International Journal of Computer Science and Applications
© Technomathematics Research Foundation
Vol. 10 No. 1, pp. 1 - 10, 2013

AUDIO WATERMARKING BASED ON ECHO HIDING WITH ZERO ERROR PROBABILITY

VALERY KORZHIK
State University of Telecommunications, St. Petersburg, Russia
val-korzhik@yandex.ru

GUILLERMO MORALES-LUNA
Computer Science, CINVESTAV-IPN, Mexico City, Mexico
gmorales@cs.cinvestav.mx
http://delta.cs.cinvestav.mx/∼gmorales

IVAN FEDYANIN
State University of Telecommunications, St. Petersburg, Russia
ivan.a.fedyanin@gmail.com

A new audio watermarking technique is proposed. It executes wet paper codes in embedding procedure of watermarks into audio cover objects using echo signals. Technique of echo signals results in a high quality of audio covers after embedding and robustness against natural signal transforms. Wet paper codes usage provides zero extraction error probability even for blind decoder at the cost of a very negligible embedding rate degradation. The technique of cepstrum analysis is used for hidden message extraction along with its parameter optimization.

1. Introduction

Digital watermarking(WM) is an important technique for copyright protection of digital media content including audio files [Cox *et al* (2002)]. The WM goal is to design such systems which are robust against all possible deliberate attacks. Apparently, such attacks have to maintain an acceptable audio signal fidelity while making the embedded WM disabled with respect to hidden message extraction. This problem is very hard and its solution is not found so far for all possible attacks, though there exist such situations where a resistance of WM to deliberate attacks is not required. In this case, WM plays the role of some additional imperceptible information (e.g. possible contacts with the files owner). Nevertheless, a WM system should be resistant to any natural transforms as channel filtering, channel noise and MPEG compression. Many novel techniques have been proposed for WM audio systems. For instance, techniques based on masking [Boney *et al* (1996)], phase coding [Bender *et al* (1996)], phase modulation [Nishimura *et al* (2001)], echo hiding and reverberation [Gruhl *et al* (1996)], among others.

Since echo is just a delayed copy of the original sound with decreased amplitude, it does not sound unnatural; moreover echo is robust to natural transforms. We claim that both imperceptibility and robustness to natural sound file transforms are getting practically "for free" because "echo" does not affect on sound comprehension under some echo parameter restrictions. It is worth to note that some extension of echo hiding, known as reverberation, has the same and even better properties, but we will consider only simple echo based embedding in the current paper to make the investigation of this topic as complete as possible.

2. Watermark embedding and extraction procedures

The embedding procedure is:

$$\mathbf{x} = \mathbf{S} * \mathbf{h}_b \tag{1}$$

where $\mathbf{x} = (x(n))_{n=0}^{N-1}$ is the watermarked signal after embedding, $\mathbf{S} = (S(n))_{n=0}^{N-1}$ is the input audio signal (say in wav format), and for $n = 0, \ldots, N-1$

$$h_b(n) = \delta(n) + \alpha_b \delta(n - \tau_b), \text{ where } \delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

$*$ is the operation of convolution, and N is the number of samples in which one bit $b \in \{0, 1\}$ of WM is embedded.

We have chosen a constant α_b and different delays τ_0 and τ_1 because the use of a constant τ_b and two values $\alpha_0 = 0$ and $\alpha_1 > 0$ results in some problems in the choice of the threshold for the extraction scheme. At a single glance, it seems to be very natural to use the correlation receiver directly in the sample domain, namely:

$$\tilde{b} = \arg\max_b \sum_{n=1}^{N} x(n) \cdot x(n - \tau_b) \tag{2}$$

but the decision rule (2) results in a very large bit error probability as a consequence of a non-zero correlation between the delay on the interval τ_b sound samples $\mathbf{S}$. Therefore, it seems to be much better to exploit the *cepstrum* transform of sound signals [Cvejic *et al* (2007); Oppenheim *et al* (1989)].

