Quantum Hashing for Multimedia

Minho Jin, *Student Member, IEEE*, and Chang D. Yoo, *Member, IEEE*

***Abstract*—In this paper, a novel multimedia identification system based on quantum hashing is considered. Many traditional systems are based on binary hash which is obtained by encoding intermediate hash extracted from multimedia content. In the system considered, the intermediate hash values extracted from a query are encoded into quantum hash values by incorporating uncertainty in the binary hash values. For this, the intermediate hash difference between the query and its true-underlying content is considered as a random process. Then, the uncertainty is represented by the probability density estimate of the intermediate hash difference. The quantum hashing system is evaluated using both audio and video databases, and with marginal increment in computational cost, the quantum hashing system is shown to be more robust against various distortions than the binary hashing system using the same intermediate hash values.**

***Index Terms*—Content identification, hashing, signal models.**

Nomenclature

$\lvert\mathbf{0}\rangle, \lvert\mathbf{1}\rangle \in \mathbb{H}^2$	Orthonormal basis vectors defined in a complex Hilbert space $\mathbb{H}^2$;
$D \in \mathbb{Z}$	Dimensionality of the intermediate hash vector: D depends on the intermediate hash type;
$K \in \mathbb{Z}$	Number of intermediate hash vectors extracted from the query: K depends on the length of the query multimedia content;
$\mathbf{v}[k] \in \mathbb{R}^D$	D-dimensional real-valued, kth intermediate hash vector extracted from the query;
$v_d[k] \in \mathbb{R}$	dth element of $\mathbf{v}[k]$;
$\hat{v}_d[k] \in \mathbb{R}$	dth element of the kth intermediate hash vector extracted from the undistorted (original) multimedia content associated with the query;
$\mathbb{B}$	Binary space $\{0, 1\}$;
$f_\kappa : \mathbb{R} \rightarrow \mathbb{B}$	Binary encoding function: the intermediate hash is encoded into bit 0 if it is smaller than a threshold κ and bit 1 otherwise;
$\mathbf{b}[k] \in \mathbb{B}^D$	D-dimensional binary hash vector encoded from $\mathbf{v}[k]$;
$b_d[k] \in \mathbb{B}$	dth element of $\mathbf{b}[k]$;
$\hat{b}_d[k] \in \mathbb{R}$	dth element of the kth binary hash vector extracted from the undistorted multimedia content associated with the query;
$e_d[k] \in \mathbb{R}$	$v_d[k] - \hat{v}_d[k]$;
$\mathbf{q}[k] \in \mathbb{H}^{2^D}$	D-dimensional quantum hash vector extracted from $\mathbf{v}[k]$;
$q_d[k] \in \mathbb{H}^2$	dth element of $\mathbf{q}[k]$;
$\psi_d^+, \psi_d^- \in \mathbb{C}$	Complex-valued weights of $\lvert\mathbf{1}\rangle$ and $\lvert\mathbf{0}\rangle$ in $q_d[k]$;
$\mathbf{c}[j] \in \mathbb{B}^D$	jth D-dimensional binary hash vector in the binary hash database extracted from undistorted multimedia contents;
$c_d[j] \in \mathbb{B}$	dth element of $\mathbf{c}[j]$;
$\gamma_l(q_d[k], c_d[j+k])$	Dissimilarity between $q_d[k]$ and $c_d[j+k]$ whose range is the set of real numbers between 0 and 1. The subscript l denotes the parameter involved in calculating $\gamma_l(q_d[k], c_d[j+k])$ (see Section II-B2 for details);
$\tilde{\gamma}_{l,\beta}(q_d[k], c_d[j+k])$	Integer quantized version of $\gamma_l(q_d[k], c_d[j+k])$ whose range is the set of integer numbers between 0 and β. The subscripts l and β denote the parameters involved in calculating $\tilde{\gamma}_{l,\beta}(q_d[k], c_d[j+k])$ (see Section II-B2 for details);
$\tilde{\mathbf{q}}_K \in \mathbb{H}^{2^{KD}}$	(KD)-dimensional quantum hash vector created by concatenating $\mathbf{q}[0]$, $\mathbf{q}[1], \ldots, \mathbf{q}[K-1]$;
$\tilde{\mathbf{c}}_{K,j} \in \mathbb{B}^{KD}$	(KD)-dimensional binary hash vector created by concatenating $\mathbf{c}[j]$, $\mathbf{c}[j+1], \ldots, \mathbf{c}[j+K-1]$;
$\Gamma_l(\tilde{\mathbf{q}}_K, \tilde{\mathbf{c}}_{K,j})$	Dissimilarity function between $\tilde{\mathbf{q}}_K$ and $\tilde{\mathbf{c}}_{K,j}$ whose range is the integer numbers between 0 and $KD\beta$. The subscripts l and β denote the parameters involved in calculating $\Gamma_l(\tilde{\mathbf{q}}_K, \tilde{\mathbf{c}}_{K,j})$ (ee Section II-B2 for details);
$\tau \in \mathbb{R}$	Threshold value of an accept/reject decision (ee Section II-B2 for details);
$\theta_{e,d} = [\mu_{e,d}, \sigma_{e,d}^2]$	Parameter vector whose first and second elements are the mean and the variance of $e_d[k]$, respectively;
$p_{e,d}(x)$	Probability density function of $e_d[k]$;
$p_{e,d}(x; \theta_{e,d})$	Normal approximation of $p_{e,d}(x)$;

Manuscript received August 28, 2008; revised August 23, 2009. First published September 29, 2009; current version published November 18, 2009. This work was supported in part by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD), by the Basic Research Promotion Fund (KRF-2008-314-D00309), and by the Ministry of Culture, Sports and Tourism (MCST) and Korea Culture Content Agency (KOCCA) in the Culture Technology (CT) Research and Development Program 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. M. Kivanc Mihcak.

The authors are with the Division of Electrical Engineering, Department of EECS, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: jinmho@kaist.ac.kr; 1pcdyoo@ee.kaist.ac.kr).

Digital Object Identifier 10.1109/TIFS.2009.2033221

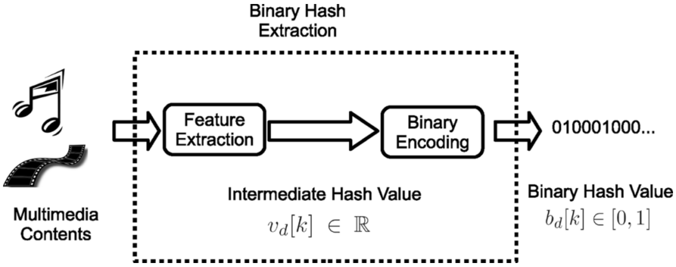


Fig. 1. Extraction of binary hash: refer to Section II-B for detailed explanation and definitions.

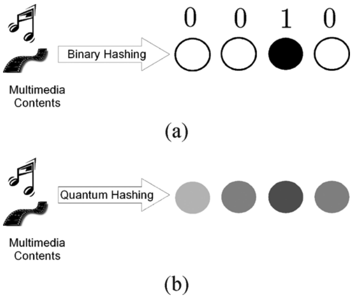
$\mathcal{K}_d$	Kurtosis of $e_d[k]$;
$V_d(t)$	Characteristic function of $v_d[k]$;
$\hat{V}_d(t)$	Characteristic function of $\hat{v}_d[k]$;
$\bar{p}_{v,d}(x)$	Empirical distribution of $v_d[k]$ that is estimated from the query;
$h \in \mathbb{R}$	Bandwidth of the Parzen window (ee Section III-B for details).

