

ACOUSTICAL LETTER

Simple watermark for stereo audio signals with modulated high-frequency band delay

Kazuhiro Kondo* and Kiyoshi Nakagawa

Graduate School of Science and Engineering, Yamagata University, 4–3–16 Jonan, Yonezawa, 992–8510 Japan

(*Received 8 April 2008, Accepted for publication 30 April 2008*)

Keywords: Watermarks, Stereo audio, High-frequency band, Modulated delay
PACS number: 43.60.Ek, 43.38.Vk [doi:10.1250/ast.29.384]

1. Introduction

There are a number of methods of embedding data into audio signals without causing significant quality degradation [1]. One of these, phase coding, has been regarded as an effective method with minimum impact on the perceived quality [2]. The phase of the original audio is replaced with a reference phase in accordance with the data. Since the Human Auditory System (HAS) is known to be relatively insensitive to small phase distortions, this method can potentially embed data without affecting the host signal quality significantly. We use this property in an attempt to code the embedded data in the relative phase between two stereo channels. We then evaluate the subjective speech quality of the embedded audio, as well as its robustness to added noise and audio codecs.

2. Digital watermarking of stereo audio signals

The spatial localization of sound sources in HAS is known to depend on both amplitude and phase characteristics for low frequency, but mostly on amplitude characteristics for high frequency [3]. We take advantage of this characteristic, and propose an audio watermark which embeds data by delaying the left or the right channel of high-frequency stereo signals. This characteristic is also exploited in the intensity stereo coding used in some audio codecs, *e.g.*, MPEG 1 Layer 3 (MP3) coders. We will also take advantage of this and alter the phase of the high-frequency band in the stereo channels in accordance with the data. The amplitude envelope of these signals will be preserved in order to keep this alteration unperceivable.

2.1. Embedding

Figure 1 shows the proposed watermark embedding algorithm. The high-frequency channels are mutually delayed by a fixed number of samples according to the embedded data. The input signal is first split into two subbands using the subband split filter. The cut-off frequency is set to approximately one-tenth of the full bandwidth, *i.e.*, approximately 2 kHz. We use a 128 tap FIR filter designed by the FFT windowing method. The left- and right-channel high band signal powers, P_{LH} and P_{RH}, are calculated on a frame-by-frame basis. Then, the high band signals are averaged to give one middle high-frequency channel. This signal will be copied for both left and right channels, but will be delayed on a frame-by-frame basis in accordance with the embedded data. We choose to delay the left channel to mark a '0' and the right channel to mark a '1.' A frame is divided into two equal parts, and only the latter half of the frame is delayed in accordance with the data. This is shown in Fig. 2. Thus, the first half of the frame is never delayed. We choose this arrangement to enable frame synchronization. If we embed a known data pattern regularly, *e.g.*, consecutive zeros, there will always be a transition in the left-channel delay at the middle and end of a frame, which should easily be detectable using correlation functions, for instance.

The amplitudes of each channel are then adjusted to match the power with that of the original signals, P_{LH} and P_{RH}. These signals are then added back to the low-frequency components to obtain full-bandwidth stereo audio signals.

We found that this works well for many frames which contain audible sounds, but some of the frames do not contain enough high-frequency components. Obviously, we are not able to encode data into these frames. We simply choose not to embed data in these frames. No delay is introduced in these frames. An example is shown in the third frame in Fig. 2.

The added delay should be small enough to be unperceivable. We empirically chose a maximum delay D of 68 μs (3 samples at 44.1 kHz sampling rate). Even with this small amount of delay, abrupt changes are noticeable. Therefore, we smoothed the delay changes introduced in accordance with a raised cosine-like pulse pattern, as shown in in Fig. 3. The smoothed delay, $d(i)$, can be defined as

$$d(i) = \begin{cases} D\sin\left(\dfrac{2\pi i}{T}\right) & \left(i < \dfrac{T}{4}\right) \\ D & \left(\dfrac{T}{4} < i < \dfrac{3T}{4}\right) \\ D\sin\left(\dfrac{2\pi\left(i - \dfrac{T}{2}\right)}{T}\right) & \left(\dfrac{3T}{4} < i < T\right), \end{cases} \quad (1)$$

where T is the half-frame (delay-altered frame) length, and i is the sample index. Non-integer delays are introduced using a sinc delay filter.