There are two notions of cepstrum: *complex cepstrum* $\mathbf{x}_c = (x_c(n))_{n=0}^{N-1}$ and *real cepstrum* $\mathbf{x}_r = (x_r(n))_{n=0}^{N-1}$, determined as follows: $\forall n = 0, \ldots, N-1$

$$x_c(n) = \frac{1}{N} \sum_{k=0}^{N-1} \left[\log|X(k)| + j\Theta(k)\right] e^{2\pi j \frac{nk}{N}} \tag{3}$$

$$x_r(n) = \frac{1}{N} \sum_{k=0}^{N-1} \log|X(k)| e^{2\pi j \frac{nk}{N}} \tag{4}$$

where $\forall k = 0, \ldots, N-1$, $X(k) = \sum_{n=0}^{N-1} x(n) e^{-2\pi j \frac{nk}{N}}$, $|X(k)|$ is the module of $X(k)$, $\Theta(k)$ is the argument of the complex number $X(k)$ and $j = \sqrt{-1}$. For both

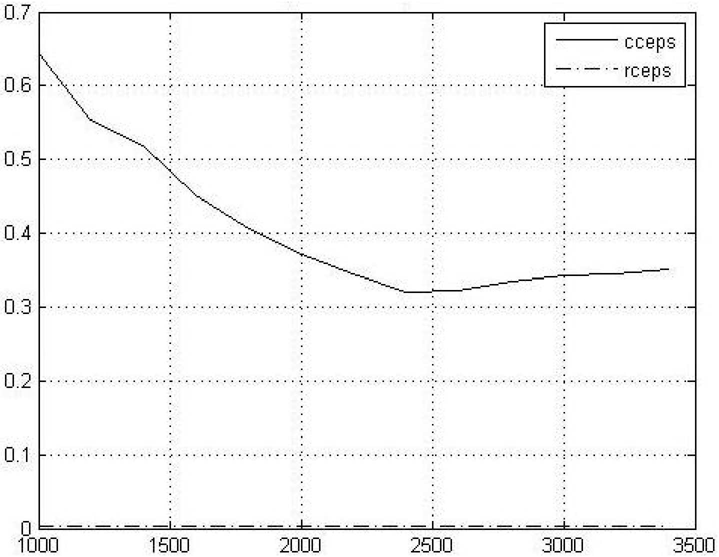

Fig. 1. The relative error Δ versus the number of appended zeros. The solid line plots the complex cepstrum, while the dotted line plots the real cepstrum.

cepstrums, the following identity holds [Cvejic *et al* (2007); Oppenheim *et al* (1989); Oppenheim *et al* (1968)]:

$$\tilde{x}(n) = \tilde{S}(n) + \tilde{h}_b(n), n = 1, 2, ..., N \tag{5}$$

where the tilde denotes the cepstrum transform of the corresponding signals. But in fact, (5) holds only for a very large N whereas in order to embed many bits into audio signal, N should be very limited, hence (5) is just an approximate equality. The relative error carried by (5) can be expressed as:

$$\Delta = \frac{\sum_n \left(\tilde{x}(n) - (\tilde{S}(n) + \tilde{h}_b(n))\right)^2}{\sum_n \left(\tilde{S}(n)\right)^2} \tag{6}$$

where $\tilde{\mathbf{S}} = \left(\tilde{S}(n)\right)_{n=0}^{N-1}$ is modeled as a white Gaussian noise and $\tilde{x} = \widetilde{\mathbf{S} * \mathbf{h}_b}$.

