Pairwise Boosted Audio Fingerprint

Dalwon Jang, *Student Member, IEEE*, Chang D. Yoo, *Member, IEEE*, Sunil Lee, *Member, IEEE*, Sungwoong Kim, *Student Member, IEEE*, and Ton Kalker, *Fellow, IEEE*

***Abstract*—A novel binary audio fingerprint obtained by filtering and then quantizing the spectral centroids is proposed. A feature selection algorithm, coined pairwise boosting (PB), is used to determine the filters and quantizers by casting the fingerprinting problem of identifying a query audio clip into a binary classification problem. The PB algorithm selects the filters and quantizers which lead to accurate classification of matching and nonmatching audio pairs: a matching pair is an audio pair that should be classified as being identical, and a nonmatching pair is a pair that should be classified as being different. By iteratively reducing the classification error of both matching and nonmatching pairs, the PB algorithm improves both the robustness and discriminating ability. In our experiments, the proposed fingerprint outperformed previously reported binary fingerprints in terms of robustness and discriminating ability. In the experiment, we compared the performances of a number of distance measures.**

***Index Terms*—Audio fingerprinting, boosting, content-based audio identification.**

I. Introduction

THE demand for protecting, managing, and indexing digital audio is growing, and as a viable solution, fingerprinting is receiving increased attention. An audio fingerprinting system extracts feature vectors known as fingerprints from a query audio clip, finds matching fingerprints in a database (DB), and retrieves the appropriate meta-data associated with the matching fingerprint in the DB. As a result of growing interest, various audio fingerprinting systems have been proposed in recent literature [2]–[9].

An audio fingerprint must 1) be able to *discriminate* between matching and nonmatching audio segments, must 2) be *robust* against expected signal degradations, and must 3) allow *efficient matching and searching*. Efficient matching and searching is critical in practical fingerprinting systems that may contain millions of entries [2], [10]. In general, matching and searching efficiency can be improved by using compact fingerprints with a suitable distance measure. Properly designed binary fingerprint will allow efficient matching and searching, and it can be obtained as a representation of the output of a multilevel quantizer.

This paper proposes a novel binary audio fingerprint obtained by quantizing the filtered outputs of spectral centroids [6]. The performance of the proposed fingerprint depends on the choice of filters and quantizers. The filters and quantizers are determined by casting the fingerprinting problem into a binary classification problem. By reducing the classification error, a feature selection algorithm, which is coined pairwise boosting (PB), selects the filters from a class of filters similar to the Haar wavelet filter appropriate for fingerprinting [12]. The PB algorithm also determines the appropriate quantizer for each selected filter. The PB algorithm is adapted from a well-known machine learning algorithm called Adaboost [13], [14] for the purpose of feature (the parameters used to extract the fingerprint) selection, and it requires the input to be a pair of data whereas Adaboost imposes no such restriction.

Given a set of labelled training data of perceptually similar (matching) and dissimilar (nonmatching) pairs of audio clips, the PB algorithm iteratively selects a classifier which leads to low classification error. In each iteration, a different training weight distribution is used: the distribution is updated at each iteration so that falsely classified data pairs are weighted more while those that are correctly classified are weighted less. The PB algorithm iteratively updates the weights of both matching and nonmatching pairs so that robustness and discriminating ability of the fingerprint are enhanced at each iteration. Previously, asymmetric pairwise boosting (APB), a variant of Adaboost, has been proposed in audio fingerprinting [7], [9] where only the weights of matching pairs are updated; however, this asymmetric update limits the discriminating ability of an audio fingerprint.

The remainder of this paper is organized as follows. Section II explains the PB algorithm and the extraction process of the proposed binary audio fingerprint. Section III explains the matching process for the proposed fingerprint. Section IV analyzes the PB algorithm. Section V presents the experimental results. Section VI concludes this paper.

II. Proposed Binary Audio Fingerprint

Fig. 1 shows the block diagram of the four stages of the extraction process of the proposed binary fingerprint: preprocessing, spectral subband centroid (SSC) computation [6], filtering, and quantization. From the viewpoint of compactly representing audio clip to perform fingerprinting, the input is being represented more compactly every stage.

Manuscript received September 15, 2008; revised September 16, 2009. Current version published November 18, 2009. This work was supported by Grant 2009-0083594 from the National Research Foundation of Korea. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Reginald Lagendijk.

D. Jang, C. D. Yoo, and S. Kim are with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea (e-mail: dal1@kaist.ac.kr; cdyoo@ee.kaist.ac.kr; leehwiso@kaist.ac.kr).

S. Lee is with Multimedia Platform Laboratory, Digital Media and Communications R&D Center, Samsung Electronics, Suwon 443-732, Korea (e-mail: sunil.lee@samsung.com).

T. Kalker is with Hewlett-Packard Laboratories, Multimedia Communications and Networking Laboratory, Palo Alto, CA 94304 USA (e-mail: ton.kalker@hp.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2009.2034452

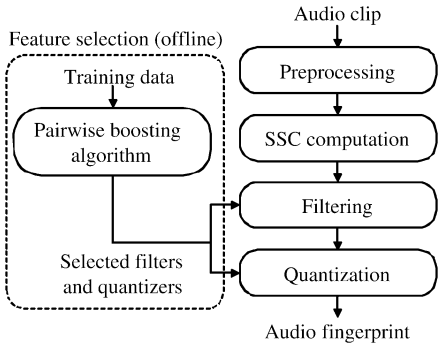

Fig. 1. Extraction process of the proposed audio fingerprint.

First, an audio clip is converted to a normalized format. Second, an M-dimensional SSC vector is calculated every frame of a preprocessed input audio clip. Third, every N consecutive M-dimensional SSC vector is appropriately filtered, and finally, the filtered output is quantized to produce the final binary audio fingerprint. The appropriate filter and quantizer are determined offline using the PB algorithm on a set of training data.

A. Preprocessing and SSC Computation [6]

An audio clip is converted to a normalized format (mono and 11 025-Hz sampling rate), and then framed by overlapping windows whose size and shift are denoted as L_w and L_s, respectively. An M-dimensional SSC vector is computed every frame: each frame is divided into M critical bands [11], and from each critical band, a spectral centroid is computed [6]. A binary fingerprint is derived from a collection of N consecutive SSC vectors referred to as an SSC image. An SSC image which is an $M \times N$ matrix is denoted as $\mathbf{x}$, and the (m,n)th element of $\mathbf{x}$ is denoted as $\mathbf{x}(m,n)$ where $1 \le m \le M$ and $1 \le n \le N$. For every frame shift, an SSC image is obtained from an audio of length $(L_s \times (N-1) + L_w)$.

B. Filter and Quantizer

From a number of candidate filters and candidate quantizers, the PB algorithm selects a set of filters and quantizers. Thus, candidate filters and quantizers must be defined prior to running the PB algorithm.

