I.J. Education and Management Engineering **2012, 1,73-80**
Published Online January 2012 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2012.01.12

Available online at http://www.mecs-press.net/ijeme

An Adaptive Audio Watermarking Scheme Method Based on Kernel Fuzzy C-means Clustering

[a] Honghong Chen, [b] Zulin Zhang

[a] School of Mathematics and Computer Engineering Xihua University Chengdu, Sichuan, 610039, China
[b] Department of Computer Science Sichuan University of Nationalities Kangding, Sichuan 626001, China

Abstract

In this paper, we propose an adaptive audio watermarking scheme according to local audio features. Firstly, the original audio signal is partitioned into audio frames and these audio frames are transformed into DWT domain respectively. Next, the local features of each audio frame are extracted respectively, and these features are used to train kernel fuzzy c-means (KFCM) clustering algorithm. According to well-trained KFCM, the audio frames to embed the watermark are selected and their embedding strengths are determined adaptively. The experimental results show the proposed method is robust to common signal processing operations such as lossy compression (MP3), filtering, re-sampling, re-quantizing, etc.

Index Terms: Audio signal; audio watermarking; adaptive watermarking; kernel fuzzy c-means clustering algorithm

1. Introduction

Digital audio watermarking technique provides efficient tools for ensuring that product ownership of audio multimedia is preserved, even if multimedia data is attracted by attackers. For an audio watermarking system, imperceptibility and robustness are its two basic requirements. Imperceptibility refers to the perceptual quality of the data being protected. For audio data, digital watermark should be inaudible. The digital watermark should also be robust to audio signal processing. Ideally, the amount of signal distortion necessary to remove the watermark can degrade the desired audio quality to point of becoming commercially valueless. Typical audio processing operations include lossy compression (such as MP3), additive noise, filtering, resampling, etc. Currently, many watermarking techniques have been developed, such as quantization-based techniques, relationship-based techniques, physiological model-based techniques, adaptive techniques, etc.

Recently, some machine learning methods, such as neural networks and support vector machines, are introduced into digital audio watermarking technique, and can simultaneously improve robustness and audible quality of the watermarked audio signal. In [1], neural networks were introduced into a nonblind audio watermarking scheme, which was used to estimate the watermark scaling factor intelligently from the host audio

Corresponding author:
E-mail address: [a] hhchen94@sina.com, [b] zhangzl@scun.edu.cn

signal, and make the watermark detection more robust against common attacks. Wang et al.[2] introduced SVM for audio watermark detection in DWT domain, which considered the watermark extraction as a two-class problem. In [3], Serap et al. presented an audio watermark decoding process based on SVM, which combined the watermark decoding and detection problems into a single classification problem. Xu et al. [4] proposed a DWT-based audio watermarking method using support vector regression (SVR) which was used to learning the correlation between the sub-audios obtained by sub-sampling technique. Peng et al.[5] proposed an audio watermarking method in multiwavelet domain based on SVM, where SVM was used to learn nonlinear relationship between local audio features in order to extract watermark signal.

In this paper, we propose an adaptive audio watermarking scheme. Firstly, we divide audio signal into audio frames, and these audio frames are transformed into DWT domain respectively. Secondly, we extract its local features for each audio frame, that is, its local energy and the maximal peaks of its all subbands. Next, through running kernel fuzzy c-means (KFCM) clustering algorithm on these features, we obtain the maximal fuzzy membership degree for each audio frame. According the fuzzy membership degree, we can select the audio frames to be embedded watermark and determine the embedding strength of each audio frame adaptively. The proposed method can simultaneously improve the robustness and audible quality of the watermarked audio signal.

The rest of this paper is organized as follows: Section II introduces the concept of kernel fuzzy c-means clustering algorithm. In Section III, we present the proposed audio watermarking approach. Simulation results for several watermarked audio manipulations are presented in Section IV. Finally, we draw our conclusions in Section V.