The simulation results of Δ, averaged on the ensemble of Gaussian signals, for $N_0 = 1000$, $\tau_b = 50$ versus the number of zeroes appended by the signal cepstrum transforms are shown in Fig. 1, where the length of input Gaussian noise is $N_0 = 1000$, and the delay is $\tau_b = 50$. From there, it can be seen that the relative error for both cepstrums cannot equal zero just by increasing the appended zeroes and that real cepstrum is superior than the complex within this regard. Nevertheless, let us assume that (5) holds approximately. A natural decision rule does follow:

$$\sum_{n=0}^{N-1} \tilde{x}(n) \cdot \tilde{h}_0(n) \underset{b:1}{\overset{b:0}{\gtrless}} \sum_{n=0}^{N-1} \tilde{x}(n) \cdot \tilde{h}_1(n) \tag{7}$$

where the symbol $\underset{b:1}{\overset{b:0}{\gtrless}}$ denotes that if the inequality $>$ holds then the embedded bit is $b = 0$ while if $<$ holds then $b = 1$. In fact let the input signal $\mathbf{S} = (S(n))_{n=0}^{N-1}$ be, by the moment, a Gaussian white noise, then [Proakis (2001)] the optimal likelihood decision rule for the model (5) is just (7). The bit error probability p is $p = 1 - F\left(\sqrt{\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}\tilde{h}_0^2(n)}\right)$where $\sigma^2 = \mathrm{Var}\,(\tilde{\mathbf{S}})$ and $F : x \mapsto F(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{t^2}{2}}\,dt$, under the conditions:

$$\sum_{n=0}^{N-1}\tilde{h}_0^2(n) \approx \sum_{n=0}^{N-1}\tilde{h}_1^2(n) \quad \text{and} \quad \sum_{n=0}^{N-1}\tilde{h}_0(n)\cdot\tilde{h}_1(n) \approx 0.$$

But the “trick” with the channel model (1) and with the use of cepstrum transform is that the decision rule (7) can be very far from the optimal one and there exists a much better decision rule based on *subintervals*, namely:

$$\sum_{k=1}^{L}\sum_{n=0}^{N_0-1}\tilde{x}_k(n)\cdot\tilde{h}_0(n) \underset{b:1}{\overset{b:0}{\gtrless}} \sum_{k=1}^{L}\sum_{n=0}^{N_0-1}\tilde{x}_k(n)\cdot\tilde{h}_1(n) \tag{8}$$

where $\tilde{\mathbf{x}}_k = (\tilde{x}_k(n))_{n=0}^{N-1}$ is the cepstrum of the signal $\mathbf{x}$ on the k-th sample subinterval, N_0 is the number of samples on each subinterval, and L is the number of subintervals. It is rather strange that, within conventional communication theory, we are able to improve the decision rule by fractioning the original interval on subintervals, but this is due to the property that $\sum_{n\in I}\tilde{h}_i^2(n)$, $i \in \{0,1\}$, does not depend on the interval length provided that its length embraces the cepstrum pulse response $\left(\tilde{h}_i^2(n)\right)_{n\in I}$ duration. Then if we assume that the cepstrums $\tilde{\mathbf{x}}_k$ are mutually independent on the different subintervals, we may expect that the signal-to-noise ratio will increase with the increasing of the number L of subintervals. The probability of bit error is: $p = 1 - F\left(\sqrt{\frac{L}{2\sigma^2}\sum_{n=0}^{N-1}\tilde{h}_0^2(n)}\right)$but in practice this does not hold because not all the required conditions in its proof are fulfilled.

If the use of complex and real cepstrum in (8) are compared, then it may be concluded that the real cepstrum is superior to the complex one. This fact has been mentioned also in [Cvejic *et al* (2007)] although without any justification. From the expressions of the error probabilities p, the probability of the bit error depends on the variance of the input cepstrum $\tilde{\mathbf{S}}$ while it is much greater for the case of complex cepstrum. This last fact can be explained by the property of *phase unwrapping* that is required for the complex cepstrum. It is worth to note that the complex log is a multiply valued function, because its imaginary part has infinite number of values differing on 2π. In order to remove this uncertainty, it is common to calculate an imaginary part modulo 2π. But it results in turn in a breaking of this function and, in order to remove this unwanted property, phase unwrapping should be used [Childers *et al* (1977)]. In Fig. 2 we display (a) the modulo 2π waveforms phase and (b) the unwrapped phase.

