{i-1} |C_N(i,j)|, \tag{1}$$

where C_N is the normalized covariance matrix of signals after KLT de-correlation. The overhead in terms of bits per KLT matrix is calculated via

$$\begin{aligned} OH_s &= B_s \times N, & (2) \\ OH_v &= B_v, & (3) \end{aligned}$$

where B_s denotes the number of bits per element for SQ, N is the number of non-redundant elements per matrix, and B_v denotes the number of bits per codeword for VQ (recall that one KLT matrix is quantized into one codeword). For 5 channel audio material, N is equal to 15.

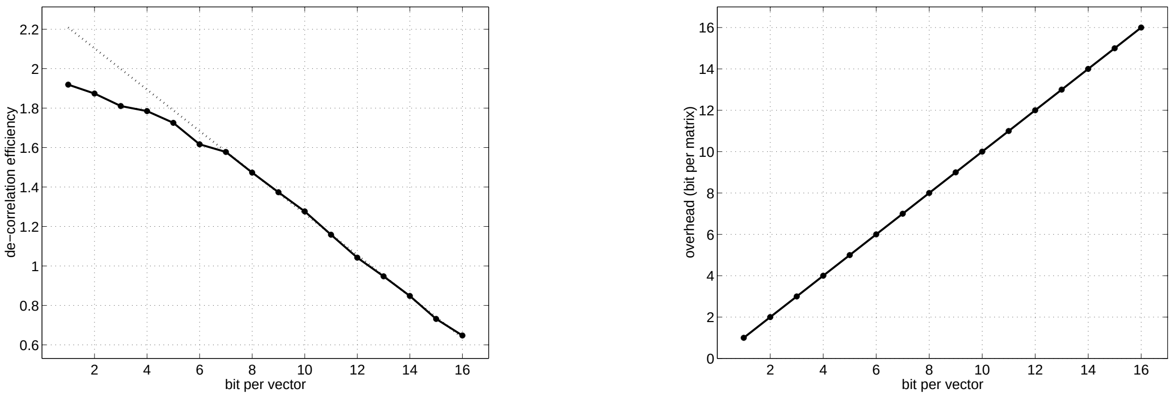


(a) (b)

Figure 2. (a) The de-correlation efficiency and (b) the overhead bit rate versus the number of bits per element.

Figure 2 (a) suggests that there is no significant degradation in de-correlation efficiency when the number of bits is reduced from 32 bits per element to 14 bits per element. However, further reduction in the number of bits per element will result in a dramatic increase of distortion D given in Equation (1). From Equation (2), we know that the overhead increases linearly as the number of bits per element increases with a gradient equals to N. This is confirmed by Figure 2 (b), in which the overhead is plotted as a function of the number of bits per element in the logarithmic scale. Compared with Figure 2 (a), we observe that the overhead OH increases much more rapidly than the decrease of the distortion D when the number of bits per element increases from 14 to 32. It indicates that when transmitting the covariance matrix with a higher bit rate, the improvement of de-correlation efficiency is actually not sufficient to compensate the loss due to a higher overhead rate.

The minimum number of bits per element for SQ is 2, since we need one bit for the sign and at least one bit for the absolute value for each element. To further reduce the number of bits per element can be achieved by using VQ. Figure 3 (a) illustrates that the average distortion D increases almost linearly when the number of bit per vector decreases from 16 bits per vector to 7 bits per vector, and then slows down when the bit per vector further decreases. Compared with Figure 2, it is verified that VQ results in smaller quantization distortion than SQ at any given bit



(b) (b)

Figure 3. (a) The de-correlation efficiency and (b) the overhead bit rate versus the number of bits per vector in VQ.

Table 4. De-correlation results with SQ.

bit/element	2	4	6	8	10	12	14	16	32
D	2.14	9.13e-1	2.91e-1	7.29e-2	2.05e-2	7.36e-3	5.24e-3	4.93e-3	4.89e-3
Overhead (bit/matrix)	30	60	90	120	150	180	210	240	480
Ave MNR (dB/sb)	N/A*	56.56	56.37	56.02	56.08	56.00	55.93	55.91	55.84

Using 2 bits per element, which quantizes each element into values of either 0 or ±1, leads to problems in later compression steps.

