LOW BITRATE AUDIO CODING USING GENERALIZED ADAPTIVE GAIN SHAPE VECTOR QUANTIZATION ACROSS CHANNELS

Sanjeev Mehrotra, Wei-ge Chen, and Kishore Kotteri

Microsoft Research, Microsoft Corporation, One Microsoft Way, Redmond, WA, 98052
{sanjeevm,wchen,kishorek}@microsoft.com

ABSTRACT

Audio coding at low bitrates suffers from artifacts due to spectrum truncation. Typical audio codecs code multi-channel sources using transforms across the channels to remove redundancy such as middle (mid) - side (M/S) coding. At low bitrates, the spectrum of the coded channels is truncated and the spectrum of the channels with lower energy, such as the side channel, is truncated severely, sometimes entirely. This results in a muffled sound due to truncation of all coded channels beyond a certain frequency. It also results in a loss of spatial image even at low frequencies due to severe truncation of the side channel. Previously we have developed a low bitrate coding method to combat the loss of higher frequencies caused by spectrum truncation. In this paper, we present a novel low bitrate audio coding scheme to mitigate the loss of spatial image. Listening tests show that the combination of the two low bitrate coding methods results in a audio codec that can get good quality even at bitrates as low as 32kbps for stereo content with low decoder complexity.

Index Terms— Audio Coding, Vector Quantization

1. INTRODUCTION

With recent interest in direct delivery of music to mobile phones and satellite radios as well as demand for being able to store larger music libraries on portable devices, there is increasing need for good quality audio at lower bitrates. It is well known that existing codecs such as MPEG II-Layer 3 (MP3) and Advanced Audio Coding (AAC) have significant artifacts when coding stereo content below 128kbps.

Typical audio codecs first take the input audio signal and segment it into transform blocks, the size of the transform block being determined by signal characteristics such as whether the signal is transient (using smaller transform block sizes) or stationary (which uses larger transform block sizes). The signal is transformed from time domain to frequency domain using an overlapping transform such as a subband filterbank or modulated discrete cosine transform (MDCT). Then, a channel transform is applied across the audio channels (for example the left and right) to remove redundancy. In the remainder of this paper, we will refer to the original channels as those prior to the channel transform, and the coded channels as those after the channel transform. In typical music sources the left and right channels are strongly correlated and thus have a lot of redundancy. A channel transform such as the mid/side (M/S) transform removes this redundancy. The mid channel can be defined as $x_m[k] = (x_l[k] + x_r[k])/\sqrt{2}$, where $x_m[k]$ is the kth spectral coefficient of the mid channel, and $x_l[k]$ and $x_r[k]$ are those of the left and right channel. The side channel can be defined as $x_s[k] = (x_l[k] - x_r[k])/\sqrt{2}$. After the channel transform, the spectral coefficients of the coded channels are quantized and entropy coded. The quantization step size used for each coefficient of each
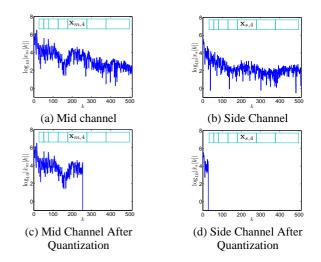

(a) Mid channel   (b) Side Channel

(c) Mid Channel After Quantization   (d) Side Channel After Quantization

Fig. 1. Magnitude of spectrum of mid and side coded channels in log domain. Also shown are vectors which tile the missing portion of the side channel spectrum.

channel is different and is found by using psychoacoustic modeling to find places where larger quantization noise can be tolerated.

At low bitrates, there are fewer bits to encode the signal, and thus large quantization step sizes are used to code many of the coefficients. For example, all channels above a certain frequency are quantized to zero in all the channels. Furthermore, at lower bitrates, some of the coded channels, for example the side channel for stereo sources, are further truncated to improve the coding of the mid channel. After quantization, the spectrum of the coded channels can look like that in figure 1(c)-(d), whereas before quantization it will look like that in figure 1(a)-(b), resulting in a muffled sounding signal with very little spatial image. In all figures which show the spectrum in this paper, we show the log of the magnitude of the spectrum.