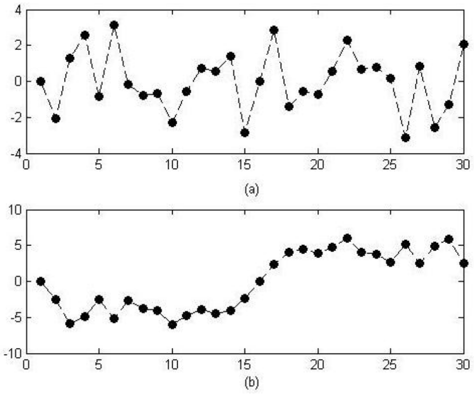

Fig. 2. Original modulo 2π phase (a), and phase after unwrapping procedure(b)

Since the probability of bit error is much lesser in the case of using real cepstrum for the extraction procedure, we will consider in the sequel only a real cepstrum implementation.

3. Computer modeling and estimation of extraction error probability

We consider simple echo-based watermarking by (1) where the cover messages(CM) $\mathbf{S} = (S(n))_{n=0}^{N-1}$ are different musical files in format `wav` with durations between 3 and 6 minutes. The delays τ_0 and τ_1 should be optimized in order to provide, on the one hand, maximum embedding rate and, on the other hand, an acceptable bit error probability after extraction by the rule (8). Our experiment shows that the optimal values are close to 27 and 32 samples respectively which correspond to delays 0.61 and 0.73 ms with frequency of samples 44.1 kHz for the `wav` format.

The parameter which has the main impact on CM quality after embedding and WM robustness is the amplitude of the echo α_b. Therefore we vary this value and estimate the CM quality by experts, assigning grades 5 for excellent, 4 for good, 3 for satisfactory, and 2 for unsatisfactory.

The type of "window" where one bit is embedded plays an important role in the extraction efficiency. There are known different types of windows (exponential, Hamming, rectangular and Hann). Our experiments showed that the best results can be achieved with the Hann window although difference with Hamming window is not too large.

The important parameter that should be optimized is the number of subintervals L (see eq. (8)). In Table 1, the results of simulation for different musical files presented, under the condition that $\tau_0 = 27$, $\tau_1 = 32$ and a Hann window is used.

We vary the echo amplitude α, the number of samples N_0 in which one bit is embedded and we optimize the number of subintervals L in order to provide minimal

Table 1. The bit error probabilities p and CM quality Q (in 1-5 grades) depending on the optimal number of subintervals L_{opt}, echo amplitude α, and the number of samples N_0 in which one bit is embedded. **ER**: The embedding rate (bit/sec). The delays are $\tau_0 = 27$, $\tau_1 = 32$

Name of file	N_0	**ER**	L_{opt}	α	Q	p,%
music1.wav (classical)	4410	10	3	0.2	5	0.04
				0.1	5	1.68
	980	45	1	0.3	4.8	0.06
				0.25	4.9	0.16
	294	150	1	0.45	2.8	0.09
				0.4	3.5	0.32
music2.wav (hard rock)	4410	10	3	0.2	5	0.22
				0.1	5	3.58
	980	45	1	0.3	4.9	0.07
				0.25	5	0.33
	294	150	1	0.45	3.3	0.09
				0.4	3.5	0.31
music3.wav (rock)	4410	10	3	0.2	5	0.07
				0.1	5	4.78
	980	45	1	0.3	4.8	0.14
				0.25	4.9	0.59
	294	150	1	0.45	2.8	0.17
				0.4	3.5	0.43
music4.wav (pop)	4410	10	3	0.2	5	0.17
				0.1	5	3.71
	980	45	1	0.3	4.8	0.25
				0.25	4.9	0.7
	294	150	1	0.45	2.7	0.15
				0.4	3.5	0.36
music5.wav (jazz)	4410	10	3	0.2	5	0.75
				0.1	5	7.85
	980	45	1	0.3	4.8	0.42
				0.25	4.9	1.04
	294	150	1	0.45	2.7	0.67
				0.4	3.5	1.2

probability of bit error.

