

An audio watermarking method robust against time- and frequency-fluctuation

Ryuki Tachibana, Shuichi Shimizu, Taiga Nakamura, and Seiji Kobayashi

Tokyo Research Laboratory, IBM Japan
1623-14, Shimotsuruma, Yamato-Shi, Kanagawa-Ken, Japan

ABSTRACT

In this paper, we describe an audio watermarking algorithm that can embed a multiple-bit message which is robust against wow-and-flutter, cropping, noise-addition, pitch-shift, and audio compressions such as MP3. The embedding algorithm calculates and manipulates the magnitudes of segmented areas in the time-frequency plane of the content using short-term DFTs. The detection algorithm correlates the magnitudes with a pseudo-random array that corresponds to two-dimensional areas in the time-frequency plane. The two-dimensional array makes the watermark robust because, even when some portions of the content are heavily degraded, other portions of the content can match the pseudo-random array and contribute to watermark detection. Another key idea is manipulation of magnitudes. Because magnitudes are less influenced than phases by fluctuations of the analysis windows caused by random cropping, the watermark resists degradation. When signal transformation causes pitch fluctuations in the content, the frequencies of the pseudo-random array embedded in the content shift, and that causes a decrease in the volume of the watermark signal that still correctly overlaps with the corresponding pseudo-random array. To keep the overlapping area wide enough for successful watermark detection, the widths of the frequency subbands used for the detection segments should be increased as frequency increases. We theoretically and experimentally analyze the robustness of proposed algorithm against a variety of signal degradations.

Keywords: Digital watermarking, audio watermark, synchronization attack, geometric distortion

1. INTRODUCTION

Recent work on audio watermarking has shown significant progress in inaudibility, reliability, and robustness.¹⁻⁵ Audio watermarking techniques have achieved robustness against MPEG compression, additive noise, lowpass filtering, etc.

Wu⁵ pointed out that a *random sample cropping attack* can efficiently interfere with watermark detection process. This is because cropping displaces the detection windows from the original embedding windows. Though one method to solve this synchronization problem is an exhaustive search of the embedding windows, that requires excessive computational time.

Another serious attack for audio watermark is pitch shifting. This is because it causes mis-synchronization between the frequencies of the pseudo-random sequences (PRS) embedded in the content and the frequencies of the PRS that is used for detection. This problem could be caused by intentional attacks and by unintentional side effects of analogue editing as well. Although Tilki⁶ proposed a method that embeds five additional sinusoids for frequency synchronization, less attention has been paid to fluctuation of frequency. Since most forms of degradation effect watermark detection in the same way as cropping or pitch shifting does, we believe that robustness against these problems is important for analysis of audio watermarks. In fact, the effects of cropping and pitch shifting on audio watermark is similar to the effects of geometric distortions on image watermark, and the robustness of image watermarking techniques against translation, rotation, and scaling has also been receiving much attention recently.

In this paper, we describe an audio watermarking method that is robust against time and frequency fluctuations. For example, a 64-bit message can be detected in a 30-second music sample and survive cropping, wow-and-flutter, and pitch shifting as well as MPEG compression, additive noise, echo, and digital-analog conversions. A psychoacoustic model calculates the amount of inaudible modification for watermark embedding and hence assures high transparency. The detection algorithm does not need to refer to the original content.

One of the key ideas of this method is that it deals with a two-dimensional pseudo-random array (PRA) in the time-frequency plane of the content. The embedding algorithm modifies the magnitudes of segmented areas in the plane according to the PRA and the detection algorithm correlates the PRA and the magnitudes of the content. The two-dimensional array allows use of a longer array, which makes the watermark robust against time-fluctuation caused by duplicate or missing audio samples. This is because loss in some portions of the array can be recovered using other portions. The watermark robustness against pitch-shifted content can be estimated based on the amount of the segmented areas that can maintain the correspondence between the PRA and the watermark signal* embedded in the content.

For robustness against pitch shifting, Tilki's method synchronized frequencies by searching for five additional sinusoids. On the other hand, our strategy is to make the detected watermark's strength insensitive to frequency fluctuation. For that reason, the widths of the subbands used for the detection segments should be increased in higher frequencies, so that the correspondences for all subbands are maintained to the same degree.

As for robustness against random cropping, our method utilizes the shift-invariance characteristics of magnitudes in the Fourier domain that is also utilized by Wu's method and for image watermarking techniques.⁷⁻⁹ Wu solved the mis-synchronization problem by synchronizing the embedding and detection windows at salient points and by limiting watermark embedding and detection to the regions following the salient points. Unlike Wu's method, our method embeds watermark into the entire content. Since magnitudes are smoothly modified using windowing and overlapping, the effect of watermark embedding can be observed even in displaced detection windows. Consequently, a sample-by-sample synchronization procedure for aligning the embedding and detection windows is not necessary.

This paper is organized as follows. After the key ideas for the method are described in the first half of Sect. 2, the algorithms for watermark embedding and detection are presented in the second half. Theoretical and experimental analysis of the method is given in Sect. 3. Finally, conclusions are drawn in Sect. 4.

2. WATERMARKING ALGORITHM

In this chapter, after the basic ideas of the method are described, the embedding and detection algorithms are presented.

2.1. Basic Concepts

2.1.1. Multiple-bit embedding

This method can embed a multiple-bit message. For an instance, a 64-bit message is embedded in the robustness tests shown in Sect. 3. The multiple-bit message is divided into short messages, which are embedded separately. The algorithm should apply its error correction and/or detection method to the multiple-bit message and must embed and detect several additional bits to indicate the beginning of the message. The error correction and detection method used in Sect. 3 can be found in¹⁰.