Also, as will be described in the next section, we will use correlation to detect delay between channel signals. In other words, we compare the cross correlation between channels with delay and without delay. If the former is larger than the latter, we detect embedded data. However, in some frames, the host signal itself shows a relatively flat correlation, even

*e-mail: kkondo@yz.yamagata-u.ac.jp

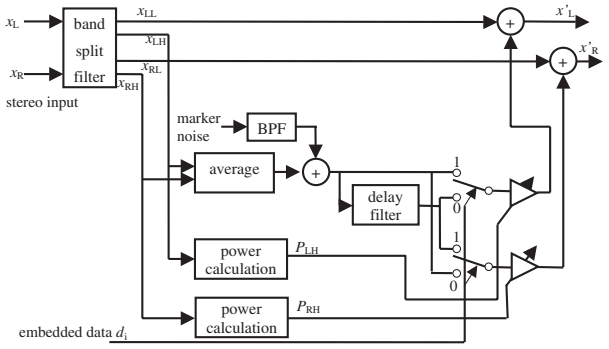

Fig. 1 Proposed watermark embedding scheme.

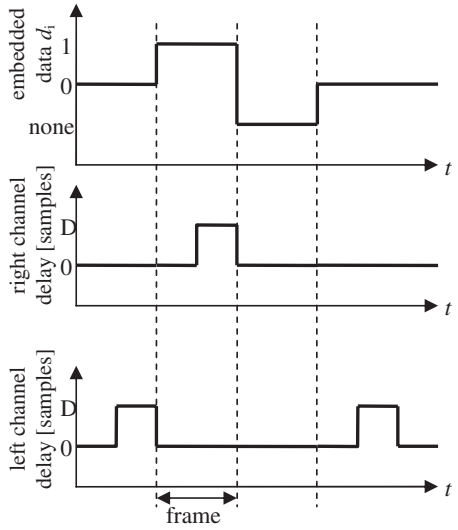

Fig. 2 Example of embedded data and channel delay.

for long lags. In these frames, we will not be able to detect delays using correlation. Thus, we add a small amount of band-pass filtered noise as markers to enhance the difference between correlation without delay and with delay. The amount of added noise is empirically chosen as 10% of the original frame power. However, even with the addition of markers, some frames do not display enough correlation difference. We choose not to embed data in these rare frames altogether.

2.2. Detection

Figure 4 shows the proposed watermark detection algorithm. This algorithm does not require the original source as a reference, and thus is blind detection. The input stereo audio signal with embedded data is first split into subbands using the same band split filter as in the embedding process. The low-frequency portions are discarded. The high frequency components are delayed by D samples. Frame boundaries are synchronized. Three normalized correlation values for the latter half of the frame are calculated.

(1) Left- and right-channel correlation with no delay (**Corr1**)
(2) Correlation between delayed left channel and right channel with no delay (**Corr2**)
(3) Correlation between delayed right channel and left channel with no delay (**Corr3**)

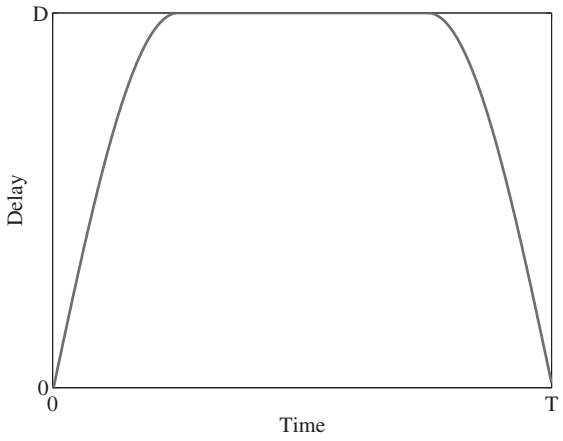

Fig. 3 Raised cosine pulse delay pattern.