In [1], we presented a novel audio codec which combined traditional waveform audio coding with a low bitrate coding scheme to create a codec which performed on average as good as the best audio codec at that bitrate, but performed better on some clips. It also had a far lower decoder complexity than other audio codecs of similar quality. This codec used traditional audio coding methods to code some portions of the spectrum. The remaining portion of the spectrum, which was quantized to zero by the traditional waveform codec, was coded using a low bitrate coding using an adaptive gain-shape vector quantizer with the adaptive codebook being formed by taking modified and unmodified portions of the spectrum which have already been coded. Although this coding works very well for coding the higher portion of the spectrum when both coded channels are

truncated beyond a large frequency, it does not perform well at lower bitrates when, for example, the side channel spectrum is being truncated very severely, such as below 8kHz. In such cases, an alternate coding method is needed to reconstruct the spatial image.

Existing solutions to solve this problem are along the lines of perceptual stereo coding. In intensity stereo coding [2], for example, a stereo source is only coded as a mono source. The effect of stereo is simply perceptually recreated by applying different scale factors to shape the energy in various bands to get the energy envelope of the two channels correct. This has been generalized in binaural cue coding (BCC) in which audio cues such as interaural level difference (ILD), interaural time difference (ITD), and interaural coherence (IC) are sent as parameters from which a downmixed source (such as a single channel source) is reconstructed to stereo [3, 4]. The reconstruction is done so as to preserve these parameters as in the original. Techniques such as this are used in the parametric stereo coding in the High-Efficiency AAC (HE-AAC) audio codec.

In this paper, we present an alternate coding method which performs far better than traditional audio codecs, but performs as good as the best audio codecs such as HE-AAC. However, decoder complexity is found to be lower. The basic idea is similar to that used in [1] in that we use an adaptive gain-shape vector quantizer where the codebook is adaptively formed from portions of the spectrum that have already been coded.

2. GENERALIZED GAIN SHAPE VECTOR QUANTIZATION

Gain shape vector quantization (GSVQ) is a modified version of vector quantization (VQ) that can be used when it is important to preserve the energy of the vector. For example, if the vector $\mathbf{x}$ is to be coded using VQ, then we can write $\mathbf{x} = \|\mathbf{x}\| \frac{\mathbf{x}}{\|\mathbf{x}\|}$, where the first term $s = \|\mathbf{x}\|$ is the gain and the second term $\mathbf{y} = \mathbf{x}/\|\mathbf{x}\|$ is a unit norm vector. Now, the gain s can be quantized separately using some number of bits so that the gain is more accurately represented. The shape $\mathbf{y}$ is a unit norm vector which can be coded using a codebook with unit norm vectors. Although GSVQ allows for better preservation of the energy (or gain) of a vector, it still is not able to control the cross-correlation across two vectors which is important when coding two vectors consisting of spectral coefficients, one from the left channel, and another from the right channel. For example, one of the cues in binaural cue coding is related to cross-correlation [3, 4].

Here we generalize the concept of gain-shape vector quantization across any arbitrary number of vectors. Consider that N vectors, $\mathbf{x}_n, n = 0, \dots, N-1$, have to be coded and we wish to be able to accurately represent the energy and cross- correlation of the vectors. Then we can form a matrix $\mathbf{X} = [\mathbf{x}_0 \ \mathbf{x}_1 \ \dots \ \mathbf{x}_{N-1}]$, which has a correlation matrix given by $\mathbf{R}_{XX} = \mathbf{X}^* \mathbf{X}$, where $\mathbf{X}^*$ is the Hermitian transpose of the matrix. Then we can code a sufficient number of parameters to represent the symmetric matrix $\mathbf{R}_{XX}$. Now since $\mathbf{R}_{XX}$ is symmetric, we can write $\mathbf{R}_{XX} = \mathbf{U}_X \mathbf{D}_X \mathbf{U}_X^*$, where $\mathbf{D}_X$ is a diagonal matrix, and $\mathbf{U}_X$ is an orthonormal matrix. Then, if