2. Kernel Fuzzy C-means Algorithm

Given an unlabeled data set $X=\{x_1,x_2,\ldots,x_m\}\subseteq R^d$, and a nonlinear mapping ϕ: $R^d \rightarrow F$ from this input space to a high-dimensional feature space F. By applying the nonlinear mapping ϕ, the dot product $x_i \cdot x_j$ in the space is mapped to $\phi(x_i) \cdot \phi(x_j)$ in the feature space. The key notion in kernel-based learning is that the mapping ϕ need not be explicitly specified. The dot product $\phi(x_i) \cdot \phi(x_j)$ in the high-dimensional feature space can be calculated through the kernel function $K(x_i, x_j)$ in the input space. The kernel fuzzy c-means algorithm in the feature space F by a mapping ϕ minimizes the function J_r [6],[7]:

$$J_r(X)=\sum_{i=1}^{c}\sum_{j=1}^{m}(\mu_{ij})^r \,||\, \phi(x_j)-v_i^{\phi} \,||^2 \qquad (1)$$

where μ_{ij} is the membership degree of data point x_j to the ith fuzzy cluster, and r is a fuzziness coefficient. The ith cluster centroid is $v_i^{\phi}=n_i^{-1}\sum_{j=1}^{m}(\mu_{ij})^r\phi(x_j)$ and $n_i=\sum_{j=1}^{m}(\mu_{ij})^r$. The key notion in the kernel fuzzy c-means algorithm lies in the calculation of the distance in the feature space. The distance between $\phi(x_j)$ and v_i^{ϕ} in the feature space is calculated through the kernel in the input space:

$$\begin{aligned}
&\| \phi(x_j) - v_i^{\phi} \|^2 \\
&= \phi(x_j)\cdot\phi(x_j) - 2\phi(x_j)\cdot\frac{\sum_{k=1}^{m}(\mu_{ik})^r\phi(x_k)}{\sum_{k=1}^{m}(\mu_{ik})^r} \\
&\quad + \frac{\sum_{k=1}^{m}(\mu_{ik})^r\phi(x_k)}{\sum_{k=1}^{m}(\mu_{ik})^r}\cdot\frac{\sum_{l=1}^{m}(\mu_{il})^r\phi(x_l)}{\sum_{l=1}^{m}(\mu_{il})^r} \\
&= \phi(x_j)\cdot\phi(x_j) - \frac{2\sum_{k=1}^{m}(\mu_{ik})^r\phi(x_k)\cdot\phi(x_j)}{\sum_{k=1}^{m}(\mu_{ik})^r} \\
&\quad + \frac{\sum_{k=1}^{m}\sum_{l=1}^{m}(\mu_{ik})^r(\mu_{il})^r\phi(x_k)\cdot\phi(x_l)}{(\sum_{k=1}^{m}(\mu_{ik})^r)^2} \\
&= K(x_j, x_j) - \frac{2\sum_{k=1}^{m}(\mu_{ik})^r K(x_k, x_j)}{\sum_{k=1}^{m}(\mu_{ik})^r} \\
&\quad + \frac{\sum_{k=1}^{m}\sum_{l=1}^{m}(\mu_{ik})^r(\mu_{il})^r K(x_k, x_l)}{(\sum_{k=1}^{m}(\mu_{ik})^r)^2},
\end{aligned} \tag{2}$$

Where $K(x_k, x_l) = \phi(x_k)\cdot\phi(x_l)$ By using $n_i = \sum_{j=1}^{m}(\mu_{ij})^r$, we have

$$\begin{aligned}
\left\|\phi(x_j) - v_i^{\phi}\right\|^2 &= K(x_j, x_j) - \frac{2}{n_i}\sum_{k=1}^{m}(\mu_{ik})^r K(x_k, x_j) \\
&\quad + \frac{1}{n_i^2}\sum_{k=1}^{m}\sum_{l=1}^{m}(\mu_{ik})^r(\mu_{il})^r K(x_k, x_l)
\end{aligned} \tag{3}$$

Therefore, the objective function can be rewritten as follows:

$$\begin{aligned}
J_T(X) = \sum_{i=1}^{c}\sum_{j=1}^{m}(\mu_{ij})^T \Big(K(x_j, x_j) \\
- \frac{1}{n_i^2}\sum_{k=1}^{m}\sum_{l=1}^{m}(\mu_{ik})^T(\mu_{il})^T K(x_k, x_l) \Big)
\end{aligned} \tag{4}$$

The kernel fuzzy c-means algorithm iteratively updates the new membership degree μ_{ij} at each iteration. The update of μ_{ij} in the feature space is defined through the kernel in the input space as follows:

$$\mu_{ij} = \left(\sum_{k=1}^{c} \left(\frac{\| \phi(x_j) - v_i^{\phi} \|^2}{\| \phi(x_j) - v_k^{\phi} \|^2} \right)^{1/(r-1)} \right)^{-1} \tag{5}$$

From (3), the kernel fuzzy c-means algorithm does not need to calculate the cluster centroids because the centroid information is considered in updating the membership degree μ_{ij} .