The experimental fact that the optimal number of subintervals L is between 1 and 3 contradicts the eq. (**??**) because it entails that in order to minimize p, L should be as large as possible but always less than $N/\max(\tau_0, \tau_1)$(otherwise each subinterval would be less than the pulse response of the echo channel). This contradiction can be explained by a violation of the relation (5), by the presence of some statistical dependency between the random sequences $\tilde{\mathbf{x}}_k$ and their non-Gaussian probability distributions. Although the errors can be decreased by the use of forward error-correcting codes (FEC) (say convolutional or low parity density codes [Moreira *et al* (2006)]) there is an opportunity to decrease the error probability quite significantly because the interference in bit extraction is just CM (musical files) which are exactly known in the embedding procedure. We consider such approach in the following Section.

4. The embedding algorithm with use of wet paper codes

The wet paper codes (WPC) [Fridrich *et al* (2006)] were used initially in the embedding procedure for a special stegosystem known as *perturbed steganography.* Later, they were used in an embedding procedure with binary images and with other applications [Fridrich *et al* (2005)]. We propose a rather *unusual application of WPC* below. Let us recall some main concepts of WPC.

Let us denote by $\mathbf{m} = (m_\mu)_{\mu=1}^{M}$ the binary message string which should be embedded , and by $\mathbf{b} = (b_\kappa)_{\kappa=1}^{\tilde{N}}$ the binary string in which it is necessary to embed $\mathbf{m}$. Let $F \subseteq \{1, 2, ..., \tilde{N}\}$ be a subset of indexes in the set $\{1, 2, ..., \tilde{N}\}$ pointing the positions in which the embedding is allowed. This means that we can change bits only at positions in F during the embedding procedure. It is necessary to encode the string $\mathbf{m}$ into the string $\mathbf{b}$ in such a way to change just bits positions at F. Moreover, in the extraction procedure it is assumed that the subset F is unknown.

The embedding procedure is executed as follows: Initially an $M \times \tilde{N}$ binary matrix $\mathbf{H}$ has to be generated. (It can be considered either as a *stegokey* or as function of a stegokey). The encoded binary vector $\mathbf{b}'$ is $\mathbf{b}' = \mathbf{b} \oplus \boldsymbol{\nu}$ where $\boldsymbol{\nu}$ has $\tilde{N} - k$ zero positions i for $i \notin F$ and k unknown positions i for $i \in F$, where $k = |F|$ is the number of elements in the subset F.

Let us denote by $\tilde{\boldsymbol{\nu}}$ the binary vector of the length k that can be obtained from the vector $\boldsymbol{\nu}$ after removal of all zero positions in $\boldsymbol{\nu}$. Then $\boldsymbol{\nu}$ can be found if both $\tilde{\boldsymbol{\nu}}$ and F are known. In order to find $\tilde{\boldsymbol{\nu}}$ it is necessary to solve the linear system

$$\tilde{\mathbf{H}}\tilde{\boldsymbol{\nu}} = \mathbf{m} \oplus (\mathbf{H} \cdot \mathbf{b}) \tag{9}$$

where $\tilde{\mathbf{H}}$ is a submatrix of $\mathbf{H}$ obtained after a removal of all columns corresponding to zero elements of $\boldsymbol{\nu}$. The system (9) has an unique solution whenever

$$\text{rank } \tilde{\mathbf{H}} = M. \tag{10}$$

In practical applications the parameter M is not initially fixed and the matrix $\mathbf{H}$ is generated by rows until the condition (10) fails. If M' is the maximum number of rows for which (10) holds then it is possible to embed M' bits and this value M' is also embedded as a head of $\mathbf{m}$.