A filter $F : \mathbb{R}^{M \times N} \rightarrow \mathbb{R}$ is determined by the type, width (W), height (H), and reference point. The width and height determine the time-frequency support of the filter with respect to the reference point. Six filter types, W, H, and reference point are shown in Fig. 2. The filter computes the difference in sum of the SSC features in different time-frequency support, and $F(\mathbf{x})$ is mathematically expressed as

$$F(\mathbf{x}) = \sum_{(m,n)\in \mathbf{R_W}} \mathbf{x}(m,n) - \sum_{(m,n)\in \mathbf{R_B}} \mathbf{x}(m,n) \qquad (1)$$

where the set $\mathbf{R_W}$ and set $\mathbf{R_B}$ denote the white and black regions of a filter, as shown in the figure. The reference point $(\tilde{m}, \tilde{n})$ denotes the offset of the filter support in an SSC image

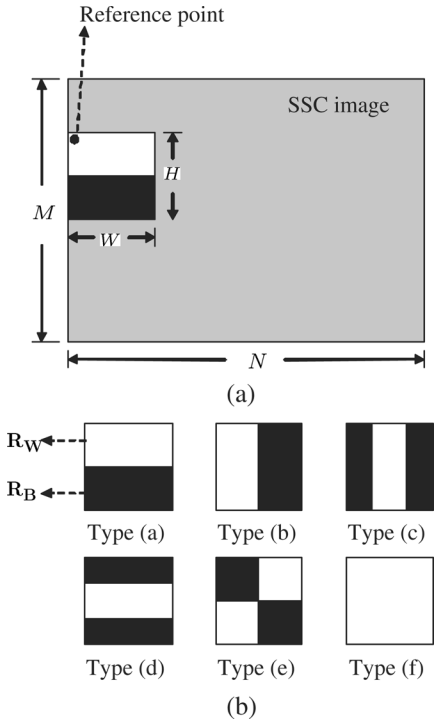

Fig. 2. Filtered output is computed as the difference between the sum of $\mathbf{x}(m,n)$ in white region and the sum of $\mathbf{x}(m,n)$ in black region. (a) Width (W), height (H), and reference point in an $M \times N$ SSC image. (b) Six filter types used in the proposed PB algorithm.

where $\tilde{m}$ and $\tilde{n}$ denote subband index and frame index, respectively. An SSC image is extracted for every frame shift, thus $\tilde{n} = 1$ since shifting the filter is equivalent to shifting the SSC image. Given an $M \times N$ SSC image, $\tilde{m}$, W, and H can vary. Thus, $1 \le \tilde{m} \le M, 1 \le W \le N$, and $1 \le H \le (M - \tilde{m} + 1)$. A filter type limits the ranges of W and H. For Type (b) and (e) filters, W must be a multiple of 2. For Type (c), W must be a multiple of 3. For Type (a) and (e) filters, H must be a multiple of 2. For Type (d), H must be a multiple of 3. To evaluate the performance of the proposed fingerprint, 3808 different candidate filters were considered with $M = 16$ and $N = 10$.

A quantizer $Q_{\mathbf{T}} : \mathbb{R} \rightarrow \{0, 1, \cdots, T\}$ is defined by a set of T thresholds, denoted as $\mathbf{T} = \{t_1, t_2, \cdots, t_T\}$, where $t_1 < t_2 < \cdots < t_T$. The quantizer $Q_{\mathbf{T}}(\cdot)$ is defined as

$$Q_{\mathbf{T}}(a) = \begin{cases} 0 & a < t_1 \\ j & t_j \le a < t_{j+1}, \quad (j = 1, \cdots, T-1) \\ T & t_T \le a. \end{cases} \qquad (2)$$

For a single candidate filter, a number of candidate quantizers are considered based on a number of candidate thresholds which are determined depending on the range of filtered outputs of the training data. For a filter, we set N_T candidate thresholds $(N_T \gg T)$ that minimize the mean squared quantization error of filtered outputs of the training data. Then, from possible ${}_{N_T}C_T = (N_T!)/(T!(N_T - T)!)$ combinations of T thresholds, one combination is selected. In our experiment, $N_T = 19$ and $T = 3$. Thus, 969 candidate quantizers were considered for each filter.

C. Binary Fingerprint Extraction

As shown in Fig. 1, an audio fingerprint is obtained by quantizing the filtered outputs. The binary fingerprint of an SSC image $\mathbf{x}$ is obtained by representing $Q_{\mathbf{T}}(F(\mathbf{x}))$ in binary form. Since a quantizer can have $(T+1)$ different outputs, the output of a quantizer can be represented in binary form using $L_B = \lceil \log_2(T+1) \rceil$ bits, where $\lceil r \rceil$ returns the smallest integer larger than r. Henceforth, the binary representation function is denoted as $B : \{0, 1, \cdots, T\} \rightarrow \{0,1\}^{L_B}$, where $T = 2^{L_B} - 1$. In our experiment, the Gray code is used for $B(\cdot)$ [15].

D. Pairwise Boosting

Using a collection of labelled pairs of SSC images as training data, the PB algorithm iteratively selects a set of classifiers. Each classifier is parameterized by a filter and quantizer, and these filters and quantizers are used for the fingerprint extraction. Each training data is labelled as either a matching (denoted as "+1") or a nonmatching pair (denoted as "−1"). With the criterion of minimizing the weighted classification error, the PB algorithm iteratively selects I number of classifiers.

A classifier defined by a particular filter and quantizer classifies a pair of SSC images into either a matching or a nonmatching pair. For a given SSC image pair $\mathbf{x_1}$ and $\mathbf{x_2}$, the classifier classifies the pair by determining whether the quantized outputs of $F(\mathbf{x}_1)$ and $F(\mathbf{x}_2)$ are equal or not. This is mathematically expressed as

$$h_{F,\mathbf{T}}(\mathbf{x_1}, \mathbf{x_2}) = \begin{cases} +1, & \text{if } Q_{\mathbf{T}}(F(\mathbf{x_1})) = Q_{\mathbf{T}}(F(\mathbf{x_2})) \\ -1, & \text{otherwise.} \end{cases} \tag{3}$$

A candidate classifier is parameterized by a different filter and quantizer, thus there are 3808×969 candidate classifiers. Selecting a classifier is equivalent to selecting a filter and its associated quantizer.

The PB algorithm is described in Fig. 3. Given D training data, the weight assigned to the dth data in the ith iteration $w_d^{(i)}$ is initialized to $(1)/(D)$ for $d = 1, \ldots, D$. In each iteration, the classifier $h_{F,\mathbf{T}}(\cdot)$ that minimizes the weighted error is selected from a number of possible candidate classifiers. The classifier, the filter, and the threshold set selected in the ith iteration are denoted as $h_{F^{(i)},\mathbf{T}^{(i)}}(\cdot)$, $F^{(i)}(\cdot)$, and $\mathbf{T}^{(i)}$, respectively. For notational simplicity, $h_{F^{(i)},\mathbf{T}^{(i)}}(\cdot)$ will be denoted as $h^{(i)}(\cdot)$. The weighted error of the ith iteration $\epsilon^{(i)}$ is defined as the sum of weights assigned to the pairs which are falsely classified by the ith classifier. The weight $w_d^{(i)}$ is updated and normalized such that the weights of falsely classified data are increased, and vise versa for correctly classified data. This is the same as in Adaboost [13], [14]. Weight normalization limits the range of the weighted error so that $0 \leq \epsilon^{(i)} \leq 1$.

After I iterations, I number of filters and quantizers are obtained, and these are used for the fingerprint extraction. By concatenating $B(Q_{\mathbf{T}^{(i)}}(F^{(i)}(\mathbf{x})))$ for $i = 1, \ldots, I$, the binary fingerprint vector of length $I \times L_B$ is obtained from an SSC image $\mathbf{x}$ every frame.

Input D training examples $(\mathbf{x}_{1,1}, \mathbf{x}_{2,1}, y_1), \cdots, (\mathbf{x}_{1,D}, \mathbf{x}_{2,D}, y_D)$ with label $y_d \in \{-1, +1\}$.

Initialization Weights $w_d^{(1)} = \frac{1}{D}$, $d = 1, \cdots, D$.

Do for $i = 1, \cdots, I$

1) Choose the ith classifier $h_{F^{(i)},\mathbf{T}^{(i)}}(\cdot)(= h^{(i)}(\cdot))$ that minimizes weighted error
$$\epsilon^{(i)} = \sum_{d=1}^{D} w_d^{(i)} \cdot \delta(h^{(i)}(\mathbf{x}_{1,d}, \mathbf{x}_{2,d}) \neq y_d)$$
where $\delta(\theta) = 1$ when event θ occurs and $\delta(\theta) = 0$ otherwise.
2) Update weights:
$$w_d^{(i+1)} = w_d^{(i)} \cdot \left(\frac{1-\epsilon^{(i)}}{\epsilon^{(i)}}\right)^{-h^{(i)}(\mathbf{x}_{1,d}, \mathbf{x}_{2,d}) \cdot y_d}.$$
3) Normalize weights so that $\sum_{d=1}^{D} w_d^{(i+1)} = 1$.