I. Introduction

With the advancement of various software and hardware tools to copy, distribute and generate multimedia data, the number of multimedia data on the web has grown in leaps and bounds. As a result, it is impossible to manually identify these data for copyright protection. A multimedia hashing (MH) system aims to automatically identify a query by searching a hash database (DB). Many applications of an MH system include file-sharing service and broadcast monitoring service [1]. For these applications, a binary DB is incorporated to reduce the computational cost and storage [1]–[7].

This paper considers an MH system based on a binary hash DB. Fig. 1 illustrates a general binary hash extraction incorporated in the aforementioned systems [1]–[7]. In these systems, various feature vectors are encoded as binary hash values. In this paper, the feature vectors to be encoded into binary hash are referred to as *intermediate hash*. In Haitsma and Kalker [1], the intermediate hash value is the temporal difference of energy difference between adjacent bands of an audio data, and the intermediate hash is encoded into binary hash using an 1-bit quantizer. In Oostveen *et al.* [2], the intermediate hash value is the temporal difference of block mean luminance difference between adjacent blocks of a video data, and the intermediate hash is encoded into binary hash using an 1-bit quantizer.

The input to an MH system is assumed to be perceptually equivalent but slightly different from the contents used in the DB construction. Henceforth, the content used in the DB construction will be referred to as *original content*. The input is considered as a degraded version of original content, for example, the original content that has undergone some lossy compression such as MP3, WMA, DivX, etc. The intermediate hash value of the original content is most likely to be different from that of its degraded version. In this paper, we call the difference between the original and its degraded version as *distortion*. Hereafter, for simplicity, we call the intermediate hash difference between the original and its degraded version as *intermediate hash difference*.



(a)

(b)

Fig. 2. Example of (a) 4-bit binary hash and (b) 4-qubit quantum hash: the opacity of each ball denotes the uncertainty of each hash element taking the value 0. (a) Binary Hash. (b) Quantum Hash.

The proposed quantum hashing (QH) system aims to improve the robustness of the binary hashing (BH) systems by explicitly considering the effects of the distortion in the binary encoding. Whereas, aforementioned BH systems have concentrated on developing robust intermediate hash against distortions. As illustrated in Fig. 2(a), a BH system extracts a bit sequence as binary hash, and the bits are represented by transparent and opaque balls denoting the bit 0 and 1, respectively. Rather than using definitive hash values, the QH system incorporates uncertainty in the hash values, and this is represented in Fig. 2(b) by the opacity of a ball denoting the uncertainty of being bit 0. For mathematical representation, we use the qubit notation which originates from the quantum information theory [8], [9].

A qubit $|\mathbf{x}\rangle$ in a Hilbert space $\mathbb{H}^2$ is a superposition of two orthonormal qubit bases $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$, and it is mathematically represented as follows [8]:

$$|\mathbf{x}\rangle = \psi_x^- |\mathbf{0}\rangle + \psi_x^+ |\mathbf{1}\rangle \in \mathbb{H}^2 \tag{1}$$

where $|\psi_x^-|^2$ and $|\psi_x^+|^2$ denote the probabilities that $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$ are observed in $|\mathbf{x}\rangle$, respectively. Using the qubit notation, Deutsch [10] established the concept of quantum computer. Based on this concept, Shor [11] proposed an efficient factorization algorithm, and Grover [12] proposed an efficient search algorithm. In neural networks, Kak [13] proposed quantum neural computation, and Li and Zheng [14] and Zhou *et al.* [15] proposed concepts such as quantum neuron and perceptron. Unlike previous studies which are based on quantum computers, Eldar [16] proposed quantum signal processing by imposing some properties of quantum mechanics to conventional signal processing algorithms.

In the system considered, the weights of two qubit bases $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$ are derived from the probability that the binary hash value of the true-underlying content of the query (the undistorted original content associated with the query) is 0 or 1. For this, the intermediate hash difference is modeled as a random process, and its probability density function (PDF) is estimated from the sampled data of the query. Finally, the probability of the binary hash value being 0 or 1 is computed using the PDF of the intermediate hash difference. The QH system is evaluated using both audio and video DB, and it is found that the QH system is more robust against various types of distortions than a BH system without much loss in complexity and computational load.

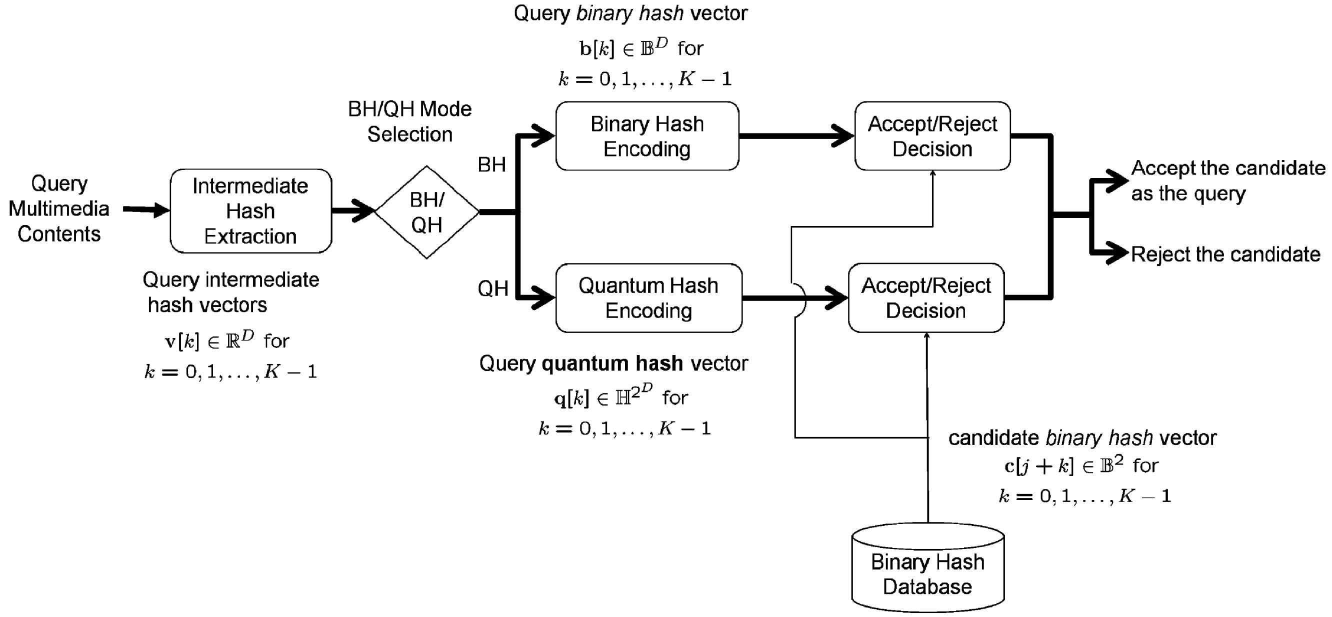

Query *binary hash* vector
$\mathbf{b}[k] \in \mathbb{B}^D$ for
$k = 0, 1, \ldots, K-1$
BH/QH Mode Selection
Query Multimedia Contents
Intermediate Hash Extraction
BH/ QH
BH
QH
Binary Hash Encoding
Quantum Hash Encoding
Accept/Reject Decision
Accept/Reject Decision
Accept the candidate as the query
Reject the candidate
Query intermediate hash vectors
$\mathbf{v}[k] \in \mathbb{R}^D$ for
$k = 0, 1, \ldots, K-1$
Query **quantum hash** vector
$\mathbf{q}[k] \in \mathbb{H}^{2^D}$ for
$k = 0, 1, \ldots, K-1$
candidate *binary hash* vector
$\mathbf{c}[j+k] \in \mathbb{B}^2$ for
$k = 0, 1, \ldots, K-1$
Binary Hash Database

Fig. 3. Multimedia hashing system with BH and QH modes. $\mathbf{b}[k] \in \mathbb{B}^D$ and $\mathbf{c}[j+k] \in \mathbb{B}^D$ are the kth D-dimensional binary hash vector of a query and the $j+k$th D-dimensional binary hash vector in the DB, respectively. $\mathbf{q}[k] \in \mathbb{H}^{2^D}$ is the quantum hash vector of the query. Refer to Sections II-A and II-B for detailed explanation and definitions.