$$\mathbf{Y} = \mathbf{D}_X^{-1/2} \mathbf{U}_X^* \mathbf{X}, \tag{1}$$

then the matrix $\mathbf{Y} = [\mathbf{y}_0 \ \mathbf{y}_1 \ \dots \ \mathbf{y}_{N-1}]$ consists of vectors $\mathbf{y}_n$ which are of unit norm and uncorrelated. That is $\mathbf{R}_{YY} = \mathbf{Y}^* \mathbf{Y} = \mathbf{I}$, the identity matrix. Instead of coding each vector $\mathbf{y}_n$ independently using a codebook, we code $\mathbf{y}_n$ jointly using a codebook where each entry l in the codebook is a matrix which is a set of N vectors $\mathbf{c}_{n,l}$, where $\mathbf{C}_l = [\mathbf{c}_{0,l} \ \mathbf{c}_{1,l} \ \dots \ \mathbf{c}_{N-1,l}]$ satisfies

$$\mathbf{C}_l^* \mathbf{C}_l = \mathbf{I}. \tag{2}$$

If K is the number of entries in the codebook used to jointly code the set of N vectors, the best entry in the codebook is found to be $l_{opt} \in \{0, 1, \dots, K-1\}$ using search. The decoder simply reconstructs as

$$\hat{\mathbf{X}} = \mathbf{U}_X \mathbf{D}_X^{1/2} \mathbf{C}_{l_{opt}}, \tag{3}$$

which results in $\mathbf{R}_{\hat{X}\hat{X}} \approx \mathbf{R}_{XX}$.

3. AUDIO CODING USING GSVQ

In this section we present how generalized adaptive gain-shape vector quantization is used in an audio codec. The encoder first encodes the audio signal using a traditional waveform audio codec. For transform blocks where there is strong correlation between the left and the right channels, the traditional audio codec uses mid/side channel coding. For transform blocks where there is not strong correlation, the traditional codec will use left/right stereo coding. In those blocks where left/right stereo coding is used, we simply use the ideas presented in [1] to code the spectrum.

In transform blocks which use mid/side channel coding, the original spectrum shown in figure 1(a)-(b) can look like that in figure 1(c)-(d) after quantization using a traditional audio codec at low bitrates. The mid and side channels are both truncated, with the side channel severely truncated. To fill in the missing portion of the spectrum in the mid channel, we can use ideas from [1]. Then the mid channel will look similar to that in figure 1(a). If we inverted the channel transform and reconstructed the left and right channels at this point, we would not get a muffled sounding signal, since the spectrum is full bandwidth. However, due to the side channel spectrum truncation, we would still get a signal with the spatial image lost at most of the frequencies. However, we cannot use the ideas from [1] to fill in the missing portion of the spectrum in the side channel as there is only a small portion of the spectrum that is available to form a codebook. In order to form an effective codebook to code the side channel, we could use portions of the spectrum from the mid channel to augment it. Although this can work, it will only accurately represent the energy of the side channel. Other statistics such as the cross correlation between the two channels will not be preserved. Cross-correlation is also an important statistic to preserve in order to create an accurate stereo image, at least in the lower frequency range. For this we use ideas from section 2.

