Nearly Lossless Audio Watermark Embedding Techniques to be Extracted Contactlessly by Cell Phone

Toshio Modegi and Makoto Chiba*

*Communication and Information Operations, Dai Nippon Printing Co., Ltd.
3-5-20, Nishigotanda, Shinagawa-ku, Tokyo, 141-8001, Japan,
{Modegi-T, Chiba-M10}@mail.dnp.co.jp

ABSTRACT

We are proposing "Ubiquitous Acoustic Spaces", where each sound source can emit some address information and make us automatically access to its related cyber space, using mobile devices. In order to realize this, we have proposed a novel audio watermarking technique, which enables contactless asynchronous detection of embedded watermarks through speaker and microphone devices. However its embedding data rate was around 10 [bps], which was not sufficient for embedding such as URL texts.

Therefore, we have extended the embedding frequency range and proposed a duplicated embedding algorithm, which uses both previously proposed frequency division method and temporal division method. By these improvements, possible embedding data rate could be extended to 61.5 [bps], and we could extract watermarks through public telephone networks, even from a cell phone speaker.

In this paper, we describe abstracts of our improved watermark embedding algorithms, and experimental results of extraction precision on several signal capturing conditions.

Keywords: audio watermark, lossless, contactless extraction, cell phone, analogue robustness, sound source location

1 INTRODUCTION

We are proposing "Ubiquitous Acoustic Spaces[4]", where each sound source can emit some address information with audio signals and make us automatically access to its related cyber space, using handheld devices such as cellphones. In order to realize this concept, we have considered three types of extraction methods, which were an acoustic modulation, an audio fingerprint, and an audio watermark technique. Then we have proposed a novel audio watermarking technique[1]-[3], which enables contactless asynchronous detection of embedded audio watermarks through speaker and microphone devices. However its embedding data rate was around 10 [bps], which was not sufficient for embedding generally used URL address texts [4].

Therefore, we have extended the embedding frequency range and proposed a duplicated embedding algorithm, which uses both previously proposed frequency division method and temporal division method together. By this newly proposed methods, embedding data rate can be extended by increasing number of divided bands in the frequency dimensions. In our experiments, possible embedding data rate could be extended to 61.5 [bps].

In this paper, we describe abstracts of our improved watermark embedding and extracting algorithms, and experimental results of watermark extraction precision on several audio signal capturing conditions.

2 PREVIOUSLY PROPOSED AUDIO WATERMARK EMBEDDING METHOD

2.1 Stereo Watermark Embedding Method

At first, we describe an algorithm of our previously proposed stereo watermark embedding method [1]. Figure 1 shows our proposed watermark embedding method for stereo audio signals. This embeds a set of given bitstream data by changing two-channel stereo locations of low frequency components in an embedding given audio signal. In each separated audio frame, a single tri-state code (–1, 0, +1) is embedded. The panning location of lower frequency components (around less than 400Hz) will be changed to the left in case of the embedded code –1 (a bit 0 is embedded), changed to the right in case of the code +1 (a bit 1 is embedded), and changed to the center in case of the code 0 (no bit is embedded).

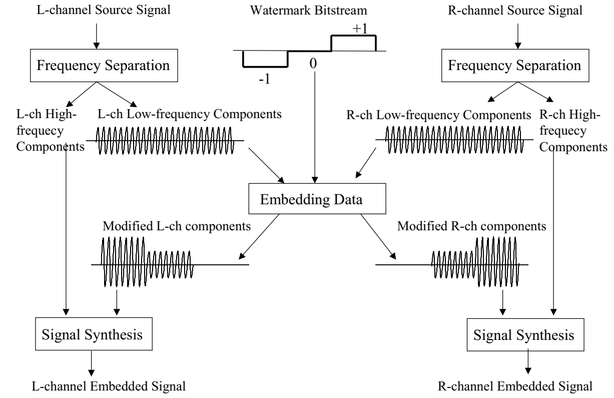

Figure 1: Stereo watermark embedding method.