The remainder of this paper is organized as follows: Section II describes the proposed QH system. Section III describes how the weights in a qubit are computed. Section IV presents experimental results, and Section V concludes this paper.

II. Quantum Hashing for Multimedia

A. Binary Multimedia Hashing System

Fig. 3 illustrates the block diagram of an MH system with BH and QH modes. When a query is fed into the system, the system extracts intermediate hash from the query. If the BH mode is selected, then the system encodes the intermediate hash into binary hash like conventional BH systems used in many applications [1], [2], [4]–[7], [17], [18]. Afterwords, DB search is performed to retrieve candidate binary hash vectors in the DB that are likely to matched with the query. Then, the Hamming distance between the query binary hash vector and the candidate binary hash vector is computed. Finally, the system accepts the candidate as the query if the Hamming distance is smaller than a preset threshold or rejects otherwise. If a candidate is accepted, then the system outputs the meta-data associated with the candidate. This paper will focus on the accept/reject decision using the quantum hash, and details concerning DB search are considered outside the scope of this paper. Among various hashing algorithms, this paper considers the normalized spectral sub-band moments (NSSM) [19] as the intermediate hash values for audio hashing and the centroid of gradient orientation (CGO) [20] as the intermediate hash values for video hashing: see Appendix for details regarding NSSM and CGO.

1) Binary Hashing: Let us denote a D-dimensional intermediate hash vector $\mathbf{v}[k] \in \mathbb{R}^D$ extracted from the kth frame of multimedia content as follows:

$$\mathbf{v}[k] = [v_0[k],\ v_1[k],\ \ldots, v_{D-1}[k]]^T \quad k = 0, 1, \ldots. K-1 \tag{2}$$

where K is the number of intermediate hash vectors in the query. Let $\mathbf{b}[k] \in \mathbb{B}^D$ be the binary hash vector encoded from $\mathbf{v}[k]$, where $\mathbb{B} = \{0, 1\}$. Also let $b_d[k] \in \mathbb{B}$ be the dth element of $\mathbf{b}[k]$. In this paper, it is assumed without loss of generality that $v_d[k] \in \mathbb{R}$ is encoded into a bit $b_d[k]$ as follows:

$$b_d[k] = f_\kappa(v_d[k]) \quad d = 0, 1, \ldots, D-1 \tag{3}$$

where $f_\kappa : \mathbb{R} \rightarrow \mathbb{B}$ is defined by

$$f_\kappa(x) = \begin{cases} 1, & \text{if } x \geq \kappa \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where κ is a certain threshold (for example, the median of intermediate hash values).

B. Quantum Hashing

Depending on the severity of the distortion, the Hamming weight[1] of D-bit binary hash difference between a query and its true-underlying content can vary from 0 to D. The proposed QH system evaluates the probability that each bit of the D-bit binary hash difference is 1. When the system in Fig. 3 operates in the QH mode, the intermediate hash from the query is encoded into the quantum hash. As illustrated in Fig. 3, the QH algorithm is designed to use the same intermediate hash and the binary hash DB as the BH algorithm. The QH system is different from the BH system in that the quantum hash is incorporated in encoding of the intermediate hash and making accept/reject decision as illustrated in Fig. 3. This section discusses two issues of 1) extracting quantum hash from a query and 2) making an accept/reject decision using quantum hash values.

[1] The Hamming weight is defined as the sum nonzero elements of a bit sequence [21].

1) Extracting Quantum Hash From a Query: In the QH system, $v_d[k]$ is modeled as follows:

$$v_d[k] = \hat{v}_d[k] + e_d[k] \tag{5}$$

where $\hat{v}_d[k] \in \mathbb{R}$ is the intermediate hash value of the true-underlying content associated with the query content, and $e_d[k] \in \mathbb{R}$ represents the intermediate hash difference. We assume that $e_d[k]$ and $\hat{v}_d[k]$ are independent. Let $\hat{b}_d[k] = f_\kappa(\hat{v}_d[k])$. Then, the query intermediate hash value $v_d[k] = \mathrm{v}$ is encoded into qubit $q_d[k] \in \mathbb{H}^2$ as follows:

$$q_d[k] = \psi_d^-[k] \ \ |\mathbf{0}\rangle + \psi_d^+[k] \ \ |\mathbf{1}\rangle \tag{6}$$

where

$$\begin{aligned} |\psi_d^-[k]|^2 &= P(\hat{b}_d[k] = 0 | v_d[k] = \mathrm{v}) \\ |\psi_d^+[k]|^2 &= P(\hat{b}_d[k] = 1 | v_d[k] = \mathrm{v}) \end{aligned} \tag{7}$$

and where

$$\begin{aligned} P(\hat{b}_d[k] = 0 | v_d[k] = \mathrm{v}) &= P(\hat{v}_d[k] < \kappa | v_d[k] = \mathrm{v}) \\ &= P(v_d[k] - e_d[k] < \kappa | v_d[k] = \mathrm{v}) \\ &= P(e_d[k] > \mathrm{v} - \kappa). \end{aligned} \tag{8}$$

In Section III, we describe how $P(e_d[k] > \mathrm{v} - \kappa)$ is computed.

2) Making an Accept/Reject Decision Using Quantum Hash Values: The QH system uses the same binary hash DB as the BH system. Therefore, it is necessary to define the dissimilarity between the quantum hash value from a query and a binary hash value in the DB. Let $\mathbf{c}[j] \in \mathbb{B}^D$ be the jth binary hash vector in the DB which is given by

$$\mathbf{c}[j] = \left[c_0[j], c_1[j], \ldots, c_{D-1}[j]\right]^T \tag{9}$$

for $j = 0, 1, \ldots, J-1$, where J is the number of binary hash vectors in the DB. Based on the qubit notation, we define the dissimilarity $\gamma_l : \mathbb{H}^2 \times \mathbb{B} \to [0,1]$ between $q_d[k] \in \mathbb{H}^2$ and $c_d[j] \in \mathbb{B}$ as follows:

$$\begin{aligned} \gamma_l(q_d[k], c_d[j]) &= \begin{cases} |\psi_d^+[k]|^l, & \text{if } c_d[j] = 0 \\ |\psi_d^-[k]|^l, & \text{otherwise} \end{cases} \\ &= \begin{cases} \left(1 - P(e_d[k] > \mathrm{v} - \kappa)\right)^{l/2}, & \text{if } c_d[j] = 0 \\ P(e_d[k] > \mathrm{v} - \kappa)^{l/2}, & \text{otherwise} \end{cases} \end{aligned} \tag{10}$$

where l is a preset parameter. Furthermore, in the system considered, (10) is quantized as an integer value as follows:

$$\begin{aligned} \tilde{\gamma}_{l,\beta}(q_d[k], c_d[j]) &= \lfloor \beta \cdot \gamma_l(q_d[k], c_d[j]) \rfloor && (11) \\ &= \begin{cases} \lfloor \beta |\psi_d^+[k]|^l \rfloor, & \text{if } c_d[j] = 0 \\ \lfloor \beta |\psi_d^-[k]|^l \rfloor, & \text{otherwise} \end{cases} && (12) \end{aligned}$$

where β is a preset scaling factor, and $\lfloor x \rfloor$ denotes the largest integer that is equal or less than x.

