Lossless Transform Coding of Audio Signals

Tilman Liebchen, Marcus Purat, and Peter Noll
Institute for Telecommunications, Technical University Berlin
Einsteinufer 25, D-10587 Berlin, Germany
email: liebchen@ee.tu-berlin.de, purat@ee.tu-berlin.de, noll@ee.tu-berlin.de

Abstract

Recent papers have proposed linear prediction as a useful method for lossless audio coding. Transform coding, however, hasn't been investigated so far, although it seems to be more adapted to the harmonic structure of most audio signals. In this paper we present first results on lossless transform coding of CD-quality audio data. One main aspect lies on a suitable quantization method to obtain perfect reconstruction. Using a codebook with different entropy codes for the transform coefficients we achieve bitrates, slightly better then those obtained by the lossless linear prediction schemes mentioned above.

1 Introduction

Lossless audio coding is a topic of high interest for both professional and customer applications. Modern lossy coding standards (e.g. ISO MPEG I+II) can achieve large compression ratios with high subjective quality, however, multiple coding can reveal the originally masked distortion. Anyway, reproducing critical music items shows that even the best systems cannot be considered as truly transparent.

Applying entropy coding methods as Huffman, LZW, or arithmetic coding directly to the audio signal is not very efficient due to the long-term correlations in a 16-bit quantized signal. Therefore, common data compression tools fail for digitalized music. A preprocessing stage, which eliminates the statistical dependencies within the signal, leads to an almost uncorrelated source which is easier to code. While there have been several papers which proposed linear prediction for this preprocessing stage [1][2][3], the other common tool for decorrelating signals, signal transforms, hasn't been investigated yet.

In this paper we discuss the application of discrete orthonormal transforms for lossless audio coding. At first, a general coding scheme is presented to explain how the transform can be employed for lossless coding. Then some essential properties concerning the quantization of the transform coefficients are derived. Finally, we describe a practical coding algorithm and give some results.

2 Lossless Transform Coding

In general, the spectrum resulting from a signal transform will be real valued, even for an integer value signal. For an efficient transmission the coefficients have to be quantized, which causes inevitable errors in the output signal. Thus, lossless transform coding has to be considered as a combination of conventional lossy transform coding and additional transmission of the coding error [4]. **Fig. 1** shows the coding scheme. The signal code c is derived form the input signal x using the lossy compression algorithm. To achieve lossless compression, the difference e between the input signal x and the reconstructed signal y is generated in the encoder by local decoding, and both the lossy compressed code c and the error e are transmitted. In the decoder, the error signal is added to the decoded approximation, resulting in an output signal which is a perfectly reconstructed version of the input signal.

It is obvious that the error signal strongly depends on the lossy coding algorithm. If the compression ratio is very high, the error signal will be large and correlated, and the coding problem passes from the original signal to the error signal. If on the other hand compression is very small, the error will be zero, but not all of the signals redundancy is removed. Thus, as a main goal in lossless transform coding we have to find the best compromise between a high compression ratio in the lossy branch and an easily codeable signal in the correction branch.

We now consider a system as it is shown in **Fig. 2**. From the quantized input signal $x(n) \in \mathbf{Z}$ a set of transform coefficients $t(k)$ is calculated using an arbitrary orthonormal transform $\mathbf{A}$ with block length N. The coefficients are scaled by α and quantized with an unitary quantization step size $\Delta = 1$, leading to the integer value spectrum $c(k)$, which - after entropy coding - is finally transmitted. Using a suitable transform, many of the coefficients are very small or even zero, and, moreover, they constitute an uncorrelated source. Hence, the integer value spectrum can be easily entropy coded without taking into account joint probabilities.

As a result of the quantization, which is equivalent to integer truncation, decoding of $c(k)$ does not guarantee perfect reconstruction, although the integer spectrum itself is coded losslessly. Therefore, we have to check for possible errors by decoding $c(k)$ in the encoder and to generate an error correction signal, if necessary.