2.1.2. Pattern blocks and pseudo-random arrays

A short message is embedded in a *pattern block*, which is defined as a two-dimensional segmented area in the time-frequency plane of the content. The time-frequency plane is a two dimensional array constructed from the sequence of power spectrums calculated using short-term DFTs. Fig. 1 illustrates four consecutive pattern blocks in the time-frequency plane. The background image of the figure is a spectrogram of a music sample. A pattern block is further divided into N_W and N_H tiles in rows and columns, respectively. Hence, the total number of tiles in a pattern block, N_T , is given by $N_H \times N_W$. We call N_W tiles in row a "subband". A tile is the primitive for magnitude modification and contains several frequency components of four consecutive DFT frames. The N_T tiles in a pattern block share a synchronization signal and N_B bits (Fig. 2(a)). In the figure, N_H and N_W are 6 and 4, respectively. The synchronization signal is needed so the detection process can search for the beginning of a pattern block. It will be explained in Sect. 2.3 why this search is not computationally expensive. Hereafter, the subscript 'S' is used for the synchronization signal. One bit is encoded in D_B tiles, and D_S tiles are used for the synchronization signal. Therefore we can write $N_T = D_B \times N_B + D_S$. A pseudo-random number corresponds to a tile; the embedding algorithm slightly modifies the magnitude of a tile according to the pseudo-random number assigned to the tile, and the detection algorithm correlates the magnitudes of tiles with the pseudo-random numbers. Therefore the position

* We define a "watermark signal" as a signal that is added onto the original host signal to obtain the watermarked signal.

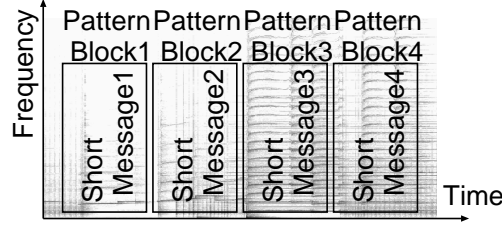


Figure 1. A pattern block has a two-dimensional area in the time-frequency plane and conveys a short message.

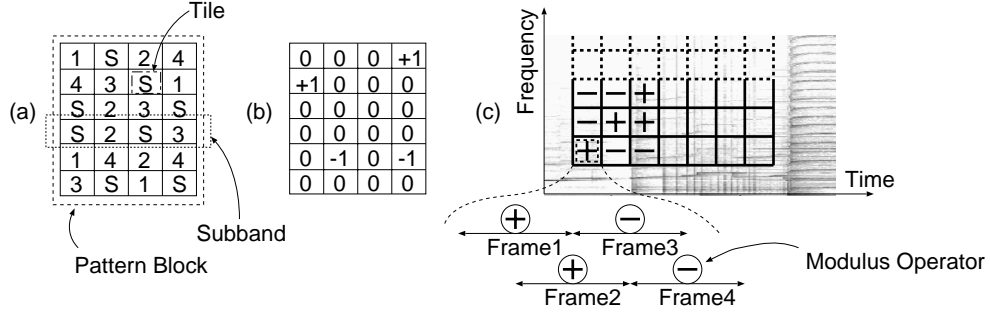


Figure 2. A pattern block consists of tiles and is shared by the synchronization signal and bits. A tile contains four overlapping DFT frames.

and value of the pseudo-random array is a sort of a secret key that must be shared by the embedder and the detector. We suppose ω_j^B is a pseudo-random array for the j -th bits, and $\omega_{j,k}^B$ is the k -th element of the array with a value of $+1$ or -1 . On the other hand, ω^S is the pseudo-random array for the synchronization signal. Fig. 2(a) illustrates a pattern block with four bits and a synchronization signal, and (b) is an example of ω_4^B of (a).

Using a two-dimensional pseudo-random array, the method is robust against time fluctuation caused by duplicated or missing of audio samples. This is because the loss in some portions of the array can be recovered by the other portions. The duration of a pattern block is an important factor for robustness of the method. That will be shown in Sect. 3. Another advantage of the two-dimensional array is security. Varying the modification pattern of the magnitudes from frame to frame makes it difficult for crackers to analyze the secret pseudo-random array. A tile has four frames of discrete Fourier transforms (DFT) each of which overlaps the adjacent frames by a half window (Fig. 2(c)). The embedding algorithm increases or decreases the amplitudes of the four frames in a tile according to the sign of the pseudo-random value multiplied by the “Modulus operator”,

$$C_t = (+1, +1, -1, -1). \quad (1)$$

The t -th element of the sequence corresponds to the t -th frame of the four frames in a tile. The meaning of the modifiers is that, if the pseudo-random value assigned to a tile is positive, the embedding algorithm increases the magnitudes of frequencies of the former two frames in the tile and decreases those of the latter two frames. This is done to allow the detector to weaken the effect of the host signal on the watermark strength by the subtracting magnitudes of adjacent frames, while keeping the effect of the watermark signal high. More detail will be provided in Sect. 2.3.

2.2. Embedding Algorithm

The embedding algorithm calculates a watermark signal in the frequency domain, converts it to the time domain using an inverse DFT (IDFT) and adds it into the host signal (Fig. 3). The amplitudes of the watermark signal are calculated using a psychoacoustic model. Whether amplitudes of the host signal are increased or decreased is decided by a pseudo-random array and the bits to be embedded. The embedding algorithm uses the same phases as the phases of the host signal for constructing the watermark signal to avoid introducing unnecessary phase change.

