Audio Engineering Society
Convention
Paper

Presented at the 115th Convention
2003 October 10–13 New York, NY, USA

Error Mitigation in MPEG-4 Audio Packet Communication Systems

Schuyler Quackenbush[1], Peter F. Driessen[2]

[1]*Audio Research Labs, Scotch Plains, NJ, 07076, USA*

[2]*Dept. of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada*

Correspondence should be addressed to Schuyler Quackenbush (srq@audioresearchlabs.com)

ABSTRACT

This paper investigates techniques for mitigating the effect of missing packets of MPEG-4 Advanced Audio Coding (AAC) data so as to minimize perceived audio degradation. Applications include streaming of AAC music files over the Internet and wireless packet data channels. A range of techniques are presented, but statistical interpolation in the time/frequency domain is found to be the the most effective. The novelty of the work is to use statistical interpolation techniques intended for time domain samples on the frequency domain coefficients. A means of complexity reduction is presented, after which the error mitigation is found to require on average 17% additional computation for a channel with 5% errors as compared to a clear channel. In an informal listening test, all subjects preferred this technique over a more simplistic technique of signal repetition, and for one signal item statistical interpolation was preferred to the original.

1. INTRODUCTION

Streaming audio over the Internet is a popular application, and currently two proprietary systems [1][2] dominate the market. Standardized technology [3][4] [5] is in widespread use for music download for storage and playback, and is gaining in popularity for music streaming. Such systems typically include a perceptual audio coder and a packet transmission protocol designed for streaming over IP networks.

The audio coders may be MPEG-1 layer 3 (MP3), MPEG-2 or MPEG-4 Advanced Audio Coding (AAC), or other proprietary algorithms. Typical coders achieve a compression ratio of about 10:1, and thus requires 1-2 bits per sample, or in the range of 32-128 Kbps for a stereo signal, depending on the audio sampling rate. The Internet Streaming Media Alliance [7] is developing open standards for streaming media over the Internet. Streaming audio over wireless IP networks such as 3rd/4th generation cellular wireless networks [8][9] is an emerging application, which is expected to grow as these networks are deployed.

The delay requirements for streaming data are different than for voice, interactive data or bulk data transfer. For streaming data, each packet has a finite lifetime, i.e. it is considered lost (late, missing, erased) if not delivered within a certain time window. Packet losses are caused by network congestion, misrouted packets, or fading and interference on wireless links. The perceived audio quality suffers as a result of these packet losses, to the point that if losses are excessive the result is unacceptable quality of service. Since most practical networks cannot guarantee zero packet loss, techniques are required to conceal the effect of those packets that are lost.

Error concealment is done by generating packets that are perceptually similar to the missing packets. This paper describes an approach for generating the replacement packets using statistical interpolation of the MDCT frequency domain coefficients.

1.1. MPEG Advanced Audio Coding

MPEG AAC supports a wide range of sampling frequencies (from 16 kHz to 96 kHz) which enables it to have an extremely wide range of bitrates. This permits it to support applications ranging from professional or home theater sound systems to Internet music broadcast systems. A simplified block diagram of the AAC encoder is shown in Fig. 1, and the decoder is shown in Fig. 2.

Efficient source coding is achieved by exploiting correlations between audio samples and the statistics of the quantized representation (removal of redundancies) as well as models of auditory perception (removal of irrelevancies). Since the most important auditory masking effects are best described in the frequency domain, perceptual audio coding is done in the time-frequency domain.

AAC uses a high frequency-resolution, 1024-band Modified Discrete Cosine Transform (MDCT) for maximum statistical signal gain, and can increase its time resolution by switching to 128 bands when the signal exhibits non-stationarity. This resolution-switching, or “block switching” capability serves to contain the backward spread of quantization noise in the time domain. At 48 kHz sampling rate, this corresponds to a frequency resolution of 23 Hz and a time resolution of 21.3 ms for the high-frequency resolution blocks (so-called “long blocks” because they are processed as one block of 1024 samples), and a frequency resolution of 187 Hz and a time resolution of 2.7 ms for the high-time resolution blocks (so-called “short blocks” because they are processed as eight blocks of 128 samples). There are four window shapes associated with the transform: long, start, short and stop. The start and stop facilitate transition between the long and short block types. The long, start and stop windows are “long blocks” in that they produce a single time-sample of 1024 frequency coefficients. In contrast, the short window is associated with eight applications of the 128-band transform and hence produces a sequence of eight samples of 128 frequency coefficients. An example of a long-start-short-stop-long window sequence is shown in Fig. 3.

The psycho-acoustic model specifies the quantizer step size per scale factor band, the scale factor bands being a partitioning of the frequency spectrum with each band roughly equal in width to one-half critical band. Quantization noise is set separately in each scale factor band so as to fall below the masking threshold.