After descaling with α^{-1} and applying the inverse transform $\mathbf{A}^{-1} = \mathbf{A}^{t}$, where $\mathbf{A}^{t}$ stands for the transposed matrix, we obtain the real valued signal $y'(n)$ which is, due to the quantization in the transform domain, not the original $x(n)$. However, since we consider integer signals, there is no need to reconstruct the input signal exactly by a real valued signal. If $|y'(n) - x(n)| < 0.5$, integer truncation of $y'(n)$ leads to a reconstructed integer signal $y(n)$ which is identical with $x(n)$. Otherwise, the input signal is not perfectly reconstructed, i.e. we have an error $e(n) \neq 0$. Of course, $e(n) = x(n) - y(n)$ is an integer signal, because $x(n)$ and $y(n)$ are integer as well. The correction signal $e(n)$ has to be transmitted in addition to the integer valued coefficients $c(k)$.

On the decoder side, $y(n)$ is calculated identically by descaling, inverse transformation and integer trucation. After adding $e(n)$, we get $y(n) + e(n) = y(n) + (x(n) - y(n)) = x(n)$, which is the desired original input signal.

3 Quantization effects

3.1 Theoretical bounds

Obviously, the scalefactor α has great influence not only on the coefficients $c(k)$, but also on the error signal $e(n)$, and, of course, there is a close connection between the error bit rate R_e and the bit rate R_c for the coefficients. Let R_0 denote the coefficient bit rate for the case $\alpha = 1$, then R_c rises with $\mathrm{ld}(\alpha)$:

$$R_c = R_0 + \mathrm{ld}(a).$$

To derive a bound for the error bit rate, the process of scaling, integer truncation and descaling is considered as a linear quantization with step size $\Delta = 1 / \alpha$, leading to an equally distributed quantization error $q(k) = t(k) - t'(k)$ in the transform domain. The first order entropy of the error $q(k)$ is certainly $H_1(q) = \mathrm{ld}(\Delta) = -\mathrm{ld}(\alpha)$.

The unquantized error $\varepsilon(n) = x(n) - y'(n)$ in the time domain has the same variance $\sigma_\varepsilon^2 = \sigma_q^2$ due to the energy conservation property of any orthonormal transform. Since for a large transform order N $\varepsilon(n)$ can be seen as a superposition of many equally distributed random variables, the assumption of a gaussian probability density function (PDF) for $\varepsilon(n)$ is justified. For a large step size Δ, i.e. for $\alpha << 1$, the quantized error $e(n)$ has also a (discrete) gaussian PDF with same variance, so its entropy $H_1(e)$ is about 0.26 bit higher than $H_1(q)$. Hence, for $\alpha << 1$, R_e is bounded by

$$R_e \geq 0.26 - \mathrm{ld}(a),$$

if we assume an uncorrelated error signal. The overall bit rate is made up by $R = R_c + R_e$, and so the following condition can be derived:

$$R \geq R_0 + 0.26.$$

Thus, the overall bitrate is at least 0.26 bits higher than the coefficient bit rate for $\alpha = 1$. These results can be practically verified, as it is shown in **Fig. 3**. Generally, the bit rate R_e will be higher than $H_1(e)$, but we assume that with a suitable entropy code this bound can be approximated, so that the sum $R_c + H_1(e)$ is a pretty good approximation for the achievable overall bit rate R. It can be seen that for $\alpha < 1$ we obtain a nearly constant overall bit rate, which is slightly higher than R_0, whereas for $\alpha > 1$, the overall bit rate increases with $\mathrm{ld}(\alpha)$.

3.2 Error probabilities

We already mentioned that the unquantized error $\varepsilon(n)$ in the time domain has a gaussian PDF

$$f(\varepsilon) = \frac{1}{\sqrt{2\pi\sigma_\varepsilon^2}} \cdot \mathrm{e}^{-\frac{\varepsilon^2}{2\sigma_\varepsilon^2}}$$

with the variance of the equally distributed quantization error

$$\sigma_\varepsilon^2 = \sigma_q^2 = \frac{\Delta^2}{12} = \frac{1}{12\alpha^2}.$$

As we consider integer input signals, only $|\varepsilon(n)| \geq 0.5$ leads to an error $e(n) \neq 0$. Therefore, the probability for the occurrence of an error in the reconstructed signal $y(n)$ is

$$p_e = P\{|e| \geq 0.5\} = 2 \cdot \int_{0.5}^{\infty} f(e)\,\mathrm{d}\,e = \mathrm{erfc}\left(a\,\sqrt{3/2}\right).$$

The error probability decreases with a growing scalefactor, but according to the above equation it won't be zero even for a very large α. This is due to the fact that the assumption of a gaussian PDF for the error signal is theoretically correct only for $N \to \infty$. Actually, if the scalefactor is very large, the error probability is zero. As an example, it can be shown that for an N-point DCT and $\alpha > \sqrt{N}$ definitely no error will occur. But, regarding **Fig. 3**, it is obvious that such a large value of α is not relevant for practical applications, because it leads to a very high overall bit rate. Therefore, the above equation will be used to derive the following results.