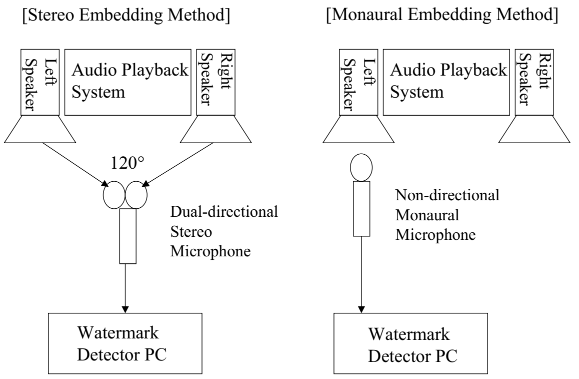

Figure 2: Two kinds of proposing contactless watermark

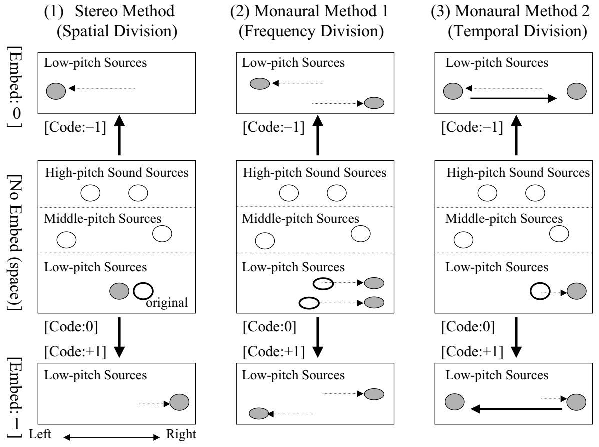

Figure 3: Three concepts of our proposing watermark embedding methods.

This method will not modify any frequency components except for the lower-frequency sound source direction, which we (human beings) cannot clearly recognize. Even if we modify the quality of each channel component, the embedded codes will not be disappeared as far as the volume balance of two channels is maintained. Therefore, this method features nearly lossless embedding, robustness against lossy data compression or analogue conversion. And we have confirmed a contactless detection capability of embedded watermarks through speaker and microphone devices. The following sections describe a specific algorithm based on this method.

2.2 Two Kinds of Monaural Supported Watermark Embedding Method

In order to contactlessly extract the embedded watermarks on the previously proposed method, we have to use stereo dual-directional microphone like shown in the left side of Fig.2. This restricts strictly sensible positions of a stereo directional microphone and requires enough sound source output energy because of somewhat distance to both speakers. In this section, we describe two kinds of our proposed extended methods, by which we can detect watermarks with a monaural microphone as shown in the right side of Fig.2. These two methods enable detection by non-directional monaural miniature microphone units embedded in such as mobile voice recorder or cellphone devices.

Figure 3 shows concepts of our previously proposed stereo embedding method and two kinds of monaural embedding methods. These three can be categorized by an embedding area division method as a spatial division (stereo supported method), a frequency division and a temporal division (two kinds of monaural supported methods).

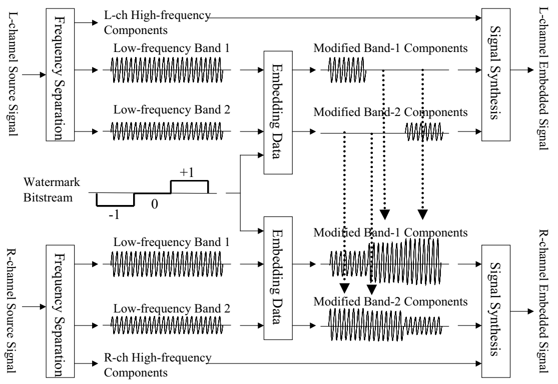

Figure 4: Frequency division type monaural watermark embedding method.

Figure 4 shows the first extended monaural watermark embedding method based on the middle concept shown in Fig.3. This embeds a set of given bitstream data by changing the level balance of two-band divided low frequency components in a monaural signal or stereo left-channel signal. Figure 4 indicates, the component of the first band will be removed in case the embedded code is +1, the second band will be removed in case of the code –1, and both bands will be removed in case of the code 0. The removed components will be transferred to the corresponding frequency position of the right channel if exists. This method has robustness against an extracting frame phase shift, but may be easily influenced by lower frequency distortions of acoustic transmission systems such as speaker and microphone device characteristics.

Figure 5 shows the second extended monaural watermark embedding method based on the right side concept shown in Fig.3. This embeds a set of given bitstream data by changing the level balance between two temporally divided sub-frame low-frequency components, in a monaural signal or stereo left-channel signal. The component of the first sub-frame will be removed in case the embedded code is +1, the second sub-frame will be removed in case of the code –1, and both sub-frame will be removed in case of the code 0. The removed components will be also transferred to the corresponding frequency position of the right channel if exists. Different from the Fig.4. This method has robustness against lower frequency characteristic distortions but is likely to be influenced by a phase shift.

As shown in Fig.5, reverse component transfer operations to the left components on the left channel from the corresponding components of the right channel can be applied. These operations can make data extraction sensitivity of the left channel increase and can be applied also to the previous two-band frequency division type embedding method.

