Packet Loss Concealment for Audio Streaming based on the GAPES and MAPES Algorithms

Hadas Ofir and David Malah
Department of Electrical Engineering
Technion IIT Haifa 32000, Israel
hadaso@tx.technion.ac.il, malah@ee.technion.ac.il

***Abstract*— Packet loss in internet audio streaming applications is a well-known and difficult problem. Unless concealed, each such loss produces an annoying disturbance. In this work we present a new algorithm for audio packet loss concealment, designed for MPEG-Audio streaming, based only on the data available at the receiver. The proposed algorithm reconstructs the missing data in the DSTFT (Discrete Short-Time Fourier-Transform) domain using either the GAPES (Gapped-data Amplitude and Phase Estimation) or the MAPES (Missing-data Amplitude and Phase Estimation) algorithm. The algorithm was implemented on an MP3 coder but is also suitable for MPEG-2/4 AAC. Examined subjectively by a group of listeners, the proposed algorithm performs better than other previously reported methods, even at high loss rates.**

I. Introduction

Broadcasting multimedia such as audio and video in real-time over the internet is referred to as *streaming media*. This kind of delivery has become popular in recent years. Audio streaming operates by compressing short segments of a digital audio signal that are then gathered into small data-packets and are consecutively sent over the internet. When reaching their destination the packets are reassembled and decompressed into waveform samples. To maintain seamless play, several packets are downloaded to the user's computer and buffered before play. While the buffered packets are played, more packets are being downloaded and queued up for playback. This way, the data is ready to be played in real time, without having to download all of the content before use.

However, since internet delivery doesn't assure quality of service, data packets are often delayed or discarded during network congestions, creating gaps in the streamed media. Each such gap, unless concealed in some way, produces an annoying disturbance. The common approach for dealing with such cases is to fill in the gap, trying to approximate the original waveform, so that a human listener will not notice the disturbance. However, since typical audio packets correspond to 20-30 msec of audio, the gap created by even a single lost packet is relatively wide (around 1000 samples) and is therefore difficult to interpolate.

There are many techniques for packet loss recovery [1], [2], mostly designed for speech signals. Previous works start from simple methods, such as noise or waveform substitution [3] and packet repetition [4], on to advanced techniques that use interpolation in the compressed domain for MPEG audio coders [5], [6] or by using parametric audio modelling [7]. The various techniques can be roughly divided into two categories [2]: *Receiver-based* methods, which only use data available at the receiver in the concealment process, and *Sender-based* methods, which change the encoding format by adding some redundancy or additional side information for the receiver to use later. Our solution is receiver-based.

An earlier version of the proposed algorithm was presented in our previous work [8], where the missing data was reconstructed in the DSTFT domain based on the GAPES [9] (Gapped-data Amplitude and Phase Estimation) interpolation algorithm. This paper continues that work by presenting another alternative for activating the algorithm, this time based on the MAPES-CM [10] (Missing-data Amplitude and Phase Estimation - Cyclic Maximization) algorithm. MAPES-CM has lower complexity demands than GAPES, and can handle more loss patterns.

The remainder of the paper is organized as follows: Section II gives an overview of MPEG-audio compression, and Section III explains the considerations in choosing the concealment domain. Section IV describes the derivation of MAPES-CM, and Section V presents the concealment algorithm, where we focus on the practical aspects of using MAPES-CM instead of GAPES. Section VI describes the results of the quality tests, and Section VII concludes the paper.

II. MPEG-Audio Compression

In recent years, MPEG-1 Layer 3 audio coder, a.k.a. MP3, has become a common tool for internet audio streaming and is also known as an efficient way to store audio files. This is mainly due to the fact that the quality degradation arising from the MP3 lossy compression is almost negligible at typical rates [11]. MP3 and its successors (such as MPEG-2/4 AAC) are part of a family called *perceptual audio coders*. These coders achieve relatively high compression by exploiting the characteristics and limitations of the human auditory system. When a signal is coded, a psychoacoustic model is applied to it in order to determine the compression parameters. Then, compression is achieved by a quantization process that shapes the quantization noise so it is always below the auditory threshold, and hence is unnoticeable to a human listener.