The necessary bit rate for the error signal is bounded by its entropy

$$H_1(e) = -\sum_{k=-\infty}^{\infty} P\{e = k\} \cdot \mathrm{ld}\left(P\{e = k\}\right).$$

For $\alpha = 1$, the error probability is

$$p_e = \mathrm{erfc}\left(\sqrt{3/2}\right) \approx 0.08,$$

which, on average, leads to an error in one out of twelve reconstructed samples. In this case, almost all errors are either 1 or –1. Let us denote $p_k = P\{e = k)$, then we get the probabilities $p_0 = 1 - p_e$ and $p_1 = p_{-1} = p_e / 2$. Thus, the error entropy is

$$H_1(e) \approx -\sum_{k=-1}^{1} p_k \cdot \mathrm{ld}\left(p_k\right) \approx 0.5,$$

i.e. we have to spend at least 0.5 bits per input sample for the transmission of the correction signal $e(n)$. Thus, for $\alpha = 1$, an overall bit rate $R \approx R_0 + 0.5$ can be achieved. This is about 0.24 bit above the theoretical bound for $\alpha \ll 1$. Anyway, choosing $\alpha = 1$ has the advantage, that the error is very easily codeable, because its value is practically restricted to ±1. The entropy can approximated very closely by practical algorithms, which turned out to be difficult for smaller values of α. Furthermore, no scaling and descaling has to be done using an orthonormal transform.

4 Coding Algorithm

A block diagram of the implemented lossless transform coding system is shown in **Fig. 4**. The coder uses an orthonormal DCT with either fixed or variable block length. For the reasons mentioned above, no additional scaling is done. Each integer value spectrum $c(k)$ is divided into groups of 32 adjacent coefficients. This partition proved to be most efficient. Since these groups have an almost laplacian PDF, the codebook consists of several Rice codes. Each group is coded using the most convenient Rice code, i.e. the code which leads to a minimum number of bits. A Rice code is in fact a Huffman code for a laplacian PDF, which is determined by its standard deviation σ. Since Rice codes only exist for discrete values of σ, only the indices of the chosen codes have to be transmitted.

The inverse transform allows for the location of errors. With $p_e \approx 0.08$ the average distance between errors in the reconstructed signal is about twelve samples, i.e. only one out of twelve samples in $e(n)$ is different from zero. Therefore, it is easier to code the distance than to code the absolute position within the block. In fact only the deviation from the expected distance is actually coded, using also different Rice codes. Only a sign bit has to be transmitted in addition to the distance, because the probability of errors $|e(n)| > 1$ is almost zero, as already mentioned before. Exceptions are very rare; they are handled by an escape code.

In fixed block length mode, each block of N input samples is transformed using a DCT with the same length. In adaptive block length mode, the input signal $x(n)$ is divided into blocks of 2048 samples. Each combination of 2048-, 1024- and 512-point transforms is calculated, and for this block the most suitable combination is finally selected.

Experiments with different block transforms, e.g. the DFT, DST and the Discrete Hartley Transform showed the superiority of the DCT. Lapped Transforms, e.g. LOT and MLT [5], have been tested as well. The marginally better results are in the order of up to 0.1 bits per sample and are obtained on the expense of a higher complexity and the loss of editabilty of seperate signal segments. For this reason, we have chosen the DCT for our coder.

The decoder applies the inverse transform after decoding of the coefficients. By adding the decoded correction signal, the original signal is perfectly reconstructed. The computational complexity of the decoder is about half of that of the encoder, because only one transformation has to be performed.

5 Coding results

Results, obtained for several SQAM [6] files, are shown in **Fig. 5**. Each file corresponds to about 20 or 30 seconds of monophonic music, with 16-bit linear quantization at a sampling rate of 44.1 kHz.