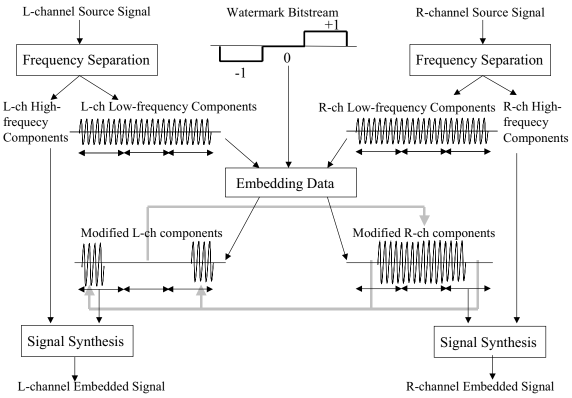

Figure 5: Temporal division type monaural watermark embedding method.

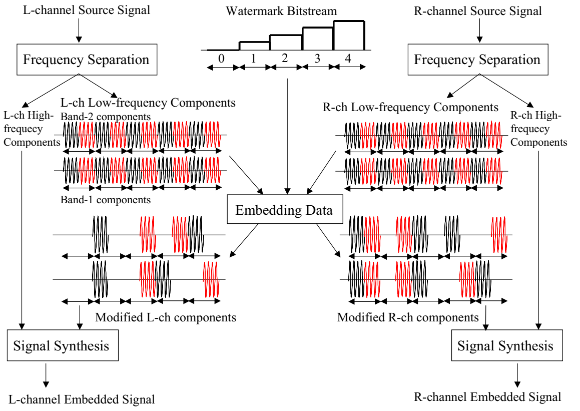

Figure 6: Double-rate monaural watermark embedding method by combining two kinds of previously proposed monaural methods.

3 IMPROVED AUDIO WATERMARK EMBEDDING METHOD

3.1 General Description

In this paper, we propose a duplicated embedding algorithm, which uses both previously proposed frequency division method and temporal division method together like shown in Fig.6. This newly proposed method can be further extended by increasing the number of divided bands in the frequecncy direction. In our experiment, we could extend up to four number of bands and embed data by 6.15 times rate. In this section, we describe specific embedding and extraction algorithms of two-band divided double-rate embedding case shown in Fig.6.

3.2 Watermark Embedding Algorithm for the Double-rate Extended Monaural Method

At first, we convert a set of N-length stereo audio frame $X_l(i)$ and $X_r(i)$ (i=0,…,N−1) sampled by the frequency fs to total 16 sets of frequency dimension data as real parts: $A_{lk}(i)$, $A_{rk}(j)$ (j=0,…,N−1, k=1,2,3,4) and imaginary parts: $B_{lk}(j)$, $B_{rk}(j)$ (j=0,…,N−1, k=1,2,3,4). They are given by the following equations using our proposed asymmetry dividing window functions $W_k(i)$ (k=1,2,3,4) as follows [4]:

$$\begin{aligned}
&A_{lk}(j)=\Sigma_{i=0,N-1} W_k(i)X_l(i)\cos(2\pi ij/N)\ .\\
&B_{lk}(j)=\Sigma_{i=0,N-1} W_k(i)X_l(i)\sin(2\pi ij/N)\ .\\
&A_{rk}(j)=\Sigma_{i=0,N-1} W_k(i)X_r(i)\cos(2\pi ij/N)\ .\\
&B_{rk}(j)=\Sigma_{i=0,N-1} W_k(i)X_r(i)\sin(2\pi ij/N)\ .\\
&\qquad\qquad\qquad (0\leq j\leq N\text{-}1,\ 1\leq k\leq 4) \qquad (1)
\end{aligned}$$

$$\begin{aligned}
&W_1(i)=0.5-0.5\cos(8\pi i/(3N))\ \ (i<3N/8)\ .\\
&W_1(i)=0.5-0.5\cos(8\pi(i\text{-}N/4)/N)\ \ (3N/8\leq i<N/2)\ .\\
&W_1(i)=0.0\ \ (N/2\leq i)\ .
\end{aligned}$$

$$\begin{aligned}
&W_2(i)=0.0\ \ (i<3N/8)\ .\\
&W_2(i)=0.5-0.5\cos(8\pi(i\text{-}3N/8)/N)\ \ (3N/8\leq i<N/2)\ .\\
&W_2(i)=0.5-0.5\cos(4\pi(i\text{-}N/4)/N)\ \ (N/2\leq i<3N/4)\ .\\
&W_2(i)=0.0\ \ (3N/4\leq i)\ .
\end{aligned}$$

$$\begin{aligned}
&W_3(i)=0.0\ \ (i<N/2)\ .\\
&W_3(i)=0.5-0.5\cos(4\pi(i\text{-}N/2)/N)\ \ (N/2\leq i)\ .
\end{aligned}$$

$$\begin{aligned}
&W_4(i)=0.0\ \ (i<N/4)\ .\\
&W_4(i)=0.5-0.5\cos(4\pi(i\text{-}N/4)/N)\ \ (N/4\leq i<N/2)\ .\\
&W_4(i)=0.5-0.5\cos(8\pi(i\text{-}N/8)/(3N))\ \ (N/2\leq i<7N/8)\ .\\
&W_4(i)=0\ \ (7N//8\leq i)\ . \qquad (2)
\end{aligned}$$

