Signal Processing 91 (2011) 1973–1984

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

# Blind and robust audio watermarking scheme based on SVD–DCT

Bai Ying Lei *, Ing Yann Soon, Zhen Li

*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore*

ARTICLE INFO

*Article history:*
Received 28 July 2010
Received in revised form
22 February 2011
Accepted 3 March 2011
Available online 11 March 2011

*Keywords:*
Audio watermarking
Singular value decomposition
Robust
Blind
Synchronization

ABSTRACT

Singular value decomposition (SVD) is a new and important transform technique in robust digital watermarking due to its different properties from the traditional transforms such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). In this paper, we propose a new, blind and robust audio watermarking scheme based on SVD–DCT with the synchronization code technique. We embed a binary watermark into the high-frequency band of the SVD–DCT block blindly. Chaotic sequence is adopted as the synchronization code and inserted into the host signal. Experimental results show that the proposed watermarking method is comparable to, if not, better than SVD based method and several selected typical audio watermarking methods, even in the presence of various common signal processing attacks.



## 1. Introduction

With the development and popularity of the Internet, duplication, modification and forgery of digital multimedia have become difficult to detect and prevent. Therefore, the issue of intellectual property rights protection and digital right management has attracted a lot of attention [1,2]. The most commonly adopted solution to the problem of copyright protection of multimedia documents is digital watermarking [3,4], which refers to the process of embedding the copyright information into a cover object. There have been intensive efforts in securing multimedia data by watermarking [5].

In general, the digital watermarking scheme can be broadly classified into robust and fragile (or semi-fragile) watermarking. For robust audio watermarking algorithms, two important issues need to be addressed. One is to provide trustworthy evidence to protect rightful ownership, and the other is to achieve an appropriate trade-off between imperceptivity and robustness against distortions due to common audio manipulations and synchronization attacks [6,7]. SVD is a powerful and desirable transformation technique for robust watermarking and has been widely applied in robust image watermarking [8–10]. Recently, SVD has been applied to robust audio watermarking [11–17] by transforming the host audio signal into a matrix with non-negative scalar entries from the perspective of linear algebra.

As we know, watermarks are generally not placed in perceptually insignificant regions of the audio signal, because many common signal and geometric processing affect these components [18]. Hence, the watermark information is embedded only in the largest singular values (SVs) which correspond to the energy of the most perceptually significant regions in the original audio. Robust watermarking methods for audio often combine short-time Fourier transform (STFT), DCT, Discrete Sine Transform (DST) or DWT with SVD [8,11–13]. However, in these methods, SVD is only an auxiliary tool to modify the transformed coefficients of STFT, DCT, DST or DWT. Effectively, such methods are essentially STFT, DCT, DST or DWT based in that these transforms are performed on the cover object directly. Therefore, the superior characteristics of SVD for robust watermarking are not fully exploited. The difficulty of applying SVD to the host audio directly is due

* Corresponding author. Tel.: +6567906546; fax: +6567911604.
*E-mail addresses:* leib0001@e.ntu.edu.sg (B.Y. Lei), eiysoon@ntu.edu.sg (I.Y. Soon), lizh0019@e.ntu.edu.sg (Z. Li).

0165-1684/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2011.03.001

to the fact that SVD transformed values are very sensitive and difficult to be manipulated by other transforms.

In this paper, we present a blind and robust audio watermarking scheme which capitalizes on the combined features of the SVD, DCT and synchronization code technique. As the distortion in the largest coefficients will not cause much effect and can hold against general signal processing attacks, it would be sufficient to embed the watermark with only a slight modification. Watermark detection is efficient and blind in the sense that only the secret key for encrypting the embedded locations but not the original audio is required. Unlike the traditional SVD based watermarking schemes where the watermark bits are embedded directly on the SVs, the proposed scheme is based on bit embedding on the high-frequency DCT coefficients of the block of SVs obtained by SVD transformed audio subblocks. To improve the fidelity and the perceptual quality of the watermarked audio, we also model the adaptive frequency mask which takes into account the Human Auditory System (HAS) [2,19] to control the strength of the embedded local watermark.

The remainder of this paper is organized as follows. In Section 2, related work of typical SVD watermarking algorithms are introduced. Some basic SVD principles are given in Section 3. In Section 4, the proposed robust watermarking method is described in detail followed by a brief description of the watermark extraction in Section 5. Section 6 discusses the error probability issues. The experimental results and performance analysis are provided in Section 7. Finally, Section 8 concludes the paper.

## 2. Related work

For the robust SVD based audio watermarking scheme, most algorithms are non-blind or semi-blind [11–13]. The typical schemes amongst them are the semi-blind watermarking method based on STFT [11] and non-blind watermarking method adaptable to different domains in [13]. For instance, in [11], Ozer et al. first proposed a non-oblivious robust watermarking based on SVD of the spectrogram of the signal. Though the audio watermarking technique is robust to some common signal processing attacks, it was a semi-blind technique and did not adopt the synchronization technique. Therefore, its robustness performance to synchronization attacks such as Flipsample, Copysample and Cutsample attacks are relatively poor and unsatisfactory.

An efficient SVD based algorithm for digital audio watermarking is presented in [13], which is domain adaptive as it can be applied in DCT, DWT and DST domains according to different requirements. Moreover, the proposed segment-by-segment implementation method has enhanced detectability compared to the simple implementation on the whole host audio signal directly. Security is increased with the use of the chaotic encrypted watermarks. However, this permutation and spread spectrum based audio watermarking scheme is not blind as the original signal and other information are needed in the watermark detection process.

The most recent SVD based blind audio watermarking method is proposed by Bhat et al. [12]. It is based on the quantization index modulation in the wavelet domain using the synchronization code technique. This quantization technique is found to be high in capacity and robust to lots of attacks. However, the imperceptibility is not very good as shown by the Signal-to-Noise Ratio (SNR) and Objective Difference Grade (ODG) results. The SNR results are just a little bit above 20 dB. Furthermore, results on the detailed synchronization attacks such as jittering, amplitude variation and time scale modification are not reported in this scheme.

There are other SVD based audio watermarking schemes applied to the other domains in the literature for different purposes [14–17]. For example, Zezula and Misurec proposed an audio watermarking algorithm in the Modulated Complex Lapped Transform (MCLT) domain based on SVD [16].

Actually, compared to image and video signals, audio signals are represented by fewer samples per time interval. Moreover, an additional problem in audio watermarking is that HAS is much more sensitive than the Human Visual System (HVS), and that inaudibility is much more difficult to achieve than the invisibility of image watermarking [2]. Therefore, the image watermarking schemes based on SVD have gained more attention and interest and there are a lot of SVD and SVD–DCT based image watermarking algorithms [20–25]. For example, in [8], the semi-blind pure SVD based watermarking scheme is proposed by Liu and Tan in 2002. In this scheme, the SVD matrices of the reference watermark have been found to be easily discredited so this scheme cannot be used for copyright protection [9,24]. There are also some blind watermarking methods which embed watermarks in SVs of the transformed coefficients [20]. In both [21] and [22], the intrinsic property of orthogonality of the SVD matrices is altered. As a result, the extracted watermark cannot be guaranteed to be correct even if no attacks are performed.