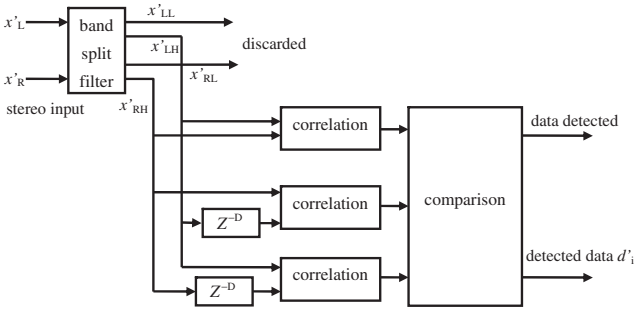

Fig. 4 Proposed watermark detecting scheme.

If **Corr1** is the largest, we assume no data has been embedded. If **Corr2** is the largest, we detect that a '1' has been embedded, and if **Corr3** is the largest, we detect a '0.' If the high-energy components are below the threshold, we assume that no data has been embedded. Thus, the detection process is a relatively simple procedure. The frame synchronization may be somewhat expensive, but this is not required once synchronization has been established. The frame synchronization can be established by calculating a sample-by-sample correlation near the embedded sync pattern.

3. Experimental evaluation

We evaluated the proposed algorithm using the four short stereo audio samples listed in Table 1. Most clips were selected from the Sound Quality Assessment Material (SQAM) CD provided by the European Broadcasting Union [4]. All sources were of CD quality, sampled at 44.1 kHz, stereo, and 16 bits/sample. The embedding frame rate was set to 1,000 samples, or about 23 ms.

Table 1 Evaluated sound sources.

Notation	Description
abba	Electric instr. (Rock)
beethoven	Ode to Joy (Classics), orch. chorus
mozart	K. 270 (Classics), wind instruments
piano	Schubert (Classics), piano solo
trumpet	Haydn (Classics), Orch. & trump. solo

Table 2 Embedded bits and bit rate.

Source	Length [samples]	Embedded bits	Bit rate [bps]
abba	667500	265	17.5
beethoven	275500	184	29.4
mozart	627500	101	7.1
piano	544000	90	7.3
trumpet	611500	132	9.5

3.1. Embedded data bit rate

The embedded number of bits for each source is shown in Table 2. The bit rate is apparently source dependent. An audio signal with high energy seems to be able to embed more data. This is probably because we choose not to embed data in frames with low high-frequency energy.

3.2. Robustness evaluation

We also evaluated the robustness of our algorithm to two types of disturbances: additive white Gaussian noise and audio codecs. We embedded three bit patterns in our host signals: **pattern 1**, a random bit pattern with the same average number of '0' and '1'; **pattern 2**, an alternating '0' and '1' pattern; and **pattern 3**, an alternating '1' and '0' pattern, *i.e.*, phase-inverted **pattern 2**.

Since the proposed algorithm can show three error modes (substitution, where '0' is replaced with '1' and vice versa, deletion, and insertion errors), we evaluated bit errors after aligning the detected bit pattern by dynamic programming. We used sclite from NIST's scoring package, freely available on the net [5]. However, most of the errors turned out to be deletion errors. Only in rare cases were substitution and insertion errors seen.

Figure 5 shows the signal-to-added-noise ratio and the bit error rate (BER) of all sources when **pattern 1** was embedded. There are differences in BER below SNR of 15 dB, but all sources show BER close to 0 above 20 dB SNR. Note that added noise larger than that giving 20 dB SNR is quite audible and renders the audio data worthless. Figure 6 shows BER for **patterns 1** through **3**. No significant difference is seen in BER with different embedded data patterns.