The frame shift value is a half-length of a frame, N/2, and we embed data only to even-numbered frames. For odd-numbered frames we use the fourth window function $W_4(i)$ and execute modifications based on the following equation (3) for k=4 and 0≤j≤M-1. The value M is the maximum embedding frequency limit typically 40 (≅400Hz)

$$\begin{aligned}
&E_k(j)=\{A_{lk}(j)^2+B_{lk}(j)^2+A_{rk}(j)^2+B_{rk}(j)^2\}^{1/2}.\\
&A_{rk}'(j)=A_{rk}(j)E_k(j)/\{A_{rk}(j)^2+B_{rk}(j)^2\}^{1/2}.\\
&B_{rk}'(j)=B_{rk}(j)E_k(j)/\{A_{rk}(j)^2+B_{rk}(j)^2\}^{1/2}.\\
&A_{lk}'(j)=B_{lk}'(j)=0. \qquad (3)
\end{aligned}$$

For even-numbered frames, we apply three window functions $W_1(i)$, $W_2(i)$, and $W_3(i)$, and execute modifications based on the equation (3) for k=2 and 0≤j≤M-1 in any case. At first we calculate the following four kinds of energy values E_1, E_2, E_3 and E_4.

$$\begin{aligned}
&E_1=\Sigma_{j=m,m+p\text{-}1}\{A_{l1}(j)^2+B_{l1}(j)^2+A_{r1}(j)^2+B_{r1}(j)^2\}\bullet F(j-m).\\
&E_2=\Sigma_{j=m,m+p\text{-}1}\{A_{l3}(j)^2+B_{l3}(j)^2+A_{r3}(j)^2+B_{r3}(j)^2\}\bullet F(j-m).\\
&E_3=\Sigma_{j=m+p,m+2p\text{-}1}\{A_{l1}(j)^2+B_{l1}(j)^2+A_{r1}(j)^2+B_{r1}(j)^2\}\\
&\qquad\qquad\bullet F(j-m-p).\\
&E_4=\Sigma_{j=m+p,m+2p\text{-}1}\{A_{l3}(j)^2+B_{l3}(j)^2+A_{r3}(j)^2+B_{r3}(j)^2\}\\
&\qquad\qquad\bullet F(j-m-p). \qquad (4)
\end{aligned}$$

Here, the value m is the minimum embedding frequency limit typically 15 (≅150Hz) and p is the width of divided band typically 12 (≅120Hz). $F(j)$ is a frequency-direction window function for improving resolution of frequency divided bands defined as follows:

$$F(j) = 1.0-4.0\bullet(j-p/2)^2/p^2\ . \qquad (5)$$

Embedding five-level data (code 0, 1, 2, 3, 4, 5) is done as following using a level slice value L_o (given level lower limit),.

1) $E_1>L_o$, $E_3>L_o$ and Embedded Code 1:
Execute modifications based on the equation (6) for k=1, $m\le j\le m+p$-1 ($j_o=m$) and $m+p\le j\le m+2p$-1 ($j_o=m+P$), and the equation (3) for k=3, $m\le j\le m+p$-1 and $m+p\le j\le m+2p$-1.

2) $E_2>L_o$, $E_4>L_o$ and Embedded Code 2:
Execute modifications based on the equation (6) for k=3, $m\le j\le m+p$-1 ($j_o=m$) and $m+p\le j\le m+2p$-1 ($j_o=m+P$), and the equation (3) for k=1, $m\le j\le m+p$-1 and $m+p\le j\le m+2p$-1.

3) $E_1>L_o$, $E_4>L_o$ and Embedded Code 3:
Execute modifications based on the equation (6) for k=1 and $m\le j\le m+p$-1 ($j_o=m$), the equation (6) for k=3 and $m+p\le j\le m+2p$-1 ($j_o=m+P$), the equation (3) for k=3 and $m\le j\le m+p$-1, and the equation (3) for k=1 and $m+p\le j\le m+2p$-1.