AAC is a block-processing coder, in that each of a sequence of blocks of 1024 input samples are compressed into a “raw data block.” In this paper, the term ”block” will be used to refer to either the 1024 input waveform samples, the corresponding compressed raw data block, the corresponding partially decoded set of time-frequency coefficients, or the corresponding 1024 coded and decoded output waveform samples. Which of these is meant is clear from the context of the discussion. AAC

is an instantaneously variable rate coder, in that it allocates sufficient bits to each block to represent the audio signal at a constant quality, rather than at a constant bit rate (although the encoder can use output buffering to smooth the instantaneous rate such that it can transmit over a constant rate channel.) At 48 kHz sampling rate and 96 kb/s for a stereo signal, the average length of an AAC raw data block is 2048 bits (256 bytes). Lower data rates can be achieved by either reducing the signal bandwidth (e.g. by reducing the sampling rate), by reducing the signal quality (i.e. decreasing the signal to mask ratio), or both.

An AAC raw data block has the desirable property that it can be decoded without knowledge of adjacent blocks. This is important in a packet data communication environment in that a single missing raw data block does not impact the reconstruction of adjacent raw data blocks.

1.2. Packet networks

Typical packet networks implement a protocol stack, in which system design issues may dictate that layers of the stack are isolated. The result is that data packetization at the transport or network layer may ignore data framing information available from the application layer. Similarly, data packetization at the link or physical layer may ignore data framing information available from the transport or network layers. This may have a significant impact on packet loss at the application layer, in that packet losses at a lower layer will typically result in loss of portions of one or more adjacent packets at higher layers, as shown in Fig. 4. Therefore if a link layer packet is lost, then (at least one) entire IP packet is lost, and the corresponding bits are erased before being passed to the AAC decoder. Since an AAC raw data block can be decoded independent of adjacent blocks, but partial blocks cannot be decoded, these erased bits will typically cause two adjacent AAC blocks to be lost. MPEG-4 error resilience techniques [6] designed to recover partially lost frames could be used to improve performance, but are not considered in this work.

1.3. Network design parameters

All packet networks make a tradeoff between throughput and delay. The longer the permitted delay, the greater the throughput. Streaming audio requires a strict upper bound on delay, and has to accept the loss of packets delayed beyond this bound and the corresponding drop in throughput, increase in average packet error rate and reduction in the perceived quality of service. Another important factor in determining quality of service is the burstiness of packet errors, with longer bursts more difficult to conceal.

2. OVERVIEW OF MITIGATION TECHNOLOGY

Estimation of missing signal intervals can be done in the time domain (at the output of an audio decoder) or in the frequency domain (internal to the audio decoder). For error concealment in the time domain, the literature of audio restoration [14] [15] offers useful insights, since click removal is analogous to concealing erasures.

Error concealment in the spectral domain is considered in [10][11]. It is found that the signal quality is much more degraded by errors that result in an increase in the magnitude of a DCT coefficient, rather than by those that result in a decrease or an inversion of sign. A simple concealment technique in [10] is to clip large DCT coefficients.

In [11], subjective signal quality is estimated by the number of times the noise-to-mask ratio exceeds 0 dB. For sample oriented concealment methods in which a single erroneous frequency-domain sample is replaced, simple methods such as repetition, linear interpolation or L/R replacement yielded no improvement over muting. Prediction was found to be most effective when the prediction gain is high, and order 16 was found to yield the best performance. The potential prediction gain is higher in the low frequency bands.

Error concealment in a intermediate or sub-band domain with a time/frequency resolution in between the MDCT and time domain is considered in [12]. Prediction is used for tonal signals and noise substitution for noise-like signals. Noise substitution is done in the MDCT domain to capture the spectral shape and prediction is used only in the lower sub-bands.

3. MITIGATION OF LOST PACKETS

MPEG AAC has several desirable properties when considering how to mitigate the effect of lost packets. First, it has an overlap-add synthesis transform (i.e. Modulated Lapped Transform, specifically IMDCT) whose tapered windows serve to smooth the transition between known and estimated time-domain output blocks (Fig. 3). Second, it has a compressed data structure that is amenable to being partitioned into packets (i.e. at raw data block boundaries).

If one AAC raw data block is exactly the payload of one transport layer or channel packet, then channel losses

will have minimum effect on the AAC decoder. Such a mapping may not always be possible. For example, when coding multi-channel signals the AAC raw data block size may be larger that the network maximum transmission unit (MTU), so that an AAC raw data block would be split across two or more channel packets. Conversely, when operating at very high compression, AAC raw data blocks may be so small as to make the network packetization overhead prohibitive, so that multiple raw data blocks are put into each channel packet. In another scenario, the channel transport may be asynchronous to the AAC encoder, such that the channel packetization is unaware of the AAC raw data block boundaries. This mapping has at best the performance of synchronized packetization and on average much worse performance, as the loss of a single channel packet typically causes losses in multiple AAC raw data blocks (Fig. 4).