Let $\mathbf{q}[k] \in \mathbb{H}^{2^D}$ be the quantum hash vector extracted from the kth frame of the query as follows:

$$\mathbf{q}[k] = \left[q_0[k], q_1[k], \ldots, q_{D-1}[k]\right]^T. \tag{13}$$

Let $\tilde{\mathbf{q}}_K$ be the concatenation of all K quantum hash vectors in the query as follows:

$$\tilde{\mathbf{q}}_K = \left[\mathbf{q}[0]^T, \mathbf{q}[1]^T, \ldots, \mathbf{q}[K-1]^T\right]^T. \tag{14}$$

In this paper, the query and the candidate in the DB are assumed to have the same length, thus $\tilde{\mathbf{q}}_K$ must be matched with K-consecutive binary hash vectors in the DB. For this, let $\tilde{\mathbf{c}}_{K,j}$ be the concatenation of K binary hash vectors starting from the jth binary hash vector in the DB as follows:

$$\tilde{\mathbf{c}}_{K,j} = \left[\mathbf{c}[j]^T, \mathbf{c}[j+1]^T, \ldots, \mathbf{c}[j+K-1]^T\right]^T. \tag{15}$$

In order to make an accept/reject decision of $\tilde{\mathbf{q}}_K$ as $\tilde{\mathbf{c}}_{K,j}$ for a given j, the following hypotheses test is performed:

$$\begin{aligned} &H_0 : \tilde{\mathbf{c}}_{K,j} \text{ corresponds to } \tilde{\mathbf{q}}_K. \\ &H_1 : \tilde{\mathbf{c}}_{K,j} \text{ does not correspond to } \tilde{\mathbf{q}}_K. \end{aligned} \tag{16}$$

Let $\tilde{\mathbf{q}}_K[m] \in \mathbb{H}^2$ and $\tilde{\mathbf{c}}_{K,j}[m] \in \mathbb{B}$ be the mth element of $\tilde{\mathbf{q}}_K$ and $\tilde{\mathbf{c}}_{K,j}$, respectively. Using (12), hypothesis testing (16) is performed as follows:

$$\begin{aligned} \Gamma_l(\tilde{\mathbf{q}}_K, \tilde{\mathbf{c}}_{K,j}) &= \beta^{-1} \sum_{m=0}^{KD-1} \tilde{\gamma}_{l,\beta}(\tilde{\mathbf{q}}_K[m], \tilde{\mathbf{c}}_{K,j}[m]) \\ &\underset{H_0}{\overset{H_1}{\gtrless}} \tau, \end{aligned} \tag{17}$$

where τ is a preset threshold. If the dissimilarity $\Gamma_l(\tilde{\mathbf{q}}_K, \tilde{\mathbf{c}}_{K,j})$ is smaller than a preset threshold τ, the system accepts the null hypothesis H_0, which claims that $\tilde{c}_{K,j}$ corresponds to $\tilde{q}_K$. Otherwise, the system accepts the alternative hypothesis H_1. The parameter l can be selected based on the performance, and we set $l = 0.8$ based on our experiments which will be shown in Section IV-C. In addition, the parameter β is selected for $KD\beta$ to be less than the maximum value of 16-bit integer for computational efficiency.

When extracting the quantum hash, we can compute both $\lfloor \beta |\psi_d^-[k]|^l \rfloor$ and $\lfloor \beta |\psi_d^+[k]|^l \rfloor$ in (12) for $k = 0, 1, \ldots, K-1$ and $d = 0, 1, \ldots, D-1$. Then, for any j, the summation in (17) can be efficiently computed with KD integer additions by adding $\lfloor \beta |\psi_d^-[k]|^l \rfloor$ if $c_d[j] = 0$ and $\lfloor \beta |\psi_d^+[k]|^l \rfloor$ otherwise. Therefore, compared to the BH system, the QH system requires only marginal increment in computational cost of computing $\lfloor \beta |\psi_d^-[k]|^l \rfloor$ and $\lfloor \beta |\psi_d^+[k]|^l \rfloor$ for $k = 0, 1, \ldots, K-1$ and $d = 0, 1, \ldots, D-1$. The marginal increment in computational cost does not depend on J, the DB size, which will be experimentally shown in Section IV-D.

3) Comparison Between Quantum Hashing and Binary Hashing: Previous MH systems use either binary hash DB or real-valued hash DB to find the matched pair of the query in the DB. If the system uses the binary hash DB, then it extracts binary hash from the intermediate hash, and then it finds the matched pair using the query binary hash. If the system uses the real-valued hash DB, it extracts real-valued hash from the query, and then it finds the matched pair using the query

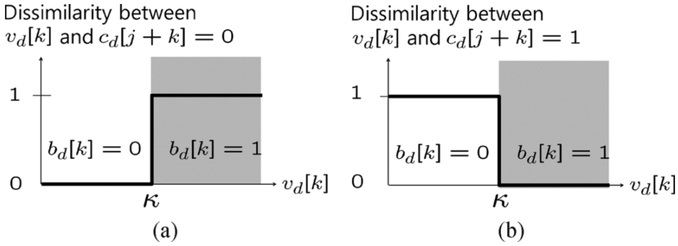


Fig. 4. The dissimilarity between $v_d[k] \in \mathbb{R}$ and $c_d[j+k] \in \mathbb{B}$ in the BH system: in the shaded region, $v_d[k]$ is encoded into $b_d[k]=1$; otherwise, $v_d[k]$ is encoded into $b_d[k]=0$. (a) The dissimilarity between $v_d[k]$ and $c_d[j+k]=0$. (b) The dissimilarity between $v_d[k]$ and $c_d[j+k]=1$.

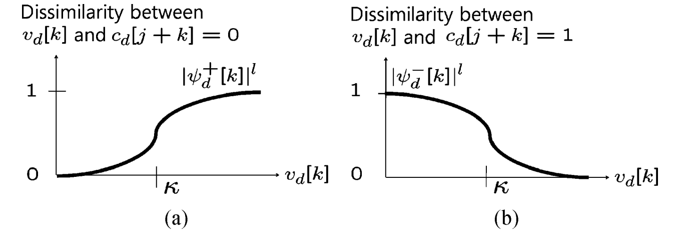


Fig. 5. The dissimilarity between $v_d[k] \in \mathbb{R}$ and $c_d[j+k] \in \mathbb{B}$ in the QH system: $v_d[k]$ is not explicitly encoded into bit 0 or 1. (a) The dissimilarity between $v_d[k]$ and $c_d[j+k]=0$. (b) The dissimilarity between $v_d[k]$ and $c_d[j+k]=1$.

real-valued hash. In real applications, the binary hash DB is more commonly used since the real-valued hash system requires huge storage and computational complexity.

The major difference between the QH and BH is the way the dissimilarity between the query and a candidate in the binary hash DB is computed. In the BH system, the system extracts intermediate hash from the query, and the intermediate hash is encoded into binary hash to find the matched pair in the binary hash DB. Then, the dissimilarity between the query and a candidate in the DB is computed using the Hamming distance between the query binary hash vector and the candidate binary hash vector. Since $v_d[k] \in \mathbb{R}$ is encoded into $b_d[k] = f_\kappa(v_d[k]) \in \mathbb{B}$, the dissimilarity between the query intermediate hash value $v_d[k]$ and the candidate binary hash value $c_d[j+k]$ can be considered as a hard-decision function of $v_d[k]$ as illustrated in Fig. 4.