4) $E_2>L_o$, $E_3>L_o$ and Embedded Code 4:
Execute modifications based on the equation (6) for k=3 and $m\le j\le m+p$-1 ($j_o=m$), the equation (6) for k=1 and $m+p\le j\le m+2p$-1 ($j_o=m+P$), the equation (3) for k=1 and $m\le j\le m+p$-1, and the equation (3) for k=3 and $m+p\le j\le m+2p$-1.

5) Otherwise (including Embedded Code 0):
Execute modifications based on the equation (3) for k=1, for k=3 and $0\le j\le M$-1.

$$\begin{aligned}
&E_k(j)=\{A_{lk}(j)^2+B_{lk}(j)^2+A_{rk}(j)^2+B_{rk}(j)^2\}^{1/2}.\\
&A_{lk}'(j)=F(j-j_o)\bullet A_{lk}(j)E_k(j)\,/\,\{A_{rk}(j)^2+B_{rk}(j)^2\}^{1/2}.\\
&B_{lk}'(j)=F(j-j_o)\bullet B_{lk}(j)E_k(j)\,/\,\{A_{rk}(j)^2+B_{rk}(j)^2\}^{1/2}.\\
&A_{rk}'(j)=(1-F(j-j_o))\bullet A_{rk}(j)E_k(j)\,/\,\{A_{rk}(j)^2+B_{rk}(j)^2\}^{1/2}.\\
&B_{rk}'(j)=(1-F(j-j_o))B_{rk}(j)E_k(j)\,/\,\{A_{rk}(j)^2+B_{rk}(j)^2\}^{1/2}. \quad (6)
\end{aligned}$$

Finally, for converting to temporal dimensions, we calculate the following equation (7) and (8), which are added to the previously modified frame data $X_l^p(i+N/2)$ and $X_r^p(+N/2i)$ with half-size of frame offset.

For odd-numbered frames:

$$\begin{aligned}
&X_l'(i)=1/N\ \{\Sigma_{j=0,N-1}A_{lk}'(j)\cos(2\pi ij/N)\\
&\qquad-\Sigma_{j=0,N-1}B_{lk}'(j)\sin(2\pi ij/N)\ \}+(1-W_4(i))X_l^p(i+N/2).\\
&X_r'(i)=1/N\ \{\Sigma_{j=0,N-1}A_{rk}'(j)\cos(2\pi ij/N)\\
&\qquad-\Sigma_{j=0,N-1}B_{rk}'(j)\sin(2\pi ij/N)\}+(1-W_4(i)). \qquad (7)
\end{aligned}$$

For even-numbered frames:

$$\begin{aligned}
&X_l'(i)=1/N\ \Sigma_{k=1,3}\{\ \Sigma_{j=0,N-1}A_{lk}'(j)\cos(2\pi ij/N)\\
&\qquad-\Sigma_{j=0,N-1}B_{lk}'(j)\sin(2\pi ij/N)\ \}\\
&\qquad+\Sigma_{k=1,3}\ (1-W_k(i))X_l^p(i+N/2).\\
&X_r'(i)=1/N\ \Sigma_{k=1,3}\{\ \Sigma_{j=0,N-1}A_{rk}'(j)\cos(2\pi ij/N)\\
&\qquad-\Sigma_{j=0,N-1}B_{rk}'(j)\sin(2\pi ij/N)\ \}\\
&\qquad+\Sigma_{k=1,3}\ (1-W_k(i))X_r^p(+N/2i). \qquad (8)
\end{aligned}$$

3.3 Watermark Extracting Algorithm for the Double-rate Extended Monaural Method

Similarly to the section 3.2, we convert a set of N-length stereo left-channel watermark embedded audio frame $X_l(i)$ (i=0,…,N−1, k=1 and 3) sampled by frequency fs to frequency dimension data as real parts: $A_{lk}(j)$ and imaginary parts: $B_{lk}(j)$ (j=0,…,N−1, k=1 and 3). Based on the following equation (9) similar to the equation (4), we calculate energy values E_1, E_2, E_3 and E_4 for four bands divided in the both frequency and temporal direction.

$$\begin{aligned}
&E_1=\Sigma_{j=m,m+p\text{-}1}\{A_{l1}(j)^2+B_{l1}(j)^2\}\bullet F(j-m).\\
&E_2=\Sigma_{j=m,m+p\text{-}1}\{A_{l3}(j)^2+B_{l3}(j)^2\}\bullet F(j-m).\\
&E_3=\Sigma_{j=m+p,m+2p\text{-}1}\{A_{l1}(j)^2+B_{l1}(j)^2\}\bullet F(j-m-p).\\
&E_4=\Sigma_{j=m+p,m+2p\text{-}1}\{A_{l3}(j)^2+B_{l3}(j)^2\}\bullet F(j-m-p). \qquad (9)
\end{aligned}$$

In some case, the above audio frames are resampled by analogue converted watermark embedded audio signals, each signal level is changed apparently from the original not embedded level. Therefore we define the lower limit level L dynamically on the actual preceding signal level. Using these values and defining two-band energy ratio as 2, we can determine an embedded code as following.