Because AAC has a time/frequency representation analysis/synthesis structure for coding audio data, it was decided to estimate lost data in the time/frequency domain. Unknown frequency coefficients are estimated from coefficients that are of identical frequency and adjacent in time. This is facilitated by creating a mitigation state buffer that is inserted into the AAC decoder block diagram just prior to the IMDCT and which contains a number of blocks of time/frequency data (Figs. 2, 5).

Mitigation requires using a buffer so that good blocks can be used to reconstruct the intervening lost blocks. This buffer imposes a additional delay beyond the nominal startup delay associated with decoding and presentation.

3.1. Simulation of channel errors

Several different artificial error patterns were used for testing. Periodic error bursts of 1-10 bad packets at 400 or 1000 msec intervals were used because the listener can predict when the next error will occur and thus critically evaluate the effectiveness of concealment on various types of musical program material. The periodic errors are typically a contiguous burst of all bad packets of the given length, although they may be any pattern of good and bad packets of the given length beginning and ending with a bad packet. These artificial patterns allow careful evaluation of the error concealment technique for various packet error patterns and rates within an error burst. The listener can compare the subjective effect of the different error rates and patterns within the error burst.

4. PROPOSED MITIGATION TECHNIQUES

4.1. Problem definition and notation

To set up the problem, we assume a buffer of $2N+1$ sets of AAC time/frequency coefficients $c_{m,k}$, each set associated with a single raw data block and decoded to the point just prior to the IMDCT (Fig. 2). When referring to specific time/frequency coefficients in this array we shall use $k = 1, ...1024$ as the frequency index and $m = 1, .., 2N+1$ as the time or block index. The $c_{m,k}$ contain q missing blocks in the interval $m = 1, ..., N+1$. $t = mt_b$ is the time corresponding to the block index m with block transmission time t_b. Assume that the q missing blocks are located at known positions $m = t_i, i = 1, ..., q$, where t_i is the integer time index of the missing blocks. Thus missing blocks are not necessarily contiguous. For each missing block, all of the frequency coefficients k are missing.

4.2. Overview

Channel transmission errors or delays will result in a gap in the sequence of AAC raw data blocks that are delivered to the AAC decoder within the hard realtime limits set by the decoder. Such missing raw data blocks will be referred to simply as block errors. An obvious choice for the output waveform corresponding to the block error intervals is silence, but whereas this might be the best estimate of the missing data based on no information, this section will present numerous techniques for estimating the data information based on the statistics of adjacent audio data.

As was stated previously, the AAC decoder employs an overlap-add IMDCT synthesis filterbank. Estimating the missing information in the time/frequency domain, just prior to the IMDCT, takes advantage of the inherent smoothing provided by the overlap-add part of the synthesis filterbank. Therefore estimation of missing data is done in the *Mitigation* block, as shown in Fig. 2.

The Mitigation block consists of a buffer containing a sequence of sets of time/frequency coefficients $c_{m,k}$, each set corresponding to an AAC raw data block, and a mechanism to estimate sets of time/frequency coefficients that may be missing. This is shown in Fig. 5, where Z^{-1} represents the storage buffer for one set of 1024 time/frequency coefficients $c_{m_0,k}$ which correspond to one AAC raw data block at a particular time index m_0. The mitigation buffer is a tapped delay line of such storage buffers, $2N+1$ in total. The most recent set of

time/frequency coefficients enter the left side of the delay line (the input where $m = 1$), while the output (passed to the synthesis filterbank) are those in the buffer where $m = N + 2$. Input data that is associated with errored blocks (i.e. missing packets) is tagged as missing and the time/frequency coefficients are set to zero.

When the delay line is shifted, if the look-ahead storage element prior to the mitigation element contains valid data, it is shifted into the mitigation element. Conversly, if it is tagged as missing, the block estimation is invoked and an estimate of the missing data is loaded into the mitigation element. The summation node after the mitigation element might be better thought of as a switch, but summation is correct since one or the other of the summation inputs always has zero value.

Each delay element stores the set of 1024 AAC time/frequency coefficients, so one can think of the delay line in Fig. 5 as a $2N + 1$ row by 1024 column storage array $c_{m,k}$, as shown in Fig. 7A. Note that for a tapped delay line that shifts from left to right, the direction of increasing time in Fig. 7A is from right to left (i.e. where the oldest block is to the right, the newest to the left).

If there is a single missing block, the mitigation buffer provides a symmetric support in time surrounding the missing block. The size of the buffer, dictated by N, has several implications. First, it introduces an additional delay in the decoder of $N + 2$ times the AAC block time (which is 21.3 ms at 48 kHz sampling rate). Second, a larger buffer requires more memory which will typically increase implementation complexity. Third, the larger the buffer the better the ability to estimate missing data based on data present in the buffer, particularly if the signal is reasonably stationary and model-based signal estimation is used. The work presented here assumes the application to be one-way communication (such as streaming audio), so that delay is of minimal concern, and chooses to explore what can be gained in the performance of error mitigation techniques at the cost of increased memory requirements. However, we constrain the delay to be less than 300 ms, which leads to a start-up delay that is well within the range of what listener would find acceptable when changing program channels.