The kernel fuzzy c-means algorithm can be summarized as follows:

Step 1: Fix $c, t_{\max}, r > 1$ and $\varepsilon > 0$ for some positive constant.

Step 2: Initialize the memberships μ_{ij}^{0}.

Step 3: For \$t=1,2, t_{maz}, do:

(a) Update all memberships μ_{ij}^{t} with (5);

(b) Compute $E_t = \max_{ij} | \mu_{ij}^{t} - \mu_{ij}^{t-1} |$, if $E_t \le \varepsilon$, stop;

end;

In this paper, we will use kernel fuzzy c-means technique to classify audio frames, and select more suitable audio frames which have larger the fuzzy membership degree μ_{ij} to embed the watermark.

3. The Proposed Watermarking Method

3.1. Local Audio Features

In watermarking technique, the embedding strength may affect both imperceptibility and robustness of the watermark. Generally, the stronger embedding strength has the higher robustness and the lower imperceptibility, whereas the weaker embedding strength has the lower robustness and the higher imperceptibility. Since the different audio frames have the different contents or local features, the watermark capacity to be embedded varies with the different audio frames, that is, the embedding strength of each audio frame should be adaptively controlled by its local audio features. In this paper, we select the following local audio features to describe an audio frame:

(i) The energy E_k of audio frame. According to time masked characteristic, the stronger energy's audio signal can conceal the stronger watermark, vice versa. Let $A_k = \{a_k(i) \,|\, i = 1,2,\ldots L\}$ be kth audio frame. The energy E_k of the audio frame is calculated by the following formula:

$$E_k = \sum_{i=1}^{L} \left| a_k(i) \right|^2 \tag{6}$$

(ii) The maximal peaks P_k^r of each sub-band of audio frame. Since the different sub-bands of audio frame provide information about the different frequency distributions, the maximal peaks P_k^r of these detail sub-bands are selected as local features. For kth audio frame, we get the wavelet coefficients of sub-band A_k^3, D_k^3, D_k^2, D_k^1, where A_k^3 is the coarse sub-band, and the detail sub-bands are D_k^3, D_k^2, D_k^1. The maximal peaks P_k^r is defined as follows:

$$\mathrm{P}_k^0 = \max\{c \,|\, c \in A_k^3\}, \mathrm{P}_k^r = \max\{c \,|\, c \in D_k^{\lambda}\} \;\; r = 1,2,3 \qquad (7)$$

In this paper, for each audio frame, we extract its local features, energy E_k, maximal peaks $P_k^r(r = 0,1,2,3)$, and form a vector $x_k = (E_k, \mathrm{P}_k^0, \mathrm{P}_k^1, \mathrm{P}_k^2, \mathrm{P}_k^3) \in R^5$ which is used as the input vector of kernel fuzzy c-means method.

3.2. Watermark Embedding

Let A={a(i)| i=1,2,…,M } be an original audio signal with length M. The watermark is a significant binary image with size $N_1 \times N_2$, denoted by $W = \{w(i, j), 1 \le i \le N_1, 1 \le j \le N_2\}$. The watermark embedding procedure can be summarized as follows:

Step 1. To be secure, the image watermark is permuted and reshaped into line order as follows.

$$V = \{v(r) = w(i, j), \quad 1 \le i \le N_1, 1 \le j \le N_2, \\ r = (i-1) \times N_2 + j\} \tag{8}$$

$$\mathrm{W'} = \{w_{\mathrm{r}}, 1 \le r \le \lfloor N_1 N_2 / 2 \rfloor\} \\ w_{\mathrm{r}} = \begin{cases} v(2r-1) & r \le \lfloor N_1 N_2 / 2 \rfloor \\ v(2(r - \lfloor N_1 N_2 / 2 \rfloor)) & r > \lfloor N_1 N_2 / 2 \rfloor \end{cases} \tag{9}$$

Step 2. Decomposing the original audio signal.

The original audio signal is decomposed into the audio frames with length L:

$$A = \bigcup_{k=1}^{m} A_k = \bigcup_{k=1}^{m} \{a_k(i) \mid i = 1,2,...,L\} \tag{10}$$

where k=1, 2, …, m, and $m = \lfloor M / L \rfloor$. Each audio frame is decomposed through DWT in three levels, A_k^3 expresses its coarse sub-band, and $D_k^r (r = 1,2,3)$ express its detail sub-bands.

Step 3. Calculating the local features of audio frames.