The decoding procedure (extraction of message $\boldsymbol{m}$) is performed quite simply:

$$\mathbf{m} = \mathbf{H} \cdot \mathbf{b}'. \tag{11}$$

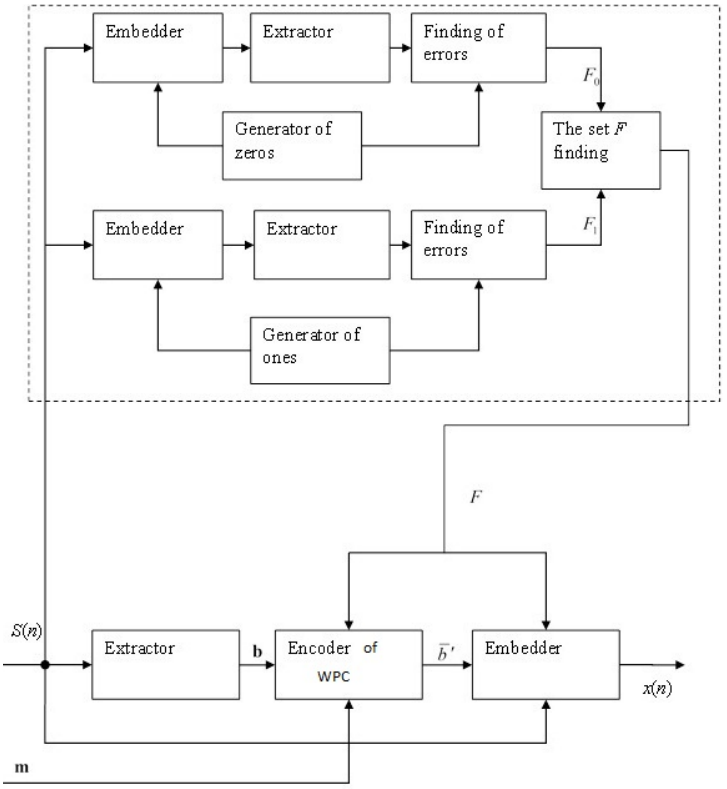

Fig. 3. The embedding algorithm with the use of WPC

Let us consider now how to implement the WPC in our case. The general idea is to embed bits only in such N_0-blocks where the interference $\left(\tilde{S}(n)\right)_{n=1}^{N_0}$ does not result in errors after extraction by the rule (8).

The embedding algorithm is presented in Fig. 3. As seen within this scheme, initially the N_0-blocks in which extractors produce errors during an embedding of both values zero and one are marked. Let F be the position subset where embedding is possible. In the following "embedding round", the message sequence **m** is encoded with the WPC and the echo embedding is performed for those N-blocks corresponding to the subset F where the errors after "virtual extraction" are absent. The extractor outputs at the input of encoder the bit string **b** decoded from audio file before embedding. Next this sequence **b** is encoded in line with the rule of WPC mentioned above.

This results in a zero bit error probability after the real extraction of the watermarked signal. The sacrifice within this approach is a decreasing of the embedding rate because some N-blocks are removed from the embedding process. The results of this method are shown in Table 2. There, it can be observed that the number T

Table 2. Simulation for a WPC-based embedding algorithm and some chosen parameters. The number of samples for an embedding of one bit is $N_0 = 980$, the delays corresponding to bit 0 and 1 are 25 and 30 respectively, the amplitude of embedding is $\alpha = 0.3$, the number of subintervals is $L = 1$, t is the potential number of embedded bits in audio file, d is the number of changeable bits in audio file, and T is number of embedded bits

Name of file	t	d	T	
			$\check{N} = 500$	$\check{N} = 1000$
music1.wav	10000	9989	9563	9781
music2.wav	12179	12156	11490	11727
music3.wav	12244	12185	11430	11685
music4.wav	8135	8093	7613	7786
music5.wav	9625	9512	8990	8994

of embedded bits is slightly less than the number t of bits that could be embedded (but with errors after extraction) without the use of WPC. T depends on the length of WPC. (The difference between d (the cardinality of F) and T can be explained by the fact that (10) does not hold for all code blocks).