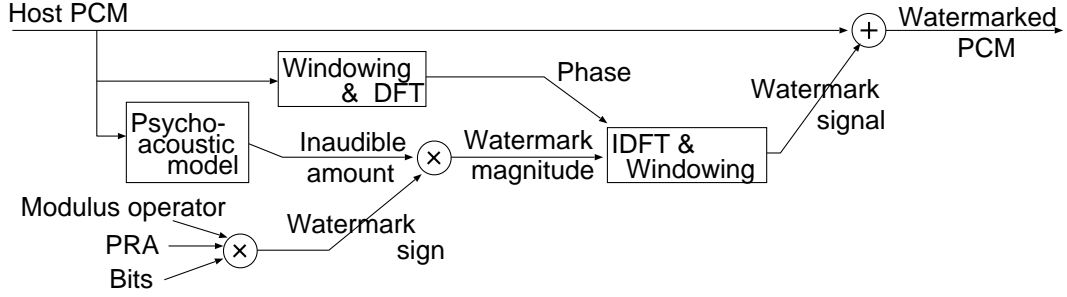


Figure 3. Diagram of watermark embedding algorithm

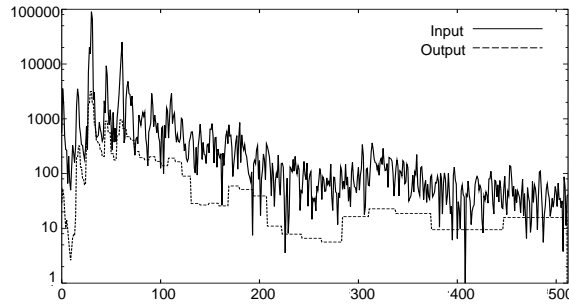


Figure 4. Input and output of the psychoacoustic model

- Amplitude

The inaudible level of the amplitude modification is calculated using a psychoacoustic model. An example of input and output of the psychoacoustic model is shown in Fig. 4. We indicate this amount of the w -th frequency of the t -th frame in a pattern block by $a_{t,w}$.

- Sign

The following rule decides whether the amplitudes of frequencies in a tile of the pattern block are to be increased or decreased. We suppose that the tile corresponds to the k -th element of the pseudo-random array for the j -th bit in the pattern block. The value of the element is $\omega_{j,k}^B$ and the value of the bit is B_j . Then the sign of the amplitude modification for the f -th frequency of the t -th frame of the pattern block is given by

$$s_{t,f} = B_j \times C_{(t \bmod 4)+1} \times \omega_{j,k}^B, \quad (2)$$

where the tile includes the f -th frequency of the t -th frame.

- Phase

The phases $\theta_{t,f}$ of the watermark signal are taken from the DFT analysis of the frame of the host signal. A frame consists of N_{PCM} consecutive pulse code modulation (PCM) samples and overlaps the adjacent frames by a half window. The samples should be multiplied with a windowing function such as a sine window before evaluating the DFT.

Using $s_{t,f}$, $a_{t,f}$, and $\theta_{t,f}$, the watermark signals in the frequency domain are reconstructed. The real and imaginary parts of the signal are calculated by $F_{t,f}^R = s_{t,f} a_{t,f} \cos \theta_{t,f}$ and $F_{t,f}^I = s_{t,f} a_{t,f} \sin \theta_{t,f}$, respectively. The watermark signal in the time domain is obtained by transforming these using IDFTs. To avoid generating clicking sounds at the borders of adjacent IDFT frames, the watermark signal is multiplied by a windowing function and overlapped with the adjacent frames after each IDFTs. Finally, the watermarked PCM samples are obtained as the summation of the host signal and the watermark signal in the time domain.

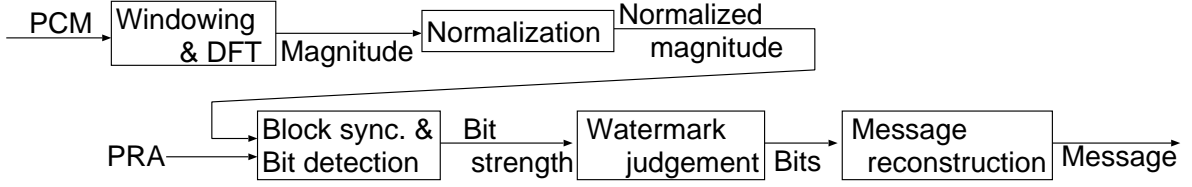


Figure 5. Diagram of watermark detection algorithm

2.3. Detection Algorithm

The detection algorithm calculates the magnitudes for all tiles of the content and correlates them with the pseudo-random array (PRA) by applying the following steps (Fig. 5).

1. Windowing DFT

The magnitude $a_{t,f}$ of the f -th frequency in the t -th frame of a pattern block of the content is calculated by the DFT analysis of a frame of the host signal. A frame consists of N_{PCM} consecutive PCM samples and overlaps the adjacent frames by a half window. The samples should be multiplied with a windowing function such as a sine window before the DFT.