In the QH system, the dissimilarity between the query and a candidate in the DB is computed using (10) which measures the dissimilarity between the query quantum hash value $q_d[k]$ and the candidate binary hash value $c_d[j+k]$. Thus, the dissimilarity between $v_d[k]$ and $c_d[j+k]$ is a soft-decision function of $v_d[k]$ as illustrated in Fig. 5. Among the many possible soft-decision functions, we proposed a soft-decision function based on the distortion modeling described by (5). Section IV demonstrates the effectiveness of the proposed QH algorithm experimentally. Since the QH algorithm uses the same binary hash DB used by the BH algorithm, it does not require additional storage compared to the BH system. In addition, the accept/reject decision can be made with marginal increment in computational cost.

III. Distortion Statistics for Quantum Hashing

The weights $\psi_d^-[k]$ and $\psi_d^+[k]$ in qubit $q_d[k]$ are derived from the probability $P(e_d[k] > \mathrm{v} - \kappa)$. In the system considered, $e_d[k]$ is assumed to be independent and identically distributed (i.i.d.) over $k = 0, 1, \ldots, K-1$. Hereafter, we denote the PDF of $e_d[k]$ as $p_{e,d}(x)$.

A. Parametric Distortion Statistics

Let $\boldsymbol{\theta}_{e,d} = [\mu_{e,d},\ \sigma_{e,d}^2]$ be a parameter vector whose first and second elements are the mean and the variance of $e_d[k]$, respectively. Using $\boldsymbol{\theta}_{e,d}$, the normal approximation $p_{e,d}(x;\boldsymbol{\theta}_{e,d})$ of $p_{e,d}(x)$ can be written as follows:

$$p_{e,d}(x;\boldsymbol{\theta}_{e,d}) = \frac{1}{\sqrt{2\pi\sigma_{e,d}^2}} \exp^{-(x-\mu_{e,d})^2/2\sigma_{e,d}^2}. \tag{18}$$

Given $v_d[k]$ for $k = 0, 1, \ldots K-1$, $\boldsymbol{\theta}_{e,d}$ can be estimated in the maximum likelihood (ML) sense as follows:

$$\begin{aligned}\boldsymbol{\theta}_{e,d} &= [\mu_{e,d},\ \sigma_{e,d}^2] \\ &= \left[\frac{1}{K}\sum_{k=0}^{K-1} v_d[k] - \hat{\mu}_{v,d},\ \frac{1}{K}\sum_{k=0}^{K-1}(v_d[k]-\mu_{e,d})^2 - \hat{\sigma}_{v,d}^2\right]\end{aligned} \tag{19}$$

where $\hat{\mu}_{v,d}$ and $\hat{\sigma}_{v,d}^2$ are the mean and the variance of the dth element of true-underlying intermediate hash vectors: these ensemble mean and vector parameters can be precalculated from a set of development data of true-underlying contents, and are note estimated from the query. Finally, $P(e_d[k] > \mathrm{v} - \kappa)$ in (8) can be computed as follows:

$$\begin{aligned}P(e_d[k] > \mathrm{v}-\kappa) &= \int_{\mathrm{v}-\kappa}^{\infty} p_{e,d}(x;\boldsymbol{\theta}_{e,d})dx \\ &\approx 1 - \frac{1}{1+\exp^{-\sqrt{\pi}\left(0.9((\mathrm{v}-\kappa)-\mu_{e,d})/\sigma_{e,d}\right)}}.\end{aligned} \tag{20}$$

In (20), the cumulative density function is approximated by a first order sigmoid for computational efficiency [22].

B. Nonparametric Distortion Statistics

In Section III-A, $p_{e,d}(x)$ is approximated with an assumption that $e_d[k]$ is normally distributed. However, it can be shown by using the kurtosis that $e_d[k]$ is not normally distributed for various distortions. Let $\mathbb{E}_d = \{e_d[n] | 0 \le n \le N-1\}$ be a set of the dth element of the intermediate hash difference. The kurtosis of $e_d \in \mathbb{E}_d$ is computed as follows [23]:

$$\mathcal{K}_d = \frac{\sum\limits_{e_d\in\mathbb{E}_d} e_d^4}{\left(\sum\limits_{e_d\in\mathbb{E}_d}(e_d-\mu_d)^2\right)^2} - 3 \tag{21}$$

where μ_d is given by

$$\mu_d = \frac{1}{N}\sum_{e_d\in\mathbb{E}} e_d. \tag{22}$$

Fig. 6 illustrates $\mathcal{K}_1$. Details regarding the distortions labeled in the horizontal axis can be found in Section IV. When $e_d \in \mathbb{E}_d$ follows a normal distribution, $\mathcal{K}_d$ must be near 0. However,

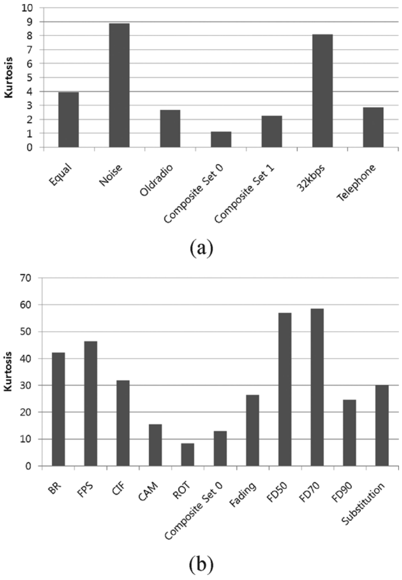


Fig. 6. The kurtosis of the first element of the intermediate hash difference: details regarding the distortions labeled in the horizontal axis can be found in Section IV. (a) Audio. (b) Video.

as shown in Fig. 6, $\mathcal{K}_1$ is usually much larger than 0, which implies that the $e_d \in \mathbb{E}_d$ does not follow a normal distribution. For this reason, a nonparametric distribution is considered to approximate $p_{e,d}(x)$.

In this paper, it is assumed that $e_d[k]$ is unobservable, and the estimate of $p_{e,d}(x)$ must depend only on the observable $v_d[k]$ and not $e_d[k]$. Let $V_d(t)$ and $\hat{V}_d(t)$ be the characteristic functions of $v_d[k]$ and $\hat{v}_d[k]$, respectively. From the assumption that $\hat{v}_d[k]$ and $e_d[k]$ are independent, $p_{e,d}(x)$ can be written as follows [24]:

$$p_{e,d}(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \exp^{-itx}\left(\frac{V_d(t)}{\hat{V}_d(t)}\right)^{*} dt. \tag{23}$$