Table 5. De-correlation results with VQ.

bit/vector	1	2	3	4	5	6	7	8
D	1.92	1.87	1.81	1.78	1.73	1.62	1.58	1.47
Overhead (bit/matrix)	1	2	3	4	5	6	7	8
Ave MNR (dB/sb)	56.73	56.61	56.81	56.87	56.12	56.23	56.88	56.96
bit/vector	9	10	11	12	13	14	15	16
D	1.37	1.28	1.16	1.04	0.948	0.848	0.732	0.648
Overhead (bit/matrix)	9	10	11	12	13	14	15	16
Ave MNR (dB/sb)	56.42	55.97	56.08	56.28	55.83	55.72	56.19	55.87

rate. Figure 3 (b) shows how the overhead varies with the number of bits per covariance matrix. Note that VQ reduces the overhead bit rate more than a factor of N (which is 15 for 5 channel audio) with respect to SQ.

Tables 4 and Table 5 show the average distortion D, the overhead information and the average MNR value for SQ and VQ, respectively, where the average MNR value is calculated as below:

$$\text{mean MNR}_{subband} = \frac{\sum_{channel} \text{MNR}_{channel,subband}}{\text{number of channels}}, \tag{4}$$

$$\text{average MNR} = \frac{\sum_{subband} \text{mean MNR}_{subband}}{\text{number of subband}}. \tag{5}$$

The test audio material used to generate results in this section is a 5 channel performance of "Messiah" with the

KLT matrix updated every one second. As shown in Table 4, a fewer number of bits per element results in a higher distortion in exchange of a smaller overhead and, after all, a larger MNR value. Thus, although a smaller number of bits per element of the covariance matrix results in a larger average distortion D, the decrease of the overhead bit rate actually compensates this distortion and improves the MNR value for the same coding rate.

A similar argument applies to the VQ case as shown in Table 5 with only one minor difference. That is, the MNR value is nearly a monotonic function for the SQ case while it moves up and down slightly in a local region (fluctuating within 1 dB per subband) for the VQ case. However, the general trend is the same, i.e. a larger overhead in KLT coding degrades the final MNR value. We also noticed that even when using 1 bit per vector, vector quantization of the covariance matrix still gives good MNR results.

Our conclusion is that it is beneficial to reduce the overhead bit rate used in the coding of the covariance matrix of KLT, since a larger overhead has a negative impact on the rate-distortion tradeoff.

5. TEMPORAL ADAPTATION EFFECT

A multichannel audio program in general comprises of several different periods, each of which has its unique spectral signature. For example, a piece of music may begin with a piano preclude followed by a chorus. In order to achieve the highest information compactness, the de-correlation transform matrix must adapt to the characteristics of different sectoins of the program material. The MAACKL algorithm utilizes a temporal-adaptive approach, in which the covariance matrix is updated frequently. On one hand, the shorter the adaptive time, the more efficient the inter-channel de-correlation mechanism. On the other hand, since the KLT covariance matrix has to be coded for audio decoding, a shorter adaptive time contributes to a larger overhead in bit rates. Thus, it is worthwhile to investigate the tradeoff so that a good balance between this adaptive time and the final coding performance can be reached.

In Figure 4, we show the magnitude of the lower triangular elements of the normalized covariance matrix calculated from de-correlated signals by using different adaptive periods, where no quantization has been applied yet. These figures suggest that there is no significant improvement of the de-correlation efficiency when the KLT adaptive time decreases from 10 seconds to 0.05 second. As the overhead dramatically increases with the shorter adaptive time, the final coding performance may be degraded. In order to find the optimal KLT adaptive time, a thorough investigation is performed for both SQ and VQ.