2. Normalization

The magnitudes are then normalized by the average of the magnitudes in the frame so that contributions of all frames to the watermark strength are equal. A normalized magnitude is

$$\hat{a}_{t,f} = \frac{a_{t,f}}{\frac{1}{N_{PCM}/2} \sum_{f=1}^{N_{PCM}/2} a_{t,f}}. \quad (3)$$

The logarithmic magnitude $P_{t,f}$ is calculated from a normalized magnitude as $P_{t,f} = 20 \log_{10} \hat{a}_{t,f}$. The difference between the magnitudes of a frame and the frame after the next one is taken as $\hat{P}_{t,f} = P_{t,f} - P_{t,f+2}$. Since $P_{t,f}$ and $P_{t,f+2}$ have similar values in most cases, this subtraction weakens the influence of the host signal on the detected watermark strength. On the other hand, it increases the effect of the watermark signal because the opposite signs are embedded into the two-adjacent frames due to the modulus operator, C_t .

3. Magnitude of tile

The magnitude of a tile located at the b -th subband of the t -th frame in the block is calculated by

$$Q_{t,b} = \frac{\sum_{f=f_b^L}^{f_b^H} \hat{P}_{t,f}}{f_b^H - f_b^L + 1}, \quad (4)$$

where f_b^L and f_b^H are the lowest and highest frequencies in the b -th subband, respectively.

4. Pattern block synchronization

Because there is a possibility that the content has been trimmed before detection, the start of the content is not necessarily the beginning of a pattern block. Therefore, the beginning of a pattern block has to be located. We call this procedure “pattern block synchronization”.

Note that the minimum step of the search is a frame, which is much larger than a PCM sample. If a search is required for all samples, it would be very computationally expensive. However, pattern block synchronization requires only $4N_W$ calculations of synchronization strength, and additional DFTs are unnecessary because the search is after the DFTs.

First, a “pattern block synchronization strength”, S_s , is calculated for each frames based on the assumption that the s -th frame is the beginning of the pattern block. The frame that gives the maximum synchronization strength is accepted as the beginning of the pattern block. The S_s is defined as

$$S_s = \frac{\sum_{k=1}^{D_s} \omega_k^S (Q_{t+s,b} - \bar{Q})}{\sqrt{\sum_{k=1}^{D_s} \{\omega_k^S (Q_{t+s,b} - \bar{Q})\}^2}}, \quad (5)$$

where

$$\bar{Q} = \frac{1}{D_s} \sum_{k=1}^{D_s} Q_{t,b}, \quad (6)$$

and ω_k^S is the k -th pseudo-random number for the synchronization signal corresponding to the tile at b -th subband in the $(t+s)$ -th frame.

Synchronization strengths are calculated with s from 0 up to $4N_W - 1$ because a pattern block has N_W tiles in row and a tile contains 4 frames. The s satisfying the following equation is the synchronization position.

$$S_s = \max_{s=0}^{4N_W-1} S_s \quad (7)$$

Assuming that several consecutive pattern blocks have synchronization positions that are separated by the same number of frames, then the successful synchronization rate can be improved. This “linear assumption method” searches for synchronization positions for consecutive N_S pattern blocks at the same time by searching the number of frames in a pattern block, d , and the synchronization position for the first pattern block, s . The “multiple pattern block synchronization strength” is given by

$$\bar{S}_{d,s} = \frac{1}{N_S} \sum_{u=1}^{N_S} S_{u \times d + s}. \quad (8)$$

If the pitch of the content is known not to have been shifted, d must be within $4N_W$, and the search for d is not necessary. Similarly, if the content has not been trimmed, s must be 0. The d and s satisfying

$$\bar{S}_{d,s} = \max_{d=d_{lower}}^{d_{upper}} \max_{s=0}^{4N_W-1} \bar{S}_{d,s}, \quad (9)$$

give the synchronization positions for the blocks, where d_{lower} is the shortest possible length of a block, and d_{upper} the longest possible. The resulting synchronization position for the u -th block is $u \times d + s$.

In a case where the true synchronization positions for consecutive pattern blocks are shifted, choosing a local maximum value of S_s within a few neighboring frames improves the reliability of synchronization. We call this procedure “local adjustment”.

5. Bit detection

“Bit strengths”, X_j , are calculated at the obtained synchronization position.

$$X_j = \frac{\sum_{k=1}^{D_B} \omega_{j,k}^B (Q_{t+s,b} - \bar{Q})}{\sqrt{\sum_{k=1}^{D_B} \{\omega_{j,k}^B (Q_{t+s,b} - \bar{Q})\}^2}} \quad (10)$$

$\omega_{j,k}^B$ is the k -th pseudo-random number for j -th bit. The sign of X_j indicates the value of the j -th bit in the pattern block.

$$B_j = \begin{cases} 1 & (X_j \geq 0) \\ 0 & (X_j < 0) \end{cases} \quad (11)$$

Note that the distribution of bit strengths, X_j , for unwatermarked content can be approximated by a standard Gaussian distribution due to the Central Limit Theorem.

6. Watermark decision

Decision on whether the content has been watermarked or not can be done at this stage using X_j or S_s . Though various methods can be considered, we used the following rule for the tests in Sect. 3.

If the content has not been watermarked, the distribution of the bit strengths can be approximated by a standard Gaussian distribution. It is known that the distribution of the standard deviations of Gaussian variables follows a χ^2 distribution. Therefore the decision can be made by examining the hypothesis that the standard deviation of bit strengths over a predefined period follows a χ^2 distribution.

Furthermore, when the number of bits that is weaker than a predefined threshold exceeds a certain number, the detector should not output a message. This is because it is likely that there are too many bit errors to recover using error-correcting codes.

7. Reconstruction of the multiple-bit message

Finally, the multiple-bit message is reconstructed from the detected bits. Error correction and detection should also be performed at this stage.