1) If $E_1>L$, $E_3>L$, $E_1>2E_2$ and $E_3>2E_4$, an embedded code is supposed to be 1.

2) If $E_2>L$, $E_4>L$, $E_2>2E_1$ and $E_4>2E_3$, an embedded code is supposed to be 2.

3) If $E_1>L$, $E_4>L$, $E_1>2E_2$ and $E_4>2E_3$, an embedded code is supposed to be 3.

4) If $E_2>L$, $E_3>L$, $E_2>2E_1$ and $E_3>2E_4$, an embedded code is supposed to be 4.

5) Otherwise, an embedded code is 0.

3.4 Extension Method of Watermark Embedding Rate

Figure 7 shows data embedding frequency bands based on our previously or newly proposed audio watermark embedding methods. In all of these figures, we suppose the source audio signal is CD-quality (44.1 kHz sampled) and the frame size N is 4096. Data embedding frequency area is defined between 15 and 40 [unit: sample], which means between 150 and 400 [Hz]. The higher bands more than 40 will not be modified, whereas the lowest band less than 15 is defined as a gap area and the left-channel signal components are transferred to the right channel in any case.

Figure 7-(a) is a single-band embedding, indicating our previously proposed temporal division method. If the source audio signal is CD-quality (44.1 kHz sampled) and the frame size *N* is 4096, embedding data rate is around 10 [bps]. Figure 7-(b) is a double-band embedding described in the previous section, newly proposed in this paper. In this case, embedding data rate can be extended to around 30 [bps], although 20 [bps] embedding case is described in the previous section.

Figure 7-(c) is a triple-band extended embedding method, whose data rate can be extended to around 40 [bps]. And Fig. 7-(d) is a quadruple-band extended embedding method, whose data rate can be extended to around 61.5 [bps].

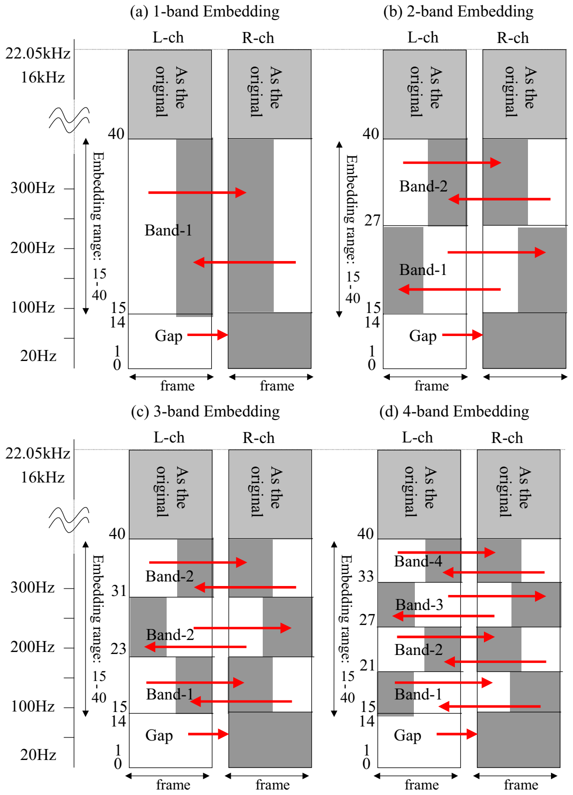

Figure 7: Data embedding frequency bands in previously or newly proposed methods.

3.5 Improvement of 8-bit Byte Data Embedding Sequence

Figure 8 shows two-byte data embedding sequence examples. All bits of the first byte have been normally embedded whereas some bits of the second byte have been embedded multiple times owing to host signal level conditions.

Figure 8-(1) is a previously proposed 10 [bps] method [4], which requires at least two additional frames for discriminating between bytes. Figure 8-(2) is our first improved method, which requires only one additional frame for discriminating between bytes. Figure 8-(3) is our second improved method, which reduces an additional frame for discriminating between bytes. In this method, the last data bit-8 should be a parity, we will search the last bit of a byte by parity checking method.

Figure 8 (4)-(6) are extending the sequence of Fig.8-(3) method to higher data-rate described in the previous section. Therefore in these methods, the last data bit-8 should be also a parity.