First, let us look at how adaptive time affects the overhead bit rate. Suppose n channels are selected for simultaneous inter-channel de-correlation, the adaptive time is K seconds, i.e. each sub-block contains K seconds of audio, and M bits are transmitted to the decoder for each KL transform. The overhead bit rate $r_{overhead}$ is

$$r_{overhead} = \frac{M}{nK} \tag{6}$$

in bits per second per channel (bit/s/ch). This equation suggests that the overhead bit rate increases linearly with the number of bits used to encode and transmit the KL transform matrix. The overhead bit rate is, however, inversely proportional to the number of channels and the adaptive time. If SQ is used in the encoding procedure, each non-redundant element has to be sent. For n channel audio material, the size of the covariance matrix is $n \times n$, and the number of non-redundant elements is $n \times (n+1)/2$. If B_s bits are used to quantize each element, the total bit requirement for each KLT is $n(n+1)B_s/2$. Thus the overhead bit rate $r^{SQ}_{overhead}$ for SQ is equal to

$$r^{SQ}_{overhead} = \frac{(n+1)B_s}{2K}.$$

The overhead bit rate $r^{vQ}_{overhead}$ for VQ is simpler. It is equal to

$$r^{VQ}_{overhead} = \frac{B_v}{nK},$$

where B_v represent the number of bits used for each KLT covariance matrix.

The average MNR value (in dB) and the overhead bit rate (in the logarithm scale) versus the adaptive time for both SQ and VQ are shown in Figures 5 (a) and (b), respectively. The test material is 5-channel "Messiah", with 8 bits per element for SQ and 4 bits per vector for VQ for KLT de-correlation. The total coding bit rate (including