To compute (23), $\hat{V}_d(t)$ and $V_d(t)$ must be determined. When $\hat{V}_d(t)$ is unknown, it can be computed from a set of development data of true-underlying contents with an assumption that $\hat{v}_d[k]$ is iid over k. In the system considered, the characteristics function $V_d(t)$ of the query intermediate hash value $v_d[k]$, is computed as follows:

$$V_d(t) = \alpha\int_{-\infty}^{\infty} \bar{p}_{v,d}(x;h)\exp^{itx} dx + (1-\alpha)\hat{V}_d(t) \tag{24}$$

where $\bar{p}_{v,d}(x)$ is the empirical PDF computed from $\{v_d[k] | 0 \leq k \leq K-1\}$. Finally, using $\hat{V}_d(t)$ and $V_d(t)$, $p_{e,d}(x)$ in (23) is computed. In our experiments, $\bar{p}_{v,d}(x)$ was estimated in two different ways. First, $\bar{p}_{v,d}(x)$ is directly computed using the histogram. Second, $\bar{p}_{v,d}(x)$ is obtained using the Parzen window with the Gaussian kernel as follows:

$$\bar{p}_{v,d}(x) = \frac{1}{K\sqrt{2\pi h^2}}\sum_{k=0}^{K-1} \exp^{-(x-\mathrm{v}_d[k])^2/2h^2} \tag{25}$$

where h is a bandwidth. To find the optimal h, we use a fast optimal bandwidth selection algorithm proposed by Raykar and Duraiswani [25]. In their algorithm, the plug-in optimal bandwidth selection has been efficiently performed using the Gaussian kernel density derivative estimate: details regarding the optimal bandwidth selection and the plug-in method can be found in the following papers [25], [26]. In our experiments, we used $\alpha = 0.06$ since the number of samples is smaller than the number of bins to compute $\bar{p}_{v,d}(x)$: the empirical probability is computed using 100 histogram bins, and we will use $K = 50$ for audio and $K = 100$ for video hashing experiments. Finally, using (23), $P(e_d[k] > \mathrm{v} - \kappa)$ is computed as follows:

$$P(e_d[k] > \mathrm{v} - \kappa) = \int_{\mathrm{v}-\kappa}^{\infty} p_{e,d}(x)dx. \tag{26}$$

IV. Experiments

In this paper, the NSSM and the CGO are used as intermediate hash for audio hashing experiments and video hashing experiments, respectively. The NSSM and the CGO values are symmetrically distributed around 0, and these are normalized beforehand so that its dynamic range is from −1 to 1 [19], [20]. Thus, we use $f_{\kappa=0}(x)$ in (4) for binarization.

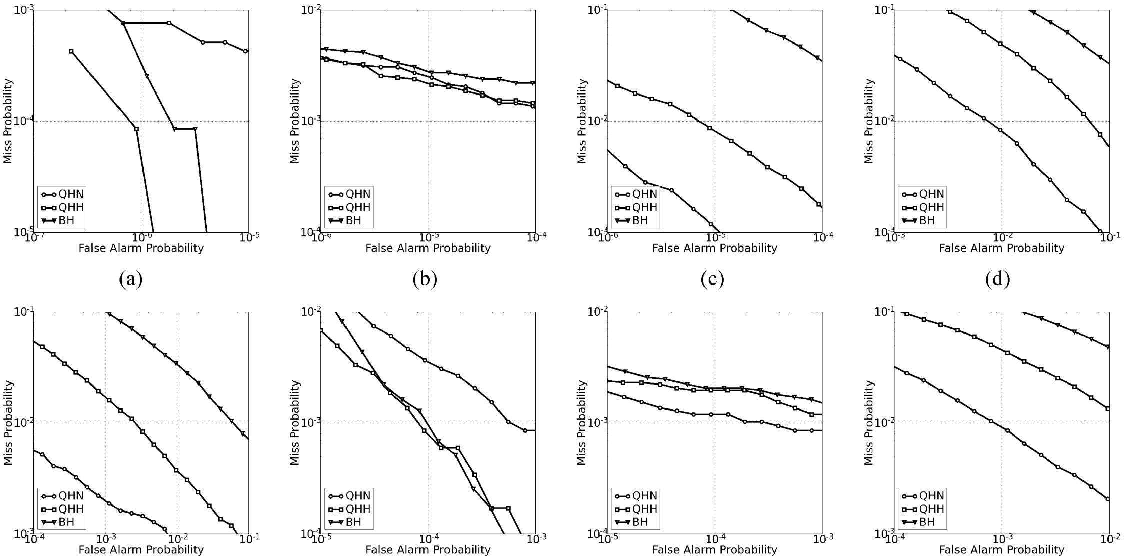
A. Audio Hashing

The QH and BH systems were evaluated using the audio hash DB generated from 1000 songs of various genres such as classic, jazz, pop, rock, etc., which amounts to 62-h playing time. For audio hashing experiments, the hypotheses testing (16) is performed using 11677 matched pairs (the original content and its distorted version) and 3461343 mismatched pairs (different contents). The length of contents in each pair is 9.29 s.

In our audio experiments, we used the 16-dimensional NSSM from 16 critical bands between 300 and 5300 Hz, which were extracted at every 185.8 ms [19]. Thus, the dimensionality of each hash vector is $D = 16$, and the number of hash vectors in the query is $K = 50$. In the BH system, the $(KD = 800)$-dimensional query binary hash vector is matched with an 800-dimensional candidate binary hash vector. Also, in the QH system, the 800-dimensional query quantum hash vector is matched with an 800-dimensional candidate binary hash vector. In addition, we set $\beta = 40$, thus $KD\beta$ is less than the maximum of 16-bit integer. The following distortions were considered in our audio hashing experiments.[2]

1) *Equalization*: adjacent band attenuation set to a random gain between -6 dB and 6 dB + 96 kbps lossy MP3 compression;
2) *Noise*: white noise of signal-to-noise ratio (SNR) 25 dB + 96 kbps lossy MP3 compression;
3) *Echo*: echo insertion with filter-emulating old-time radio + 96 kbps lossy MP3 compression;
4) *Composite Set 0*: *Equalization* + *Noise* + *Echo* + Envelop tremors + 96 kbps lossy MP3 compression;
5) *Composite Set 1*: echo insertion with filter-emulating ambient metal room + pitch decrement by 1% + 92.9 ms shift + 96 kbps lossy MP3 compression;
6) *32 kbps*: 32 kbps lossy compression;
7) *Telephone*: telephone channel filtering.

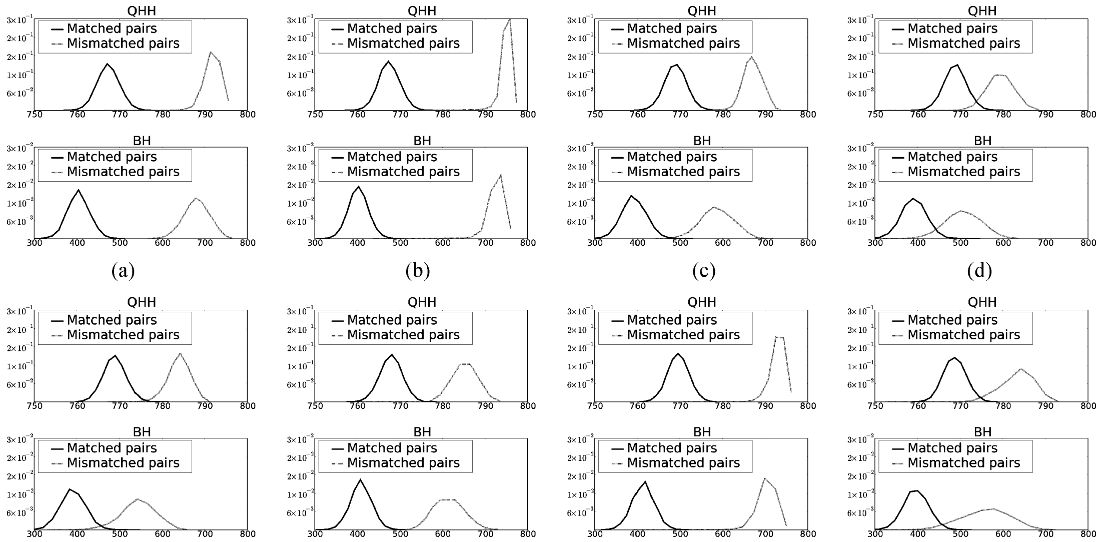
[2] Examples of distorted audio clips are available at http://mmp.kaist.ac.kr/qh



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 7. Quantum audio hashing: DET curves. (a) *Equalization*. (b) *Noise*. (c) *Echo*. (d) *Composite Set 0*. (e) *Composite Set 1*. (f) *32 kbps*. (g) *Telephone*. (h) *Mixed Query*.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 8. Quantum audio hashing: the histograms of the dissimilarity $\Gamma_l(\tilde{\mathbf{q}}_K, \tilde{\mathbf{c}}_{K,j})$ in (17) computed using matched and mismatched pairs. (a) *Equalization*. (b) *Noise*. (c) *Echo*. (d) *Composite Set 0*. (e) *Composite Set 1*. (f) *32 kbps*. (g) *Telephone*. (h) *Mixed Query*.