## 3. SVD watermarking

The traditional transform techniques such as DST, DCT and DWT just decompose a signal in terms of a standard basis set, which is not an optimal representation in some sense. As a kind of orthogonal transforms and a numerical technique for diagonalizing matrix, SVD [26] is a numerical technique for linear algebraic in the transformed domain comprising basis states that are optimal in some sense. 1-D audio signal can be reshaped to 2-D and viewed as a non-negative real matrix. The SVD of the $N \times N$ matrix $I$ can be described as follows:

$$I = USV^T = [u_1, u_2, \ldots, u_N] \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} \quad (1)$$

where $U \in R^{N \times N}$ and $V \in R^{N \times N}$ are unitary matrixes, $S \in R^{N \times N}$ is a diagonal matrix and the superscript $T$ denotes matrix transposition. The diagonal elements of $S$ are called the SVs of $I$ and assumed to be arranged in decreasing order, that is, $\lambda_1 > \lambda_N$. The columns of the $U$ matrix are called the left singular vectors while the columns of the $V$ matrix are called the right singular vectors of $I$. In fact, the slight modification of the components in matrix $S$ rarely affects the perception features of cover object which can be exploited for audio watermarking technique to meet the transparency as well as the robustness requirements.

In SVD based watermarking, a frame is treated as a matrix decomposed into three matrices. Calculating the SVD consists of finding the eigenvalues and eigenvectors of $II^T$ and $I^TI$. The eigenvectors of $I^TI$ consist of the columns of matrix $V$, and the eigenvectors of $II^T$ compose the columns of matrix $U$. The SVs in matrix $S$ are square roots of eigenvalues from $II^T$ or $I^TI$. From Eq. (1), we know that the host audio reshaped to 2-D can be interpreted as a summation of $N$ layers, each termed as an eigenlayer. The SVs indicate the energy intensity in its corresponding eigenlayer.

## 4. Watermark embedding

### 4.1. Synchronization code

The synchronization code [6,7,19,27,28] is used to locate the position of hidden informative bits, thus resisting the cropping and shifting attacks. Suppose $\{Syn(k)\}$ is an original synchronization code and $\{Seq(k)\}$ is an unknown sequence with the same length. If the number of different bits between $\{Syn(k)\}$ and $\{Seq(k)\}$, when compared bit by bit, is less than or equal to a predefined threshold $T$, the sequence $\{Seq(k)\}$ will be determined as the synchronization code. The advantage of embedding the synchronization code in the time domain is the low cost incurred in searching. Therefore, we embed the synchronization code in the time domain to reduce the computation cost. Before embedding, the synchronization code $\{Syn(k)\}$ should be arranged into a binary data sequence.

In our scheme, the host signal is partitioned into two portions for synchronization code insertion and watermark embedding. Synchronization insertion involves splitting the audio host signal into proper segments according to the length of $\{Syn(k)\}$. This algorithm utilizes a chaotic sequence as the synchronization code in front of the watermark to locate the position where the watermark is to be embedded. The logistic chaotic sequence with initial value in the interval [0,1] is selected to generate the synchronization code and denoted as

$$y(k+1) = g(y(k)) = \lambda y(k)(1-y(k)) \tag{2}$$

where $3.57 < \lambda \le 4$, $y(k)$ is mapped into the synchronization sequence $\{Syn(k) | k = 1,\ldots,Lsyn\}$ with the following rule:

$$Syn(k) = \begin{cases} 1 & \text{if } y(k) > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The synchronization code insertion part is cut into $Lsyn$ audio segments and each audio segment has $P$ samples denoted as

$$SA(k) = A(k \bullet P + u), \quad 1 \le k \le Lsyn, 1 \le u \le P \tag{4}$$

Then each bit of the synchronization code is embedded into each $SA(k)$ as follow:

$$SA'(k) = \begin{cases} round\left(\frac{SA(k)}{Q}\right) \bullet Q, & \text{if } Syn(k) = 1 \\ floor\left(\frac{SA(k)}{Q}\right) \bullet Q + \frac{Q}{2}, & \text{if } Syn(k) = 0 \end{cases} \tag{5}$$

where $Q$ denotes the embedding strength, $round(\cdot)$ means rounding to the nearest integer, and $floor(\bullet)$ is rounding to the minus infinity.

After embedding, the embedded and attacked signal $SA''(k)$ is also split into $Lsyn$ segments, and then the synchronization code is extracted by the following rule:

$$Syn'(k) = \begin{cases} 0, & \text{if } \frac{Q}{4} \le \mathrm{mod}(SA''(k), Q) < \frac{3Q}{4} \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

where mod $(\bullet)$ denotes modulus after division.

### 4.2. Embedding method

The combined features of SVD and DCT are exploited in our proposed watermarking algorithm. The detailed steps of the watermark embedding procedure are shown in Fig. 1. The watermark embedding algorithm proposed here hides several bits of the watermark in every SVD–DCT block in the positions selected on a pseudorandom basis as we permute it randomly. At first, the SVD–DCT block is acquired by the following steps:

*Step* 1: Rearrange the 1-D audio signal into a 2-D signal, $H(m_1,n_1)$.
*Step* 2: Partition the 2-D host signal into non-overlapping subblocks, $H_j$, in the row-column order.

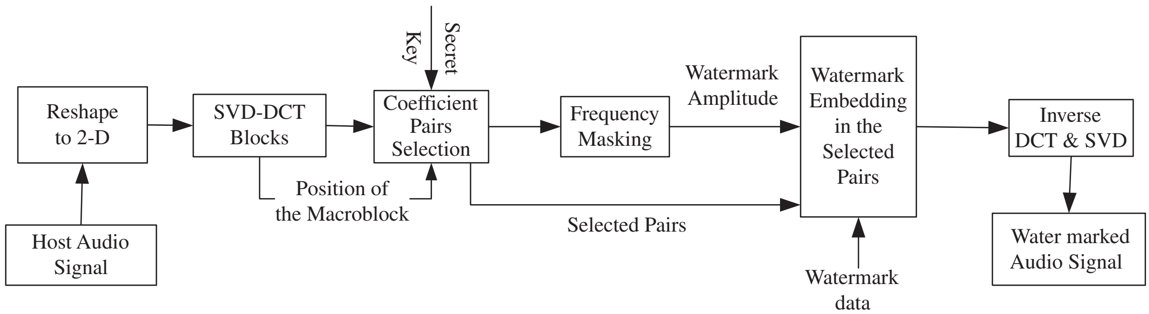


**Fig. 1.** Diagram of watermark embedding process.

| | | | |
|---|---|---|---|
| 1 | 2 | 6 | 7 |
| 3 | 5 | 8 | 13 |
| 4 | 9 | 12 | 14 |
| 10 | 11 | 15 | 16 |

**Fig. 2.** Sketch of SVD–DCT macroblock.

*Step* 3: For each block, $H_j$, ($j=1, 2, \ldots, M_1 \times N_1$, where $M_1 \times N_1$ is the total number of blocks in the 2-D host audio signal), apply $8 \times 8$ block based SVD to get its $U$, $S$ and $V$, respectively

$$H_j(r,c) = U_j(r,c)S_j(r,c)V_j(r,c)(1 \le r \le M_1, 1 \le c \le N_1). \quad (7)$$

As we know, the SVs represent the energy distribution of the watermark in different SVD layers and most energy is concentrated in the bigger SVs. The first SVD coefficient, $S_j(1,1)$ in every block is grouped together to form a new matrix, $DC$, which is of size $M_1 \times N_1$

$$DC(r,c) = S_j(1,1)(1 \le r \le M_1, 1 \le c \le N_1). \quad (8)$$

*Step* 4: The $DC$ matrix is in turn sub divided into $4 \times 4$ subblocks and the DCT is then applied to each subblock. Hence after the SVD–DCT transform, a $4 \times 4$ SVD–DCT coefficient block is obtained from the cover signal. A SVD–DCT block in the zigzag order is shown in Fig. 2, where the shaded positions are potential locations for watermark embedding.