The MPEG-audio encoding process is as follows: The digital audio file is divided into overlapping frames which are then transformed to the MDCT domain. For each frame, the encoding parameters are determined by applying the

1-4244-0230-1/06/$20.00 ©2006 IEEE

psychoacoustic model: One of these parameters is the type of window to be used in the MDCT transform, according to the frame's short- or long-term characteristics. Next, the MDCT coefficients are quantized. Each critical band has a different quantization step, which is determined by an iterative algorithm that controls both the bit-rate and the distortion level, so that the *perceived* distortion is as small as possible, within the limitations of the desired bit-rate. The quantized values and other side information (such as the window type for the MDCT) are encoded using Huffman code tables to form a bit-stream. In the MP3 standard, each frame contains 576 samples and each pair of frames forms an MP3 packet.

A. MDCT

The MDCT is a real-valued transform, turning $2N$ time samples into N MDCT coefficients, and is defined by:

$$X^M[k] = \sum_{n=0}^{2N-1} x[n] \cdot h[n] \cdot \cos\left(\frac{\pi}{N}\left(n + \frac{N+1}{2}\right)\left(k + \frac{1}{2}\right)\right) \quad (1)$$

where $0 \leq k \leq N-1$, $x[n]$ is the original signal and $h[n]$ is a window function.

The MDCT is a *lossless* transform if certain conditions are satisfied: First, consecutive MDCT segments should have 50% overlap between them. Second, the window functions of consecutive segments should relate to each other, as specified in [12]. The reconstruction of the time-domain samples from the MDCT coefficients is done by applying an overlap-and-add (OLA) procedure on the output of the inverse transform. Hence, in order to perfectly reconstruct a whole segment of $2N$ samples, 3 consecutive MDCT frames are required.

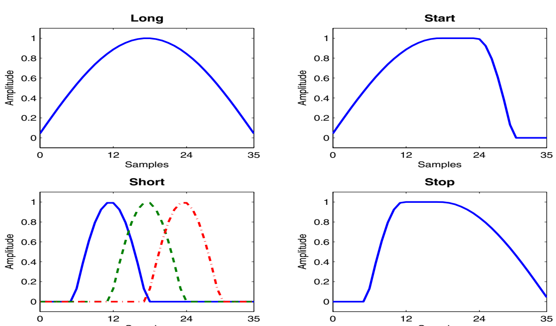

Fig. 1. The 4 window types defined by the MP3 standard.

The MP3 standard defines 4 window functions for the MDCT (see Fig. 1), which are denoted [4]: 'Long', 'Start' (long-to-short), 'Short', and 'Stop' (short-to-long). The psychoacoustic model decides which of them to use according to the statistical characteristics of the signal in each frame: The Long window allows better frequency resolution for audio segments with stationary characteristics, while the Short window, which actually contains 3 short MDCT transforms, provides better time resolution for transients. The 'Start' and 'Stop' windows are transition windows which separate between Long and Short windows. The windows' ordering rules can sometimes help in case of a packet loss, since for a single packet loss, the window types of the lost frames can be recovered, in most cases, by observing the window types of neighboring packets. For example: Long-X_1-X_2-Stop could only match a single possible pattern, where X_1 is a Start window and X_2 is a Short window.

III. Concealment Domain

As already mentioned, MPEG-audio coders compress the audio signal in the MDCT domain. Specifically in the MP3 coder, a loss of a single packet creates a gap of 1152 samples in the time domain, or equivalently, 2 consecutive MDCT coefficients at each of the 576 frequency bins. Since a smaller gap is easier to interpolate, it makes sense to apply the concealment directly in the MDCT domain, rather than the time domain, as suggested in [6]. In that work, an adaptive missing-samples restoration algorithm for auto-regressive time-domain signals was applied in the MDCT domain, where the coefficients of each frequency bin along time were considered as an independent signal with missing values.