From a signal estimation point of view, the mitigation buffer need be only as big as is needed to build a good signal model, which in turn depends on signal statistics, i.e. the extent of signal stationarity. There is no point having a buffer whose extent is greater than the support used to build a signal model.

Estimated blocks are tagged as not missing and subsequently treated as good blocks. Hence in Fig. 5, blocks at and to the left of the mitigation element might be missing, but blocks to the right of the mitigation element are good (i.e. marked as not missing). The exception is that if a block cannot be estimated (e.g. after an extended outage in which the entire buffer contains errored blocks) then it remains tagged as missing.

4.3. Mitigation techniques

A number of techniques for estimating missing sets of time/frequency coefficients have been investigated (see Fig. 6). In all cases, the techniques estimate a missing coefficient c_{m_0,k_0} at time index m_0, and a frequency k_0 from the set of coefficients $\{c_{m,k_0}\}, m = 1, ..., 2N + 1$ that are not missing. The techniques are briefly listed here and subsequently described in detail:

- **Statistical Interpolation - SI** Statistical interpolation estimates missing coefficients as linear combinations of known ones. It is assumed that for given frequency bin (with spectral coefficient index k), the samples $s_j = c_{j,k}$ are a realization of an autoregressive process of order p such that

$$s_j = e_j - \sum_{l=1}^{p} a_l s_{j-l} \tag{1}$$

  where e_j is white and $\mathbf{a}$ is the vector of prediction coefficients with $a_0 = 1$. The missing samples are determined so as to minimize the variance of the estimation error e_j. The algorithm, from [13], is described in Appendix A. Extensions of the algorithm are given in [14]. Best results are obtained with $p > 3q$, with mostly acceptable results for $p > q$.

  The novelty in the present work is to use statistical interpolation techniques of [14] intended for time domain samples on the frequency domain coefficients.

  One advantage of this approach is reduced computational complexity. The gaps in the frequency domain are typically only a few samples (blocks), corresponding to a few thousand samples in the time domain. Since the matrix inversions in the algorithm will grow as the square or cube of the number of missing samples q, 1024 distinct interpolations of q missing samples is more efficient than a single interpolation of $1024q$ missing samples.

Another advantage is that the time domain overlap of adjacent blocks built into AAC helps to smooth the transition between the good block and mitigated block.

- **Predict - P** Prediction of one or more contiguous missing coefficients from prior or subsequent coefficients. The prediction can be either causal or anti-causal and typically uses a distinct autoregressive model at each frequency. Prediction coefficients are estimated using the covariance method and estimates are computed as $s_j = \sum_{l=1}^{p} a_l s_{j-l}$ for causal prediction or $s_j = \sum_{l=1}^{p} a_l s_{j+l}$ for anti-causal prediction.

- **Repeat - R** Repetition of adjacent set of time/frequency coefficients, but with two additional features. First, a technique for decorrelation is used so as to avoid the unnatural "buzziness," caused by repetition of the 21.3 msec blocks at a 48 Hz rate. Second, each repetition has exponentially increasing attenuation, thus fading gradually to mute in the case of extended outages. Repetition can be causal, anti-causal or both.

As explained in Sec. 1.1, the AAC synthesis filterbank has variable resolution. This has a profound effect on the execution of mitigation strategies, in that whenever the filterbank transitions through a resolution switch sequence, as shown in Fig. 3, the tiling of the time-frequency plane has a discontinuity. Fig. 7A illustrates the time/frequency structure in a portion of the buffer for the case that it contains only long blocks. Here adjacent coefficients in the array represent the same time/frequency resolution. However Fig. 7B illustrates the time/frequency structure for the case that the middle block is a short block, i.e. is 8 sets of 128 time/frequency coefficients. Mitigation strategies of repeat, predict or interpolate make no sense when operating across such a discontinuity. Although one could convert the required blocks to a common time/frequency resolution (e.g. all short blocks), this can be computationally expensive. Instead, we have chosen to adopt strategies that avoid operating across time/frequency discontinuities, those strategies being

- **Relabel** This case employs delayed decision, and is the reason that estimation is done one block prior to output (i.e. in Fig. 5 the mitigation block is distinct from and immediately to the left of the output block). If there is a single short block that is a missing block, then the immediately adjacent blocks are a start block and a stop block (start makes the transition from long to short and stop makes the transition from short to long). Since the short block data is missing, the relabel strategy relabels each of the start, short and stop blocks as long blocks and uses the prediction or interpolation techniques to estimate the missing data. Relabeling works for the following block sequences, in which the middle block is the missing block:
  - **Start - short - stop** This was the example presented.
  - **Start - short - short** The relabel strategy relabels start to long and the missing short to start and uses causal prediction.
  - **Short - short - stop** The relabel strategy relabels stop to long and the missing short to stop and uses anti-causal prediction.