3. THEORETICAL AND EXPERIMENTAL ANALYSIS

Theoretical and experimental analysis of the robustness of the method is shown using an experimental system. We also discuss the crucial parameters for robustness.

3.1. Parameter Design

We implemented the method in a software system that can embed and detect a 64-bit message in 30-second pieces of music. The message is encoded in 128 bits using a Cyclic Redundancy Check (CRC) code and a Bose-Chaudhuri-Hocquenghem (BCH) code. Each pattern block has 4 bits embedded, and the block has 30 columns and 9 rows of tiles. The bandwidths of the 30 frequency subbands are described in Sect. 3.3. 150 tiles out of the 270 tiles are assigned for the synchronization signal, 30 tiles for a bit. For the pattern block synchronization, 7 consecutive blocks are used for the linear assumption method. For the local adjustment, the four neighboring frames are evaluated. The length of a DFT frame is 1024 samples. A sine window is used for windowing the DFT frames. The thresholds for deciding whether the content is watermarked or not are set so that the false positive error ratio is under 10^{-6} .

Three pieces of music are used for the analysis: a violin sonata by Bach, a symphony by Debussy, and a female jazz vocalist with strings and a piano. All signals are sampled at a frequency of 44.1 kHz, and each piece is 100 seconds long. The resulting Signal-to-Noise ratio by watermark embedding was 35.0dB on average.

3.2. Effect of Amplitude Modification

One of the important characteristics of the method is modifying magnitudes that are independent of phases. Because magnitudes are less influenced than phases by displacement of the analysis windows, the watermark can be detected even after cropping.

Figure 6(a) shows an example of magnitudes of a frequency component before and after watermark embedding. The abscissa is the number of samples by which the observing DFT window is displaced, and the ordinate is the observed magnitude. The arrows and signs below the graph indicates the observing DFT windows and signs used for embedding. It can be seen that the observed magnitudes of the frequency component smoothly change along the time axis and that the effect of watermark embedding are maintained even at intermediate positions. This is the reason the detection algorithm does not need a sample-by-sample exhaustive search for the embedding windows. Figure 6(b) shows the average of the detected bit strength[†], \bar{X}_j , for the displacement from 0 up to 512 samples, which is the interval between two consecutive DFT frames. The average drops slightly but remains sufficiently high up to 512 samples of displacement. Note that the next frame will be selected for more than 255 samples of displacement by the pattern block synchronization process.

[†] To obtain a meaningful average, 1 is embedded for all watermark bits.

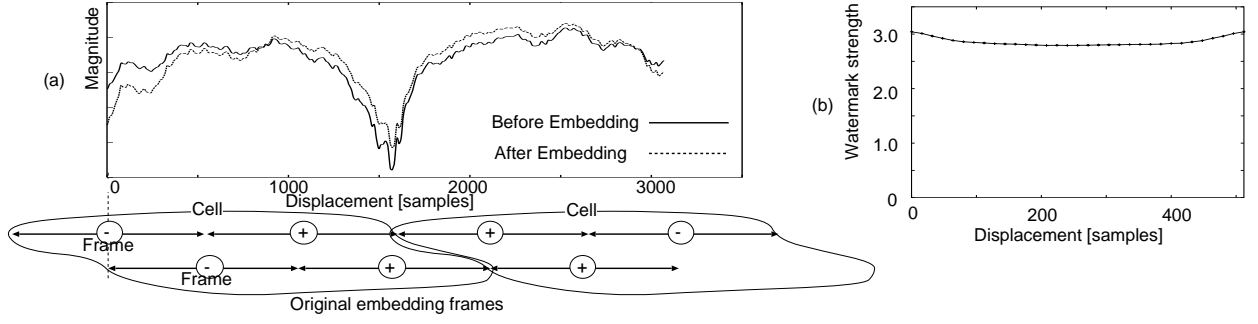


Figure 6. Robustness against displacement of detection windows

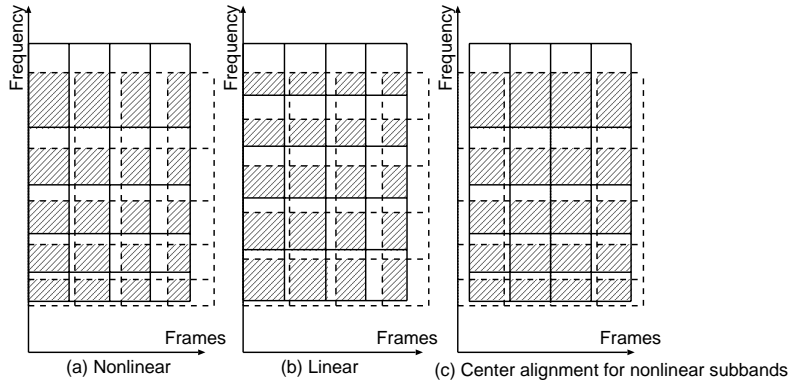


Figure 7. Design of pattern blocks and its influence on robustness

3.3. Bandwidth

The design of the pattern blocks is crucial for robustness. When the content is transformed somehow, the shapes of the blocks are frequently changed. When the pitch of the content is shifted upward, the duration of a block becomes shorter and the frequencies of the block become higher. When wow-and-flutter affects the content, the duration and the height of the blocks becomes different. In these cases, the detector cannot synchronize the pseudo-random array to the blocks that were embedded in the content, and the detected watermark strength consequently decreases. In order for detection succeed in spite of these sorts of signal transformations, pattern blocks must have a shape that maintains the correspondence between the embedded watermark signal and the pseudo-random array.