Then the watermark is embedded into the host audio through the following steps:

*Step* 1: Select the coefficients pairs to be modified in the potential positions of the SVD–DCT block.
*Step 2*: Compute the frequency mask. To determine the level of tolerance against distortions caused by the watermark embedding, a linear model for frequency mask is generated as follows:

$$mask = \alpha \times edge + \beta \quad (9)$$

where the parameter $\alpha$ is for controlling the local embedding strength of the watermark, and $\beta$ is the base strength of the watermark which is needed to resist the effect of the rounding and truncation operations in the spatial domain. In our experiment, $\alpha$ and $\beta$ are set to 0.125 and 4 respectively. Since blocks with edge characteristics often have a lot of frequency components, the parameter *edge* is introduced to reduce artifacts: *edge* is the sum of the absolute values of the DCT-coefficients of indexes 9–16 that represent the higher DCT frequencies as marked in Fig. 2. High values in these components in the blocks of the first SVs indicate that the macroblocks have edge characteristics. In fact, the parameter *edge* is formulated as follows:

$$edge = \sum_{i=9}^{16} |F(i)| \quad (10)$$

where $F$ is the SVD–DCT block matrix in the zigzag order.

*Step* 3: Use the frequency mask to weight the watermark amplitude.
*Step* 4: Modify the relationship between the selected coefficients pairs according to the watermark embedding rule.

In fact, at the start of each SVD–DCT block embedding, a pseudo-random position sequence is generated based on a secret key for the purpose of choosing the coefficients pairs to be modified. In this way, the probability for an attacker to detect or alter the watermark through indentifying the embedded positions is significantly reduced. In the embedding procedure, the difference between the magnitudes of two coefficients in a selected pairs is computed as follows:

$$D(x_1,y_1,x_2,y_2) = |F(x_1,y_1)| - |F(x_2,y_2)| \quad (11)$$

where $F$ is the SVD–DCT block matrix, and $(x_1,y_1)$, $(x_2,y_2)$ are the coordinates of the selected pairs. Here we assume that the high frequencies in the SVD–DCT block are close enough so that $D(x_1,y_1,x_2,y_2)$ is positive if the watermark bit to be embedded is "1", and negative otherwise.

To embed a watermark bit "1", the procedure is as follows:

(1) If $D(x_1,y_1,x_2,y_2) \ge mask$, no operation is needed.
(2) If $D(x_1,y_1,x_2,y_2) < mask$, the following operations are needed:
If $F(x_1,y_1) \ne 0$ or $F(x_2,y_2) \ne 0$, then:

$$F'(x_1,y_1) = sign(F(x_1,y_1))[(|F(x_1,y_1)| + |F(x_2,y_2)|)/2 + 0.5 \times mask]$$
$$F'(x_2,y_2) = sign(F(x_2,y_2)) \bullet [(|F(x_1,y_1)| + |F(x_2,y_2)|)/2 - 0.5 \times mask] \quad (12)$$

If $F(x_1,y_1) = F(x_2,y_2) = 0$, then:

$$F'(x_1,y_1) = 0.5 \bullet mask$$
$$F'(x_2,y_2) = -0.5 \bullet mask. \quad (13)$$

The parameter *mask* in Eqs. (12) and (13) is the frequency mask for changing the watermarking strength according to the sharpness of the macroblocks.

To embed a watermark bit "0", the newly computed difference $D(x_1,y_1,x_2,y_2)$ should be negative, thus the procedure is as follows:

(1) If $D(x_1,y_1,x_2,y_2) \ge mask$, no operation is needed.
(2) If $D(x_1,y_1,x_2,y_2) < mask$, the following operations are needed:
If $F(x_1,y_1) \ne 0$ or $F(x_2,y_2) \ne 0$, then:

$$F'(x_1,y_1) = sign(F(x_1,y_1)) \bullet [(|F(x_1,y_1)| + |F(x_2,y_2)|)/2 - 0.5 \times mask]$$
$$F'(x_2,y_2) = sign(F(x_2,y_2)) \bullet [(|F(x_1,y_1)| + |F(x_2,y_2)|)/2 + 0.5 \times mask] \quad (14)$$

If $F(x_1,y_1) = F(x_2,y_2) = 0$, then:

$$F'(x_1,y_1) = -0.5 \bullet mask$$
$$F'(x_2,y_2) = 0.5 \bullet mask. \quad (15)$$

To achieve an appropriate trade-off between the robustness and transparency after watermark embedding, the potential locations of macroblocks are in the high frequency band. This differs from the traditional method of working in the mid-frequency band of DCT blocks because in this work the DCT block is derived from the perceptual important components of the signal concerned.

The watermarking is achieved by changing the difference of the magnitudes of a selected pair to a defined extent. In this scheme, only 2 pairs of coefficients in the 8 potential positions in Fig. 2 are selected for embedding watermark bits. There are two reasons for this selection. One is to maintain the imperceptibility and the other is to achieve high security. Since the attacker does not know which 2 pairs are embedded with watermark bits, attacking on all the 8 coefficients will cause intolerable distortions to the signal concerned.

After the watermark embedding, inverse DCT of the macroblock is performed on the modified SVD–DCT blocks, and then the blocks consisting of $4\times 4$ SVs are reconstructed. Only the first SVs of the original $8\times 8$ blocks contain watermark information, and other SVs are kept unchanged.

## 5. Watermark extraction

The watermark extraction procedure is rather simple and described as follows:

*Step 1*: Retrieve the positions of the coefficients pairs in each SVD–DCT block according to the secret key, the host signal characteristics, and the positions of different SVD–DCT blocks which are similar in the watermark embedding procedure.

*Step 2*: The difference between coefficients of each selected pair is computed:

$$D'(x_1,y_1,x_2,y_2)=|F'(x_1,y_1)|-|F'(x_2,y_2)| \tag{16}$$

where $|F'(x_1,y_1)|$ and $|F'(x_2,y_2)|$ are the selected coefficients of the watermarked signal.

*Step 3*: Sum the value of the difference $D'(x_1,y_1,x_2,y_2)$ corresponding to all pairs of coefficients where the same bit is repeatedly embedded as follows:

$$S'(i)=\sum_{\Phi_i}D'(x_1,y_1,x_2,y_2) \tag{17}$$

where $\Phi_i$ is the set of the selected pairs for the $i$th bit.

*Step 4*: The watermark is extracted with the following rules:

$$w'(i)=\begin{cases}1 & if\ S_i'\geq 0\\ 0 & if\ S_i'<0\end{cases}. \tag{18}$$

## 6. Error analysis

Two types of errors may occur while searching for the watermark sequence: the false positive error (i.e., false watermark detection) and the false negative error (i.e., failure to detect an existing watermark). It is rather difficult to give an exact probabilistic model of false positive and negative errors. Here, we adopt a simplified model based on binomial probability distribution to provide an analysis to estimate the probability of a false positive and negative error for our proposed technique. For comparing the similarities between the original and extracted watermark signals, the normalized cross-correlation coefficient (NC) is used, which is computed as:

$$NC(w,w')=\frac{\sum_l w(i)w'(i)}{\sqrt{\sum_i w^2(i)}\sqrt{\sum_i w'^2(i)}} \tag{19}$$