- **Repeat Shorts** This is the straightforward case of having a missing short block adjacent to a good short block. The strategy does not repeat the 8 sets of 128 good coefficients, but rather repeats and decorrelates the one immediately adjacent set of 128 coefficients. This can be causal, anti-causal or both. If the missing short has a good short on each side, then both causal and anti-causal is used, estimating 4 sets from each of the adjacent good sets.

The preferred method of signal estimation is statistical interpolation (SI), since it estimates missing data from both prior and subsequent surrounding data. Next preferred is prediction from either prior data (causal prediction) or subsequent data (anti-causal prediction). For both SI and prediction the model order and interval from which the model is estimated can be varied based on the number of missing samples, e.g. a lower order model and a smaller interval of support for fewer missing samples. The final method is repeat with decorrelation, either of long or short blocks. Note that an extended outage results in an extended period of repeated blocks each of which has increasing attenuation thus fading gradually to mute. For any burst of more than five consecutive missing blocks repeat is used rather that SI.

5. COMPLEXITY

As already noted, model-based estimation in the time/frequency domain has a very large computational

advantage relative to modelling in the time domain. However this computational advantage can be increased by noting that the purpose of transform-based signal compression is to concentrate the signal variance in as few bins as possible, and that further complexity reduction can be realized by estimating the missing time/frequency samples only for those bins in which there is appreciable signal energy.

Inspection of the cumulative distribution function of the variance of the time/frequency coefficients in the AAC decoder for a representative set of musical excerpt reveals that, on average, more than 95% of the signal variance is represented by fewer than 128 of the total 1024 MDCT bins. Hence, estimation via statistical interpolation need not be done for every bin.

A straight-forward strategy to capitalize on this was adopted in this work. At the time that estimation of missing information is called for, the system

- Computes the signal energy in each bin based on the set of blocks that would be used to support the SI estimate.
- Sorts those energies
- Computes a cumulative energy distribution
- Runs the SI algorithm on the bins having the highest energy until a total of 95% of the energy has been estimated.

Decoding time was measured for three conditions: clear channel, 15% block error rate with estimation of each bin through 20 kHz (first 920 bins), 15% block error rate with estimation of only the bins that represent 95% of the signal variance. The results are shown in Table 1, in which all execution times are normalized so that average decode time for the clear channel case is 1.0. In the table the rows are numbers associated with specific signal files (Filename) and the columns are normalized execution time for the 15% errored channel case for each of the two estimation strategies: bins representing the first 20 kHz of frequency and bins representing the first 95% of signal variance. The Speedup column is the ratio of the first two columns, and indicates the speedup delivered by the 95% variance method. The last three rows show minimum, maximum and average relative execution time.

Filename	20 kHz	95%	Speedup
flute3.raw	2.56	1.35	1.90
musicman6a.raw	2.91	1.43	2.03
porgy3b.raw	2.46	1.57	1.57
winston3.raw	2.77	1.26	2.20
hawkins21.raw	2.46	1.61	1.53
porgy6a.raw	2.46	1.47	1.67
svega4.raw	2.37	1.33	1.78
dire2s.raw	2.17	1.49	1.46
organ.raw	2.43	1.50	1.62
valdes2s.raw	2.38	1.54	1.55
Min	2.17	1.26	
Max	2.91	1.61	
Average	2.50	1.46	1.72

Table 1: Algorithm Complexity

Clearly the computation required to run SI depends on the local signal statistics. However the technique of estimating only those bins that contribute significantly to the signal variance (up to a limit of 95% of the signal variance) leads to an modest additional computation load for mitigation, as shown in Table 2. As the table shows, while the normalized load is 1.5 for the 15% error rate, it drops to only 1.17 for a 5% error rate.

Block Error Rate	Normalized Computational Load
2.5%	1.08
5%	1.17
10%	1.33
15%	1.50

Table 2: Normalized Complexity as a Function of Error Rate

6. PERFORMANCE

The performance of the new error concealment algorithm was determined via a subjective listening test. The tested algorithm incorporated complexity reduction techniques, terminating signal estimation after 95% of signal viariability had been accounted for. A total of 14 listeners with varying degrees of experience with audio coding participated.

A forced-choice paired-comparison method was used, in which the listener responded on the scale shown in Table 3.

score	Descriptor
2	A is much better than B
1	A is better than B
0	A is the same as B
-1	B is better than A
-2	B is much better than A

Table 3: Listener Response Scale

There were three systems under test: the original with no errors (O), a simple mitigation technique that merely repeated the previous good frame (R), and statistical interpolation (SI) as presented in this paper. Since the goal was to assess the performance of SI, only two comparisons were presented: SI vs R and SI vs O. In order to control the effects of presentation order, both orders for each comparison were presented (i.e. A/B and B/A). The test consisted of a total of 16 trials for each of the 14 listeners, (4 signals x 2 comparisions x 2 presentation orders) for a total of 224 trials (16 trials x 14 listeners).

Four 15-second music excerpts were used as test items, indicated in the first column of Table 4.