In our experiments, the false alarm and the detection probabilities of each distortion set are computed using different threshold τ in (17). The performance of an MH system depends on how the query is distorted. In real situation, the queries to the system may be differently distorted, and the system does not know how the query is distorted. Thus, we also evaluated the QH and BH systems with a mixed query, which is a union of the five distortion sets (*Echo*, *Composite Set 0*, *Composite Set 1*, *32 kbps*, and *Telephone*). When evaluating the mixed query, the system is not provided with how each query is distorted. Thus, it can be considered as a blind retrieval test.

Hereafter, we abbreviate the QH systems whose distortion distributions are estimated using normal, histogram, and Parzen window as QHN, QHH, and QHP, respectively. Table I summarizes the equal-error rate (EER) of our audio hashing experiments, where the EER is defined as the probability that the false alarm probability is equal to the miss probability (one minus detection probability). Fig. 7 illustrates the detection error trade-off (DET) curves [27] of the QH and the BH systems. Fig. 8 illustrates the histograms of the dissimilarity $\Gamma_l(\tilde{\mathbf{q}}_K, \tilde{\mathbf{c}}_{K,j})$ in (17) computed using matched and mismatched pairs of the audio data. The DET curves and EER values in Fig. 7 and Table I are empirically obtained from the abovementioned matched and mismatched pairs. When computing the EER, we first find the threshold where the absolute difference between the false alarm and the miss probabilities is a minimum. Then, the EER values are computed as the maximum of the false alarm probability and the miss probability. In these experiments, we set $l = 0.8$.

The NSSM is robust against *Equalization*, *Noise* and *Telephone*, thus the performance improvement of the QH system over the BH systems was not significant, as shown in

TABLE I
EERs (%) From Audio Hashing Experiments

	Proposed (Normal)	Proposed (Histogram)	Proposed (Parzen)	Binary Hashing
Equalization	2.57E-02	1.13E-04	3.37E-05	8.56E-03
Noise	6.85E-02	5.45E-02	1.71E-02	1.20E-01
Echo	3.43E-02	5.99E-02	1.88E-01	2.83E-01
Composite Set 0	9.00E-01	2.53E+00	4.51E+00	5.27E+00
Composite Set 1	1.63E-01	5.69E-01	1.59E+00	2.15E+00
32kbps	8.56E-02	2.78E-02	9.32E-03	2.57E-02
Telephone	8.56E-02	1.20E-01	7.71E-02	1.46E-01
Mixed Query	3.70E-01	1.19E+00	3.49E+00	2.73E+00

Fig. 7(a), (b), and (g). When the query is seriously distorted by *Echo*, *Composite Set 0*, and *Composite Set 1*, the QHN and the QHH outperformed the BH system as shown in Fig. 7(c)-(e), and the EER values in Table I. In *Equalization* and *Noise*, the QHH outperformed the BH as shown in Fig. 7(a) and (b). In *32 kbps*, the QHH performed as well as the BH, and the QHN performed the worst. The reason for this can be found in Fig. 6(a), where the *Noise*, *32 kbps* and *Equalization* have the first, second and third largest kurtosis, respectively. Thus, their distribution is far from being normal distribution (whose kurtosis is 0), and modeling the distribution with a normal distribution is inappropriate for these distortions. Therefore, the QHN did not perform well in these cases. However, the distortion in the *Equalization* and *Noise* is not severe, thus both the BH and QHH systems performed well. A supporting evidence for this can be obtained from the signal-to-distortion ratio (SDR), which was computed using the original and its distorted version. The SDR of the *Noise* was 17.33 dB while that of the *32 kbps* was 7.17 dB. The largest SDR of 22.14 dB was observed in the *Equalization*, and the smallest SDR of 3.72 dB was observed in the *Composite Set 0*.

In Fig. 7 and Table I, the QHN outperformed the QHH in most cases except for *Equalization*, *Noise* and *32 kbps*. Our conjecture for this is that the QHH requires more intermediate hash vectors than $K = 50$ to appropriately estimate $p_{e,d}(x)$. The normal approximation of $p_{e,d}(x)$ is effective for small kurtosis but ineffective for large kurtosis. Therefore, even though $K = 50$ may be insufficient to accurately estimate the histogram, the QHH outperformed the QHN when the kurtosis is large. The performances of the QHP, which are summarized in Table I, are inferior to the performances of the BH, QHH and QHN. Our conjecture is that the number of intermediate hash vectors in the query is too small to accurately estimate the optimal bandwidth.

Very often, in an MH system, the distortion is unknown. Thus, we evaluated the system performance based on the mixed set, a union of different distortions. In our audio experiments, the QHN and the QHH performed the best, followed by the BH and the QHP. As shown in Fig. 8, to separate the matched and the mismatched pairs, the BH system is required to change a threshold τ in the hypotheses testing (16) according to distortions. However, the QHH can operate with a fixed threshold, say 620, for various distortions. This results in the performance improvement of the QHH over the BH system in the mixed query. This indicates that the QH system is robust against various kinds of distortions.