Consider dividing up the missing portion of the side channel spectrum into M non-uniform size vectors as shown in figure 1(b). Similarly divide this portion of the spectrum in the mid channel as shown in figure 1(a). Let these vectors be $\mathbf{x}_{m,i}$ and $\mathbf{x}_{s,i}$ for the ith vector for the mid and side channels respectively, where $i = 0, 1, \dots, M-1$. For example, the $i = 4$ vector is shown in figure 1. The ith vector simply consists of the original spectral coefficients and can be written as

$$\mathbf{x}_{m,i} = [x_m[S[i]] \ x_m[S[i]+1] \ \dots \ x_m[S[i]+L[i]-1]]^T, \tag{4}$$

where $S[i]$ is the starting position of the ith vector, $L[i]$ is the length of the vector, and $x_m[k]$ is the kth spectral coefficient of the mid channel. We can similarly define the vector $\mathbf{x}_{s,i}$, with $S[i]$ and $L[i]$ being the same for both channels for all i. Define the matrix $\mathbf{X}_i$ as

$$\mathbf{X}_i = [\mathbf{x}_{m,i} \ \mathbf{x}_{s,i}], \tag{5}$$

which is a $L[i]$x2 dimensional matrix. Now we can use ideas presented in section 2 with $N = 2$ to code this matrix. That is, we can code $R_{XX} = \mathbf{X}_i^* \mathbf{X}_i$ for this matrix, then normalize the matrix using equation 1 to obtain $\mathbf{Y}_i$ and then code $\mathbf{Y}_i$ using a codebook with entries which satisfy equation 2.
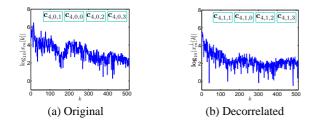
(a) Original   (b) Decorrelated

Fig. 2. Portions of the mid channel spectrum used to form codebook for the $i = 4$ vector. $\mathbf{c}_{4,0,0}$ is in same location as $\mathbf{x}_{l,4}$ and $\mathbf{x}_{r,4}$.

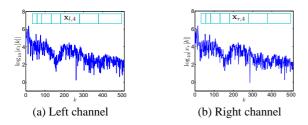(a) Left channel   (b) Right channel

Fig. 3. Magnitude of spectrum of original channels and the vectors used to code the portion of the spectrum missing in the side channel.

Let the matrix $\mathbf{C}_{i,l}$ be the lth entry in the codebook used to code the ith set of vectors. We can write this entry as $\mathbf{C}_{i,l} = [\mathbf{c}_{i,0,l} \; \mathbf{c}_{i,1,l}]$, which is a $L[i]$x2 dimensional matrix satisfying (2). To obtain the vectors $\mathbf{c}_{i,0,l}$ we use ideas similar to that used to obtain the codebook in [1]. That is, we take a portion of the spectrum already coded, as shown in figure 2(a) for $i = 4$, and normalize it to be of the desired norm. The spacing between the codevectors is chosen to maximize the amount of spectrum that is covered and is determined by the amount of spectrum that has been coded and the number of entries in the codebook. We can write $\mathbf{c}_{i,0,l} = \alpha_0 [x_m[S_l[i]] \; \ldots \; x_m[S_l[i] + L[i] - 1]]$, where α_0 is used to normalize the spectrum to have desired norm to satisfy (2) and $S_l[i]$ is the starting position of the lth entry in the codebook. We also make sure that the first vector in the codebook is colocated with the vector being coded, that is $S_0[i] = S[i]$ as shown in figure 2(a) since this is a common selection for the codevector to use. To form the second column of the matrix $\mathbf{C}_{i,l}$, we use any of the common decorrelating techniques to obtain a spectrum which is approximately statistically decorrelated from the original spectrum. Such techniques involve the use of all pass IIR filters and are commonly used in creating artificial reverberation [5, 6]. Then, we can write $\mathbf{c}_{i,1,l} = \alpha_1 [x_m^{\perp}[S_l[i]] \; \ldots \; x_m^{\perp}[S_l[i] + L[i] - 1]]$, where $x_m^{\perp}[k]$ is the kth coefficient in the decorrelated spectrum, and α_1 is a scale factor used to normalize the spectrum. Thus equation 2 approximately holds. An example is shown for $i = 4$ in figure 2(b). If we truly wanted to satisfy (2) we could actually obtain the matrix and then apply a decorrelating transform on the matrix as is done for $\mathbf{X}$ in (1). However, this results in too much complexity as it has to be done for every codevector, hence it is not used. The best matching codebook entry $l_{opt}[i]$ can be found for each i. For the search, we use MSE but strongly favor the colocated codevector, $l = 0$.

