IEICE Electronics Express, Vol.6, No.14, 1057–1063

High capacity audio watermarking using FFT amplitude interpolation

Mehdi Fallahpour[a,b] **and David Megias**[c]

Universitat Oberta de Catalunya, Rambla del Poblenou, 156, Barcelona, Spain

a) *MFallahpour@UOC.edu*

b) *Fallahpour@gmail.com*

c) *DMegias@UOC.edu*

Abstract: An audio watermarking technique in the frequency domain which takes advantage of interpolation is proposed. Interpolated FFT samples are used to generate imperceptible marks. The experimental results show that the suggested method has very high capacity (about 3 kbps), without significant perceptual distortion (ODG about −0.5) and provides robustness against common audio signal processing such as echo, add noise, filtering, resampling and MPEG compression (MP3). Depending on the specific application, the tuning parameters could be selected adaptively to achieve even more capacity and better transparency.

Keywords: watermarking, FFT, spline interpolation, frequency domain

Classification: Science and engineering for electronics

References

[1] J. J. Garcia-Hernandez, M. Nakano-Miyatake and H. Perez-Meana, "Data hiding in audio signal using Rational Dither Modulation," *IEICE Electron. Express*, vol. 5, no. 7, pp. 217–222, 2008.

[2] H. Kang, K. Yamaguchi, B. Kurkoski, K. Yamaguchi, and K. Kobayashi, "Full-Index-Embedding Patchwork Algorithm for Audio Watermarking," *IEICE Trans. Inf. & Syst.*, vol. E91-D, no. 11, pp. 2731–2734, 2008.

[3] D. Megías, J. Herrera, and J. Minguillón, "Total Disclosure of the Embedding and Detection Algorithms in a Robust Digital Watermarking Scheme for Audio," *LNCS*, vol. 3783, pp. 427–440, 2005.

[4] S. Xiang, H. J. Kim, and J. Huang, "Audio watermarking robust against time-scale modification and MP3 compression," *Signal Processing*, vol. 88, pp. 2372–2387, 2008.

[5] M. Fan and H. Wang, "Chaos-based discrete fractional Sine transform domain audio watermarking scheme," *Comput. Electrical Engineering*, Elsevier, vol. 35, no. 3, pp. 506–516, May 2009.

[6] R. Fujimoto, M. Iwaki, and T. Kiryu, "A Method of High Bit-Rate Data Hiding in Music Using Spline Interpolation," *IIH-MSP*, pp. 11–14, 2006.

[7] A. Deshpande and K. M. M. Prabhu, "A substitution-by-interpolation algorithm for watermarking audio," *Signal Processing*, Elsevier, vol. 89, no. 2, pp. 218–225, 2009.

[8] C. J. Weinstein, *Programs for Digital Signal Processing*, IEEE Press, 1979.

DOI: 10.1587/elex.6.1057
Received June 19, 2009
Accepted June 29, 2009
Published July 25, 2009

[9] T. Thiede, W. C. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J. G. Beerens, C. Colomes, M. Keyhl, G. Stoll, K. Brandenburg, and B. Feiten, "PEAQ - The ITU Standard for Objective Measurement of Perceived Audio Quality," *J. AES*, vol. 48(1/2), pp. 3–29, 2000.
[10] [online] http://www.jamendo.com/en/album/7365
[11] [online] http://www.opticom.de/products/opera.html
[12] [online] http://wwwiti.cs.uni-magdeburg.de/~alang/smba.php

1 Introduction

With the broad use of information security applications and the rising growth of the watermarking schemes, various signal processing techniques are being used to improve audio watermarking methods. Audio watermarking methods exploit the insensitivity of the human auditory system (HAS) in various techniques such as embedding algorithms based on low-bit coding, echo, rational dither modulation [1], patchwork [2], Fourier transform [3, 5], wavelet transform [4] or spread spectrum and interpolation [6, 7].