The same channel error pattern was imposed on each mitigation scheme, and consisted of three consecutive errored blocks every 20 blocks, thus yielding periodic errors of 70 msec every 464 msec (where a block is 1024 samples at a sampling rate of 44.1 kHz).

In compiling the results, a score of "1" was given to a system that was evaluated as "better than" the system to which it was compared, and a score of "2" was given if it was evaluated as "much bettern than." The test results are shown in Table 4. Each row shows the results per test item, with the last row showing the average results. Columns are for the original (O), simple repetition mitigation (R), and statistical interpolation mitigation (SI). Entries show the the total score per system (O, R, SI) for all listeners, normalized to be between 0 and 100.

item	O	R	SI
piano	73.0	1.6	25.4
brass	69.5	0.0	30.5
violin	38.6	0.0	61.4
vocal	57.9	0.0	42.1
average	61.4	0.4	38.2

Table 4: Performance results

These results were very encouraging. Statistical interpolation error concealment worked particularly well on the largely stationary violin passage, where it was preferred over the original! It worked less well on music with sharp attacks, since errored blocks containing onsets are not well estimated based on statistics of surrounding blocks. Statistical interpolation clearly performed better than the simple repeat strategy.

Informal tests using the error concealment on an AAC streaming webcast received via the internet and cable modem showed that the number of noticeable dropouts was significantly reduced. Thus we conclude that the concealment technique increases the "up-time" of a streaming music service.

7. CONCLUSIONS

Statistical interpolation of AAC frequency domain coefficients is found to be an effective means of minimizing the perceived audio degradation caused by missing packets, and thus is a useful tool for mitigating the effect of packet loss. The key novelty is the application of statistical interpolation to the AAC frequency domain coefficients (transform bins). Since each bin is a bandlimited signal that occupies the time of an entire 1024 sample AAC block, the statistical interpolation assumes a low order autoregressive process for each bin, thus avoiding the need for high order models as must be done in the time domain.

Subjective tests show that statistical interpolation yields acceptable error concealment even at an error rate of 15 percent. At a 15 percent error rate, statistical interpolation increases computational complexity by a factor of only 1.5. The error mitigation described here has been successfully applied in the "Verdi" AT&T streaming media player [17].

8. REFERENCES

[1] www.microsoft.com/windwos/windowsmedia/EN/default.asp

[2] www.real.com

[3] www.mpeg.org/MPEG/MP3.html

[4] www.mpeg.org/MPEG/aac.html

[5] www.apple.com/quicktime

[6] H. Purnhagen, "An Overview of MPEG-4 Audio Version 2," AES 17th International Conference, Sep. 2-5, 1999, Florence, Italy

[7] www.ism-alliance.com

[8] www.3gpp.org

[9] www.3gpp2.org

[10] J. Herre, E. Eberlein, “Error Concealment in the spectral domain”, *93rd AES Convention*, 1992 Oct 1-4, preprint 3364.

[11] J. Herre, E. Eberlein, “Evaluation of concealment techniques for compressed digital audio”, *94th AES Convention*, 1993 March 16-19, preprint 3460.

[12] Lauber,P., Sperschneider,R., “Error concealment for compressed digital audio”, *111th AES Convention*, September 2001, preprint 5460.

[13] R. Veldhuis, *Restoration of lost samples in digital signals*, Prentice-Hall, 1990.

[14] S.J. Godsill, P.J.W. Rayner, *Digital Audio Restoration*, Springer, 1998

[15] J.J.K. O Ruanaidh, W.J. Fitzgerald, *Numerical Bayesian Methods Applied to Signal Processing*. Springer, 1998

[16] J.J.K. O Ruanaidh, W.J. Fitzgerald, “Interpolation of mssing samples for audio restoration”, *IEE Electronics Letters*, vol. 30, no. 8, 14 April 1994, pp. 622-623.

[17] M. Kretschmer, “Get a KISS - Communication Infrastructure for Streaming Services in a Heterogeneous Environment”, *Proceedings ACM Multimedia ’98*, pp. 401-410, Bristol, UK, September 12-16, 1998.