Although the previously described scheme works, it does not preserve the energy and cross-correlation of the original left and right channels, instead it preserves the energy of the mid and side channel and the cross-correlation between them. Only a slight modification is needed to obtain this. In order to correct this, we first apply an inverse channel transform to go from mid-side coded chan-

nel domain to left-right original channel domain. Once the inverse transform is applied the spectrum does not have missing spectral coefficients, however, it does have an incorrect spectrum past the truncation point of the side-channel. We code the portion of the spectrum with missing spatial image using the generalized adaptive gain-shape vector quantization presented in section 2. First, we divide this portion of the spectrum of the left and right channels into vectors as shown in figure 3. Define $\mathbf{x}_{l,i}$ and $\mathbf{x}_{r,i}$ to the left and right ith vectors similar to equation 4. Define the matrix $\mathbf{X}_i = [\mathbf{x}_{l,i} \; \mathbf{x}_{r,i}]$ to be the matrix to code similar to equation 5. We code $\mathbf{R}_{XX}$ and send it in the bitstream for each i. The normalized matrix $\mathbf{Y}_i$ is coded using the same codebook formed from the mid channel as described.

One simplification is to note that not all components of the matrix $\mathbf{R}_{XX}$ need to be sent and thus we can save bits used to code this matrix. The matrix is symmetric so only $N(N + 1)/2$ components are unique. If the spectrum x is complex, then N of these parameters are real and $N(N + 1)/2 - N$ are complex. The complex parameters can be coded using polar coordinates as magnitude and phase, with phase being more coarsely quantized. Also since we already have some information on the spectrum, we don't need to send all of these parameters. For example, in the two channel case, the mid channel is given by $\mathbf{x}_{m,i} = (\mathbf{x}_{l,i} + \mathbf{x}_{r,i})/\sqrt{2}$ and thus $\mathbf{x}_{m,i}^* \mathbf{x}_{m,i} = (\mathbf{x}_{l,i}^* \mathbf{x}_{l,i} + \mathbf{x}_{r,i}^* \mathbf{x}_{r,i})/2 + \text{Real}(\mathbf{x}_{l,i}^* \mathbf{x}_{r,i})$. Since the mid channel spectrum is already coded, if we know $\mathbf{x}_{l,i}^* \mathbf{x}_{l,i}$ and $\mathbf{x}_{l,i}^* \mathbf{x}_{r,i}$, we can compute $\mathbf{x}_{r,i}^* \mathbf{x}_{r,i}$ without actually sending any parameters.

In general, suppose that the codec uses a channel transform $\mathbf{B}$ to code the signal as $\mathbf{Z} = \mathbf{BX}$, and then suppose that some of these coded channels are being quantized to zero. Let $\mathbf{Z}_0$ be the portion that is not quantized to zero, and let $\mathbf{Z}_1$ be the portion that is quantized to zero. That is, $\mathbf{R}_{ZZ} = \mathbf{Z}^* \mathbf{Z} = \mathbf{B}^* \mathbf{X}^* \mathbf{XB}$, where

$$\mathbf{R}_{ZZ} = \left[\begin{array}{cc} \mathbf{Z}_0^* \mathbf{Z}_0 & \mathbf{Z}_0^* \mathbf{Z}_1 \\ \mathbf{Z}_1^* \mathbf{Z}_0 & \mathbf{Z}_1^* \mathbf{Z}_1 \end{array} \right]. \tag{6}$$