We also tested the robustness of the embedded data when coded with MPEG audio codecs. We encoded each source by

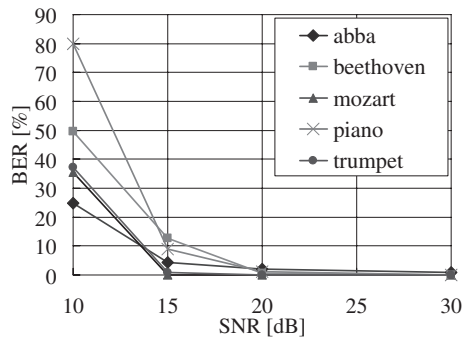

Fig. 5 BER for random embedded bit pattern (pattern 1) with additive noise.

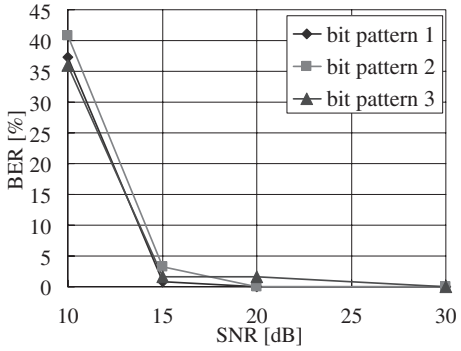

Fig. 6 BER for various embedded bit pattern with additive noise (trumpet).

Table 3 BER of MP3 and AAC encoded audio.

Source	BER [%]			
	MP3		AAC	
	96 kbps	128 kbps	96 kbps	128 kbps
abba	1.5	0.4	0.4	0.0
beethoven	0.6	0.0	0.0	0.0
mozart	0.0	1.0	0.0	0.0
piano	0.0	1.1	1.1	0.0
trumpet	0.8	0.0	0.8	0.0

MPEG 1 Layer 3 (MP3) coding and MPEG 4 AAC codecs. Lame version 3.98 beta 5 with a fixed rate of 128 and 96 kbps with joint stereo coding was used to code sources to MP3 and back to PCM. Apple Computer's iTunes version 7.4.3.1 was used to code sources to AAC and back. The AAC codec uses the Low Complexity Profile AAC to code sources to fixed rates of 128 and 96 kbps. Table 3 shows BER when random bits were embedded. In the majority of cases, no bit errors were seen, suggesting that the algorithm is robust to MPEG coders, particularly at 128 kbps. We also tested BER for other embedded bit patterns. However, no significant difference in BER with different bit patterns was seen for these audio codecs.

3.3. Subjective quality evaluation

The embedding process introduced a small noticeable alteration in quality for some of the sources, while for others, no degradation was perceived. We conducted the MUSHRA subjective tests [6] to quantify these degradations. We tested the audio clips described above with the proposed data embedding method using bit pattern 1 (**wp1**) and bit pattern 3 (**wp3**). For reference, we included the original audio clip (**ref**), a 3.5 kHz low-pass-filtered audio clip as an anchor (**lp35**), and MP3 transcoded audio clips at 128 kbps (**m128**), 96 kbps (**m96**), 64 kbps (**m64**), and 48 kbps (**m48**). Twelve subjects participated in the tests.

Figures 7 and 8 show the MUSHRA scores and the 95% confidence intervals for **abba** and **piano**. The embedded audio clip showed approximately the same quality as MP3 at 64 and 128 kbps for **abba**. For **piano**, however, the embedded audio clip showed significantly lower quality than that of MP3 at

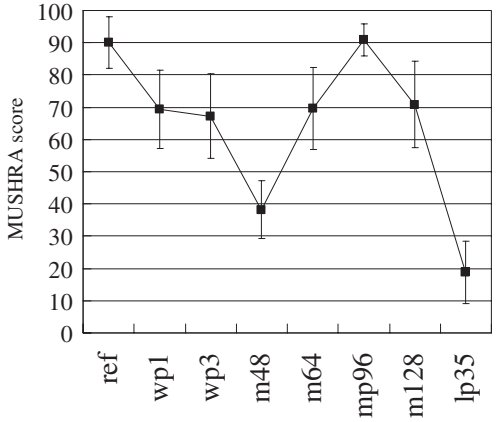

Fig. 7 Subjective quality test results (**abba**).