### 6.1. False positive error

We define the probability of false watermark detection as

$$P_{fp}=P\{NC(w,w')\geq T_P|no\ \ watermark\} \tag{20}$$

where $P\{A|B\}$ is the probability of event $A$ given event $B$, $T_P$ is a threshold. Since $w(i)$ and $w'(i)$ are either 0 or 1, correspondingly, $w^2(i)$ and $w'^2(i)$ are either 0 or 1, NC can be rewritten as:

$$NC(w,w')=\frac{\sum_l w(i)w'(i)}{\sqrt{\sum_i w^2(i)}\sqrt{\sum_i w'^2(i)}}\geq\frac{\sum_i w(i)w'(i)}{M} \tag{21}$$

where $M$ is the length of the binary watermark in 1-D format. To decide whether the watermark is present or not, we need to compare NC with a threshold. If NC is larger than a minimum threshold, then the watermark is present. Suppose $n$ bits of errors occur when extracting watermark data, then $m=M-n$ bits are identical to the original watermark data. It is easy to derive:

$$\sum_i w(i)w'(i)=m-n=M-2n \tag{22}$$

If Eqs. (21) and (22) are substituted into Eq. (20), the probability of false positive error can be further derived as:

$$\begin{aligned}P_{fp}&=P\left\{NC(w,w')\geq T_\rho|no\ \ watermark\right\}\\ &=\left\{\frac{\sum_i w(i)w'(i)}{M}\geq T_\rho|no\ \ watermark\right\}\\ &=\left\{\frac{M-2n}{M}\geq T_\rho|no\ \ watermark\right\}\end{aligned} \tag{23}$$

when $n\leq M(1-T_\rho)/2$, the false positive error will occur. Thus, we have

$$\begin{aligned}P_{fp}&=\sum_{n=0}^{\lceil M(1-T_\rho)/2\rceil}P\left\{\sum_i w(i)w'(i)=M-2n\middle|no\ \ watermark\right\}\\ &=\sum_{n=0}^{\lceil M(1-T_\rho)/2\rceil}\binom{M}{n}p^n(1-p)^{M-n}\end{aligned} \tag{24}$$

where $\binom{M}{n}=\frac{M!}{n!(M-n)!}$, $p$ is the probability of error in Bits Error Rate (BER) when some watermark extraction is performed, which is defined as

$$BER=\frac{1}{M}\sum_{i=1}^{M}w(i)\otimes w'(i) \tag{25}$$

where $\otimes$ is the exclusive or ($XOR$) operator.

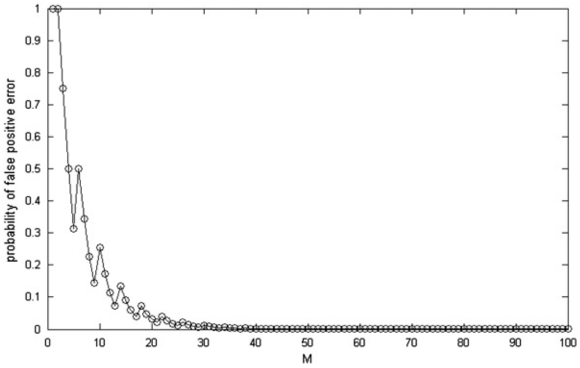


**Fig. 3.** Probability of false positive error under various $M$.

Since watermark values are either 0 or 1, thus $p = 0.5$, then

$$P_{fp} = \sum_{n=0}^{\lceil M(1-T_\rho)/2 \rceil} \binom{M}{n} 0.5^M \tag{26}$$

A plot of the probability of false positive error for $M \in (0,100]$ is shown in Fig. 3. It can be seen that probability of the false positive error approaches 0 when $M$ is larger than 30. In fact, in our scheme, $M=256$, $T_\rho = 0.5$, then $P_{fp} = 2.4486 \times 10^{-16}$.

### 6.2. False negative error

False negative error is the probability of declaring a watermarked audio as an unwatermarked one. Less false negative errors imply a better watermarking scheme. Similarly, the probability of false negative error can be determined as

$$P_{fn} = P\{NC(w,w') < T_\rho \mid no\ watermark\}$$
$$= \sum_{n=\lceil M(1-T_\rho)/2 \rceil + 1}^{M} \binom{M}{n} p^n (1-p)^{M-n} \tag{27}$$

where $p$ is the BER of the extracted watermark. Fig. 4 plots the probability of false negative error for $M \in (0,100]$. It is noted that the probability of false negative error is close to zero when $M$ is larger than 30. In our scheme $M=256$, thus the false negative error in our scheme is near to 0. In fact, if $M=256$, $T_\rho = 0.5$, $p = 0.9$, then $P_{fn} = 2.9347 \times 10^{-321}$.

Generally, if a desired probability of false positive error is given, the detection threshold $T_\rho$ can be determined by $P_{fp}$. The determination of threshold should consider the trade-off between false positive and negative errors. An increase in $T_\rho$ will reduce $P_{fp}$, but will increase $P_{fn}$. In our simulation, $T_\rho$ is set to 0.5.

## 7. Experimental results and discussions

In this section, some experimental results demonstrating the performance of the proposed SVD–DCT algorithm will be presented. In this experiment, the well-known SQAM files [29] are selected as test samples to analyze the performance of the proposed watermarking algorithm. Each audio signal is a 16-bit mono file in the WAVE

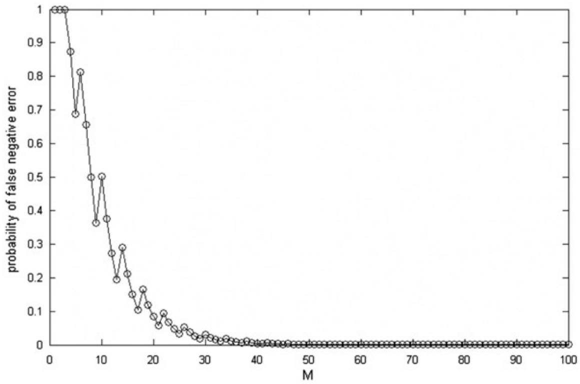


**Fig. 4.** Probability of false negative error under various $M$.

**Table 1**
Test sequences of SQAM database (16 bits).

| Index | Signal description | Item (.wav) | Index | Signal description | Item (.wav) |
|---|---|---|---|---|---|
| 1 | Bass | bass47_1 | 7 | Quartet | quar48_1 |
| 2 | Electronic tune | frer07_1 | 8 | Sopranor | sopr44_1 |
| 3 | Glockenspiel1 | gspi35_1 | 9 | Trumpt | trpt21_2 |
| 4 | Glockenspiel2 | gspi35_2 | 10 | Violoncello | vioo10_2 |
| 5 | Harpsichord | harp40_1 | 11 | Female speech | spfe49_1 |
| 6 | .Horn | horn23_2 | 12 | Male speech | spme50_1 |


**Fig. 5.** Watermark signal (a) original watermark; (b) extracted watermark without attack.

format with the 44.1 kHz sampling rate. A total of 12 host audio signals including music, percussive instruments, tonal instruments, vocal and speech signals are used. The detailed description of each signal is listed in Table 1.

$16 \times 16$ binary image is used as watermark signal for all audio signals. The original watermark signal and the extracted one without any attack are shown in Fig. 5. Actually, Fig. 5 verifies that the watermark can be extracted clearly without any attack. Performance of audio watermarking algorithms is usually evaluated with respect to imperceptibility (inaudibility) and robustness. All the parameters have been experimentally selected to achieve a good trade-off among the conflicting requirements of imperceptibility, robustness and capacity (Fig. 6).