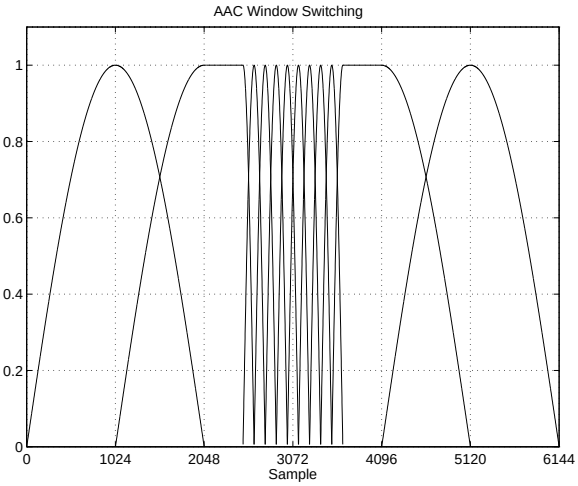

Fig. 3: AAC window switching sequence

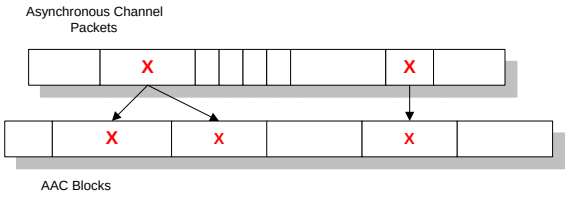

Fig. 4: Application and link level packetization

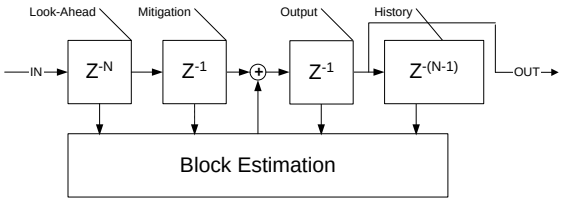

Fig. 5: Mitigation Buffer

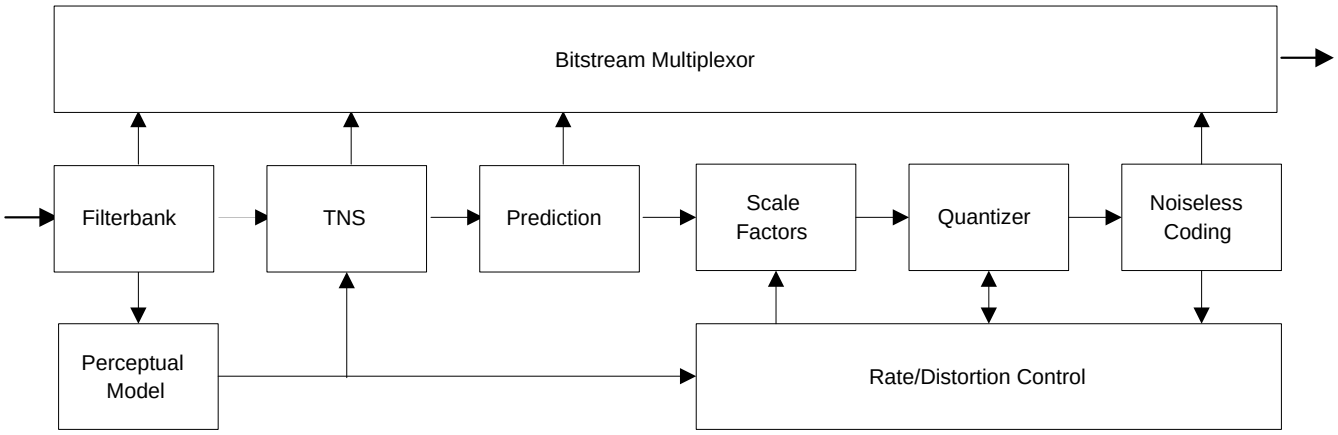

Fig. 1: AAC encoder block diagram

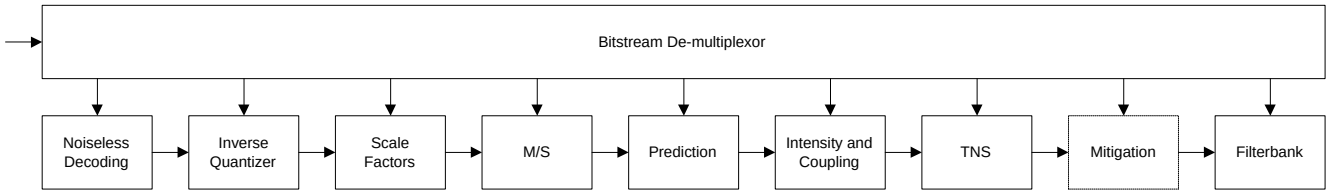

Fig. 2: AAC decoder block diagram

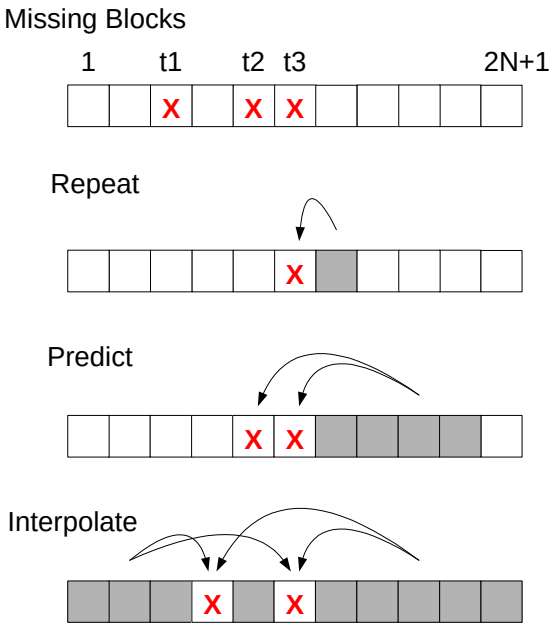

Fig. 6: Mitigation Techniques

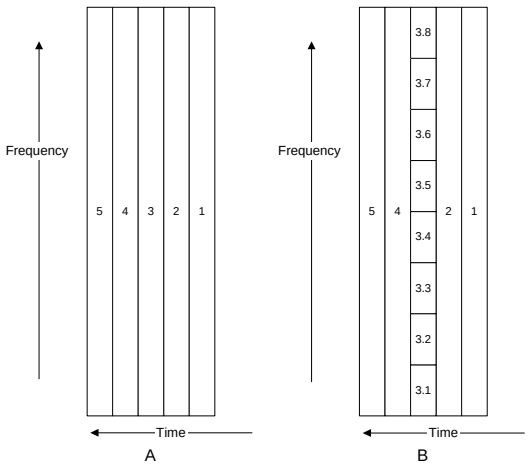

Fig. 7: Mitigation buffer time-frequency tiling during resolution switching

1. STATISTICAL INTERPOLATION ALGORITHM

In this section, the algorithm for statistical interpolation is described, and an enhancement to improve the perceived quality of the restored audio is outlined.

1.1. Interpolation of AR process

Following [13] (which uses j instead of m), the segment of data is $s_j, j = 1, ...N$, where the samples $s_j = c_{j,k}$ and $t = jt_s$ is the actual sampling time. The samples s_j are modelled as arising from an autoregressive (AR) process $s_j = e_j - \sum_{l=1}^{p} a_l s_{j-l}$ with AR coefficients a_l. The missing (unknown) samples are located at known positions $j = t_i, i = 1, ..., q$, where t_i is the integer time index of the known position and q is the number of missing samples.

The missing samples s_{t_i} are estimated using the known samples s_j and coefficients $h_{i,j}$ via

$$s_{t_i} = \sum_{j \in W/V} h_{i,j} s_j, i = 1, ..., q \tag{2}$$

where W/V is the set of known samples, i.e. all the time indices $j = 1, ..., N$ not including the missing sample indices $t_i, i = 1, ..., q$. In order to get the best RMS performance, the coefficients $h_{i,j}$ must be optimized over all realizations of the noise process e_j (and thus the stochastic process s_j), i.e. chosen so they minimize the expected variance of the statistical restoration error e_j.