Output $F^{(i)}$ and $\mathbf{T}^{(i)}$ for $i = 1, \cdots, I$

Fig. 3. Summary of the PB algorithm.

III. Fingerprint Matching

For a fingerprinting system to find a matching fingerprint, the distance measure between two fingerprints must be defined. For the proposed fingerprint, two distance measures are considered: the Hamming distance (i.e., the number of bit errors) [2] and a distance considering the output of $Q_{\mathbf{T}}(\cdot)$. The functions to compute the two distances are denoted as $D_H(\cdot)$ and $D_Q(\cdot)$, respectively. The Hamming distance between two $(I \times L_B)$-bit binary fingerprint vectors $\mathbf{b}_1$ and $\mathbf{b}_2$ is computed as

$$D_H(\mathbf{b}_1, \mathbf{b}_2) = \sum_{i=1}^{I \times L_B} \delta(\mathbf{b}_1(i) \neq \mathbf{b}_2(i)) \tag{4}$$

where $\mathbf{b}_k(i)$ denotes the ith element of $\mathbf{b}_k$ for $k = 1, 2$, and $\delta(\cdot)$ is an indicator function defined by

$$\delta(\theta) = \begin{cases} 1, & \text{if } \theta \text{ is true} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The distance $D_Q(\cdot)$ between two binary fingerprints $\mathbf{b}_1$ and $\mathbf{b}_2$, extracted from SSC images $\mathbf{x}_1$ and $\mathbf{x}_2$, is computed as

$$D_Q(\mathbf{b}_1, \mathbf{b}_2) = \sum_{i=1}^{I} |B^{-1}(C_{L_B}(\mathbf{b}_1, i)) - B^{-1}(C_{L_B}(\mathbf{b}_2, i))| \tag{6}$$

where $B^{-1} : \{0,1\}^{L_B} \rightarrow \{0, 1, \cdots, T\}$ is the inverse function of $B(\cdot)$, and $C_{L_B}(\mathbf{b}_k, i)$ is an L_B-bit sequence defined as

$$C_{L_B}(\mathbf{b}_k, i) = [\mathbf{b}_k((i-1) \times L_B + 1) \cdots \mathbf{b}_k(i \times L_B)] \tag{7}$$

for $k = 1, 2$ and $i = 1, 2, \cdots, I$. We can rewrite $D_Q(\cdot)$ as

$$D_Q(\mathbf{b}_1, \mathbf{b}_2) = \sum_{i=1}^{I} \left| Q_{\mathbf{T}^{(i)}}\left(F^{(i)}(\mathbf{x}_1)\right) - Q_{\mathbf{T}^{(i)}}\left(F^{(i)}(\mathbf{x}_2)\right) \right|. \tag{8}$$

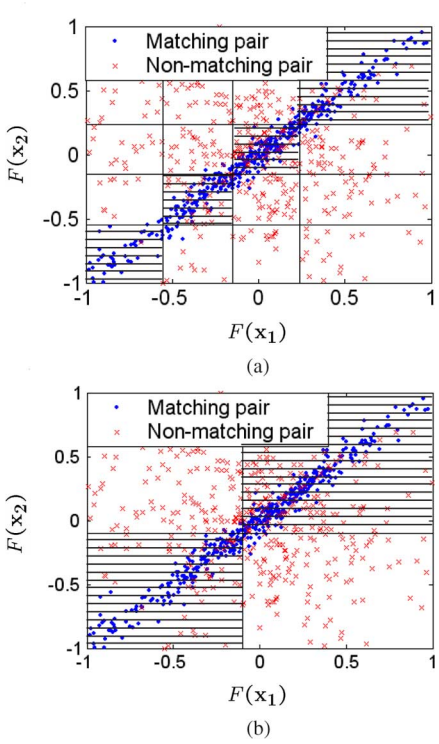

Fig. 4. Scatter plot of $(F(\mathbf{x_1}), F(\mathbf{x_2}))$ for matching and nonmatching pairs and (a) a classifier with three thresholds and (b) a classifier with a single threshold. Horizontal and vertical lines indicate threshold, and the shaded region indicates region where a pair is classified as a matching pair.

Thus, $D_Q(\cdot)$ measures the distance between two quantized outputs. $D_H(\cdot)$ is performed at the bit-level while $D_Q(\cdot)$ is performed at the integer-level and requires $B^{-1}(\cdot)$ mapping. Thus, $D_H(\cdot)$ requires less computational complexity than $D_Q(\cdot)$. But, $D_Q(\cdot)$ performs slightly better than $D_H(\cdot)$. This is experimentally verified in Section V.

We assume that the audio identification is performed based on K consecutive SSC images which are obtained for every frame shift. The fingerprint vectors of K SSC images are extracted from a query audio clip, and the matching fingerprint is determined using K number of $I \times L_B$-bit fingerprint vectors.

IV. Analysis of the PB Algorithm

It should be realized that the notion of classifying matching pairs and nonmatching pairs relates respectively to the robustness and the discriminating ability. For correct classification of matching pairs, the two fingerprints extracted from a matching pair should be the same, and this relates to the robustness. For correct classification of nonmatching pairs, the two fingerprints extracted from a nonmatching pair should be different, and this relates to the discriminating ability of the fingerprint.

The PB algorithm iteratively selects a classifier such that the classifier selected at each iteration can better classify those training data which the classifier selected at the previous iteration could not classify. By updating and normalizing the weight distribution of the training data, the weights of falsely classified data are increased, and the weights of correctly classified data are decreased. In order to minimize the weighted classification error, the selected classifier will place more emphasis on data with high weights than data with low weight. Thus, $h^{(i+1)}$ will better classify those data which were incorrectly classified by $h^{(i)}$. This will only happen when $\epsilon^{(i)} < 0.5$ as in the case of Adaboost [13]. The weight update equation in Fig. 3 indicates that weights are not changed when $\epsilon^{(i)} = 0.5$ and that the weights of correctly classified data are increased when $\epsilon^{(i)} > 0.5$. To update the weights of nonmatching pairs with $\epsilon^{(i)} < 0.5$, $T \geq 2$. Let us suppose that an SSC image $\mathbf{x}$ is randomly drawn from a distribution such that $\Pr(Q_{\mathbf{T}}(F(\mathbf{x})) = k) = p_k$ for $k = 0, 1, \ldots, T$, where $\Pr(\theta)$ is the probability of event θ. If two nonmatching SSC images $\mathbf{x_1}$ and $\mathbf{x_2}$ are independently and randomly selected, then the probability that the pair $(\mathbf{x_1}, \mathbf{x_2})$ is classified as a matching pair is given by

$$\begin{aligned}&\Pr(h_{F,\mathbf{T}}(\mathbf{x_1}, \mathbf{x_2}) = +1)\\ &\quad = \sum_{k=0}^{T} \Pr(Q_{\mathbf{T}}(F(\mathbf{x_1})) = k) \times \Pr(Q_{\mathbf{T}}(F(\mathbf{x_2})) = k)\\ &\quad = \sum_{k=0}^{T} p_k^2 \geq \frac{1}{T+1}\end{aligned} \tag{9}$$

where the inequality is derived using $\sum_{k=0}^{T} p_k = 1$ and the Cauchy–Schwartz inequality.[1] When $T = 1$, the PB algorithm performs no better than a random coin toss for classifying nonmatching pairs, and this does not lead to $\epsilon^{(i)} < 0.5$ for nonmatching pairs. Thus, the APB algorithm [7] does not update the weights of nonmatching pairs. This asymmetry limits the discriminating ability. It is necessary to update weights of both matching and nonmatching pairs in order to prevent asymmetry. To update weights of nonmatching pairs, $\epsilon^{(i)} < 0.5$ should be satisfied for nonmatching pairs. From (9), the necessary condition that $h_{F,\mathbf{T}}(.)$ satisfies $\epsilon^{(i)} < 0.5$ for nonmatching pairs is $T \geq 2$.