### 7.1. Perceptual audio quality assessment

Audio watermarking intends to embed an unperceivable and secure watermark into host signals, thus it is essential that the watermarking scheme should be

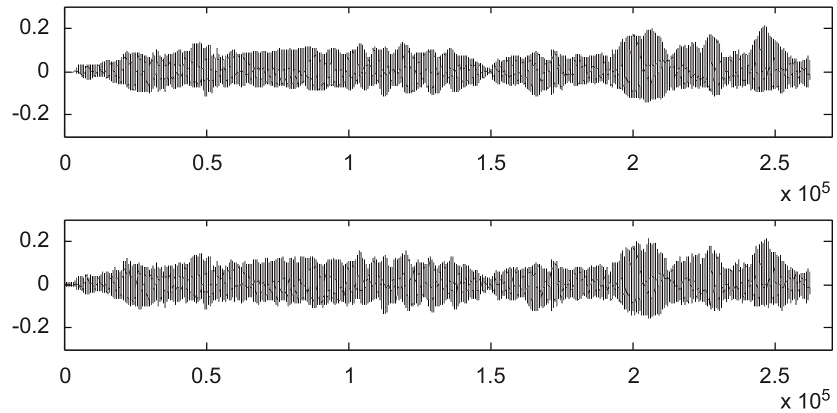


**Fig. 6.** Original audio signal and the watermarked host audio signal (SNR=36.98, SegSNR=41.05, MOS=4.86, ODG=−0.53, SDG=−0.51, LLR=0.012, CD=0.44).

perceptually transparent (imperceptibility). Generally, there are two approaches to perform the perceptual audio quality assessment [30,32]: (1) objective evaluation tests by perceptual evaluation of audio signals. (2) Subjective listening tests by human's acoustic perception.

#### 7.1.1. *Objective evaluation test*

Based on the results in the existing literature, there are several audio objective quality measures such as SNR, Segmental Signal-to-Noise Ratio (SegSNR), Cepstral Distance (CD) [13] and Log-Likelihood Ratio (LLR) measures [31]. These objective metrics are selected to evaluate perceptual performance between the host and the watermarked signals.

*7.1.1.1. SNR and SegSNR.* SNR is a statistical difference metric which is used to measure the perceptual similarity between the undistorted original and the distorted watermarked audio signal. SNR is computed by

$$SNR = 10\log_{10}\left(\sum_{i=1}^{L} S(i) \Big/ \sum_{i=1}^{L} (S'(i)-S(i))^2\right) \tag{28}$$

where $S(i)$ and $S'(i)$ correspond to the original and watermarked signal, respectively.

SegSNR is defined as the average of the SNR values of short segments of the watermarked signal and used to give an objective indication of the watermarking effect on the original audio. It is a good estimator for audio signal quality which is calculated as

$$SegSNR = \frac{10}{K}\sum_{m=0}^{K-1} \log_{10}\frac{\sum_{i=1}^{r} S^2(i)}{\sum_{i=1}^{r} (S'(i)-S(i))^2} \tag{29}$$

where $K$ is the number of frames in the watermarked audio signal and $r$ is number of samples per frame.

*7.1.1.2. CD.* CD is easy to implement with few computational requirements and has the capability for accurately predicting subjective listening scores. The CD measures cepstral coefficients computed from the original and watermarked signals on a frame-by-frame basis. For the $k$th frame, the CD is defined as

$$CD = \frac{1}{N}\sum_{k=1}^{N}\sqrt{\sum_{i=1}^{16}[real(IDFT\{\log|DFT(S(k))|\}) - real(IDFT\{\log|DFT(S'(k))|\})]^2} \tag{30}$$

where $real(IDFT\{\log|DFT(S(k))|\})$ and $real(IDFT\{\log|DFT(S'(k))|\})$ are the $i$th real cepstral coefficients of the original and watermarked signals, respectively, DFT and IDFT are the discrete Fourier transform and its inverse. The *zero*-th coefficient is not generally included in Eq. (30) since it is primarily affected by system gain rather than system distortion. Furthermore, by excluding high order coefficients, CD represents a cepstrally smoothed spectral-difference measure.

*7.1.1.3. LLR.* The LLR metric for an audio segment is based on the assumption that the segment can be represented by a $p$th order all-pole linear predictive coding model of the form:

$$S(n) = \sum_{m=1}^{P} a_m S(n-m) + G_x u(n) \tag{31}$$

where $S(n)$ is the $n$th audio sample, $a_m$ (for $m=1, 2, \ldots, P$) are the coefficients of an all-pole filter. $G_x$ is the gain of the filter and $u(n)$ is an appropriate excitation source for the filter. The audio signal is windowed to form frames with length of 15–30 ms. The LLR metric is then defined as

$$LLR = \left|\log\left(\frac{\vec{a}_S \vec{R}_{S'} \vec{a}_S^{\,T}}{\vec{a}_{S'} \vec{R}_{S'} \vec{a}_{S'}^{\,T}}\right)\right| \tag{32}$$

where $\vec{a}_S$ and $\vec{a}_{S'}$ are the LPC coefficient vectors for the original and watermarked audio signals, respectively, $\vec{R}_{S'}$ is the auto-correlation matrix of the watermarked audio signal. The closer to zero the LLR is, the better the quality of the watermarked audio signal is.

Classical objective measures such as SNR, SegSNR, CD and LLR do not use the characteristics of the HAS. On the

**Table 2**
MOS, SDG and ODG descriptions and results from our proposed scheme.

| MOS | SDG | ODG | Description | Quality |
|---|---|---|---|---|
| 5 | 0 | 0 | Imperceptible | Excellent |
| 4 | −1 | −1 | Perceptible, but not annoying | Good |
| 3 | −2 | −2 | Slightly annoying | Fair |
| 2 | −3 | −3 | Annoying | Poor |
| 1 | −4 | −4 | Very annoying | Bad |

other hand, the Perceptual Evaluation of Audio Quality (PEAQ) which is specified in ITU-Recommendation standard BS.1387 incorporates a psychoacoustic model. The PEAQ was originally proposed in 1998 and last updated in 2001. The output of the PEAQ algorithm is the ODG which corresponds to the subjective grade used to measure the differences between the original and watermarked audio signals. To implement PEAQ and compute the ODG, EAQUAL software is utilized [33]. The computed ODG values are in a range of [−4, 0] and the details is shown in Table 2.

*7.1.2. Subjective listening test*

Besides the objective test, the essential subjective quality assessment is also utilized to evaluate the audibility of the watermarked audio as the ultimate judgment is made by human acoustic perception. In the subjective listening tests, MUSHRA described in ITU-BS.1534, stands for "Multi Stimulus test with Hidden Reference and Anchors" are selected to evaluate the scheme. The watermarked signal is graded with respect to the host signal according to a five-grade impairment scale defined in ITU-R BS.562, called Subjective Difference Grade (SDG). SDG is the difference between the absolute quality grade assigned to the distorted and the original signals, both measured on the five-grade impairment scale. If all listeners identify the hidden reference correctly, the SDG is identical to the grade on the five-grade impairment-scale assigned to the processed signal. The value of an SDG should therefore never be outside a range of minus four to zero. SDG details are also specified in Table 2. A listening test was actually performed with ten listeners who were both experienced and familiar with the used set of test items to estimate the SDG grades of the watermarked signals. The ten people listened to each test samples for 10 times and gave a grade. Then the average grade for each test samples from all listeners is the final SDG grade.

Mean Opinion Score (MOS), specified by ITU-T recommendation P.800, is also a good indication of the perceived audio quality after watermarking distortion. Thus the MOS scores, which are described in Table 2, will be used in our algorithm to measure the performance as well.