For each audio frame A_k, we calculate its local features, energy E_k and maximal peaks $P_k^r (r = 0,1,2,3)$, and form a vector $x_k = (E_k, \mathrm{P}_k^0, \mathrm{P}_k^1, \mathrm{P}_k^2, \mathrm{P}_k^3) \in R^5$ which is used as the input vector of kernel fuzzy c-means method.

Step 4. Preprocessing by kernel fuzzy c-means clustering.

After calculating the local features of all audio frames, we obtain a data set $D = \{x_1, x_2, \dots, x_m\}$. Let c is the number of clustering. We run the kernel fuzzy c-means algorithm, and then obtain the fuzzy membership degrees μ_{ij}, $1 \le i \le c, 1 \le k \le m$. Thus, for each audio frame A_k, we obtain the corresponding fuzzy membership degrees $\mu_k = \max\{\mu_{ik} \mid 1 \le i \le c\}, 1 \le k \le m.$

Step 5. Selecting the audio fames to embed watermark.

Firstly, we select n' audio frames from the m audio frames, which have larger fuzzy membership degrees μ_k. Then, for each selected audio frame, we select two larger coefficients in D_k^3 as the embedding position. These selected coefficients are denoted by $V = \{c_i \mid i = 1,2,\dots,n\}$, where $n = 2 \times n'$, and $n \ge N_1 \times N_2$.

Step 6. Embedding the watermark.

For each coefficient in \$V\$, watermark embedding is accomplished by using following rule:

$$c' = c_i (1 + \mu_k \cdot \beta \cdot w_i) \tag{11}$$

where c_i is the coefficient of original audio signal, and $c_i{}'$ is the coefficient of watermarked audio signal. β is a constant, and μ_k is the fuzzy membership degrees of kth audio frame.

Step 7. Finally, each audio frame is reconstructed by applying the inverse DWT transform respectively, and then all audio frames are combined into the final watermarked audio signal A'.

3.3. Watermark Extraction

The watermark extraction is similar to the embedding procedure. The watermark extraction does not require the original audio signal. Suppose $\hat{A} = \{\hat{a}(i) \mid i = 1,2,3,\dots,M\}$ is an tested audio signal with length M.

The watermark extraction procedure can be summarized as follows:

Step 1. Let $\hat{A}$ be the watermarked audio signal with length M, which is decomposed into audio frames with length L, and each audio frame is decomposed through DWT in three levels respectively. $\overline{A}_k^3$ expresses the coarse sub-band, and $\overline{D}_k^3 (k = 1,2,3)$ express the detail sub-bands.

Step 2. Calculating the local features of audio frames.

For each audio frame $\hat{A}_k$, we calculate its local features, energy $\hat{E}_k$ and maximal peaks $\hat{P}_k^r (r = 0,1,2,3)$, and form a vector $\hat{x}_k = (\hat{E}_k, \hat{P}_k^0, \hat{P}_k^1, \hat{P}_k^2, \hat{P}_k^3) \in R^5$.

Step 3. Preprocessing by kernel fuzzy c-means clustering.

After calculating the local features of all audio frames, we obtain a data set $\hat{D} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_m\}$. By using the kernel fuzzy c-means algorithm in the same manner as in the embedding process, we obtain the fuzzy membership degrees μ_k of each audio frame $\hat{A}_k, 1 \le k \le m$.

Step 4. Extracting the watermark.

We select the embedding position of the watermark in the same manner as in the embedding process, and the selected coefficients are denoted by $\hat{V} = \{\hat{c}_i \mid i = 1,2,\ldots,n\}$. The watermark extraction can be described by the following equation:

$$w_i = \frac{(\hat{c}_i - c_i)}{(\mu_k \cdot \beta \cdot c_i)} \tag{12}$$

Where $i = 1,2,\ldots, N_1 \times N_2$.

Step 5. Finally, the one-dimensional sequence $w_1 w_2 \ldots w_{N_1 \times N_2}$ is converted into the two-dimensional logo watermark image w'.

4. Experimental Results

The original audio what we tested is the "*svega.wav*" file [8], a female singing, with 44.1kHz sampling rate and 16 bits/sample (length=20.67s, mono), shown in Figure 1(a). The signal *svega* is significant because it contains noticeable periods of silence, and the watermark should not be audible during these silent periods. The Haar 3-levels wavelet is used for DWT decomposition of each audio frame respectively.

The watermark is a binary image of size 32×32, shown in Figure 1(b).