In [13], a procedure is derived for iteratively solving for the estimated AR parameters $\hat{a}_i$ and the estimated missing samples $\hat{x}_i = \hat{s}_{t_i}$. This is an iterative algorithm that alternately applies steps 1 and 2 described below, until some criterion is met, which could be as simple as a fixed number of iterations. This procedure is summarized below and the derivation is available in [13]. An alternate presentation appears in [15] Chapter 6. It is assumed that the order of prediction $p \geq 3q$, or at least 3 times the number of missing samples.

Initialization

Set $s_j = 0$ for $j = t_i, i = 1, ...q$.

Step 1 - estimation of AR coefficient vector $\hat{\mathbf{a}}$ *with elements* $\hat{a}_i, i = 1, ..., p$

Find the $p \times p$ autocorrelation matrix $\mathbf{C}$ with elements [1]

$$c_{i,j} = \sum_{k=p+1}^{N} s_{k-i} s_{k-j} \tag{3}$$

for $i, j = 1, ..., p$. Similarly, find the $p \times 1$ vector

$$\mathbf{c} = [c_{0,1}, ..., c_{0,p}] \tag{4}$$

using (3) for $i = 0, j = 1, .., p$. Solve for the $p \times 1$ vector $\hat{\mathbf{a}}$ using

$$\mathbf{C}\hat{\mathbf{a}} = -\mathbf{c} \tag{5}$$

Step 2 - estimation of vector of q missing samples $\hat{\mathbf{x}}$ *with elements* $\hat{x}_i = s_{t_i}, i = 1, ..., q$, using

$$b_k = \sum_{j=0}^{p} a_j a_{j+k} \tag{6}$$

Find the $m \times N$ matrix $\mathbf{G}$ with elements

$$g_{i,j} = b_{j-t_i} \tag{7}$$

Define the $N \times 1$ vector $\mathbf{v}$ with elements $v_j = s_j$.

Find the $q \times 1$ vector $\mathbf{z}$ from

$$\mathbf{z} = \mathbf{G}\mathbf{v} \tag{8}$$

Find elements of the $q \times q$ matrix $\tilde{\mathbf{G}}$ using

$$\tilde{g}_{k,j} = b_{|t_j - t_k|} \tag{9}$$

Solve for the $q \times 1$ vector $\hat{\mathbf{x}}$ using

$$\tilde{\mathbf{G}}\hat{\mathbf{x}} = -\mathbf{z} \tag{10}$$

Replace $s_{t_i}, i = 1, ..., q$ with $\hat{x}_i$.

[1] Note that the elements $c_{i,j}$ here are unrelated to the T/F coefficients $c_{m,k}$ in the main text.