*7.1.3. Test results*

A short portion of an original violoncello audio signal and the corresponding watermarked audio signal are shown in Fig. 6, which demonstrates that the imperceptibility of our scheme is good. The corresponding SNR, SegSNR, CD and LLR values, along with MOS, SDG and ODG grades obtained by conducting the subjective and objective tests are tabulated in Table 3. It can be noted that the embedding distortion caused by our proposed watermarking algorithm expressed with the ODG value is very close to the threshold of becoming audible. The average ODG score is -0.57 from all test items which confirms that our watermarked audio signals are perceptually similar to the original audio signals. Besides, it is evident that all the MOS scores are within [4.51, 4.90], and high MOS scores indicate that our proposed scheme provides good imperceptibility of the watermarked audio signals. The LLR is almost close to zero and the average is 0.019, which is obviously an indication of the good performance of our proposed scheme. The CD results in our experiment are very good for all test samples which can be seen from Table 2. It is clear that the average result of CD is satisfactory. Furthermore, as can be seen from the above results, the audio quality distortion of the watermarking system is very small. The average results demonstrate that there is no significant distortion introduced by this scheme. The main reason of having the satisfactory imperceptibility results is due to the fact that only a small number of samples are modified in the watermarking algorithm and all parameters are carefully chosen to have a better performance.

**Table 3**
Objective and subjective assessment results of our proposed scheme.

| Index | SNR | SegSNR | MOS | ODG | SDG | LLR | CD |
|---|---|---|---|---|---|---|---|
| 1 | 25.90 | 28.62 | 4.51 | −0.79 | −0.73 | 0.039 | 0.63 |
| 2 | 26.18 | 27.44 | 4.54 | −0.77 | −0.74 | 0.033 | 0.58 |
| 3 | 41.96 | 44.36 | 4.90 | −0.28 | −0.28 | 0.006 | 0.36 |
| 4 | 37.93 | 40.04 | 4.86 | −0.36 | −0.35 | 0.018 | 0.42 |
| 5 | 35.39 | 37.88 | 4.83 | −0.31 | −0.31 | 0.016 | 0.49 |
| 6 | 34.21 | 35.59 | 4.81 | −0.52 | −0.50 | 0.015 | 0.47 |
| 7 | 28.71 | 30.78 | 4.59 | −0.69 | −0.65 | 0.018 | 0.53 |
| 8 | 27.49 | 30.84 | 4.56 | −0.72 | −0.69 | 0.021 | 0.51 |
| 9 | 39.37 | 39.93 | 4.89 | −0.42 | −0.41 | 0.009 | 0.38 |
| 10 | 36.98 | 41.05 | 4.86 | −0.53 | −0.51 | 0.012 | 0.44 |
| 11 | 28.81 | 31.63 | 4.61 | −0.68 | −0.64 | 0.021 | 0.52 |
| 12 | 27.39 | 29.59 | 4.58 | −0.71 | −0.67 | 0.023 | 0.50 |
| **Average** | **32.53** | **34.81** | **4.71** | **−0.57** | **−0.54** | **0.019** | **0.48** |

*7.2. Robustness*

Watermarked audio signals may undergo common signal processing operations which may affect the perceived quality of the host signal and corrupt the watermark image embedded within the signal. In order to illustrate the robustness of our watermarking scheme, we implemented a great number of common attacks on audio signals in our experiments to assess the robustness of our scheme. Some attacks are performed using MATLAB 2008, Adobe Audition 1.0 and GoldWave 5.18, which are all popular tool-sets for professional audio processing and editing. The common signal processing attacks are described as follows:

(1) Additive noise: White Gaussian noise is added to the watermarked signal until the resulting signal has a SNR of 20 dB.

(2) Amplitude variation: The watermarked signal is attenuated up to 120%.
(3) Cropping: Segments of 100 samples are removed from the watermarked audio signal at three randomly selected positions.
(4) Denoising: The watermarked audio signal is denoised by using the "Hiss removal" function of GoldWave.
(5) Echo addition: An echo signal with a delay of 50 ms and a decay of 10% is added to the original audio signal.
(6) Expanding: Expand the watermarked signal with increment of 3 dB and −3 dB respectively.
(7) Jittering: The watermarked audio signal undergoes small rapid variations.
(8) Low-pass filtering: The low-pass filter with the 44.1 kHz cutoff frequency is applied to watermarked audio signal.
(9) MP3 Compression: The coding/decoding is performed using a software implementation of the ISO/MPEG-1 Audio Layer III coder with different bit rates (32, 48, 64, 96 and 128 kbps) and then decoded back to the WAVE format.
(10) Pitch shifting: Tempo-preserved pitch shifting is a tough attack for audio watermarking algorithms as it causes frequency fluctuation. In our experiment, the pitch is shifted 1% higher and then 1% lower.
(11) Re-quantization: 16-bit watermarked audio signal is quantized to 8-bit and then back to 16-bit.
(12) Re-sampling: As the original audio signal is sampled with a sampling rate of 44.1 kHz, thus the watermarked audio signal is down-sampled to 11.025 kHz, 22.05 kHz, and then up-sampled back to 44.1 kHz; watermarked audio signal is also up-sampled to 88.2 kHz, and then down-sampled back to 44.1 kHz.
(13) Reverse amplitude: Reverse the signs of the sample amplitudes.
(14) Time-scale modification: The watermarked audio signal is lengthened by 4% while preserving the pitch.

**Table 4**
Results of robustness against common signal processing attacks achieved by our proposed scheme vs. that in [12].

| Attacks | Algorithm in [12] | | Ours | | Extracted watermarks |
|---|---|---|---|---|---|
| | BER (%) | NC | BER (%) | NC | |
| No attack | 0 | 1 | 0 | 1 |  |
| Additive noise | 0 | 1 | 0 | 1 |  |
| Amplitude variation | – | – | 0 | 1 |  |
| Cropping | 0 | 1 | 0 | 1 |  |
| Denoising | 5 | 0.9634 | 0 | 1 |  |
| Echo addition | 2 | 0.9847 | 0 | 1 |  |
| Expanding | – | – | 0 | 1 |  |
| Jittering | – | – | 0 | 1 |  |
| Low-pass filtering | 0 | 0.9993 | 0 | 1 |  |
| MP3 (32 kbps) | 1 | 0.9907 | 3 | 0.9878 |  |
| MP3 (48 kbps) | – | – | 1 | 0.9959 |  |
| MP3 (64 kbps) | 0 | 0.9980 | 0 | 1 |  |
| MP3 (96 kbps) | – | – | 0 | 1 |  |
| MP3 (128 kbps) | – | – | 0 | 1 |  |
| Pitch shifting | – | – | 0 | 1 |  |
| Re-quantization | 0 | 1 | 0 | 1 |  |
| Re-sampling(44.1-22.05-44.1) | 2 | 0.9854 | 0 | 1 |  |
| Re-sampling(44.1-11.025-44.1) | 2 | 0.9854 | 0 | 1 |  |
| Re-sampling(44.1-88.2-44.1) | 2 | 0.9854 | 0 | 1 |  |
| Time-scale modification | – | – | 0 | 1 |  |
| **Average** | **1.0667** | **0.9918** | **0.2000** | **0.9992** | |

Table 4 summarizes the watermark detection results against various common signal processing attacks. It is obvious that NC values after attacks are very high while the values of BER (%) are very low. As BER (%) is used in [12], we use BER (%) in Table 4 for easy comparison with the algorithm in [12]. The average values of NC and BER after all attacks are 0.9992 and 0.2000, respectively, which are better than the values of NC and BER (%) in [12]. The better results in our scheme vindicate the robustness of our proposed audio watermarking scheme. Besides, the extracted watermark images are very visually similar to the original watermark, which further verify the good performance of our scheme. It goes without saying that our proposed algorithm is slightly better than the algorithm in [12] as can be seen from the robustness test results.