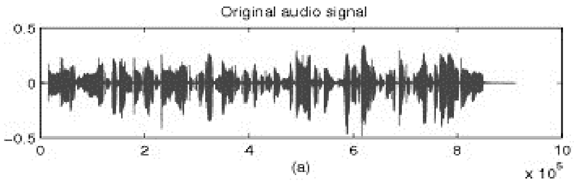

Figure 1. Digital watermark embedding process (a) Original audio signal (a female singing about 20.67 seconds in length, mono, 16 bits/sample, 44.1kHz sample rates). (b) Original watermark (a binary image of size 32×32).

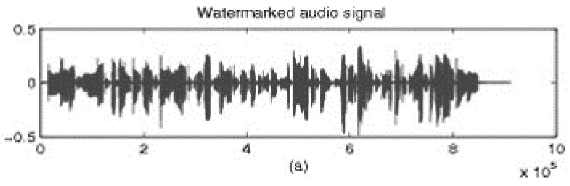

Figure 2. Digital watermark extracting process (a) Watermarked audio signal (PSNR=50.9385dB). (b) Extracted watermark (BER=0).

TABLE I. BER COMPARISON OF SEVERAL METHODS UNDER DIFFERENT ATTACKS

Signal Processing and Attacking	PSNR (dB)	Our BER	Wu's BER	Xu's BER
Attack free	50.9385	0	0	0
Echo addition	37.2245	0.0102	0.2038	0.0112
Blowup (50%)	23.3123	0	0	0
Reduce (50%)	27.7225	0	0	0
Re-sampling (11.025KHz)	31.1906	0.0312	0.5131	0.0514
Re-quantizing	45.2836	0	0	0
MP3 compression (128kb)	46.5217	0	0	0
MP3 compression (56kb)	45.0329	0.0035	0.5168	0.0058
Low-pass filtering (10KHz)	36.4519	0.0163	0.6142	0.0251
Additive noise	46.6596	0.0069	0.0212	0.0095

Firstly, we should decide some parameters in the experiment. For KFCM, we select Gaussian kernel as the kernel function of KFCM, and its the width parameter σ =210. Set the number of clustering c=64, and the constant β=0.015.

Figure 2(a) depicts the watermarked audio signal, which has PSNR value of 50.9385dB. If the original and the watermarked audio signals are observed, we cannot find any audible degradation. When there is no attack, the watermark can be extracted without a bit errors (BER=0), and the extracted watermarks are shown in Figure 2(b).

5. Conclusion

In this paper, we proposed an adaptive audio watermarking scheme. We extract local audio features from audio frames, which can well reflect the local audio contents or audible characteristics of audio signal. According to these local features, KFCM is used to select the audio frames and determine the embedding strength of each audio frame adaptively. The proposed approach can simultaneously improve robustness and audible quality of the watermarked audio signal. The experimental results show that the proposed method possesses significant robustness against the various attacks. After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

References

[1] H.J. Yang, J.C. Patra, C.W. Chan, “An artificial neural network-based scheme for robust watermarking of audio signals”, ICASSP'02, 1, 2002, pp.I-1029-1032.

[2] J. Wang, F.Z. Lin, “Digital audio watermarking based on support vector machine”. Journal of Computer Research and Development, 2005 Vol.42, No.9, pp.1605-1611. (in Chinese)

[3] S. Kirbiz, B. Gunsel, “Robust audio watermark decoding by supervised learning”, Proceedings of ICASSP 2006, 5, 2006, pp.V-761- V-764.

[4] X.-J. Xu, H. Peng, C.-Y. He, “DWT-based audio watermarking using support vector regression and subsampling”, In F.Masulli, S.Mitra, and G.Pasi (Eds.): WILF2007, LNAI 4578, 2007, pp. 136-144.

[5] H. Peng, X. Wang, W.X. Wang, J. Wang, D.Y. Hu, “Audio watermarking approach based on audio features in multiwavelet domain”, Journal of Computer Research and Development, 2010, 47(2), pp.216-222. (in Chinese)

[6] D.-W. Kima, K.Y. Leeb, D. Leea, K.H. Leea, “Evaluation of the performance of clustering algorithms in kernel-induced feature space”, Pattern Recognition, 38, 2005, pp.607-611.

[7] J.W. Liu, M.Z. Xu, “Kernelized fuzzy attribute c-means clustering algorithm”, Fuzzy Sets and Systems, 159, 2008, pp.2428-2445.

[8] http://www.petitcolas.net/fabien/steganography/mp3stego/index.html.

[9] S.Q. Wu, J.W.Huang, Y.Q. Shi, “Efficiently self-synchronized audio watermarking for assured audio data transmission”, IEEE Trans. on Broadcast, 2005, Vol 51, No.1 pp. 69-76.