5. Conclusion

Our first important contribution is in proposing the use of a WPC in order to provide zero bit error probability after extraction, together with the embedding algorithm suited for such a WPC application. (see Fig. 3). The use of a WPC decreases the embedding rate only on 6% in average. It seems to be better than the use of ordinary FEC codes in order to correct errors with a probability of 10^{-2}. However the use of a WPC is impossible if some errors occur just after embedding because the WPC results in an error extension.

Our second important contribution is in proving the fact that through the decision rule based on subintervals (see eq. (9)) a significant advantage is obtained in comparison with the decision rule based on a single echo interval (see eq. (7)) that is unusual by conventional communication theory. Moreover the number of subintervals should be optimized and this fact can be justified by a breaking of eq. (5) for echo-modulated audio signals (see (6) and Fig. 1) and significant correlation of audio signal samples. The simulation results show that for optimally chosen parameters of audio echo based watermarking it is possible to provide excellent quality of audio signal after embedding, an embedding rate in $[30, 45]$ bit/sec and the bit error probability after extraction close to 10^{-2}.

Also it has been proved that the use of real cepstrum is superior than the use of complex cepstrum in extraction procedure because it results in smaller bit error probability.

In the future we are going to investigate the use of syndrome-trellis codes in the audio watermarking scheme in order to decrease CM distortion and improve robustness.

Our second important contribution is in proposing the use of a WPC in order to provide zero bit error probability after extraction. The embedding algorithm suited for such a WPC application was proposed. (see Fig. 3).

The use of a WPC decreases the embedding rate only on 6% in average. It seems to be better than the use of ordinary FEC codes in order to correct errors with a probability of 10^{-2}. However the use of a WPC is impossible if some errors occur just after embedding because the WPC results in an error extension.

In the future we are going to investigate a combination of both WPC and FEC.

References

Bender, W., Gruhl, D., Morimoto, N., and Lu, A., Techniques for data hiding, *IBM Syst. J.*, vol. 35, pp. 313–336, September 1996.

Boney, L., Tewfik, A. H., and Hamdy, K. N., Digital watermarks for audio signals, in *ICMCS*, 1996, pp. 473–480.

Childers, D. G., Skinner, D. P., and Kemerait, R. C., The cepstrum: A guide to processing, *Proceedings of the IEEE*, vol. 65, pp. 1428–1443, 1977.

Cox, I. J., Miller, M. L., and Bloom, J. A., *Digital Watermarking.* Morgan Kaufman Publishers, 2002.

Cvejic, N. and Seppänen, T., *Digital Audio Watermarking Techniques and Technologies: Applications and Benchmarks.* Information Science Reference, Hershey, PA, USA, 2007.

Fridrich, J. J., Goljan, M., and Soukal, D., Wet paper codes with improved embedding efficiency, *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 1, pp. 102–110, 2006.

——, Perturbed quantization steganography, *Multimedia Syst.*, vol. 11, no. 2, pp. 98–107, 2005.

Gruhl, D., Lu, A., and Bender, W., Echo hiding, in *Information Hiding*, ser. Lecture Notes in Computer Science, R. J. Anderson, Ed., vol. 1174. Springer, 1996, pp. 293–315.

Moreira, J., and Farrell, P., *Essentials of error-control coding.* John Wiley & Sons, 2006.

Nishimura, R., Suzuki,M., and Suzuki, Y., Detection threshold of a periodic phase shift in music sound, in *Proc. International Congress on Acoustics, Rome, Italy, vol. IV*, 2001, pp. 36–37.

Oppenheim, A. V., and Schafer, R. W., *Discrete-time signal processing.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

Oppenheim, A., Schafer, R., and Stockham, T., Nonlinear filtering of multiplied and convolved signals, *Proceedings of the IEEE*, vol. 56, pp. 1264–1291, 1968.

Proakis, J., *Digital Communications, Fourth Edition.* Mc Graw Hill, 2001.