### 7.3. Robustness to Stirmark attacks

Performance of the proposed scheme is evaluated under the standardized Stirmark attacks for audio. Stirmark Benchmark for Audio (SMBA) is a standard and common robustness evaluation tool for evaluating the robustness of audio watermarking method [34]. In this experiment, different attacks are applied to the watermarked and unwatermarked host signal by using the Stirmark software. The parameters of the Stirmark attacks are the default parameters included in the version of the tool available on the Web.

The Stirmark attack tests are performed on the test set and the results are presented in Table 5. From the results of the SMBA test, we can see that the scheme can resist those common attack operations. The BER values are very low after attacks, so this scheme is robust and secure. Furthermore, the watermarks extracted after various attacks on the watermarked signal are given in Table 5 too. It should be noted that since synchronization techniques are adopted in our proposed algorithm, it is robust to synchronization attacks such as Flipsample, Copysample and Cutsample. Besides, the proposed method demonstrates its superiority over the algorithm in [3] and [11].

**Table 5**
Results of robustness against Stirmark attacks for audio achieved by our proposed scheme vs. SVD based [11] and DCT based [3] schemes.

| Attacks | DCT based method in [3] | SVD based method in [11] | Ours | Extracted watermarks |
|---|---|---|---|---|
| Addbrumm | 1.25 | 0 | 0 |  |
| AddDynNoise | 1.56 | 0 | 0 |  |
| AddFFTNoise | 51.25 | 0 | 0 |  |
| Addnoise | 0.78 | 0 | 0 |  |
| Addsinus | 0.77 | 0 | 0 |  |
| Amplify | 52.32 | 0.75 | 0 |  |
| Bassboost | 0 | 0 | 0 |  |
| Compressor | 0 | 0 | 0 |  |
| Copysample | 100 | 0.5 | 0.195 |  |
| Cutsamples | 100 | 0 | 0 |  |
| Echo | 23.43 | 0 | 0 |  |
| Exchange | 0 | 0 | 0 |  |
| Extrastereo | 0 | 0 | 0 |  |
| Fft_hlpass | 0.31 | 0 | 0 |  |
| Fft_invert | 52.6 | 0 | 0 |  |
| Fft_real_reverse | 0.78 | 0 | 0 |  |
| Fft_stat1 | 19.84 | 0.5 | 0.0234 |  |
| Fft_test | 19.80 | 0.4 | 0.0078 |  |
| Flipsample | 21.66 | 0.75 | 0.0156 |  |
| Invert | 52.42 | 0 | 0 |  |
| Lsbzero | 0 | 0 | 0 |  |
| Normalize | 0 | 0 | 0 |  |
| Nothing | 0 | 0 | 0 |  |
| Rc_highpass | 2.03 | 0 | 0 |  |
| Rc_lowpass | 0 | 0 | 0 |  |
| Smooth | 0 | 0 | 0 |  |
| Stat1 | 0 | 0 | 0 |  |
| Stat2 | 0 | 0 | 0 |  |
| Voiceremove | 52.1 | 0 | 0 |  |
| Zerocross | 0 | 0 | 0 |  |
| Zerolength | 60.5 | 0 | 0 |  |
| Zeroremove | 100 | 0 | 0 |  |
| **Average of all attacks** | **22.2937** | **0.0906** | **0.0076** | |

The BER values after attacks are compared with those obtained from two published algorithms: the SVD–STFT based algorithm in [11] and the DCT based algorithm in [3]. It is obvious that the performance of the proposed algorithm is much better than the method in [3] and slightly better than the method in [11].

**Table 6**
Comparison results of capacity performance achieved by our proposed scheme vs. SVD-based algorithms.

| Algorithms | SNR | Synchronization | Capacity |
|---|---|---|---|
| [6] | 42.8 | Yes | 2 |
| [7] | 29.5 | Yes | 4.27 |
| [26] | 43.0 | Yes | 36 |
| [27] | 43.1 | Yes | – |
| [35] | 25 | No | 43 |
| [36] | 30–45 | No | 86 |
| **Ours** | **32.53** | **Yes** | **43** |

### 7.4. Embedding capacity

Capacity is the amount of information that may be embedded and recovered in the audio stream, which is also called data payload. There are several methods for measuring watermarking capacity. In our scheme, suppose that the sampling rate of the audio signal is $Fs$ (Hz), and the number of samples of each segment is $N_L$, then the embedding capacity$P_c$of the proposed scheme can be expressed as

$$P_c = Fs/N_L \tag{33}$$

where the unit of embedding capacity $P_c$ is bit/s. From this equation, it can be seen that the embedding capacity increases as $N_L$ decreases. However, a smaller $N_L$ causes higher distortion. In principle, this method is independent of the watermarked signal in the sense that two watermarked signals with the same sampling time and the same watermarking parameters would have exactly the same capacity. In this way, it could be argued that the true capacity of our proposed scheme is 43 bit/s.

For the embedding capacity, we also compared our proposed scheme with other recent audio watermarking scheme with and without synchronization methods [6,7,27,28,35,36]. Table 6 shows the capacity performance comparison results of our proposed hybrid SVD–DCT based audio watermarking scheme with several state-of-the-art audio watermarking strategies with and without synchronization techniques. Many other methods are not listed as they did not report the specific number of embedding capacity. We can see that the capacity of our method is satisfactory compared to the other methods. The capacity of our algorithm is the same as the algorithm in [35]. However, the imperceptibility of our method is better than that in [35].

### 7.5. Discussions

A great variety of traditional and state-of-the-art watermarking approaches such as echo, spread spectrum, Least Significant Bit (LSB) and frequency masking methods have been presented in the literature. Recently, some SVD based audio watermarking methods have been developed too. As we know, SNR is just a measure of the noise power relative to the signal power and it is not

**Table 7**
Comparison results of SNR and MOS values achieved by our proposed scheme vs. traditional and SVD based methods.

| Reference | Algorithm | SNR | MOS |
|---|---|---|---|
| [1] | Echo | 21.47 | 3.60 |
| [1] | Phase | 12.20 | 2.44 |
| [1] | LSB | 67.91 | 4.90 |
| [2] | Frequency masking | 12.87 | 2.93 |
| [3] | Spread spectrum | 28.59 | 4.46 |
| [11] | SVD–STFT | 28.36 | 4.70 |
| in [12] | SVD–DWT | 26.84 | 4.60 |
| [13] | SVD | 27.13 | 4.60 |
| **Ours** | **SVD–DCT** | **32.53** | **4.71** |

closely correlated to human perception as SNR does not take into account perceptual models. For example, the SNR between a signal $x$ and the same signal amplified a 10% ($1.1x$) is 20 dB. However, the MOS grades in such case would be 0 (no error), since the amplitude change only implies a volume amplification of the signal without any significant degradation of audio quality. Therefore, in order to make an appropriate comparison with the traditional watermarking methods and the SVD based methods, SNR results and MOS grades are both selected. The comparison results among our method, the traditional and SVD based methods are given in Table 7 which are based on the reported results in the references [1–3,11–13]. From the comparison results, it can be seen that our proposed method outperforms some traditional methods and SVD based methods in terms of SNR and MOS. Only LSB method is better than our method, but our method is obviously better than the SVD based methods.

Finally, it should be pointed out that the proposed algorithm satisfies the desired features of optimal audio watermarking, which have been set by the International Federation of Photographic Industry (IFPI). IFPI states that the watermarking scheme should not degrade perception of audio, that is, SNR result of the algorithm should be more than 20 dB; the watermarking should be able to resist most common audio processing operations and attacks; and the watermarking should prevent unauthorized removal unless the quality of audio becomes very poor. Referring to the figures and tables above, it is easy to conclude that the performance of the proposed algorithm fulfills the IFPI performance requirement as the SNR of our method is far beyond 20 dB.