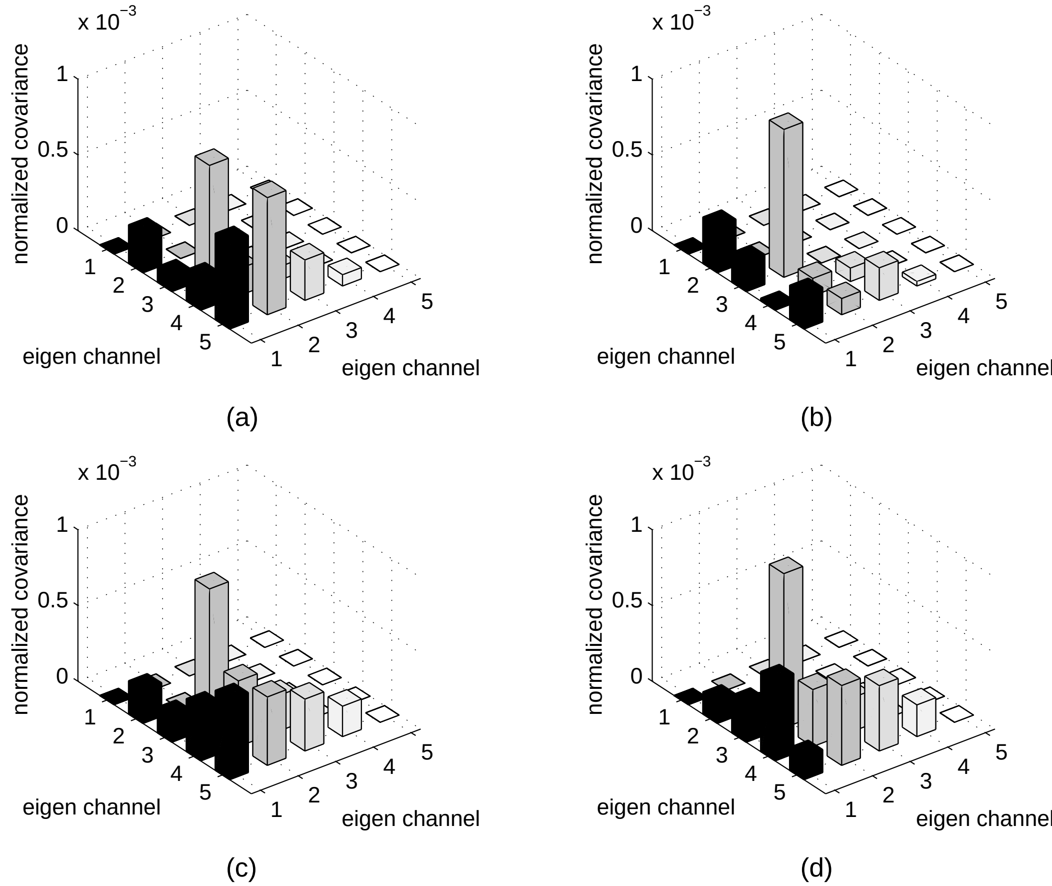


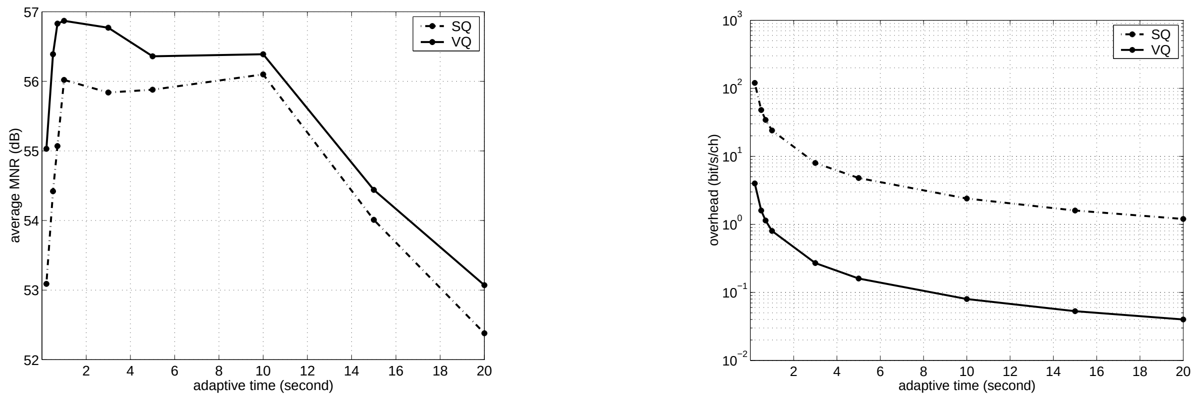
Figure 4. The magnitude of the lower triangular elements of the normalized covariance matrix calculated from de-correlated signals, where the adaptive time is equal to (a) 0.05, (b) 0.2, (c) 3, and (d) 10 seconds.

bits for the overhead and the content) is kept the same for all points in the two curves in Figure 5 (a). We have the following observations from these figures.

First, for the SQ case, the average MNR value remains about the same with less than 0.3 dB variation per subband when the adaptive time varies from 1 to 10 seconds. However, when the adaptive time is decreased furthermore, the overhead effect starts to dominate, and the average MNR value decreases dramatically. On the other hand, when the adaptive time becomes larger than 10 seconds, the average MNR value also decreases, which implies that the coding performance degrades if the KLT matrix is not updated frequently enough. For VQ, the changing pattern of the average MNR value versus the adaptive time is similar as that of SQ. However, compared with the scalar case, the average MNR value starts to degrade earlier at about 5 seconds. This is probably due to the effect that VQ gives less efficient de-correlation, so that more frequent adaptation of the KLT matrix will generate a better coding result. As shown in Figure 5 (a), it is clear that the average MNR generated by using VQ is always better than that of SQ and the difference becomes significant when the overhead becomes the dominant factor for KLT adaptive time less than 1 second.