However, there are some limitations to working in the MDCT domain: First, since different window types have different frequency resolutions, the MDCT coefficients of two consecutive frames at a certain frequency bin might represent different frequency resolutions. In such a case it doesn't make sense to estimate the coefficients of one frame from the other. Second, the MDCT coefficients typically show rapid sign changes from frame to frame, at each frequency bin, which makes it more difficult to interpolate them. These sign changes reflect phase changes in the complex spectral domain [13].

A possible way to cope with the first limitation above would be to convert the MDCT coefficients back into the time domain and then again to the frequency domain, this time using windows of constant resolution. A solution to the second limitation would be to work in a domain that has a less fluctuating representation of the signal, thus providing better interpolation results. Interpolation in the DSTFT domain overcomes both limitations and therefore was chosen as our concealment domain.

Working in the DSTFT domain requires to first convert the data from its compression domain: the MDCT. For this purpose we developed a computationally efficient procedure for direct conversion between the two domains, applying the DSTFT to the segments originally used by the MDCT. A detailed description of this conversion procedure exists in our previous reports [8], [14]. For example, here is the expression for the conversion from the MDCT to the DSTFT:

$$\begin{aligned} X^D_{(p)}[m] = & \sum_{k=0}^{N-1} X^M_{(p)}[k] \cdot \left(g^1_d[m,k] + (-1)^m \cdot g^2_r[m,k]\right) \\ & + \sum_{k=0}^{N-1} X^M_{(p-1)}[k] \cdot g^2_d[m,k] \\ & + \sum_{k=0}^{N-1} X^M_{(p+1)}[k] \cdot \left((-1)^m \cdot g^1_r[m,k]\right) \end{aligned} \quad (2)$$

where $0 \leq m \leq N$, since the output is conjugate-symmetric in the frequency domain. $p \in \mathbb{Z}$ is the current block time-index, and $(p-1)$, p, $(p+1)$ denote 3 consecutive MDCT blocks. Table I compares the complexity of our conversion to conventional conversion, assuming FFT cannot be used because the transform length is not an integer power of 2.

TABLE I
COMPLEXITY COMPARISON.

	Efficient Conversion	Conventional Conversion
MDCT to DSTFT	$6N^2$ mults, $8N^2$ adds	$10N^2$ mults, $10N^2$ adds
DSTFT to MDCT	$8N^2$ mults, $20N^2$ adds	$28N^2$ mults, $26N^2$ adds

However, in the DSTFT domain, the gap at each frequency bin, created by lost packets, is bigger than in the MDCT domain: Q consecutive lost packets, which means $2Q$ lost MDCT frames, affect $2Q+2$ DSTFT frames. This is because 3 consecutive overlapping MDCT frames are required to reconstruct a single segment of $2N$ time-domain samples. In the case of packet loss this means that a lost frame affects the reconstruction of not only its corresponding segment, but also its two closest neighbors. This concept is illustrated in Fig. 2.

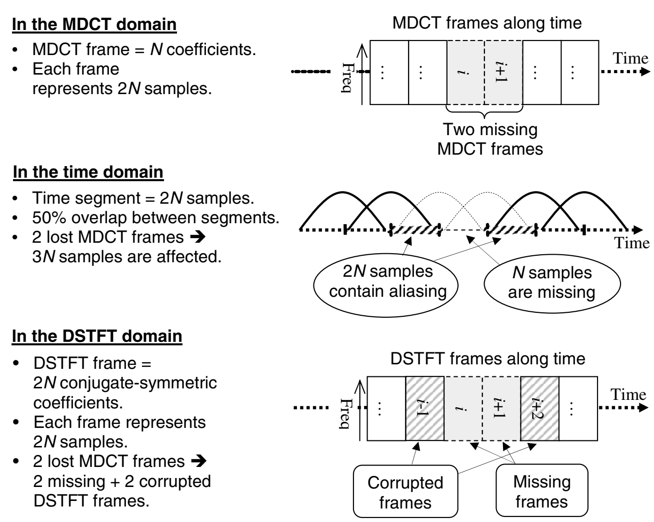

Fig. 2. Propagation of the error.