Interpolation techniques are often designed to provide a good perceptual quality from known sample values [8]. In [6], an original audio signal is divided into distinct frames and then a secret bit is embedded in each frame by using spline interpolation. [7] proposes a spline interpolation-based watermarking scheme with more robustness against attacks than the one suggested in [6].

The aim of the proposed method is to develop a high-bit-rate audio watermarking technique with robustness against common attacks and good transparency. This algorithm is based on the difference between the original and the interpolated amplitudes of the FFT samples as obtained by spline interpolation. If the difference is lower than a given fraction of the interpolated value, it is selected for embedding secret information. To obtain the marked FFT samples, the interpolated value is changed according to the secret bit. The experimental results show that the method provides high data bit rate, about 3 kbps, with good perceptual transparency (ODG about -0.5) and robustness against common attacks. Better capacity and transparency may be achieved with appropriate tuning for specific applications.

2 Proposed method

Interpolation [8] is a technique of constructing new data points within the range of a discrete set of known data points. Linear interpolation is obtained by passing a straight line between two data points. Polynomial interpolation is the best known one-dimensional interpolation scheme. Its advantages consist of its simplicity of implementation and the good quality of the interpolants obtained from it. However, it has a relatively low performance. In the spline interpolation, the interpolation interval is divided into small subintervals and each of these is interpolated by using a third-degree polynomial. The main advantages of the spline interpolation, which is used in the

DOI: 10.1587/elex.6.1057
Received June 19, 2009
Accepted June 29, 2009
Published July 25, 2009

proposed algorithm, are its stability and calculation simplicity.

2.1 Embedding algorithm

We use some of the original FFT samples as source data in the coder and do not alter them. Hence, these original values can be used in decoder. In the algorithm, we use the odd FFT samples to generate the interpolated values of the even samples which are used for embedding the secret bits. In the receiver, the original values of the odd FFT samples and the interpolated even samples are the same as in the coder. The embedding steps are follows:

$k = 1;$
for $i = low_{band}$ **to** $high_{band}$
  if $\mathrm{mod}(i, 2) == 0$
    $e_i = f_i - I_i\,;$
    if $|e_i| > 2\alpha I_i$
      $f_i^{'} = f_i;$
    else if $b_k == 0$
      $f_i^{'} = I_i; k = k + 1;$
    else if $\{(b_k == 1)$ **and** $(e_i \geq 0)\}$
      $f_i^{'} = I_i(1 + \alpha); k = k + 1;$
    else
      $f_i^{'} = I_i(1 - \alpha); k = k + 1;$
    end;
  end;
end;

where f_i is the magnitude of the i^{th} sample of the FFT spectrum, low_{band} *and* $high_{band}$ are the lower and higher limits of a selected band for embedding secret information, I_i is the interpolated value of f_i, α is a threshold, b_k is the k^{th} bit of secret bit stream, and $f_i^{'}$ is the marked value of f_i.

Adaptation to capacity, transparency and robustness against attacks is the most relevant advantage of the proposed method. *i.e.* the results are altered by changing the selected frequency band and the threshold. The selected frequency band is the area which is used for embedding secret information. Another main parameter of this algorithm is threshold α, which defines the maximum allowed ratio of interpolation error (difference between original FFT sample and its interpolant) to the interpolated sample for embedding. The interpolated sample can be used for embedding secret information when the ratio between the interpolated error to the interpolated sample is less than the threshold. The frequency band 0.5–5 kHz and $\alpha = 0.3$ are suitable fixed parameters which have been selected for this algorithm after different experiments. However, depending on the application, these parameters could be modified to obtain better capacity and/or distortion.