In general, a higher block length leads to better coding results due to a better decorrelation of the source signal, except for signals with very fast varying statistics, pitched signals like speech or very transient signals like castanets. As a drawback, editability becomes more difficult for higher block lengths. Although using an adaptive block length does not improve the results so much compared to the best fixed block length for each individual signal, this feature is helpful, since the choice of an appropriate fixed block length for all signals turned out to be difficult.

In **Table 1** the results for lossless transform coding (LTC) with and without adaptive block length are compared to those obtained by linear predictive coding, namely the SHORTEN program from T. Robinson [7], used with default parameters.

	SHORTEN	LTC (best fixed block length)	LTC (adaptive block length)
POP (Abba)	7.38	7.20	7.12
CLASSIC (Haydn)	5.95	5.90	5.87
PIANO (Schubert)	4.79	4.96	4.93
SOPRAN	6.22	5.90	5.88
MALE SPEECH	6.33	6.32	6.26
CASTANETS	7.51	7.04	6.90

Table 1: Coding results for different SQAM files (bits per sample).

For most signals, except the piano sample, the results for transform coding are slightly better, but the average improvement over linear prediction is rather small. However, transform coding seems to perform better than linear prediction and should be preferred as method for lossless audio coding.

Precompiled versions of the lossless transform coder are available for Solaris 2.4 and DOS at the URL http://www-ft.ee.tu-berlin.de/~liebchen.

6 Conclusions

In this contribution we have presented a lossless audio coding algorithm, which is based on orthonormal signal transforms. Previous algorithms for lossless audio coding only used linear prediction. We have shown how perfect reconstruction of the 16-bit input signal can be achieved with integer transform coefficients and additional transmission of the error correction information. For most signals, the bit rates, obtained for entropy coded coefficients and error correction, are below the results obtained by linear prediction.

References

[1] Cellier, C.; Chênes, P.; Rossi, M.: *Lossless Audio Data Compression for Real Time Applications*. Presented at 95th AES Convention, New York, 1993

[2] Robinson, T.: *Shorten: Simple lossless and near-lossless waveform compression*. Technical report CUED/F-INFENG/TR.156, Cambridge University Engineering Department, December 1994

[3] Bruekers, Oomen, v.d. Vleuten, v.d. Kerkhof: *Lossless Coding for DVD Audio*. Presented at 101st AES Convention, Los Angeles, 1996

[4] Craven, P.; Gerzon, M.: *Lossless Coding for Audio Discs*. AES journal, September 1996.

[5] Malvar, H.: *Signal Processing with Lapped Transforms*. Artech House, 1992.

[6] *Sound Quality Assessment Material Recordings for subjective Tests*. Technical Centre of the European Broadcasting Union, 1988

[7] Robinson, T.: *Shorten compressor for waveform files, version 2.1*.

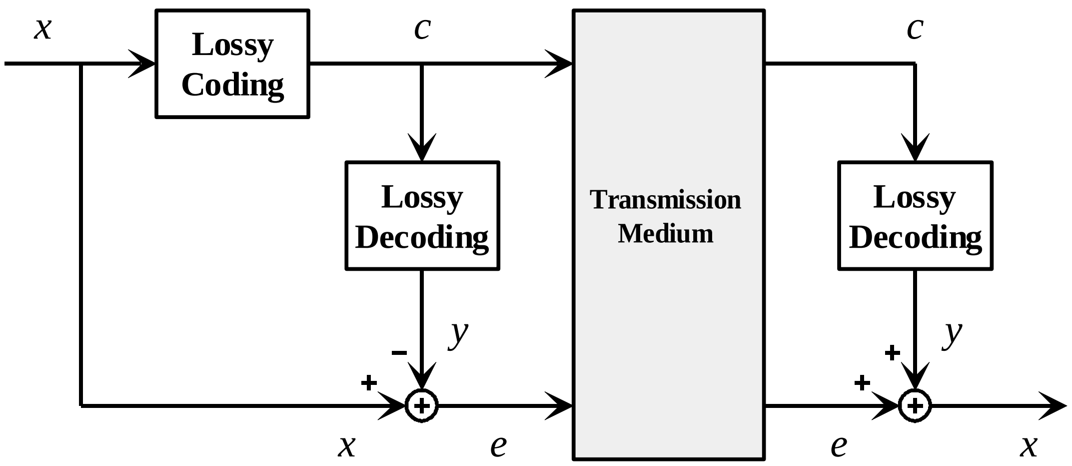

Fig. 1: Lossless coding as a combination of lossy coding and additional error transmission.