Fig. 4 compares two classifiers—one based on three thresholds $(T = 3)$ and one based on a single threshold $(T = 1)$. The figure shows the scatter plot of $(F(\mathbf{x_1}), F(\mathbf{x_2}))$ overlayed with the classification results of the classifier with three thresholds and with a single threshold. As shown in Fig. 4, using a classifier with a single threshold, about half of nonmatching pairs are classified as matching pairs (indicated by the shaded region), whereas using a classifier with three thresholds, the classification error for nonmatching pair is much less: the classification error for nonmatching pair is 0.558 in Fig. 4(b), whereas it is only 0.249 in Fig. 4(a).

The PB algorithm does not construct a combined classifier as in Adaboot [13], [14]. However, the combined classifier, that can be constructed by the PB as in the case of Adaboost, can be used as a similarity measure between two binary fingerprints. After I iterations, Adaboost [13], [14] outputs a combined classifier defined by

$$\hat{h}^{(I)}(\cdot) = \begin{cases} +1, & \text{if } \sum_{i=1}^{I} c^{(i)} h^{(i)}(\cdot) \geq 0 \\ -1, & \text{otherwise.} \end{cases} \tag{10}$$

where the confidence value $c^{(i)}$ is defined by

$$c^{(i)} = \log\left(\frac{1 - \epsilon^{(i)}}{\epsilon^{(i)}}\right). \tag{11}$$

[1] $(\sum_{k=0}^{T} p_k \times 1)^2 \leq (\sum_{k=0}^{T} p_k^2) \times (\sum_{k=0}^{T} 1^2)$.

TABLE I
SELECTED FILTERS AND QUANTIZERS

Filter				Quantizer		
Type	Reference point	H	W	t_1	t_2	t_3
(a)	(3,1)	2	9	-1.371	-0.009	1.411
(b)	(5,1)	1	10	-0.453	-0.033	0.383
(a)	(5,1)	2	10	-0.997	0.345	1.640
(e)	(3,1)	2	8	-0.544	-0.126	0.411
(a)	(6,1)	6	5	-1.025	0.022	0.852
(a)	(1,1)	2	10	-0.988	0.272	1.053
(b)	(12,1)	1	10	-0.293	0.015	0.303
(b)	(4,1)	2	6	-0.339	0.033	0.349
(f)	(5,1)	3	10	-1.503	-0.542	0.361
(d)	(7,1)	3	8	-1.635	-0.600	0.491
(e)	(1,1)	4	10	-1.022	-0.016	0.955
(e)	(7,1)	2	10	-0.670	0.048	0.436
(f)	(3,1)	2	9	-1.906	-0.212	0.729
(b)	(4,1)	1	4	-0.101	0.017	0.189
(b)	(2,1)	2	10	-0.746	0.123	0.735
(f)	(6,1)	9	6	-1.700	-0.766	0.143
(e)	(9,1)	2	10	-0.487	0.232	1.096
(e)	(13,1)	2	8	-0.393	-0.045	0.409
(b)	(12,1)	1	4	-0.124	-0.023	0.090
(e)	(5,1)	2	6	-0.696	-0.085	0.364
(d)	(8,1)	6	6	-1.410	-0.226	1.348
(f)	(1,1)	4	5	-1.262	-0.752	1.289
(f)	(8,1)	2	10	-1.577	-0.075	0.499
(e)	(3,1)	4	8	-0.676	0.109	0.841
(b)	(7,1)	2	8	-0.265	0.241	0.843
(a)	(8,1)	2	5	-1.073	0.148	0.657
(c)	(3,1)	7	9	-1.126	-0.139	0.578
(b)	(8,1)	3	10	-0.916	-0.307	0.404
(a)	(1,1)	2	4	-0.459	0.133	0.616
(e)	(9,1)	2	6	-0.625	-0.176	0.159
(b)	(14,1)	1	8	-0.299	0.023	0.138
(f)	(3,1)	1	4	-0.405	0.050	0.634

Based on the combined classifier, the similarity between two $(I \times L_B)$-bit fingerprint vectors $\mathbf{b}_1$ and $\mathbf{b}_2$, extracted from $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively, can be computed as

$$S(\mathbf{b}_1, \mathbf{b}_2) = \sum_{i=1}^{I} c^{(i)} h^{(i)}(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=1}^{I} c^{(i)} (2\delta(C_{L_B}(\mathbf{b}_1, i) = C_{L_B}(\mathbf{b}_2, i)) - 1).$$

We compared this similarity measure to $D_H(\cdot)$ and $D_Q(\cdot)$. Our experimental results in Section V-G show that $D_H(\cdot)$ and $D_Q(\cdot)$ are better than $S(\cdot)$.

V. EXPERIMENTAL RESULTS

A. Experimental Setup

In our experiment, $I = 16$ or 32, $T = 3(L_B = 2), N = 10$, $L_w = 371.52$ ms, and $L_s = 185.76$ ms, and this leads to a 32- or 64-bit fingerprint vector for an input audio of length $2.04s(= L_s \times (N-1) + L_w)$. To compute an SSC feature, 16 nonoverlapping frequency bands are selected $(M = 16)$ where they lie in the range from 300 to 5300 Hz [6]. We assume that the basic unit for audio identification is 5 s. For this to happen, $K = 16(2.04s + L_s \times (K-1) = 4.83s)$, thus a 5-s audio clip is identified using 16×32-bit or 16×64-bit fingerprint.

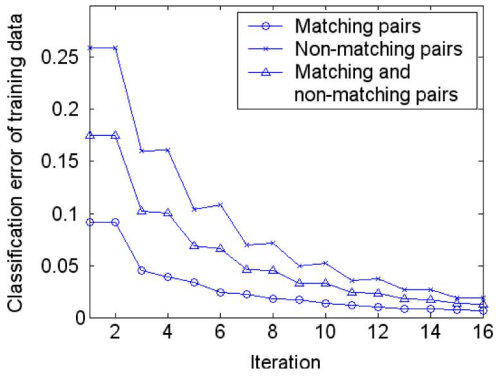

Fig. 5. Classification error of training data.

B. Training Data

A set of training data of matching and nonmatching pairs are required as input to the PB algorithm. From original and distorted audio, matching and nonmatching pairs are generated. Training data should be as large as possible to reflect the testing environment and should include various distortions, which can occur in a practical fingerprinting system. The list of audio distortions considered in our experiment is as follows [4]:

1) Time delay (TD): 92.9-ms shift.
2) Octave band equalization (EQ1): Adjacent band attenuations set to -6 and $+6$ dB in an alternating fashion.
3) Volume change (V): Envelop tremors.
4) Echo (E): Filter-emulating old time radio.
5) Bandpass filtering (BPF): 400-Hz to 4-kHz bandpass filtering.
6) WMA encoding (WMA): 64-kb/s WMA encoding.
7) 1/3 octave band equalization (EQ2): 30-band pop equalization.
8) Sampling rate change (SR): Down-sampling to 16 kHz and up-sampling to 44.1 kHz.

For all audio clips used in this experiment, 96-kb/s MP3 encoding (MP3) is performed in addition to the above distortions. From 100 original and 800 distorted songs obtained by distorting the audio with one of the above distortions, about 22 000 matching and 22 000 nonmatching pairs are selected and used in the feature selection.