IV. MAPES-CM INTERPOLATION ALGORITHM

Both the GAPES and MAPES-CM algorithms reconstruct the lost samples as part of a process originally designed for spectral estimation of data sequences with missing samples. In that sense, they are considered as missing-data derivatives of the APES algorithm [15] for spectral estimation. As opposed to other missing-data reconstruction algorithms, such as the one used in [6], these algorithms don't assume parametric modelling of the signal and are suitable for both real and complex signals. Compared to each other, MAPES-CM is a more recent version that has all of GAPES's advantages at lower complexity, and can handle more loss patterns.

Before describing the missing-data reconstruction algorithm, let us first refer to the problem of spectral estimation of a complete-data sequence, i.e., the APES algorithm: Let $\{x_n\}_{n=0}^{P-1} \in \mathbb{C}^P$ denote a discrete-time data sequence of length P. We wish to estimate the spectral component at frequency ω_0, denoted $\alpha(\omega_0)$, of this data sequence. For the frequency of interest, ω_0, we model x_n as:

$$x_n = \alpha(\omega_0) \cdot e^{j\omega_0 n} + e_n(\omega_0) \tag{3}$$

where $e_n(\omega_0)$ denotes the residual term which includes the un-modelled noise and the interference from frequencies **other** than ω_0.

This problem has two interpretations: The *adaptive filter-bank* interpretation leads later to the derivation of GAPES. Since it was already discussed in [8] we shall not repeat it. The *ML-estimator* interpretation leads to the derivation of MAPES-CM and will be described next: Partition the data vector $\underline{x} = [x_0, x_1, \ldots, x_{P-1}]^T$ into L overlapping sub-vectors (data snapshots) of size $M \times 1$ (where $L = P - M + 1$) with the following shifted structure:

$$\underline{x}_l \triangleq [x_l, x_{l+1}, \ldots, x_{l+M-1}]^T \quad , l = 0, \ldots, L-1 \tag{4}$$

where $[\cdot]^T$ denotes transpose. Then, considering the data model in (3), the lth sub-vector, $\underline{x}_l$ can be written as:

$$\underline{x}_l = \alpha(\omega_0) \cdot \underline{a}(\omega_0) \cdot e^{j\omega_0 n} + \underline{e}_l(\omega_0) \tag{5}$$

where $\underline{a}(\omega_0) \triangleq [1, e^{j\omega_0}, \ldots, e^{j\omega_0(M-1)}]^T$ and, similar to (4), $\underline{e}_l(\omega_0)$ are the data snapshots of the residual term.

Next, we assume that $\{\underline{e}_l(\omega_0)\}_{l=0}^{L-1}$ are zero-mean circularly symmetric complex Gaussian random vectors that are statistically independent of each other and that have the same unknown covariance matrix:

$$\mathbf{Q}(\omega_0) = E[\underline{e}_l(\omega_0) \cdot \underline{e}_l^H(\omega_0)] \tag{6}$$

Note that since these vectors contain overlapping data, they are obviously not statistically independent of each other, hence APES is not an exact ML estimator, but only an approximate one.

Under these assumptions, we determine the ML estimator of the $\{\underline{y}_l\}_{l=0}^{L-1}$ vectors as:

$$\max_{\mathbf{Q}(\omega_0), \alpha(\omega_0)} \quad \log\left(Pr\left\{\{\underline{y}_l\}_{l=0}^{L-1} | \alpha(\omega_0), \mathbf{Q}(\omega_0)\right\}\right) \tag{7}$$

Which, by substituting conditional probability of L uncorrelated Gaussian random vectors of length M, converges to:

$$\max_{\mathbf{Q}(\omega_0), \alpha(\omega_0)} \quad \begin{aligned} & -\ln|\mathbf{Q}(\omega_0)| \\ & -\frac{1}{L}\sum_{l=0}^{L-1}\left[\underline{y}_l - \alpha(\omega_0) \cdot \underline{a}(\omega_0) \cdot e^{j\omega_0 l}\right]^H \\ & \cdot \mathbf{Q}^{-1}(\omega_0)\left[\underline{y}_l - \alpha(\omega_0) \cdot \underline{a}(\omega_0) \cdot e^{j\omega_0 l}\right] \end{aligned} \tag{8}$$

By solving this maximization problem [10] we obtain the estimated value of the spectral coefficient, $\hat{\alpha}(\omega_0)$.