The FFT samples with an amplitude value lower than one (when the original time-domain audio signal is normalised between -1 and 1) are sensitive to a few attacks such as MP3 compression. Thus, interpolation is not used for even FFT bins with value less than one. These FFT amplitudes are fixed

DOI: 10.1587/elex.6.1057
Received June 19, 2009
Accepted June 29, 2009
Published July 25, 2009

to 0 instead and, if the corresponding embedding bit is 0, it is not changed. Otherwise, if the secret bit is one, the amplitude is increased to 0.5.

The effect of some attacks [12] on the embedding scheme has been analysed. For example, the echo attack is one of the most difficult ones to survive. Fig. 1 (a) shows the FFT spectrum of the marked signal and Fig. 1 (b) displays the echo-attacked spectrum. Fig. 1 (c) illustrates the difference between marked and echo-attacked spectra. This spectrum shows how the echo-10 attack repeats the signal after 10 samples and destroys the signal. In this case, we can use three secure spaces which are indicated by red rectangles for which the difference is very low and the echo attack does not destroy the FFT samples. Fig. 1 (d)–(f) shows the marked, attacked, error of attack and the secure embedding area (the region for which the difference between the marked and attacked signals is low) for the BassBoost attack. To find a secure area, we should consider the ratio between the error and the attacked samples, not the amplitude of the error, since we use the ratio to extract information. Different applications have different requirements. In copyright protection, robustness is more relevant than capacity, whereas for transmission of secret information it is the opposite. Thus, depending on the application, the frequency band and the threshold could be changed. In this paper, we have used a fixed frequency band and threshold for various natural audio files and different attacks to show the widespread use of the proposed method.

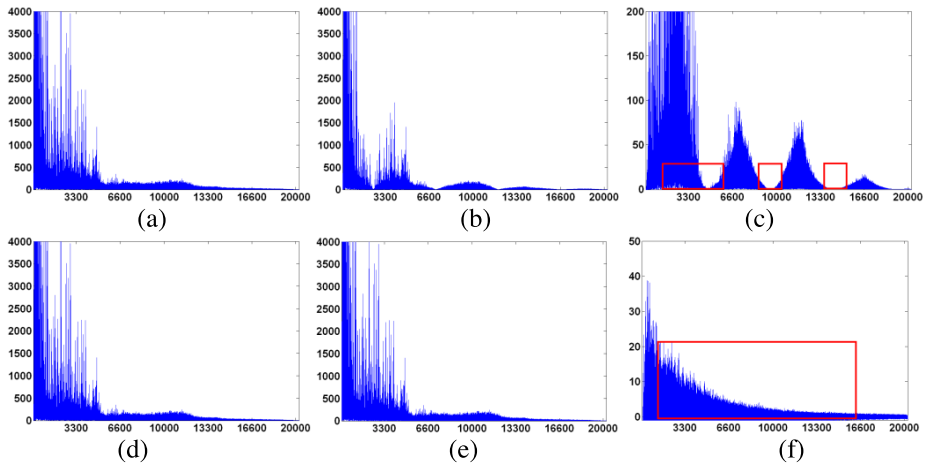

Fig. 1. FFT spectrum of (a), (d) marked audio signal (b) echo-10 attacked (c) error of echo-10 attacked (e) BassBoost attacked (f) error of BassBoost attacked

2.2 Extraction algorithm

As mentioned for the embedding steps, in the receiver the original values of odd FFT samples and the interpolated value of the even samples, I_i, are the same as in the coder. The following algorithm describes how to obtain the secret bit sequence, b'_k, by using the marked FFT sample, f'_i, the frequency bands and the threshold, α, as input values.