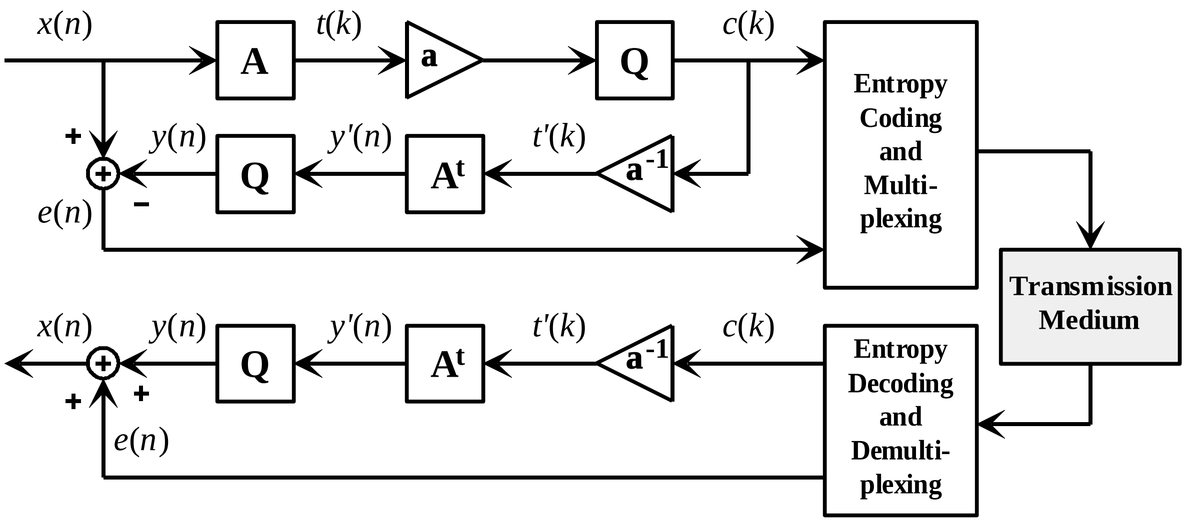

Fig. 2: Block diagram of a lossless transform coding system.