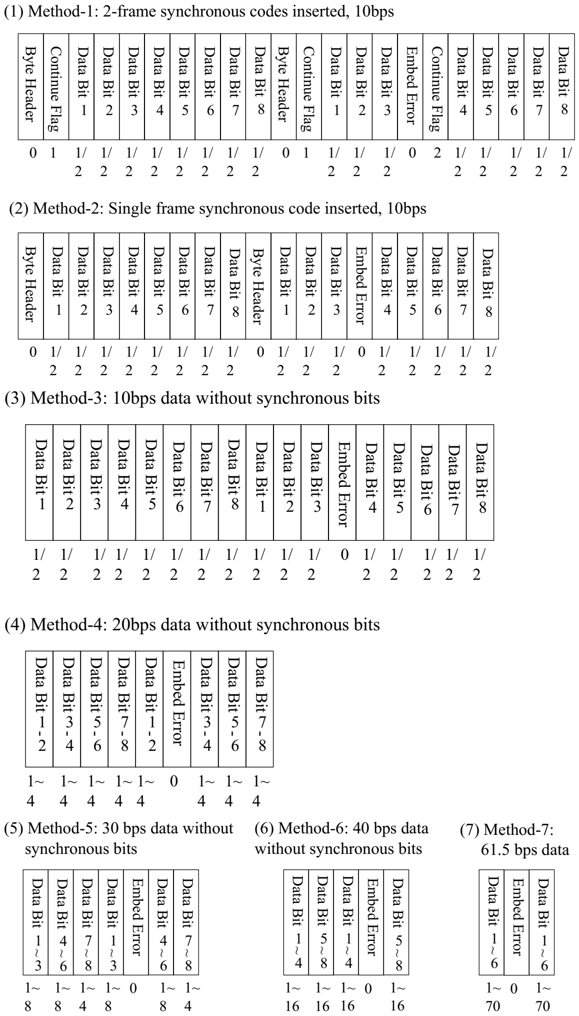
(1) Method-1: 2-frame synchronous codes inserted, 10bps
Byte Header, Continue Flag, Data Bit 1, Data Bit 2, Data Bit 3, Data Bit 4, Data Bit 5, Data Bit 6, Data Bit 7, Data Bit 8, Byte Header, Continue Flag, Data Bit 1, Data Bit 2, Data Bit 3, Embed Error, Continue Flag, Data Bit 4, Data Bit 5, Data Bit 6, Data Bit 7, Data Bit 8
0 1 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 0 1 1/2 1/2 1/2 0 2 1/2 1/2 1/2 1/2 1/2
(2) Method-2: Single frame synchronous code inserted, 10bps
Byte Header, Data Bit 1, Data Bit 2, Data Bit 3, Data Bit 4, Data Bit 5, Data Bit 6, Data Bit 7, Data Bit 8, Byte Header, Data Bit 1, Data Bit 2, Data Bit 3, Embed Error, Data Bit 4, Data Bit 5, Data Bit 6, Data Bit 7, Data Bit 8
0 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 0 1/2 1/2 1/2 0 1/2 1/2 1/2 1/2 1/2
(3) Method-3: 10bps data without synchronous bits
Data Bit 1, Data Bit 2, Data Bit 3, Data Bit 4, Data Bit 5, Data Bit 6, Data Bit 7, Data Bit 8, Data Bit 1, Data Bit 2, Data Bit 3, Embed Error, Data Bit 4, Data Bit 5, Data Bit 6, Data Bit 7, Data Bit 8
1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 0 1/2 1/2 1/2 1/2 1/2
(4) Method-4: 20bps data without synchronous bits
Data Bit 1,2, Data Bit 3,4, Data Bit 5,6, Data Bit 7,8, Data Bit 1,2, Embed Error, Data Bit 3,4, Data Bit 5,6, Data Bit 7,8
1~4 1~4 1~4 1~4 1~4 0 1~4 1~4 1~4
(5) Method-5: 30 bps data without synchronous bits
Data Bit 1~3, Data Bit 4~6, Data Bit 7~8, Data Bit 1~3, Embed Error, Data Bit 4~6, Data Bit 7~8
1~8 1~8 1~4 1~8 0 1~8 1~4
(6) Method-6: 40 bps data without synchronous bits
Data Bit 1~4, Data Bit 5~8, Data Bit 1~4, Embed Error, Data Bit 5~8
1~16 1~16 1~16 0 1~16
(7) Method-7: 61.5 bps data
Data Bit 1~6, Embed Error, Data Bit 1~6
1~70 0 1~70

Figure 8: 8-bit byte data embedding sequence in previously or newly proposed methods.

Figure 8-(7) is 61.5 [bps] byte-per-frame embedding example based on Fig.7-(d). In this method we do not consider discrimination between bytes and need not any parity bit. However, embedding byte data are restricted to ASCII 70 kinds of characters, including all of alpha-numeric and 8 symbol characters, used for such as URL texts.