DOI: 10.1587/elex.6.1057
Received June 19, 2009
Accepted June 29, 2009
Published July 25, 2009

$k = 1$;
for $i = low_{band}$ **to** $high_{band}$
  if $\mathrm{mod}(i, 2) == 0$
    $e_i^{'} = f_i^{'} - I_i$;
    if $|e_i^{'}| < 0.5\alpha I_i$
      $b_k^{'} = 0$; $k = k + 1$;
    else if $(|e_i^{'}| \geq 0.5\alpha I_i)$ **and** $(|e_i^{'}| \leq 1.5\alpha I_i)$
      $b_k^{'} = 1$; $k = k + 1$;
    end;
  end;
end;

3 Experimental results

To evaluate the performance of the proposed method and to consider the applicability of the scheme in a real scenario, five songs included in the album Rust by No, Really [10] have been selected. All audio clips are sampled at 44.1 kHz with 16 bits per sample and two channels. The Objective Difference Grade (ODG) [9] is used to evaluate the transparency where ODG = 0 means no degradation and ODG = −4 means a very annoying distortion.

Table I illustrates the perceptual distortion and the payload obtained for these songs and compares the performance of the proposed watermarking algorithm and several recent audio watermarking strategies robust against many attacks. [1] Evaluates distortion by mean opinion score (MOS), which is a subjective measurment, and achieves transparency between imperceptible and perceptible but not annoying, MOS = 4.7. [4, 5] have a low capacity but are robust against most of common attacks. [6] proposes a high bit rate data hiding, but only considers MP3 compression attacks. We have used several random bits for embedding leading to different transparency results which are shown in the ODG column. The comparison shows the superiority in both capacity and imperceptibility of this method with respect to other schemes in the literature. Note that all the results have an ODG between 0 (not perceptible) −1 (not annoying), and that capacity is around 3000 bps in all the experiments. The proposed method is thus able to provide large

Table I. Results of 5 signals (robust against table II attacks) and comparison between literature schemes

Algorithm	*Audio File*	*Time (m:sec)*	*SNR (dB)*	*ODG of marked*	*Payload (bps)*
proposed	Beginning of the End	3:16	26.5 to 33	–0.6 to –0.9	3142
	Breathing On Another Planet	3:13	26 to 35.5	–0.3 to –0.9	3047
	Citizen, Go Back to Sleep	1:57	24.5 to 30	–0.1 to –0.7	2825
	Go	1:51	29 to 37.5	–0.3 to –0.8	2938
	Thousand Yard Stare	3:57	27 to 36.5	–0.3 to –0.9	3030
	average	**2:50**	**30.55**	**– 0.58**	**2996**
[1]	Song	-	Not reported	Not reported	689
[3]	One instrument	~0:20	18 to 40	–0.5 to –2	61
[4]	One instrument	~0:20	42 to 45	–1.66 to –1.88	2
[5]	Song	-	30 - 45	Not reported	86
[6]	Classical music	0:10	25	Not reported	~ 1k

DOI: 10.1587/elex.6.1057
Received June 19, 2009
Accepted June 29, 2009
Published July 25, 2009

Table II. Robustness test results for five selected files

Attack name	*State*	ODG of attacked file	*BER %*	*parameters*
AddBrumm	best	–3.5	0	1-7000, 1-1500
	worst	–3.6	0.5	1-9000,1-3000
AddDynNoise	best	–3.25	0	1-5
	worst	–0.27	8	1
ADDFFTNoise	best	–0.59	0.5	2048, 2000
	worst	–0.68	1.5	2 , 50
Addnoise	best	–0.28	0	1-80
	worst	–0.64	0.5	1-40
AddSinus	best	–0.2	0	No Limitation
	worst	–0.1	0	
Amplify	best	–0.2	0	10-100
	worst	–0.1	0	10-100
BassBoost	best	–3.6	0	1-50 and 1-45
	worst	–3.5	0	1-45 and 1-20
Echo	best	–3.43	0	1-4
	worst	–2.93	0.5	1-10
FFT_HLPassQuick	best	–3.2	1	1024,1,7k-15k
	worst	–3.51	3.2	1024,1,9k-15k
FFT_Invert	best	–3.54	1.5	1024
	worst	–3.59	2	1024
invert	best	–3.5	0	No Limitation
	worst	–2.8	0	
Resampling	–	–1.4	4.5	44.1 to 22.05 to 44.1(kHz)
LSBZero	Best	– 0.2	0	–
	worst	– 0.1	0	
MP3	Best	– 0.2	0 (0.5,3)	>160 (160,128)
	worst	– 0.1	0.5 (1.5,4.5)	>160 (160,128)
Noise_Max	best	–3.5	0	1-10,1-500,1-300
	worst	–1.8	0.5	1-14,1-100,1-100
Pitchscale	best	– 0.1	0	1
	Worst	– 0.3	1	1
RC_HighPass	best	– 2.9	0	21k>
	worst	– 2.7	0.5	21k>
RC_LowPass	best	–3.4	0	700<
	worst	–2.9	0	1000<