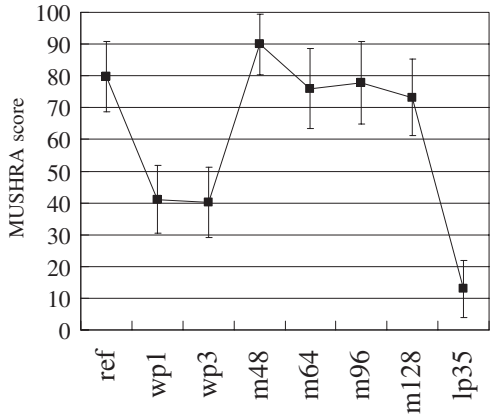

Fig. 8 Subjective quality test results (**piano**).

any rate. The quality for other sources fall between these two. The embedded quality seems to be independent of embedded bit patterns. However, the quality is apparently not transparent, and needs improvement for some sources.

4. Conclusion

We proposed an audio watermarking algorithm that exploits the fact that the perception of the auditory system for stereo audio images is relatively immune to phase in the high-frequency regions. The high-frequency stereo signals were replaced with a single average middle channel. Data was encoded into the delays between the channels. The detection of embedded data was achieved without the original signal, *i.e.*, blind detection was possible. The correlation between the right- and left-channel high-frequency signals was compared to detect the signals. The proposed algorithm showed embedded data bit rates between 7 and 30 bps depending on the host audio signal. The algorithm was also tested for robustness to added noise and MPEG coding. Little or no error was seen for additive noise with SNR above 20 dB. MPEG 1 layer 3 and MPEG AAC coding also do not seem to cause many errors. The subjective quality of the data-embedded audio clip was shown to be source dependent, with some sources showing equivalent quality to MP3 coded audio, while for others, the quality was significantly lower.

In our proposed method, we use the high-frequency energy threshold to decide whether or not to embed data. This implies a variable bit rate, and also gives rise to insertion and deletion errors. Thus, we plan to explore algorithms that allow us to embed data in every frame, which will enable us to embed data at a fixed rate. The proposed algorithm also uses blind detection. However, if the reference host signal is available, direct correlation can be used with the reference high-frequency signals to give improved robustness in detection. There is no need to change the embedding process. However, we may be able to increase the embedded data bit rate if we employ informed (nonblind) detection by changing the embedding process, perhaps by increasing the frame rate, or by using multiple delay values.

References

[1] N. Cvejic and T. Seppanen, "Introduction to digital audio watermarking," in *Digital Audio Watermarking Techniques and Technologies*, N. Cvejic and T. Seppanen, Eds. (Information Science Reference, Hershey, Pa., 2007), pp. 1–10.

[2] W. Bender, D. Gruhl, N. Morimoto and A. Lu, "Techniques for data hiding," *IBM Syst. J.*, **35**, 313–336 (1996).

[3] J. Gibson, T. Berger, T. Lookabagh, D. Linberg and R. L. Baker, *Digital Compression for Multimedia* (Morgan Kaufmann, San Francisco, 1998), p. 402.

[4] European Broadcasting Union, "Sound quality assessment material — Recording for subjective tests" (1988).

[5] The NIST Scoring Toolkit, available from http://www.nist.gov/speech/tools (2007).

[6] ITU-R BS.1543-1, "Method for the subjective assessment of intermediate quality level of coding systems" (2003).