The first design parameter for a pattern block is the bandwidth for the tiles. When the pitch of the content is shifted by a rate of p , the lowest frequency, f_b^L , and the highest frequency, f_b^H , of the b -th subband become pf_b^L and pf_b^H , respectively, and hence frequency components over f_b^H no longer contribute to the subband. Therefore the contribution of the subband becomes $(f_b^H - pf_b^L)/(f_b^H - f_b^L)$ times the original contribution. With $r_b = \frac{f_b^H}{f_b^L}$, the degradation rate can be expressed by

$$\frac{f_b^H - pf_b^L}{f_b^H - f_b^L} = \frac{r_b - p}{r_b - 1}, \quad (12)$$

which is independent of f_b^L . For this reason, by using the same r value for all r_b , the contributions of all subbands degrades at the same rate, which is better than allowing some subbands to degrade more rapidly than other strong subbands. The dotted lines of Fig. 7(a) illustrates tiles with a common r value, and the solid line is the shape of the tiles after the pitch of the content has been shifted. The hatched area of the figure is the area maintaining the correspondence and hence still contributing to the detected watermark strength. On the other hand, (b) illustrates

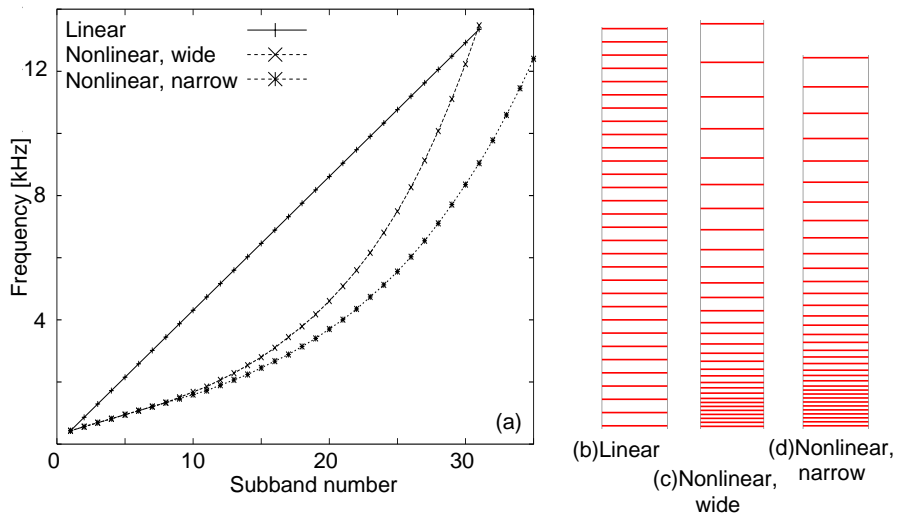


Figure 8. Three subband designs

another pattern block where every subband has the same bandwidth. The hatched areas at high frequencies are smaller than at low frequencies.

The detected watermark strengths, X_j , are proportional to the hatched areas. This is because the numerator of Eq. 10 is proportional to the maintained correspondence between $\omega_{j,k}^B$ and the pseudo-random array embedded in the $Q_{t,b}$ s while the denominator of X_j is basically independent of the correspondence.

Note that the actual correspondence sought by the pattern block synchronization process is expected to be Fig. 7(c) instead of Fig. 7(b). This is because Fig. 7(c), where the centers of the original and shifted pattern blocks match, maximizes the hatched area and the synchronization process finds the position that maximizes this area.

A robustness test was conducted using three types of subbands shown in Fig. 8. Every subband in the linear subbands shown in Fig. 8(b) has the same width of 10 frequency bins. The nonlinear subbands have wider bandwidths in higher frequencies by the following rule: (1) the lowest subband begins at the 10th bin; and (2) the bandwidth is the smallest integer with r_j larger than r_{min} , but not less than 3 bins. The value of r_{min} is 0.1 for the nonlinear wide subbands (NWS) shown in Fig. 8(c), and 0.08 for nonlinear narrow subbands (NNS) shown in Fig. 8(d). While linear and NWS have 30 subbands, NNS is given 34 subbands so that it contains approximately same number of frequency bins. Figure 8(a) shows the relationships of subband numbers versus frequencies for the three designs.

Fig. 9(a) shows experimental results on the degradation of the average detected bit strengths for these subband designs. Pitch shifting is performed using linear interpolation without anti-alias filtering. linear subbands mark stronger watermark strengths for the content just after embedding but degrades more rapidly than nonlinear subbands. NWS shows even more robustness than NNS. The right hand graph of Fig. 9 shows the same experiment differently. The abscissa is the area maintaining the correspondence, which is estimated by a simple geometrical calculation that calculates the hatched area illustrated in Fig. 7(c). It can be seen that the detected watermark strengths are proportional with the correspondence rate as long as the rate is high enough. We consider the influence of the modulus operators, which have opposite values for a frame and for the second frame after the frame, is the reason that the strengths are lower than expected when the pitch-shifting rate is high.