capacity whilst keeping imperceptibility in the admitted range (−1 to 0). We provide imperceptibility results both as SNR and ODG. SNR is provided only for comparison with other works, but ODG is a more accurate measurement of audio distortions, since it is assumed to provide a good model of the subjective difference grade (SDG) results which may be obtained by a group of human listeners. The SNR results are computed using the whole (original and marked) files, whereas the ODG results are provided using the advanced ITU-R BS.1387 standard as implemented in the Opera software [11] (the average of measurements taken in frames of 1024 samples).

Table II shows the effect of various attacks provided in the Stirmark Benchmark for Audio v1.0 [12] on ODG and BER for the five audio signals of Table I. The embedding method has been applied, the SMBA software has been used to attack the marked files and, finally, the detection method has been performed for the attacked files. The ODG in Table II is calculated between the marked and the attacked-marked files. The parameters of the attacks are defined based on SMBA web site [12]. For example, in AddBrumm, 1-7000 shows the strength and 0-1500 shows the frequency. This row illustrates that any value in the range 1-7000 for the strength and 1-1500

DOI: 10.1587/elex.6.1057
Received June 19, 2009
Accepted June 29, 2009
Published July 25, 2009

for the frequency could be used without any change in BER. In fact, this table shows the worst and best results for the five test signals based on BER and, in the case with the same BER, based on limitation of parameters. When the BER is (slightly) greater than zero, it can be made zero by using Error Correction Codes at the price of reducing the capacity. As mentioned above, depending on the specific application the parameters can be changed. E.g. by using frequency band 4–16 kHz and $\alpha = 0.25$ for the clip "Citizen, Go Back to Sleep", the obtained capacity is 6460 bps and ODG $= -0.7$, but robustness is decreased.

A very relevant issue in audio watermarking is computation time. As FFT is a fast transform, this method is useful for real-time applications. The extracting time is about 20% of the file playing time. Thus, it is perfectly possible to recover the embedded data in a real-time scenario. It is worth mentioning that these computation times have been obtained with an Intel core (TM) 2 Duo 2.2 GHz CPU and 2 GB of RAM memory with MATLAB.

4 Conclusion

In this paper, we describe a robust high-capacity watermarking algorithm for digital audio which is robust against common signal processing attacks. The ratio between the interpolated error to the interpolated sample and the selected frequency band are the two parameters of this method which can be selected adaptively to regulate the capacity, the perceptual distortion and the robustness of the scheme. Furthermore, the suggested scheme is blind. The experimental results show that this scheme has a high capacity (about 3 kbps) without significant perceptual distortion (ODG about -0.5) and provides robustness against common signal processing attacks such as echo, noise, filtering, resampling, and MPEG compression (MP3). Besides, the CPU time required by the scheme is short enough (about 20% of the playing time) to use it in real-time applications.

Acknowledgments

This work is partially supported by the Spanish Ministry of Science and Innovation and the FEDER funds under the grants TSI2007-65406-C03-03 E-AEGIS and CONSOLIDER-INGENIO 2010 CSD2007-00004 ARES.

DOI: 10.1587/elex.6.1057
Received June 19, 2009
Accepted June 29, 2009
Published July 25, 2009