6. COMPLEXITY ANALYSIS

The main concern of a VQ scheme is its computational complexity at the encoder side. For each D dimension vector, we need $O(DS)$ operations to find the best matched codeword from a codebook of size S using the full search technique. For a n channel audio, each covariance matrix is represented by a vector of dimension $n(n+1)/2$. Thus, for each KLT, we need $O(n^2S)$ operations. While for the scalar case, to quantize each element requires $O(1)$



(a) (b)

Figure 5. (a) Adaptive MNR results and (b) adaptive overhead bits for SQ and VQ for 5-channel Messiah.

operations, and for each covariance matrix, we need $O(n^2)$ operations. Suppose that the input audio is of L seconds long, and the KLT matrix is updated each K seconds. Then, there will be totally $\lceil L/K \rceil$ covariance matrices to be quantized[†]. This means we need

$$\begin{aligned} O(\lceil L/K \rceil n^2) &= O(Ln^2/K) \quad \text{for scalar quantization,} \\ O(\lceil L/K \rceil n^2 S) &= O(Ln^2 S/K) \quad \text{for vector quantization,} \end{aligned}$$

operations.

Thus, for a given test audio source, the complexity is inversely proportional to KLT adaptive time for either quantization scheme. For VQ, the complexity is also proportional to the codebook size. Compared with SQ, VQ requires more operations by a factor of S. To reduce the computational complexity, we should limit the codebook size and set the KLT adaptation time as long as possible while keeping the desired coding performance.

Experimental results shown in Section 4 suggests that a very small codebook size is usually good enough to generate the desired compressed audio. By choosing a small codebook size and keeping the KLT adaptation time long enough, we do not only limit the additional computational complexity, but also save the overhead bit requirement. At the decoder side, VQ demands just a table look up procedure and its complexity is comparable to that of SQ.

7. EXPERIMENTAL RESULTS

We tested the modified MAACKL method by using two five-channel audio sources "Messiah" and "Ftbl" at different bit rates varying from a typical rate of 64 kbit/s/ch to a very low bit rate of 16 kbit/s/ch. Figures 6 and Figure 7 show the mean MNR comparison between SQ and VQ for test audio "Messiah" and "Ftbl", respectively, where the KLT matrix adaptation time is set to 10 seconds. The mean MNR values in these figures are calculated by Equation (4). In order to show the general result of scalar and vector cases, a moderate bit rate (i.e. 8 bits per element for SQ and 4 bits per vector for VQ) is adopted here. From these figures, we see that, compared with SQ, VQ generates comparable mean MNR results at all bit rates, and VQ even outperforms SQ at all bit rates for some test sequence such as "Messiah".

8. CONCLUSIONS

To enhance the MAACKL algorithm proposed earlier, we examined the relationship between coding of the KLT covariance matrix with different quantization methods, KLT de-correlation efficiency and the frequency of KLT information update extensively in this research. In specific, we investigated how different quantization method affects the final coding performance by using objective MNR measurements. It is demonstrated that to reduce the overhead bit rate generally provides a better tradeoff for the overall coding performance. This can be achieved by adopting a

[†]where $\lceil * \rceil$ represents the smallest integer which is greater than or equal to $*$.