Next, we describe the algorithm for the missing-data case: Let $\{x_n\}_{n=0}^{P-1} \in \mathbb{C}^P$ denote again a discrete-time data sequence of length P, only this time some of the samples are missing. Let $\underline{x}_m$ denote the vector of P_m missing samples ($P_m < P$).

The MAPES-CM algorithm adapts the spectral estimation problem to the case of missing samples, by using an extended version of the ML criterion in (7):

$$\max_{\underline{x}_m, \{\mathbf{Q}(\omega_k),\alpha(\omega_k)\}_{k=0}^{K-1}} \sum_{k=0}^{K-1} \log\left(Pr\left\{\{\underline{y}_l\}_{l=0}^{L-1}|\alpha(\omega_k),\mathbf{Q}(\omega_k)\right\}\right) \tag{9}$$

where $\{\omega_k\}_{k=0}^{K-1}$ is a pre-defined frequency grid.

After substituting the corresponding expressions, we get an extended version of (8), where, for convenience, we use a short notation: α_k instead of $\alpha(\omega_k)$, etc.:

$$\max_{\underline{x}_m, \{\mathbf{Q}_k,\alpha_k\}_{k=0}^{K-1}} \sum_{k=0}^{K-1} \Big\{ -\ln|\mathbf{Q}_k| - \frac{1}{L}\sum_{l=0}^{L-1}\left[\underline{y}_l - \alpha_k \cdot \underline{a}_k \cdot e^{j\omega_k l}\right]^H \cdot \mathbf{Q}_k^{-1}\left[\underline{y}_l - \alpha_k \cdot \underline{a}_k \cdot e^{j\omega_k l}\right]\Big\} \tag{10}$$

The solution to this new maximization problem is given by an iterative algorithm [10], where each iteration has two steps:

1) Solve the problem with respect to $\{\alpha_k, \mathbf{Q}_k\}$, assuming all data samples are available, by applying APES.
2) Solve the problem with respect to the missing samples, $\underline{x}_m$, assuming that the spectral parameters are available.

Finally, regarding the missing-data case, there is the issue of initialization: In the case of GAPES, a special initialization was needed [8]. However, in the case of MAPES-CM this is not required: substituting zero values instead of the missing data prior to the first iteration is enough.

V. Concealment Algorithm

This section includes a short description of the proposed algorithm that was already described in detail in [8]. Here, we focus on the practical aspects of replacing the use of GAPES with MAPES-CM.

As was already mentioned, the proposed solution is receiver-based, and hence it requires adding a concealment block to the decoder. For example, a block diagram of an MP3 decoder containing such a block, is shown in Fig 3:

Every new MP3 packet is decoded up to the MDCT level (i.e., de-quantized) resulting in two MDCT frames. The P most recent frames, along with their associated data (i.e., each frame's window type) are stored in a buffer, which is later used as the basis for the interpolation. If a packet is lost, its corresponding MDCT values are set to zero and a flag is raised, indicating that this frame is actually missing. The window types of the missing frames are determined so that they comply with the window types of neighboring frames. Then, the next frame to be played (according to the system's delay) is copied from the buffer and decoded into waveform samples. In the case where a particular frame is missing, we estimate its MDCT coefficients before continuing the decoding process. Due to packet loss, several MDCT frames in the buffer may be missing, so in order to reduce the computational overhead that results from activating the concealment block for each missing frame separately, usually several missing frames are concealed together. Hence, we refer to a concealment of one or more MDCT frames at once as a *concealment session*.

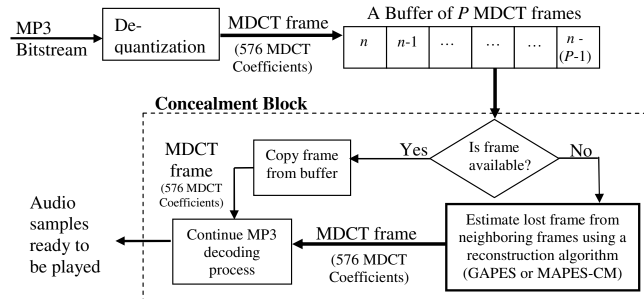

Fig. 3. An MP3 decoder including the concealment block.