Now it is sufficient to send $\mathbf{Z}_0^* \mathbf{Z}_1$ and $\mathbf{Z}_1^* \mathbf{Z}_1$ from which we know $\mathbf{R}_{ZZ}$. Now since $\mathbf{X} = \mathbf{B}^{-1} \mathbf{Z}$, we can compute

$$\mathbf{R}_{XX} = (\mathbf{B}^{-1})^* \mathbf{R}_{ZZ} (\mathbf{B}^{-1}). \tag{7}$$

For the two channel case, this corresponds to having to send one complex parameter, $\mathbf{Z}_0^* \mathbf{Z}_1$, which is the cross-correlation between the mid and side channels and one real parameter $\mathbf{Z}_1^* \mathbf{Z}_1$, which is the energy in the side channel. Other parameterizations which give similar information are also possible.

Another simplification we can make is in the common case when $l_{opt} = 0$ and if $\mathbf{Z}_0^* \mathbf{Z}_0 = s^2 \mathbf{I}$ (that is if the coded portion of the coded channels has a spherical cross-correlation matrix). If only one channel is being coded as in the two channel case, then the second condition will always be met since $s^2 = \mathbf{x}_{m,i}^* \mathbf{x}_{m,i}$. In this case, suppose we send $\mathbf{Z}_0^* \mathbf{Z}_1/s^2$ and $\mathbf{Z}_1^* \mathbf{Z}_1/s^2$ from which we can obtain $\mathbf{R}_{ZZ}/s^2$. The decoder avoids the need to calculate $\mathbf{Z}_0^* \mathbf{Z}_0$ since $\mathbf{Z}_0^* \mathbf{Z}_0/s^2 = 1$. Then, we can obtain $\mathbf{R}_{XX}/s^2$ using (7). Now, instead of satisfying (2), suppose the codevectors satisfy $\mathbf{C}_l^* \mathbf{C}_l = s^2$. Then the reconstruction $\hat{\mathbf{X}} = \mathbf{U}_X (\mathbf{D}_X^{1/2}/s) \mathbf{C}_{l_{opt}}$ will have the property that $\mathbf{R}_{\hat{X}\hat{X}} = \mathbf{R}_{XX}$. Note that the colocated entry in the codebook corresponding to $l_{opt} = 0$ already satisfies $\mathbf{C}_l^* \mathbf{C}_l = s^2$ and so no normalization is needed to form the first vector, $\mathbf{c}_{i,0,0}$, in the codebook entry. Thus only the second portion of the codebook entry, $\mathbf{c}_{i,1,0}$, needs to be normalized.

The transform domain used to code the signal using the above mentioned algorithm does not need to be the same as that used to code the base codec as shown in figure 4. In particular we allow two modifications. One is that the transform block size can be different

Table 1. First set of values shown are number of listeners with the given preference for each song, average is given as %. Second set of results are decoder complexity $R = (D/P)C$ (in MHz) for our full power, our low power, and HE-AAC averaged over 10 runs.

Song	Ours	Identical	HE-AAC	Ours	Ours low	HE-AAC
Blank baby	1	4	2	19.8	12.1	42.6
Sun Is Shining (Island Mix)	2	2	3	21.8	12.1	43.6
Eclipse	2	4	1	20.9	11.2	43.5
Boulevard Of Broken Dreams	2	1	4	20.7	12.2	45.5
Bitter Sweet Symphony	2	5	0	20.6	11.2	45.0
Average	25.7%	45.6%	28.6%	20.8MHz	11.8MHz	44.0MHz

than that used in the base. Another is that the actual transform can be different. For example, the base can use the modulated discrete cosine transform (MDCT) referred to as "Transform A" in the figure, whereas the VQ coding across the channels can use the modulated complex lapped transform (MCLT) referred to as "Transform B" in the figure. The use of a complex transform can be useful in preserving the phase of the cross-correlation between the left and the right channels. Different transform sizes can be useful depending on differing requirements of temporal resolution needed by the various coding methods.
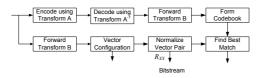

Fig. 4. Encoder block diagram.