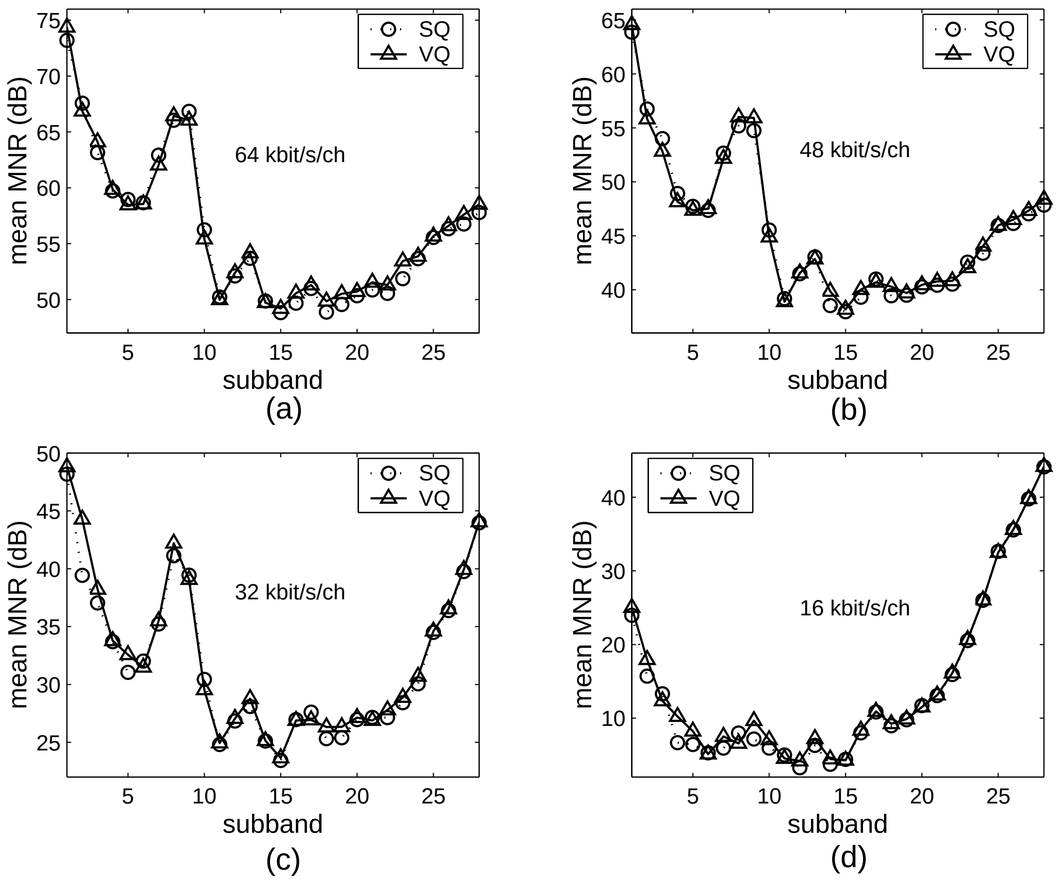


Figure 6. MNR result using test audio "Messiah" coded at (a) 64 kbit/s/ch, (b) 48 kbit/s/ch, (c) 32 kbit/s/ch, and (d) 16 kbit/s/ch.

smallest possible bit rate to encode the covariance matrix together with moderately long KLT adaptation period to generate the desired coding performance. Besides, a small codebook size in VQ do not increase the computational complexity significantly.

ACKNOWLEDGMENTS

This research has been funded by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center. Cooperative Agreement Number: EEC-9529152

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 convention*, AES preprint 4641, (New York), Spetember 1997.
3. M. Davis, “"The AC-3 Multichannel Coder",” in *AES 95th convention*, AES preprint 3774, (New York), October 1993.
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 convention*, AES preprint 3796, (Amsterdam), February 1994.
5. M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, H. F. K. Akagiri, M. Dietz, J. Herre, G. Davidson, and Y. Oikawa, “"ISO/IEC MPEG-2 Advanced Audio Coding",” in *AES 101st convention*, AES preprint 4382, (Los Angeles), November 1996.