4 EXPERIMENTAL RESULTS

The following is experimental results based on our proposed seven kinds of watermark embedding sequences shown as Fig.8. We have embedded a set of 80-byte ASCII character strings repeatedly into a 4-minute CD-quality stereo song file by the seven kinds of embedding methods. For each of embedded song, we have tried the following seven kinds of extraction experiment as shown in Fig.9. And Table 1 shows experimental results, where each number shows number of successfully extracted characters.

(1) File analysis: Analyzing directly the data embedded non-compressed song file.
(2) MP4 compressed file analysis: Analyzing digitally the data embedded MPEG-4/AAC 128 [kbp] compressed song file which can be played back in our cellphone (NTTdocomo, FOMA F901iS).
(3) Analogue line: Analyzing a resampled audio file through analogue audio line played back CD player.
(4) Microphone: Analyzing a contactlessly captured audio file using a speaker (Sony, SRS-Z510) and microphone (Sony, ECM-MS957).
(5) Cellphone: Analyzing a contactlessly recorded 3GPP/AMR-NB compressed voice file by cellphone voice recorder in the same audio playback emnvironment as (4).
(6) Phoneline: Analyzing a transferred audio signal through public phone networks using two cellphones in the same audio playback emnvironment as (4).
(7) Cellphone speaker: Analyzing a contactlessly captured audio file using the cellphone speaker playing MPEG-4/AAC compressed file as a sound source.

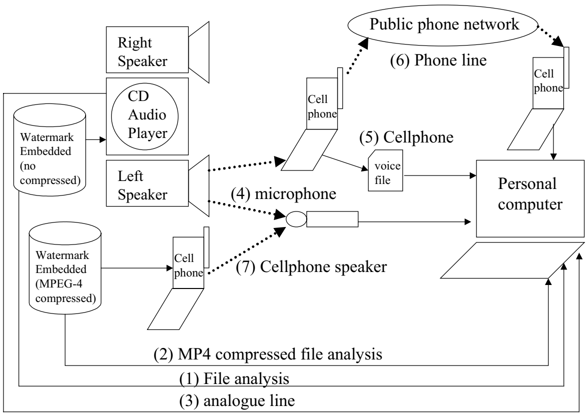

Figure 9: Watermark extracting experimental system

Table 1: Experimental results (number of characters)

Method	1	2	3	4	5	6	7
Embed Rate[bps]	10	10	10	20	30	40	61.5
Source	263	293	327	624	814	1039	2090
(1) File Analysis	262	293	327	622	783	1026	1965
(2) MP4 compressed	262	293	327	620	797	1027	1960
(3) Analogue line	262	292	327	621	759	1011	1953
(4) Microphone	260	287	327	596	668	927	1126
(5) Cellphone	263	277	324	569	610	815	1841
(6) Phone line	238	245	302	495	570	742	1530
(7) Cellphone-SP	157	147	325	593	683	610	1995

5 CONCLUSIONS

In this paper, we have increased embedding data rate up to 61.5 [bps], and in 40 [bps] emebding mode extracting precisions have been moe than 90%. Moreover, we could extract watermarks in miscellaneous audio playback environment, including through public telephone networks and even from a cell phone speaker. However, these improved methods have drawbacks of giving distortion to the right channel signal in some cases and destructing panning positions in wider band range, which we have to further evaluate and improve. And we are going to improve extraction precisions with inserting error correction codes.

In the future, we will implement our developed real-time watermark extraction software in PDA or cellphone devices, and apply to our concept of "Ubiquitous Acoustic Spaces."

REFERENCES

[1] T. Modegi, "Development of Contactless Content-ID Sensing System for Watermark Embedded Audio Signals," Proceedings of SICE Annual International Conference in Okayama, pp.1394-1399 (August. 2005).

[2] T. Modegi, "Development of Nearly Lossless Embedding Technology of Contactless Sensible Watermarks for Audio Signals," Proceedings of International Workshop on Intelligent Data Hiding and Multimedia Signal Processing, KES2005, pp.1122-1128 (Sep. 2005).

[3] T. Modegi, "Nearly Lossless Audio Watermark Embedding Techniques to be Extracted Contactlessly by Cell Phone," Proceedings of International Conference on Mobile Data Management, MDM2006, pp.36-39 (May 2006).

[4] T. Modegi, "Development of Audio Watermark Technology to be Extracted Contactlessly by Cell Phone," IEEJ Transactions on Electronics, Information and Systems, Vol.126, No.7, pp.825-831 (Jul. 2006, in Japanese).