C. Selected Filters and Quantizers

The filters and quantizers selected by the PB algorithm are shown in Table I when $I = 32$. From 3808 candidate filters, 32 filters and their associated quantizers are selected. As shown in Table I, 32 filters of various types and sizes are selected. There is very little redundancy between the selected filters. No two filters are the same. The filters with small H and large W are mainly selected. Among the 32 selected filters, 22 filters have $H \leq 2$, and 20 filters have $W \geq 8$. This selection can be interpreted as

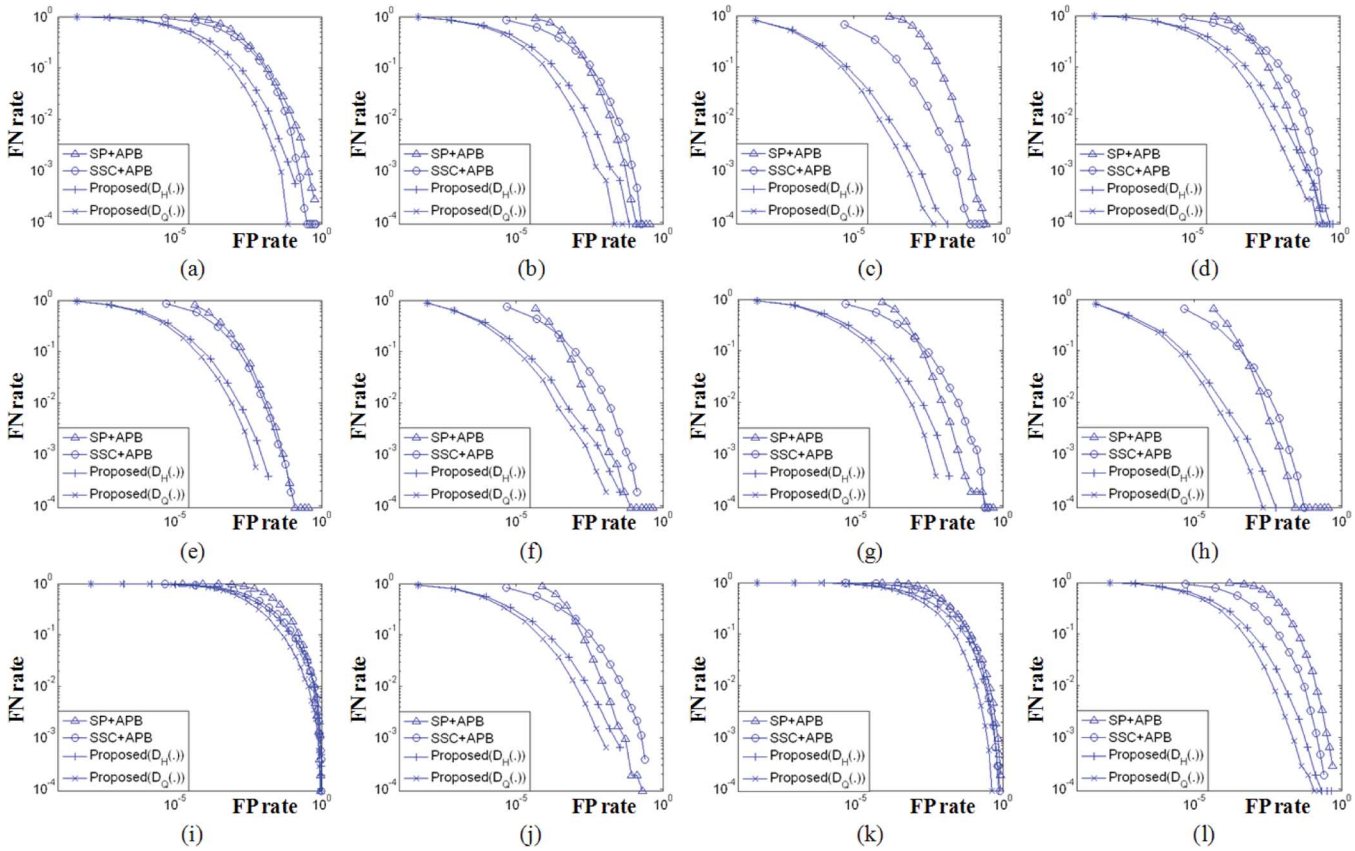

Fig. 6. Experimental results of 32-bit ($I = 16$) fingerprint against various distortions: (a) TD+MP3. (b) EQ1+MP3. (c) V+MP3. (d) E+MP3. (e) BPF+MP3. (f) WMA+MP3. (g) EQ2+MP3. (h) SR+MP3. (i) TD+EQ1+V+E+MP3. (j) WMA+EQ2+SR+MP3. (k) TD+E+BPF+EQ2+MP3. (l) EQ1+V+BPF+WMA+MP3.

follows: the fingerprints with high frequency resolution and low temporal resolution are selected.

D. Training Error

Fig. 5 shows the classification error of the training data at each iteration. The classification error is defined as the ratio of the number of the misclassified data to the total number of data. The classification error of training data at each iteration is obtained using $\hat{h}^{(i)}(\cdot)$. As written in (10), the confidence values are used as weights, and in our experiment, the weights are nearly equal across all individual classifiers selected each iteration. In the figure, we notice a staircase behavior of the error curve. When the weights are nearly equal as was obtained in our experiment and the number of iteration is small, the combined classifier performs a majority vote on the outputs of classifiers, and the overall error is particularly reduced at odd iterations. In the figure, the error for nonmatching pairs is larger than the error for matching pairs. At odd iterations, the PB algorithm focuses more on reducing the error for nonmatching pairs than on reducing the error for matching pairs in order to reduce the overall error. Thus, the error curve for nonmatching pairs looks like a staircase while the error curve for matching pairs does not. In the figure, we also notice that the error of nonmatching pairs sometimes increases. The overall classification error, which is the average of the classification error of matching and nonmatching pairs, decreases in each iteration. The PB algorithm reduces the overall classification error and does not guarantee that the classification error of the nonmatching pair will decrease. In Adaboost, the training error of the combined classifier decreases as the number of iteration increases, and this means that the classifiers are properly selected [13]. In the same manner, the reduction of training error in the PB algorithm also indicates that the classifiers are properly selected.

E. Comparative Test

The performance of the proposed fingerprint is compared with those of other binary fingerprints: the fingerprint using the spectrogram and the APB algorithm [7] (denoted as "SP+APB") and the fingerprint using the SSC and the APB algorithm [9] (denoted as "SSC+APB"). In [7] and [9], it was experimentally verified that the two fingerprints are better than the binary fingerprint of Philips fingerprinting system [2]. An identical training set is used to obtain all fingerprints, and the test sets for all fingerprints are also identical. The fingerprint rate was fixed to the following: either 32- or 64-bit fingerprint vector is extracted from 2.04 s of audio segment. The number of frequency bands, band division, frame length, and frame shift are identically set with those of the proposed fingerprint. The performances of fingerprints of "SP+APB" and "SSC+APB" are obtained using $D_H(\cdot)$, and the performance of the proposed fingerprint is obtained using $D_H(\cdot)$ and $D_Q(\cdot)$.