However, if the two transforms are similar and use same the same block size, then we can encode using "Transform B" to obtain the parameters, but instead use "Transform A" in the decoder. This allows a low complexity decoder to avoid the need for an additional forward and inverse transform at the expense of a slightly degraded sound quality. A low complexity decoder can also use a lower complexity decorrelating method when forming the codebook such as a less complicated IIR filter. This makes the decoder complexity significantly lower for a "low power" decoder as may operate on a portable music player or cell phone whereas a PC decoder can do a full power decoding of exactly the same piece of content.

4. EXPERIMENTAL RESULTS

We used the ideas from section 3 to create an audio codec for stereo content. The base audio codec used a codec similar to that presented in [1], which is a WMA Professional audio codec used to code the base with enhanced low bitrate coding using adaptive gain shape vector quantization to code some portions of the spectrum in the mid channel. The side channel was truncated entirely so that there was no energy coded in it. The spectrum was divided into 20 vectors and coded using the ideas presented. The base WMA Professional audio codec operated at 26kbps, with 3kbps used to code portions of the spectrum using the low bitrate coding presented in [1], and another 3kbps used to code the missing spatial image using the generalized adaptive GSVQ presented in this paper.

A small informal listening test was conducted on five pieces of music with 16kHz bandwidth comparing our codec to HE-AAC at 32kbps. When compared to codecs such as MP3 or AAC, all of

the listeners preferred our codec, so those results are not presented. As we can see from table 1, our codec performs as good as HE-AAC at these bitrates, with over 70% of the results saying that our codec is as good or better to HE-AAC. We also measure decoder speeds on a CPU with a 2.4GHz processor and found our decoder to take only 47% as much decoding time as the HE-AAC decoder and the low power decoder taking only 27% of the decoding time as seen from table 1. The measure of decoder speed shown in the table is $R = (D/P)C$, where D is time to decode the file, P is the playback time, and C is the CPU speed. We see that our decoder, even the full power one, is less complex than HE-AAC. A similar complexity ratio between the two is observed on an ARM processor.

5. CONCLUSION

We have presented an algorithm that can be useful for low bitrate audio coding. The algorithm consists of generalizing gain shape vector quantization to jointly coding multiple vectors by extracting the correlation matrix for the set of vectors and normalizing the vectors to be of unit norm and uncorrelated. The normalized vectors are jointly coded using an adaptive codebook with similar statistics. The resulting audio codec is as good as the state of the art audio codecs at 32kbps, but has lower decoder complexity making it ideal for use on devices such as mobile phones and portable music players.

6. REFERENCES

[1] S. Mehrotra, W. Chen, K. Koishida, and N. Thumpudi, "Hybrid low bitrate audio coding using adaptive gain shape vector quantization," in *Proc. Workshop on Multimedia Signal Processing*. Cairns, Australia: IEEE, Oct. 2008.

[2] J. Herre, K. Brandenburg, and D. Lederer, "Intensity stereo coding," in *AES Convention 96*. Audio Engineering Society, Feb. 1994, paper 3799.

[3] F. Baumgarte and C. Faller, "Binaural cue coding–Part I: Psychoacoustic fundamentals and design principles," *IEEE Trans. Speech and Audio Processing*, vol. 11, no. 6, pp. 509–519, Nov. 2003.

[4] C. Faller and F. Baumgarte, "Binaural cue coding–Part II:Schemes and applications," *IEEE Trans. Speech and Audio Processing*, vol. 11, no. 6, pp. 520–531, Nov. 2003.

[5] J. O. Smith III. Physical audio signal processing for virtual musical instruments and audio effects. Stanford, CA. [Online]. Available: http://ccrma.stanford.edu/~jos/pasp

[6] M. R. Schroeder and B. F. Logan, "Colorless artificial reverberation," *IRE Trans. on Audio*, vol. 9, no. 6, pp. 209–214, Nov. 1961.