3.4. Duration of Pattern Block

The duration of a pattern block, N_W , is another important parameter. To clarify its influence on the robustness, we implemented four systems using different values of N_W 7, 9, 11, and 13, and examined the degradation of the watermark strength from pitch shifting (Fig. 10). The number of tiles for a bit (D_B) is 24, 30, 36 and 42, respectively. Figure 10(a) shows that larger N_W results in stronger watermark strengths for the content just after embedding. The strength is proportional to the square root of the number of tiles for a bit (Fig. 10(c)), as is theoretically

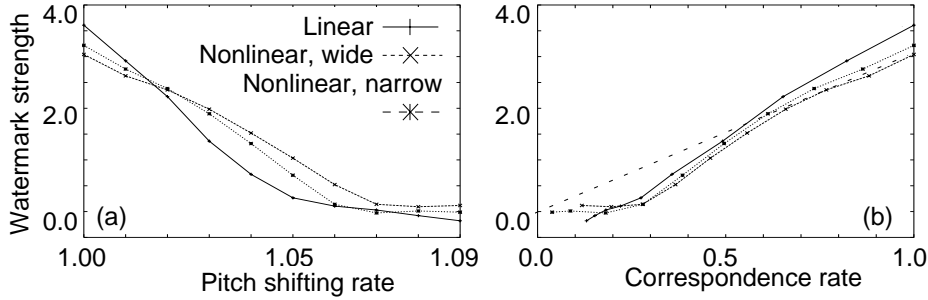


Figure 9. Robustness against pitch shifting

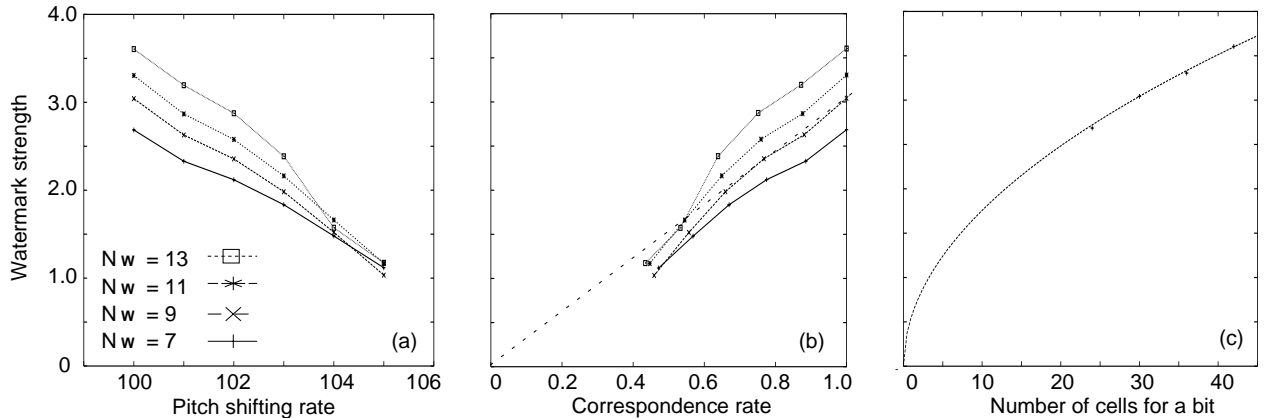


Figure 10. The influence of the duration of a pattern block on the robustness

expected. The reason is that the numerator of Eq. 10 increases proportionally with the number of tiles, while the denominator increases proportionally with the square root of the number of tiles, and hence the average of X_j increases proportionally with the square root.

Furthermore, Fig. 10(a) shows that larger pattern block duration does not result in better robustness for high pitch-shifting rates. This can be explained as follows: in the case of detecting highly pitch-shifted content using large N_W , mismatches between the original and shifted tiles accumulate over the longer duration, so tiles at both ends of the pattern block do not contribute to the watermark detection. Also the effect of the modulus operators occurs earlier. On the other hand, too small a value of N_W leads to a risk of missing a whole pattern block. The duration of pattern blocks should be set considering what sort of degradations the system must be robust against. The relationship of the correspondence rate and the watermark strengths is shown again in Fig. 10(b).

3.5. Robustness

We tested the robustness of the method against several sort of degradations. The robustness is measured by the degradation of the detected watermark strength (Fig. 11 and Fig. 12) and the ability to detect the correct 64-bit message (Table 1). For the statistical experiment, ten 100-second music samples are used. Since the message is expected to be detected three times in a 100-second music sample, “100%” in Table 1 indicates 30 correct detections of the message from the ten samples. In the table, “Original watermark” means no transformation is performed on the content after watermark embedding. “Wow-and-flutter” is a combination of two consecutive transformations: (1) “wow” is a computer simulation of a 0.707% variation of playback speed with a 5 Hz cycle time; and (2) “flutter” is a computer simulation of 0.707% variation of playback speed at a random modulation frequency up to 250 Hz. “Echo 50 msec 0.3” is echoing with maximum delay 50 msec and feedback coefficient 0.5. “Random stretching” is a

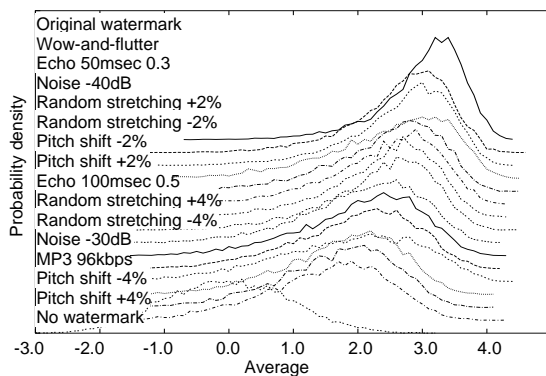


Figure 11. The shapes of the probability density functions of watermark strength detected from variously transformed content

transformation that modifies the length of the content to the target length by omitting or inserting a random number of samples from 50 up to 500. Random sample cropping can be considered as random stretching with the target length smaller than 100%.