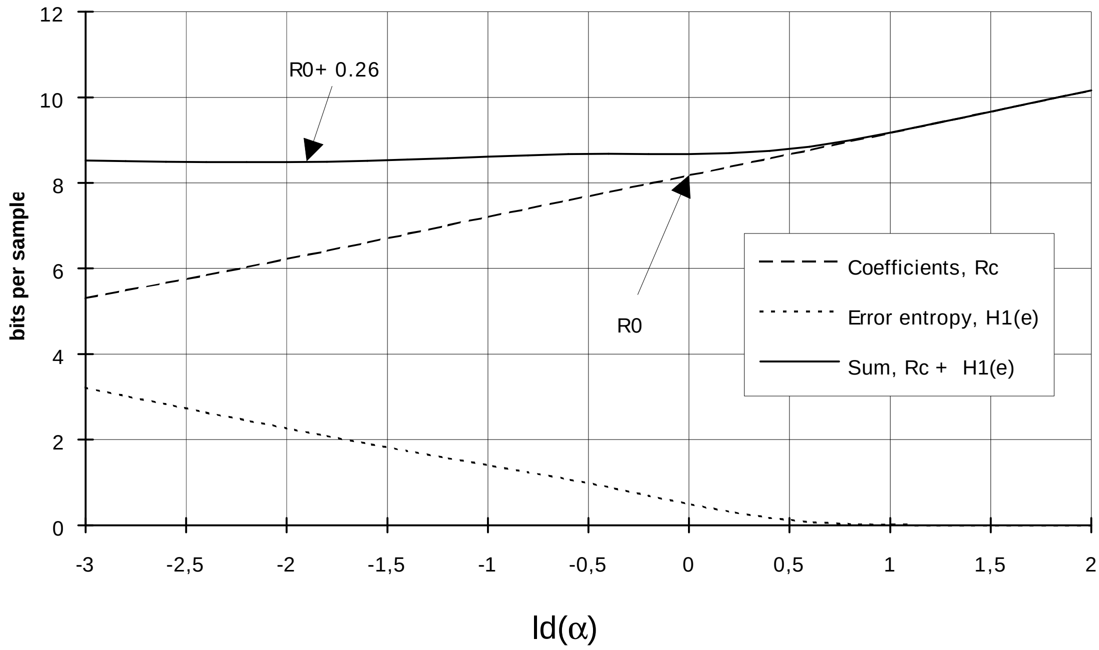

Fig. 3: Bit rates depending on the scalefactor. The values have been measured for a piece of pop music.

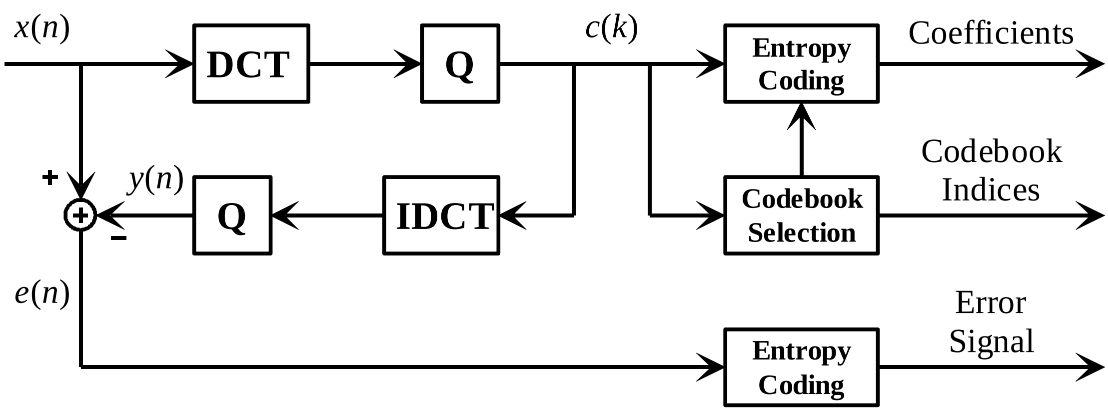

Fig. 4: Block diagram of the implemented lossless transform coder.

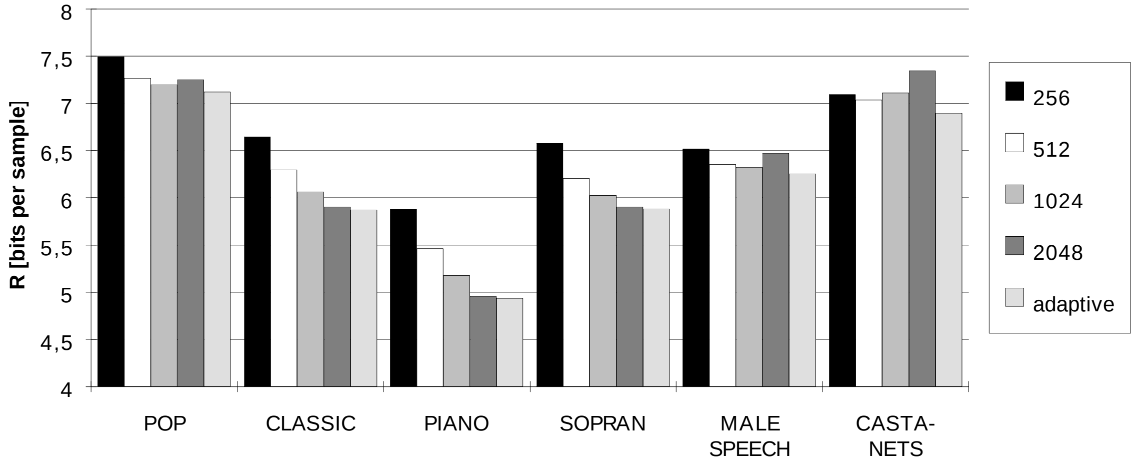

Fig. 5: Overall bit rates obtained by lossless transform coding with different block lengths.