## 8. Conclusions

In this paper, we propose a blind and robust audio watermarking technique combined with SVD, DCT and synchronization code technique. Extensive experimental works have shown that the proposed watermarking scheme possesses strong robustness to common signal processing operations. Moreover, the proposed scheme achieves very low error probability rates. The comparison of our proposed algorithm with traditional and SVD based algorithms show better performance from our algorithm. The obtained simulation results verify the effectiveness of our audio watermarking as a reliable solution to the copyright protection problem.

## References


[1] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding, IBM Systems Journal 35 (3–4) (1996) 313–336.
[2] M.D. Swanson, B. Zhu, A.H. Tewfik, L. Boney, Robust audio watermarking using perceptual masking, Signal Processing 66 (3) (1998) 337–355.
[3] I.J. Cox, J. Kilian, F.T. Leighton, T. Shamoon, Secure spread spectrum watermarking for multimedia, IEEE Transactions on Image Processing 6 (12) (1997) 1673–1687.
[4] F. Hartung, M. Kutter, Multimedia watermarking techniques, Proceedings of the IEEE 87 (7) (1999) 1079–1107.
[5] M. Arnold, Audio watermarking: features, applications and algorithms, in: Proceedings of IEEE International Conference on Multimedia and Expo, 2000, pp. 1013–1016.
[6] X.Y. Wang, H. Zhao, A novel synchronization invariant audio watermarking scheme based on DWT and DCT, IEEE Transactions on Signal Processing 54 (12) (2006) 4835–4840.
[7] S.J. Xiang, H.J. Kim, J.W. Huang, Audio watermarking robust against time-scale modification and MP3 compression, Signal Processing 88 (10) (2008) 2372–2387.
[8] R. Liu, T. Tan, An SVD-based watermarking scheme for protecting rightful ownership, IEEE Transactions on Multimedia 4 (1) (2002) 121–128.
[9] X.P. Zhang, K. Li, Comments on "an SVD-based watermarking scheme for protecting rightful ownership", IEEE Transactions on Multimedia 7 (3) (2005) 593–594.
[10] A.A. Mohammad, A. Alhaj, S. Shaltaf, An improved SVD-based watermarking scheme for protecting rightful ownership, Signal Processing 88 (9) (2008) 2158–2180.
[11] Hamza Özer, Bülent Sankur, N. Memon, An SVD-based audio watermarking technique, in: Proceedings of the Seventh Workshop on Multimedia and Security, 2005, pp. 51–56.
[12] V. Bhat, I. Sengupta, A. Das, An adaptive audio watermarking based on the singular value decomposition in the wavelet domain, Digital Signal Processing 20 (6) (2010) 1547–1558.
[13] F.E. Abd El-Samie, An efficient singular value decomposition algorithm for digital audio watermarking, International Journal of Speech Technology 12 (1) (2009) 27–45.
[14] A. Dhawan, S.K. Mitra, Hybrid audio watermarking with spread spectrum and singular value decomposition, in: Proceedings of IEEE Conference and Exhibition on Control, Communications and Automation, 2008, pp. 11–16.
[15] M. Tong, T. Yan, H.B. Ji, An efficient audio watermark algorithm with strong robustness, in: Proceedings of the International Conference on Computational Intelligence and Security, 2008, pp. 385–389.
[16] R. Zezula, J. Miurec, Audio digital watermarking algorithm based on SVD in MCLT domain, in: Proceedings of Third International Conference on Systems, 2008, pp. 140–143.
[17] S. Vongpraphip, M. Ketcham, An intelligence audio watermarking based on DWT–SVD using ATS, in: Proceedings of WRI Global Congress on Intelligent Systems, 2009, pp. 150–154.
[18] A. Robert, J. Picard, On the use of masking models for image and audio watermarking, IEEE Transactions on Multimedia 7 (4) (2005) 727–739.
[19] S.Q..Wu, J.W. Huang, D.R. Huang, Y.Q. Shi, Self-synchronized audio watermark in DWT domain, in: Proceedings of the IEEE International Symposium on Circuits and Systems, 2004, pp. 712–715.
[20] P. Bao, X. Ma, Image adaptive watermarking using wavelet domain singular value decomposition, IEEE Transactions on Circuits and Systems for Video Technology 15 (1) (2005) 96–102.
[21] F. Huang, Z.-H. Guan, A hybrid SVD–DCT watermarking method based on LPSNR, Pattern Recognition Letters 25 (15) (2004) 1769–1775.
[22] C. Kuo-Liang, Y. Wei-Ning, H. Yong-Huai, W. Shih-Tung, H. Yu-Chiao, On SVD-based watermarking algorithm, Applied Mathematics and Computation 188 (1) (2007) 54–57.
[23] Z. Peng, W. Liu, Color image authentication based on spatiotemporal chaos and SVD, Chaos, Solitons and Fractals 36 (4) (2008) 946–952.
[24] R. Rykaczewski, Comments on "an SVD-based watermarking scheme for protecting rightful ownership", IEEE Transactions on Multimedia 9 (2) (2007) 421–423.
[25] Y.D. Wu, On the security of an SVD-based ownership watermarking, IEEE Transactions on Multimedia 7 (4) (2005) 624–627.
[26] H.C. Andrews, C.L. Patterson, Singular vale decomposition and digital image processing, IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-24 (1) (1976) 26–53.

[27] J.W. Huang, Y. Wang, Y.Q. Shi, A blind audio watermarking algorithm with self-synchronization, in: Proceedings of IEEE International Symposium on Circuits and Systems, 2002, pp. 627–630.
[28] W. Li, X. Xue, P. Lu, Localized audio watermarking technique robust against time-scale modification, IEEE Transactions on Multimedia 8 (1) (2006) 60–69.
[29] EBU, "SQAM–Sound Quality Assessment Material", 〈http://sound.media.mit.edu/mpeg4/audio/sqam/〉, 2001 (last checked on October 31, 2010).
[30] Y.Q. Lin, W.H. Abdulla, Perceptual evaluation of audio watermarking using objective quality measures, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2008, pp. 1745–1748.
[31] R.E. Crochiere, J.E. Tribolet, L.R. Rabiner, An interpretation of the log likelihood ratio as a measure of waveform coder performance, IEEE Transactions on Acoustics, Speech, and Signal Processing 28 (3) (1980) 318–323.
[32] J. Beerends, J. Stemerdink, A perceptual audio quality measurement based on a psychoacoustic sound representation, Journal of the Audio Engineering Society 40 (12) (1992) 963–972.
[33] EAQUAL software, 〈http://www.mp3-tech.org/programmer/sources/eaqual.tgz〉 (last checked on October 31, 2010).
[34] L. Andreas, D. Jana, S. Ryan, V. Claus, Audio watermark attacks: from single to profile attacks, in: Proceedings of the Seventh Workshop on Multimedia and security, ACM, 2005, pp. 39–50.
[35] H. Kang, K. Yamaguchi, B. Kurkoski, K. Yamaguchi, K. Kobayashi, Full-index-embedding patchwork algorithm for audio watermarking, IEICE Transactions on Information and Systems E91.D (11) (2008) 2731–2734.
[36] M. Fan, H. Wang, Chaos-based discrete fractional Sine transform domain audio watermarking scheme, Computers and Electrical Engineering 35 (3) (2009) 506–516.