The process which takes place in a concealment session, is as follows: First, all the MDCT frames in the buffer are converted to the DSTFT domain using the efficient procedure described in section III. In the DSTFT domain, the coefficients at each frequency bin along the time axis are considered as an independent complex signal, with missing samples. The data is reconstructed by applying a single iteration of MAPES-CM, to each signal, separately. When the reconstruction is completed, the reconstructed DSTFT frames are converted back to the MDCT domain. Then, in order to improve the start position of the next iteration, the reconstructed data is merged with the MDCT frames closest to the loss, by converting the reconstructed MDCT frames and their closest neighbors from each side, to the DSTFT domain. The process above is iterated until the difference between consecutive reconstructions becomes small. Then, the reconstructed MDCT frames are used instead of the lost ones and the MP3 decoding process continues.

A. Benefits of using MAPES-CM instead of GAPES

When using GAPES for the interpolation, its special initialization requires in some of the cases to limit the buffer's size to a selected area that contains a loss pattern that GAPES can handle. MAPES-CM doesn't need that. Hence, some of the overhead caused by activating the concealment process can be saved, since more losses can be concealed by single session. Also, since GAPES is an adaptive-filtering based method, it can deal better with consecutively lost samples (i.e., gaps) than with scattered losses. MAPES-CM, on the other hand, should work well for both gaps and arbitrary loss patterns.

Regarding complexity, when compared under the conditions defined by our application, it turns out that MAPES requires less complexity. In an analysis given in [14], taking a typical case of our application as an example, the ratio between GAPES's multiplications number per iteration and the number required by MAPES-CM, is:

$$\frac{\mathbf{GAPES}}{\mathbf{MAPES-CM}} \simeq \frac{29P + 48P_m^2}{35P + 6P_m}$$

where P is the buffer's length and P_m is the number of missing frames. So, for example, for $P = 16$ and $P_m = 4$, GAPES requires twice the number of multiplications MAPES-CM needs. The differences become smaller when $P_m << P$.

VI. Results

This section reports the results of two tests: The first test evaluates the quality of the proposed concealment algorithm, by comparing it with two previously reported algorithms: *packet repetition*, suggested in the MP3 standard [4], and statistical interpolation (SI) [6]. The second test compares the algorithm's performance when using GAPES versus using MAPES-CM. Since objective measures do not reflect the sensation created in a human listener, we had to use subjective listening tests. The tests were carried out by informal listening, using inexperienced listeners with normal hearing, in the age range of 24-35 years: 16 listeners in the first test and 8 in the second. Each of the listeners was asked to compare pairs of audio files, where the packet losses in each file were concealed by a different method, and to decide which of the two he, or she, prefers. Five different audio files of different types were used in the tests: Classic, Pop and Jazz music. All the files are stereo signals ,15-17 seconds long, sampled at 44.1 kHz and coded by the LAME MP3 [16] encoder at a bit-rate of 128 kbps per channel. The methods were tested for 10%, 20% and 30% loss rates with random loss patterns.

Table II shows listeners preferences by averaging over all their votes. The numbers clearly show that the proposed algorithm performs better than the two previously reported methods. Moreover, when comparing the concealed 10% loss rate to the uncorrupted MP3 decoded signal, the proposed solution performed so well, that some of the listeners confused the concealed signal with the original coded one. In this test GAPES was used as the interpolation algorithm.

TABLE II

Comparative test results: proposed algorithm vs. previously reported works.

	Distribution of votes					
	Proposed Solution vs. Repetition		Proposed Solution vs. SI		Proposed Solution vs. Uncorrupted Original	
Loss Rate	Proposed Solution	Repetition	Proposed Solution	SI	Proposed Solution	Original
10%	**88.75%**	11.25%	**97.5%**	2.5%	18.75%	**81.25%**
20%	**95%**	5%	**100%**	0%		
30%	**88.75%**	11.25%	**97.5%**	2.5%		

TABLE III

Comparative test results: GAPES vs. MAPES-CM.