Table 1. Correct detection rate of the 64-bit message

Original watermark	100%	Noise -30dB	87%	Random stretching -4%	100%
Wow-and-flutter	100%	MP3 96kbps	100%	Pitch shifting +4%	90%
Echo 50m sec 0.3	100%	Random stretching +4%	100%	Pitch shifting +2%	100%
Echo 100m sec 0.5	97%	Random stretching +2%	100%	Pitch shifting -2%	100%
Noise -40dB	97%	Random stretching -2%	100%	Pitch shifting -4%	83%

Correct detection rates over 80% were seen for every one of the tested degradation. The error correction and detection algorithm successfully avoided detection of a wrong message.

Figure 11 shows the shapes of the probability density functions of the watermark strength detected from each degradations. The relationship of the average versus the standard deviation of each distribution is shown in Fig. 12. This figure clarifies the transition path along which the average and standard deviation move from the strongest “Embedded” status to the status where the content is completely degraded and the average and standard deviation are expected to have values of 0.0 and 1.0, respectively. The standard deviations for noise addition have large values. This is because the noise-addition software adds a noise signal with the same sound level through the entire music sample and hence the watermark strengths detected from the portions with soft sounds are seriously degraded while watermark strengths detected from the portions with loud sounds remain high values.

4. CONCLUSION

We presented watermarking method that can embed a 64-bit message onto a 30-second music sample. We tested the robustness of the method against wow-and-flutter, echo, noise addition, MP3 compression, random cropping, and pitch shifting. The method modifies the magnitudes of segmented areas in the time-frequency plane of the content according to a two-dimensional pseudo-random array assigned to the areas. Windowing and overlapping were used before and after DFTs in order to avoid generating clicking sounds at the borders of adjacent DFT frames. It was shown that the effect of the magnitude modification by the embedding algorithm was observable by the detection algorithm with displaced DFT windows. This makes the method robust against random cropping without computationally expensive searching for the embedding DFT windows.

The correspondence between the embedded watermark and the pseudo-random array used for detection plays an important role in determining the detected watermark strength. The watermark strength detected from pitch-shifted

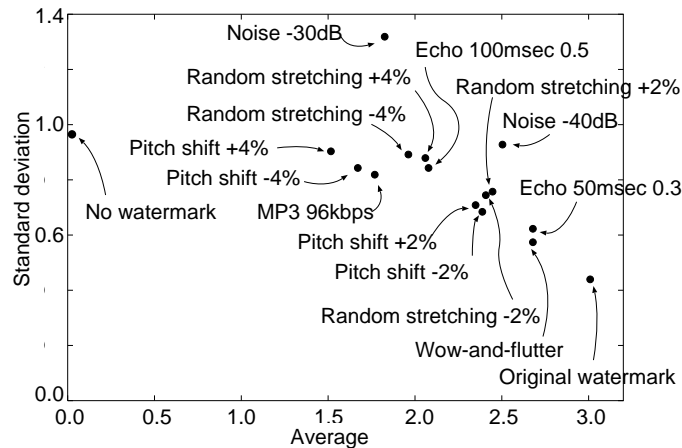


Figure 12. The watermark strength detected from variously transformed content

content was successfully estimated by a simple calculation of the area maintaining the correspondence. To keep the correspondence rate high, it was better to design subbands having wider bandwidths for higher frequencies.

The duration of the pseudo-random array is also important. The duration can be lengthened so that the loss in some portion of the array can be recovered by the other portion. Actually the watermark strengths increased proportionally with the square root of the length of the array. However it was shown that too long a duration was not effective for robustness against transformations that change the length of the content.

Further improvement is required to achieve further robustness against excessive distortions and to shorten the duration of content required to carry a message.

REFERENCES

1. L. Boney, A. H. Tewfik, and K. N. Hamdy, "Digital watermarks for audio signals," in *IEEE Int. Conf. on Multimedia Computing and Systems*, 1996.
2. M. D. Swanson, B. Zhu, A. H. Tewfik, and L. Boney, "Robust audio watermarking using perceptual masking," *Signal Processing* **66**, 1998.
3. P. Bassia and J. Pitas, "Robust audio watermarking in the time domain," in *Proceedings of EUSIPCO '98*, 1998.
4. M. Arnold and S. Kanka, "MP3 robust audio watermarking," in *DFG V^{III}D^{II} Watermarking Workshop*, 1999.
5. C.-P. Wu, P.-C. Su, and C.-C. J. Kuo, "Robust and efficient digital audio watermarking using audio content analysis," in *SPIE Conf. on Security and Watermarking of Multimedia Contents II*, 2000.
6. J. F. Tilki and A. A. L. Beex, "Encoding a hidden digital signature onto an audio signal using psychoacoustic masking," in *7th Int. Conf. on Signal Processing Applications & Technology*, 1996.
7. J. J. O. Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," *Signal Processing* **66**, 1998.
8. M. Kutter, "Watermarking resisting to translation, rotation, and scaling," in *Proc. of SPIE, Multimedia Systems and Applications*, 1998.
9. C.-Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui, "Rotation, scale, and translation resilient public watermarking for images," in *SPIE Conf. on Security and Watermarking of Multimedia Contents II*, 2000.
10. W. W. Peterson and E. J. W. Jr., *Error-correcting codes*, MIT Press, 1988.