Figs. 6 and 7 compare the performance of the binary fingerprints by showing the receiver operating characteristic

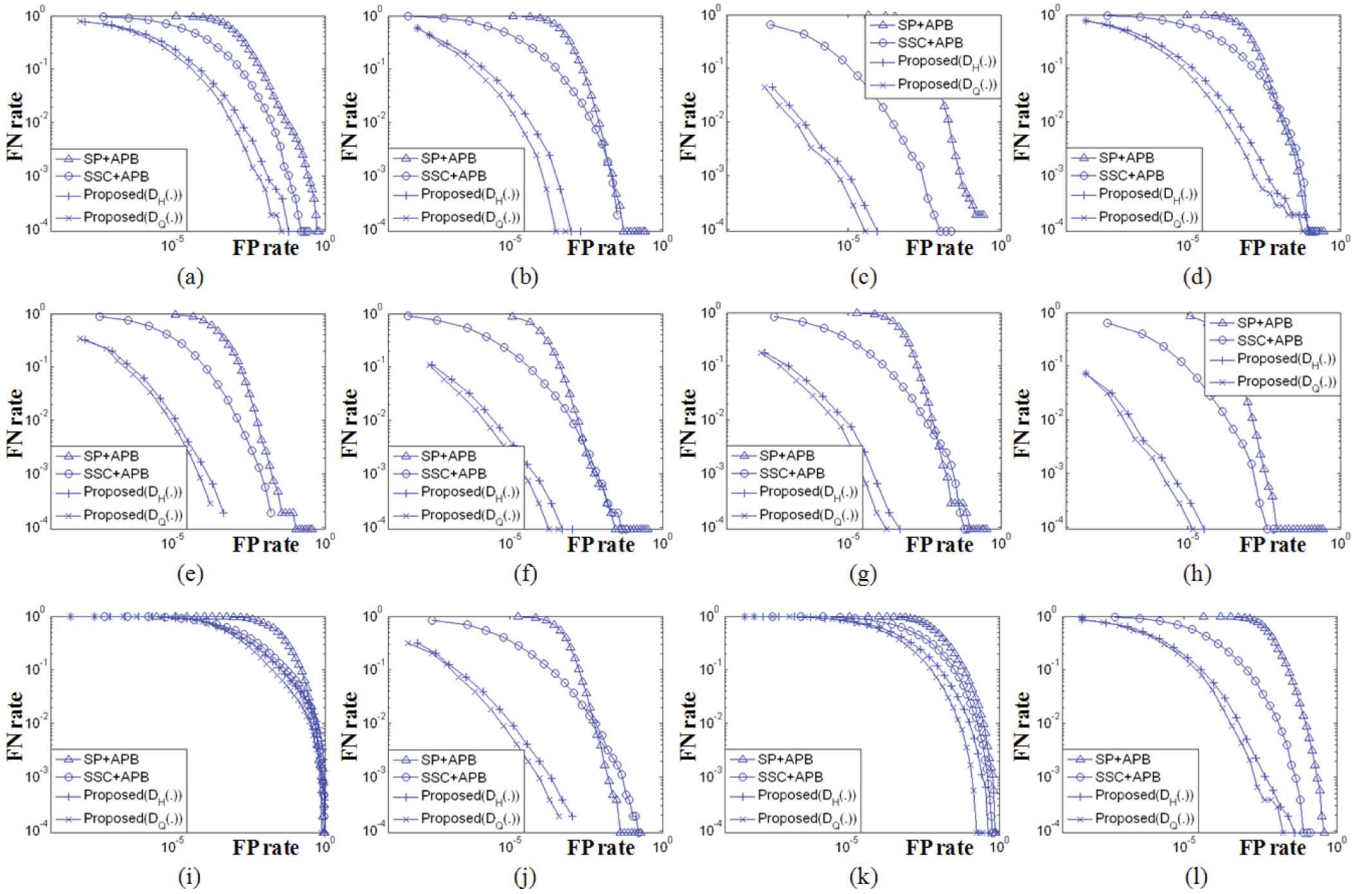

Fig. 7. Experimental results of 64-bit ($I = 32$) fingerprint against various distortions: (a) TD+MP3. (b) EQ1+MP3. (c) V+MP3. (d) E+MP3. (e) BPF+MP3. (f) WMA+MP3. (g) EQ2+MP3. (h) SR+MP3. (i) TD+EQ1+V+E+MP3. (j) WMA+EQ2+SR+MP3. (k) TD+E+BPF+EQ2+MP3. (l) EQ1+V+BPF+WMA+MP3.

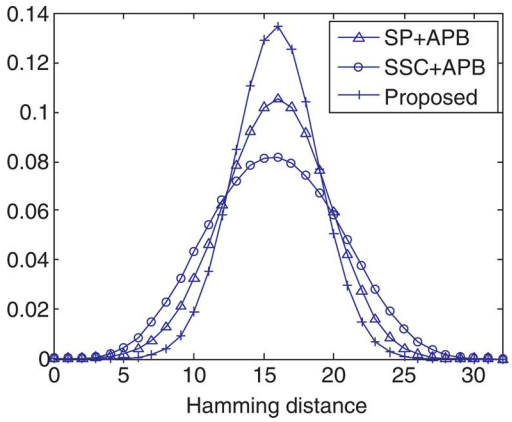

Fig. 8. Distribution of the Hamming distance for nonmatching pairs.

(ROC) curve which plots the false negative (FN) rate versus the false positive (FP) rate. The FN rate is defined as the rate that matching pairs are falsely classified, and it relates to robustness. The FP rate is defined as the rate that nonmatching pairs are falsely classified, and it relates to the discriminating ability of the fingerprint. Fig. 6 shows the performance of 32-bit fingerprints extracted from 2.04 s of audio segment, and Fig. 7 shows the performance of 64-bit fingerprints extracted from 2.04 s of audio segment. For each experiment, about 11 000 matching and 220 000 000 nonmatching pairs were used. For performance evaluation, a test set completely separate from the training set was used. Figures (a)–(h) show, respectively,

TABLE II
AUDIO IDENTIFICATION RATE (SET1: TD+EQ1+V+E+MP3; SET2: WMA+EQ2+SR+MP3; SET3: TD+E+BPF+EQ2+MP3; AND SET4: EQ1+V+BPF+WMA+MP3)

	Distortion	SP+APB	SSC+APB	Proposed($D_H(\cdot)$, $D_Q(\cdot)$)
16 × 32 bit	TD+MP3	99.4	99.8	100.0, 100.0
	EQ1+MP3	99.2	99.8	100.0, 100.0
	V+MP3	92.8	100.0	100.0, 100.0
	E+MP3	98.8	99.6	99.9, 99.9
	BPF+MP3	99.9	100.0	100.0, 100.0
	WMA+MP3	99.9	100.0	100.0, 100.0
	EQ2+MP3	98.6	99.9	100.0, 100.0
	SR+MP3	99.9	100.0	100.0, 100.0
	SET1	51.0	74.8	82.5, 85.0
	SET2	98.6	99.8	100.0, 100.0
	SET3	77.6	77.3	88.4, 89.8
	SET4	85.3	99.4	99.9, 99.9
16 × 64 bit	SET1	47.2	86.1	88.6, 88.9
	SET2	99.4	100.0	100.0, 100.0
	SET3	77.0	88.1	97.1, 96.9
	SET4	84.7	100.0	100.0, 100.0

TABLE III
FILTERS AND QUANTIZERS SELECTED WITH NO ITERATION

Filter				Quantizer		
Type	Reference point	H	W	t_1	t_2	t_3
(a)	(3,1)	2	9	-1.371	-0.009	1.411
(a)	(3,1)	2	10	-0.872	0.216	1.504
(d)	(4,1)	3	9	-1.609	-0.053	1.805
(a)	(3,1)	2	7	-1.185	-0.015	0.928
(a)	(5,1)	2	10	-0.997	0.165	1.640
(a)	(5,1)	2	6	-0.612	0.245	1.250
(a)	(5,1)	2	9	-0.620	0.305	1.509
(d)	(4,1)	3	8	-1.193	0.194	1.703
(a)	(3,1)	2	8	-1.284	0.165	1.302
(d)	(4,1)	3	10	-1.729	-0.061	1.920
(d)	(4,1)	3	7	-1.350	-0.053	1.185
(a)	(5,1)	2	7	-0.528	0.431	1.660
(a)	(5,1)	2	8	-1.101	0.137	1.126
(d)	(4,1)	3	4	-0.670	0.136	0.997
(d)	(4,1)	3	5	-1.325	-0.302	0.704
(a)	(3,1)	2	5	-0.917	-0.167	0.585

experimental results for eight distortions used in the feature selection, and (i)–(l) show experimental results for four combined distortions which are not directly used in the feature selection.