	Distribution of votes	
Loss Rate	GAPES	MAPES
10%	**75%**	25%
20%	**72.5%**	27.5%
30%	**65%**	35%

Table III compares the two alternatives of activating the proposed scheme, using either the GAPES or the MAPES-CM interpolation algorithms. It's important to note that the listeners who participated in this particular test reported that it was much harder than the previous one, since in many of the cases the results of the two methods sounded very much alike. Still, the results indicate that GAPES, which is more complex, performs slightly better than MAPES-CM, where the differences get smaller as the loss rate increases.

VII. Conclusion

We have introduced a new packet loss concealment algorithm for wide-band audio signals encoded by MPEG audio coders, based on using the GAPES and MAPES-CM algorithms in the DSTFT domain. We used comparative informal listening tests to evaluate the algorithm's performance at 10% - 30% loss rates with different music types, and obtained that the proposed algorithm performs better than two previously reported algorithms: packet repetition [4] and statistical interpolation [6]. Between the two alternative for activating the algorithm, MAPES-CM has the advantage of lower complexity, however at the expense of a slight degradation in the quality of the concealed audio signal.

References

[1] X. S. B.W. Wah and D. Lin, "A survey of error-concealment schemes for real-time audio and video transmissions over the internet," in *International Symposium on Multimedia Software Engineering. Proceedings.*, December 2000, pp. 17 – 24.

[2] O. H. C. Perkins and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 15, no. 5, pp. 40–48, 1998.

[3] O. D.J.Goodman, G.B.Lockhart and W.Wong, "Waveform substitution techniques for recovering missing speech segments in packet voice communications," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 6, pp. 1440–1448, 1986.

[4] "Mpeg-1: Coding of moving pictures and associated audio for digital storage media at up to 1.5 mbit/s, part 3: Audio," *International Standard IS 11172-3, ISO/IEC JTC1/SC29 WG11, 1992.*

[5] P. Lauber and R. Sperschneider, "Error concealment for compressed digital audio," in *AES 111th convention*, NY, September 2001, pp. 1–11.

[6] S. Quackenbush and P. Driessen, "Error mitigation in mpeg-audio packet communication systems," in *AES 115th convention*, NY, October 2003, pp. 1–11.

[7] J. Lindblom and P. Hedelin, "Packet loss concealment based on sinusoidal extrapolation," in *ICASSP '02. Proceedings.*, May 2002, pp. 173–176.

[8] H. Ofir and D. Malah, "Packet loss concealment for audio streaming based on the gapes algorithm," in *118th AES Convention*, Barcelona, May 2005, pp. 1–19.

[9] P. Stoica and E. Larsson, "Adaptive filter-bank approach to restoration and spectral analysis of gapped data," *The Astronomical Journal by the American Astronomical Society*, vol. 120, pp. 2163–2173, 2000.

[10] J. L. Y. Wang and P. Stoica, "Two-dimensional nonparametric spectral analysis in the missing data case," in *ICASSP '05. Proceedings.*, March 2005, pp. 397–400.

[11] D. Pan, "A tutorial on mpeg/audio compression," *IEEE Multimedia*, vol. 2, no. 2, pp. 60–74, 1995.

[12] Y. Wang and M. Vilermo, "Modified discrete cosine transform-its implications for audio coding and error concealment," *AES Journal*, vol. 51, no. 1/2, pp. 52–61, 2003.

[13] J. Tribolet, "Frequency domain coding of speech," *IEEE Trans. On ASSP*, vol. 27, no. 5, pp. 512–530, 1979.

[14] H. Ofir, "Packet loss concealment for audio streaming," M.Sc. thesis, Technion IIT, 2006, available at: http://www-sipl.technion.ac.il/new/Research/Publications/Graduates/Hadas_Ofir/Hadas_MSc_thesis.pdf.

[15] P. S. H. Li and J. Li, "A new derivation of the apes filter," *IEEE Signal Processing Letters*, vol. 6, no. 8, pp. 205–206, 1999.

[16] *The LAME Project: An open source of an MP3 coder: http://lame.sourceforge.net.*