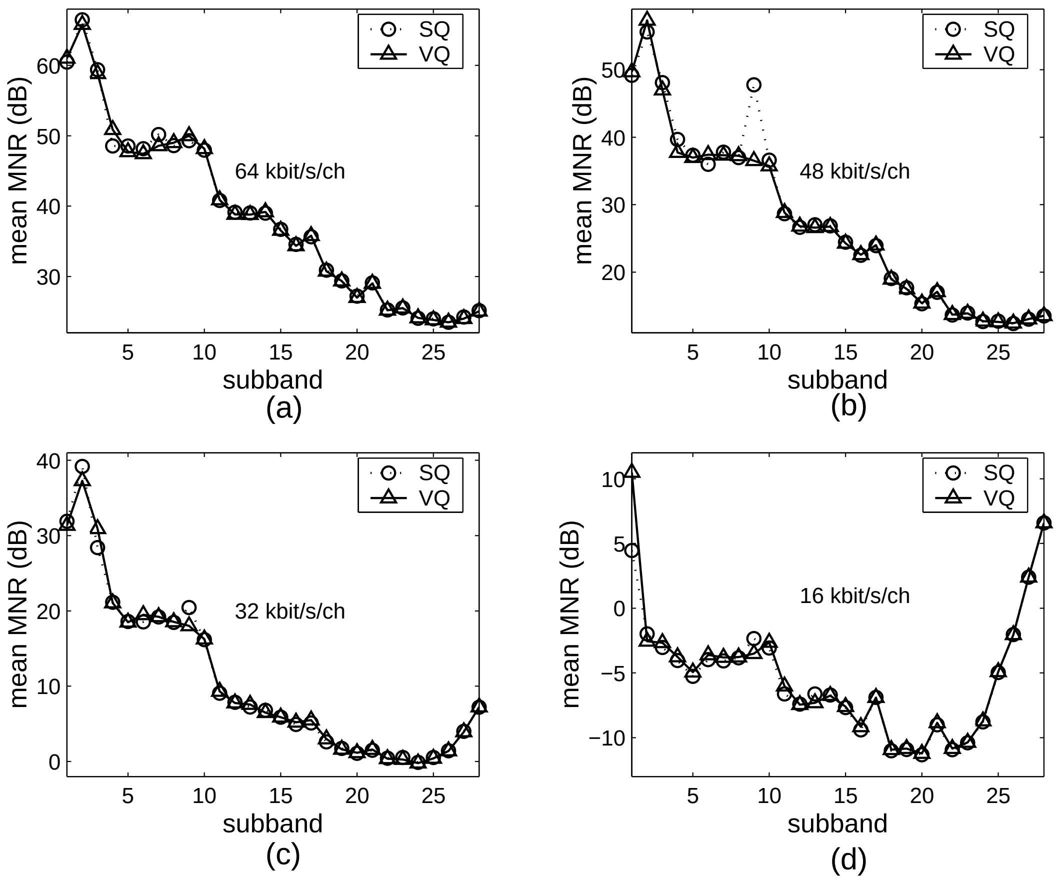


Figure 7. MNR result using test audio "Ftbl" coded at (a) 64 kbit/s/ch, (b) 48 kbit/s/ch, (c) 32 kbit/s/ch, and (d) 16 kbit/s/ch.

6. 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 convention*, AES preprint 5238, (Los Angeles, CA), September 2000.
7. D. Yang, H. Ai, C. Kyriakakis, and C.-C. Kuo, ""An Explorition of Karhunen-Loeve Transform for Multichannel Audio Coding"," in *Conference of Electronic Cinema, SPIE's International Symposium on Information Technologies*, (Boston, MA), November 5–8 2000.
8. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1991.
9. K. K. Paliwal and B. S. Atal, "Efficient vector quantization of lpc parameters at 24 bits/frame," *IEEE Transactions on Speech and Audio Processing* **1**(1), January 1993.
10. W. H. Equitz, "A new vector quantization clustering algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing* **37**(10), October 1989.
11. ISO/IEC JTC1/SC29/WG11 N1650, *"IS 13818-7 (MPEG-2 Advanced Audio Coding, AAC)"*.
12. ISO/IEC JTC1/SC29/WG11 N2262, *"ISO/IEC TR 13818-5, Software Simulation"*.
13. J. Saghri, A. Tescher, and J. Reagan, ""Practical Transform Coding of Multispectral Imagery"," *IEEE signal processing magazine* , January 1995.
14. J. Saghri and A. Tescher, ""Near-lossless bandwidth compression for radiometric data"," *Optical Engineering* **30**, July 1991.
15. J. Lee, ""Optimized Quadtree for Karhunen-Loeve Transform in Multispectral Image Coding"," *IEEE Transactions on Image Processing* **8**, April 1999.

16. W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *"Numerical Recipes in C, the Art of Scientific Computing"*, Cambridge University Press, second edition ed., 1992.
17. G. H. Golub and C. F. V. Loan, *"Matrix Computations"*, Baltimore, MD: Johns Hopkins Univ. Press, 1983.