As shown in Figs. 6 and 7, the proposed fingerprint outperforms other fingerprints for every distortion condition considered. Of the two distance measures, $D_Q(\cdot)$ slightly outperforms $D_H(\cdot)$.

The FP rate of the proposed fingerprint is much lower than the FP rates of other fingerprints using the APB algorithm: the APB algorithm which is also a variant of Adaboost cannot update the weights of nonmatching pairs because it uses a single threshold. As shown in (9), when $T = 1$, the classifier violates the condition of $\epsilon^{(i)} < 0.5$ for nonmatching pairs. This limits the discriminating ability of the fingerprint extracted using the APB algorithm, but the PB algorithm can improve discriminating ability as well as the robustness by updating the weights of both matching and nonmatching pairs by using $T \geq 2$.

To further compare the discriminating ability of the proposed fingerprint and those of other fingerprints, Fig. 8 shows the distribution of $D_H(\cdot)$ of nonmatching pairs. The distribution of $D_Q(\cdot)$ of nonmatching pair is not presented in the figure since the range of $D_Q(\cdot)$ is different from that of $D_H(\cdot)$. As shown in the figure, the mean values for three fingerprints are similar, but the distribution of $D_H(\cdot)$ of the proposed fingerprint has a smaller variance. For a fixed threshold which is lower than the mean value of $D_H(\cdot)$, the proposed fingerprint produced smaller error: the error is determined by the area of distribution below the threshold. Smaller error means a lower FP rate and better discriminating ability. The APB algorithm does not update the weight of the nonmatching pair, thus the classifiers which repeat classification error for the nonmatching pair can be selected. But, the PB algorithm prevents repeated classification error for the nonmatching pair, and this leads to lower FP rate.

To compare the audio identification performances of different binary audio fingerprints, the audio identification rate based on

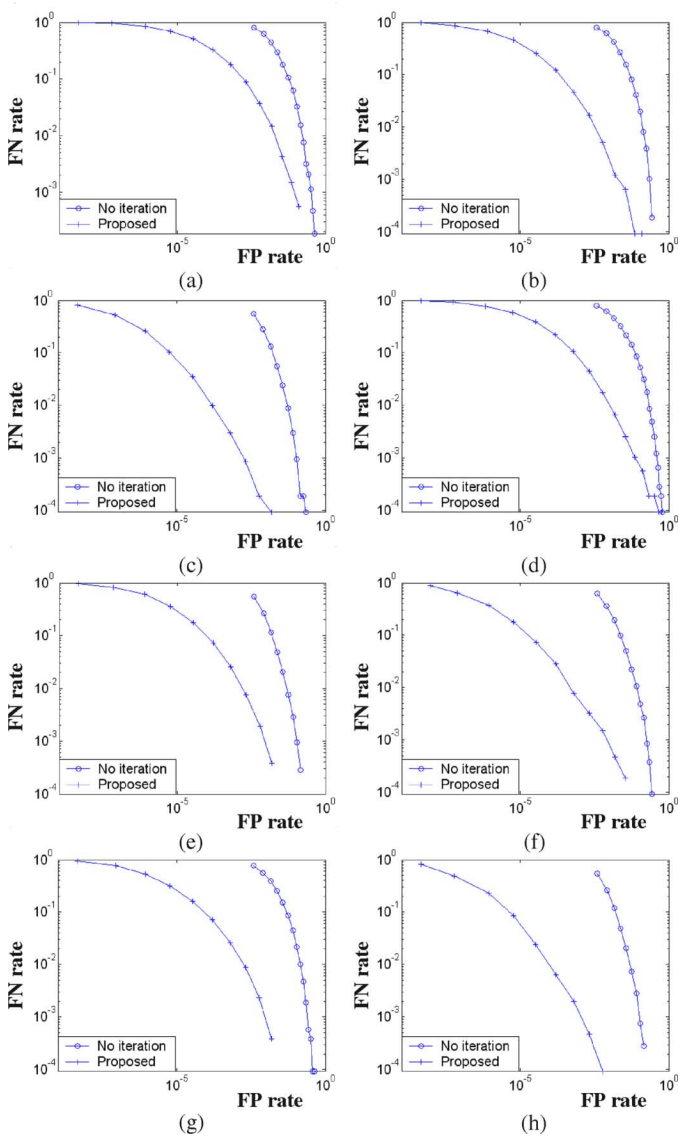

Fig. 9. Experimental results of the proposed fingerprint and fingerprint obtained from filters and quantizers which are selected with no iteration: (a) TD+MP3. (b) EQ1+MP3. (c) V+MP3. (d) E+MP3. (e) BPF+MP3. (f) WMA+MP3. (g) EQ2+MP3. (h) SR+MP3.

a 5-s audio clip $(K = 16)$ is computed using a fingerprint DB of 3000 songs. To evaluate the performance, 4400 audio clips of length 5 s were extracted randomly from 100 different songs already registered in the DB. In this experiment, we excluded the query audio which is not registered in the DB, and the 4400 audio clips are used as query audio clips. In this experiment, it is assumed that a query audio clip is correctly identified when the distance (in terms of either $D_H(\cdot)$ or $D_Q(\cdot)$) between the fingerprint of the query audio clip and the corresponding fingerprint in the DB is the smallest. Identification rate is defined as the number of correctly identified query audio clips over the number of all query audio clips. Identification rates for both 16×32-bit and 16×64-bit fingerprints are shown in Table II. As shown in the table, the proposed fingerprint outperforms other fingerprints for all distortions and fingerprint lengths considered.

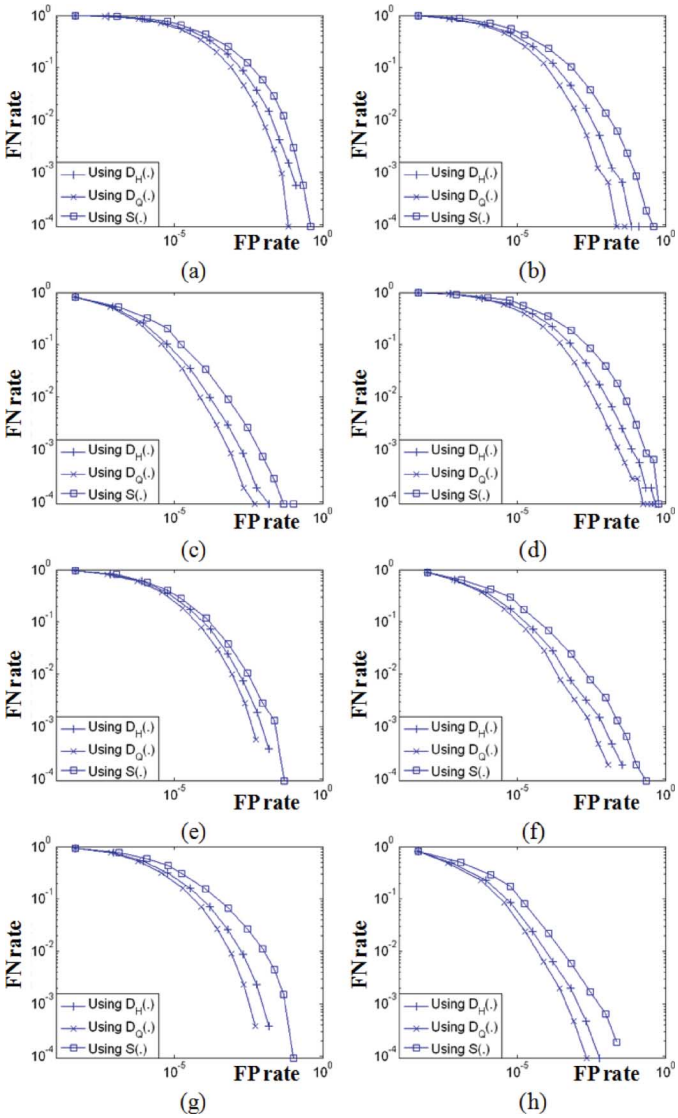

Fig. 10. Experimental results using $D_H(\cdot), D_Q(\cdot)$, and $S(\cdot)$ against various distortions: (a) TD+MP3. (b) EQ1+MP3. (c) V+MP3. (d) E+MP3. (e) BPF+MP3. (f) WMA+MP3. (g) EQ2+MP3. (h) SR+MP3.

F. Feature Selection With No Iteration

In this subsection, the proposed fingerprint is compared with the fingerprint obtained using filters and quantizers which are selected with no iteration. Rather than iteratively selecting the filters and quantizers, top I ranked filters and quantizers that give the smallest classification error are selected. Table III shows the $I(=16)$ filters and their associated quantizers, which are selected with no iteration. Unlike the filters in Table I, filters with similar characteristics in terms of type and support are repeatedly selected. For example, first, second, and fourth filters shown in Table III have the same type, the same reference point, and the same H. The similarity of filters leads to a lot of redundancy between elements of the fingerprint vector. Fig. 9 compares the performance of the proposed fingerprint with that of the fingerprint obtained using filters and quantizers which are selected with no iteration. In this experiment, a 32-bit fingerprint $(I = 16)$ extracted from an SSC image and $D_H(\cdot)$ were used. Performances against eight distortions used in the feature selection are presented in the figure. The fingerprint obtained from filters and quantizers, which are selected with no iteration, shows much higher FP rate than the proposed fingerprint, and it means the fingerprint has poor discriminating ability.

G. Similarity Measure Based on the Combined Classifier of Adaboost

Fig. 10 compares the performances of two distance measures and a similarity measure based on the combined classifier of Adaboost. In this experiment, 32-bit fingerprint $(I = 16)$ extracted from an SSC image was used. Performances against eight distortions used in the feature selection are presented in the figure. As shown in Fig. 10, $D_H(\cdot)$ and $D_Q(\cdot)$ outperform $S(\cdot)$. In terms of computation, $D_H(\cdot)$ and $D_Q(\cdot)$ outperform $S(\cdot)$ since $c^{(i)}$ is a floating point number.

VI. Conclusion

In this paper, a novel binary audio fingerprint obtained by quantizing the filtered outputs of spectral centroids has been proposed. The filters and their associated quantizers are selected by the PB algorithm which is adapted from a well-known learning algorithm called Adaboost for the purpose of feature selection. By updating the weight distributions of the training matching and nonmatching pairs, the PB algorithm improves both robustness and discriminating ability. In our experiments, the proposed fingerprint showed better performance in terms of robustness and discriminating ability than binary audio fingerprints previously reported.

References

[1] T. Kalker, J. A. Haitsma, and J. Oostveen, "Issues with digital watermarking and perceptual hashing," in *Proc. SPIE Multimedia Systems and Applications IV*, Nov. 2001, vol. 4518, pp. 189–197.

[2] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proc. Int. Conf. Music Information Retrieval*, Paris, France, 2002, pp. 107–115.

[3] J. Herre, E. Allamanche, and O. Hellumth, "Robust matching of audio signals using spectral flatness features," in *Proc. IEEE Workshop Applicat. Signal Processing Audio Acoustics*, 2001, pp. 127–130.

[4] E. Allamanche, J. Herre, O. Helmuth, B. Frba, T. Kasten, and M. Cremer, "Content-based identification of audio material using MPEG-7 low level description," in *Proc. Int. Symp. Music Information Retrieval*, Indiana, 2001, pp. 197–204.

[5] C. Burges, J. Plat, and S. Jana, "Distortion discriminant analysis for audio fingerprinting," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 3, pp. 165–174, May 2003.

[6] J. S. Seo, M. Jin, S. Lee, D. Jang, S. Lee, and C. D. Yoo, "Audio fingerprinting based on normalized spectral subband moments," *IEEE Signal Process. Lett.*, vol. 13, no. 4, pp. 209–212, Apr. 2006.

[7] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer vision for music identification," in *Proc. CVPR*, 2005, vol. 1, pp. 597–604.

[8] K. Covell and S. Baluja, "Known-audio detection using waveprint: Spectrogram fingerprinting by wavelet hashing," in *Proc. ICASSP*, Hawaii, 2007, pp. 237–240.

[9] S. Kim and C. D. Yoo, "Boosted binary audio fingerprint based on spectral subband moments," in *Proc. ICASSP*, Hawaii, 2007, pp. 241–244.

[10] Audible Magic Website [Online]. Available: http://www.audiblemagic.com

[11] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*. New York: Springer-Verlag, 1999.

[12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, 2001, pp. 511–518.

[13] Y. freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comp. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[14] R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[15] C. Savage, "A survey of combinatorial Gray codes," *Soc. Industrial Appl. Math. Rev.*, vol. 39, pp. 605–629, 1997.

Dalwon Jang (S'05) received the B.S. and M.S. degrees from Korea Advanced Institute of Science and Technology in 2002 and 2003, respectively, all in electrical engineering. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology.

His research interests are content identification, music information retrieval, multimedia analysis, and machine learning.

Chang D. Yoo (S'92–M'96) received the B.S. degree in engineering and applied science from California Institute of Technology, in 1986, the M.S. degree in electrical engineering from Cornell University, in 1988, and the Ph.D. degree in electrical engineering from Massachusetts Institute of Technology (MIT), in 1996.

From January 1997 to March 1999, he worked at Korea Telecom as a Senior Researcher. He joined the Department of Electrical Engineering at Korea Advanced Institute of Science and Technology in April 1999. From March 2005 to March 2006, he was with the Research Laboratory of Electronics at MIT. His current research interests are in the application of machine learning and digital signal processing in multimedia.

Prof. Yoo is a member of Tau Beta Pi and Sigma Xi. He currently serves on the Machine Learning for Signal Processing (MLSP) Technical Committee of the IEEE Signal Processing Society.

Sunil Lee (S'02–M'08) received the B.S. degree in electrical engineering from Yonsei University, Seoul, Korea, in 2001, and the M.S. and Ph.D. degrees, both in electrical engineering and computer science, from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2002 and 2008, respectively.

From March to August in 2008, he was a postdoctoral researcher at KAIST. He is now with Multimedia Platform Laboratory, Digital Media and Communications R&D Center, Samsung Electronics, Suwon, Korea, as a Senior Engineer. His research interests include multimedia retrieval, multimedia security, digital watermarking, multirate signal processing, and video coding.

Sungwoong Kim (S'07) received the B.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2004. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering, KAIST.

His research interest is machine learning for signal processing.

Ton Kalker (M'93–SM'99–F'01) is a Distinguished Technologist at Hewlett-Packard Laboratories. He made significant contributions to the field of media security, in particular digital watermarking, robust media identification, and interoperability of digital rights managements (DRM) systems. His history in this field of research started in 1996, submitting and participating in the standardization of video watermarking for DVD copy protection. His solution was accepted as the core technology for the proposed DVD copy protection standard. His subsequent research focused on robust media identification, where he laid the foundation of the Content Identification business unit of Philips Electronics, successful in commercializing watermarking and other identification technologies. In his Philips period, he has coauthored 30 patents and 39 patent applications. His interests are in the field of signal and audio-visual processing, media security, biometrics, information theory, and cryptography. Joining Hewlett-Packard in 2004, he focused his research on the problem of noninteroperability of DRM systems. He became one of the three lead architects of the Coral Consortium, publishing a standard framework for DRM interoperability in the summer of 2007. He also participates actively in the academic community, through students, publications, keynotes, lectures, membership in program committees, and serving as conference chair. Together with Pierre Moulin he is one of the two cofounders of the IEEE TRANSACTIONS ON INFORMATION FORENSICS. He is the former chair of the associated Technical Committee of Information Forensics and Security. He served for six years as visiting faculty at the University of Eindhoven. He is currently a visiting professor at the Harbin Institute of technology.