B. Video Hashing

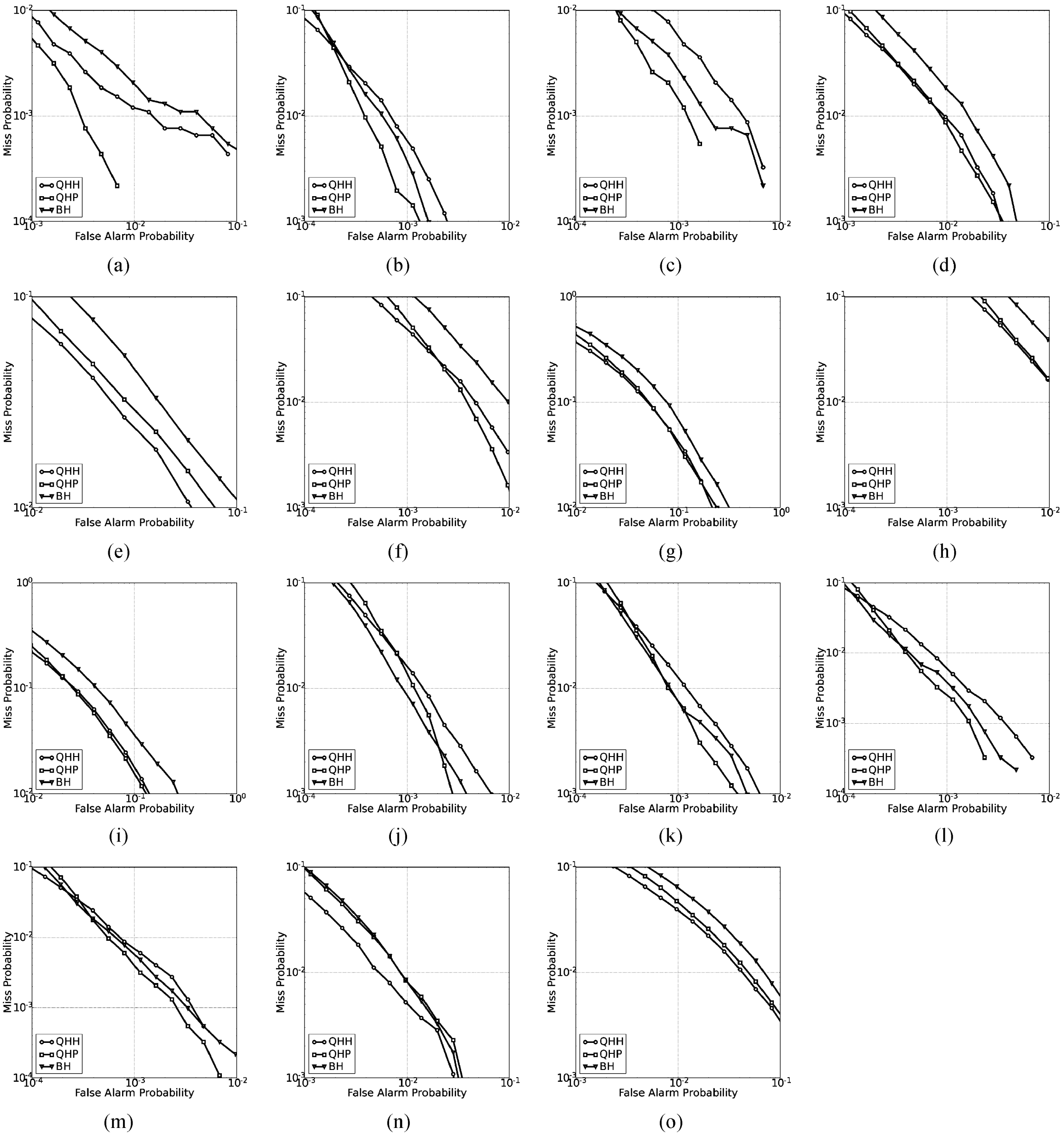
The QH and BH systems were evaluated using the video hash DB generated from 300 movie titles of various genres such as drama, humor, action, suspense, etc, which amounts to 388-hours playing time. For video hashing experiments, the hypotheses testing (16) is performed using 9176 matched pairs and 79977419 mismatched pairs. The length of contents in each pair is 10 s.

When extracting the CGO, the frame size of the video data were normalized to 320×200, and the frame rate was normalized to 10 frame per sec. From each frame, an 8-dimensional CGO vector was extracted by dividing the frame into 4×2 blocks. Thus, the dimensionality of each hash vector is $D = 8$, and the number of hash vectors in the query is $K = 100$. In the BH system, the $(KD = 800)$-dimensional query binary hash vector is matched with an 800-dimensional candidate binary hash vector. Also, in the QH system, the 800-dimensional query quantum hash vector is matched with an 800-dimensional candidate binary hash vector. In addition, we set $\beta = 40$, thus $KD\beta$ is less than the maximum of 16-bit integer. The following distortions were considered in our video hashing experiments:[3]

1) *BR*: brightness increment by $+25\%$;
2) *FPS*: frame rate change to 15 frame per sec;
3) *CIF*: common interface format (CIF) size (352×288);
4) *CAM*: DA/AD conversion using a projector and a camcoder;
5) *ROT*: rotation by 3 degree;
6) *Composite Set 0*: histogram equalization + *CIF* + 256 kbps lossy compression by DivX codec;
7) *Composite Set 1*: *CAM* + *ROT* + *FPS*+ *CIF* + Histogram equalization + 256 kbps lossy compression by DivX codec;
8) *Composite Set 2*: *CAM* + *FPS*+ *CIF* + Histogram equalization + 256 kbps lossy compression by DivX codec;
9) *Composite Set 3*: *ROT* + *FPS*+ *CIF* + Histogram equalization + 256 kbps lossy compression by DivX codec;
10) *Substitution*: 1 s of the query video clip is substituted with 1 s-long video clips from another movie + 256 kbps lossy compression by DivX codec;
11) *Fading*: faded in from black in the first 5 s part + Faded out to black in the last 5 s part + 256 kbps lossy compression by DivX codec;
12) *FD50, 70, and 90*: randomly selected 50, 70, and 90% frames are dropped; dropped frames are interpolated from adjacent frames + 256 kbps lossy compression by DivX codec.

The mixed query set is a union of the nine distortion sets (*BR*, *FPS*, *CIF*, *CAM*, *ROT*, *Composite Set 1*, *Fading*, *Substitution*, *FD90*). Table II summarizes the EERs for various distortions. Fig. 9 illustrates the DET curves of the QH and BH

[3] Examples of distorted video clips are available at http://mmp.kaist.ac.kr/qh



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o)

Fig. 9. Quantum video hashing: DET curves. (a) *BR*. (b) *FPS*. (c) *CIF*. (d) *CAM*. (e) *ROT*. (f) *Composite Set 0*. (g) *Composite Set 1*. (h) *Composite Set 2*. (i) *Composite Set 3*. (j) *Substitution*. (k) *Fading*. (l) *FD50*. (m) *FD70*. (n) *FD90*. (o) Mixed Query.

systems. Fig. 10 illustrates the histograms of the dissimilarity $\Gamma_l(\tilde{\mathbf{q}}_K, \tilde{\mathbf{c}}_{K,j})$ in (17) computed using matched and mismatched pairs of the video data. In these experiments, we set $l = 0.8$.

The CGO is based on the gradient orientation, thus it is robust against *BR*, *FPS*, and *CIF* as shown in Fig. 9(a)-(c). The CGO is known not to be robust against rotation since the gradient orientation is sensitive to rotation. Therefore, as shown in Fig. 9(e) and Table II, the BH system did not perform well for *ROT*.

For *BR*, *FPS*, *CIF*, *Substitution*, *Fading*, *FD50*, *FD70*, and *FD90*, as shown in Fig. 9(a)-(c), (j)-(n), both the BH and the QH performed well, and the performance improvement of the QH system over the BH system is not significant. When the query is seriously distorted by *CAM*, *ROT*, *Composite Set 0, 1, 2*, and *3*, the QH system outperformed the BH system as shown in Fig. 9(a), (d)-(i). Using the QHH, the EER was reduced from 1.32E+00% to 9.81E-01% for *CAM*, from 3.66E+00% to 2.78E% for *ROT*, from 9.92E-01 to 6.43E-01% for *Composite Set 0*, from 8.71% to 6.95% for *Composite Set 1*, from 1.84% to 1.20% for *Composite Set 2*, and from 6.43% to 4.91% for *Composite Set 3*, as shown in Table II. These results indicate that the proposed QH system is more robust against serious distortions than the BH system.

Unlike the audio hashing experiments, the QHN performed the worst for all cases as shown in Table II. The reason for this can be found in Fig. 6, where the kurtosis of the $e_d[k]$ is much larger than 0 (also much larger than the kurtosis of audio distortions), thus the normal distribution leads to an inappropriate model of $p_{e,d}(x)$.

C. EERs versus l

In Fig. 11(a), x^l is plotted for $0 \leq x \leq 1$ and $l = 0.5$, 0.8 and 1.5, and Fig. 11(b) illustrates the difference $(x^l - x)$. As shown in Fig. 11(b), using different l increases x^l over x in different ways, which leads to different EERs versus l in our

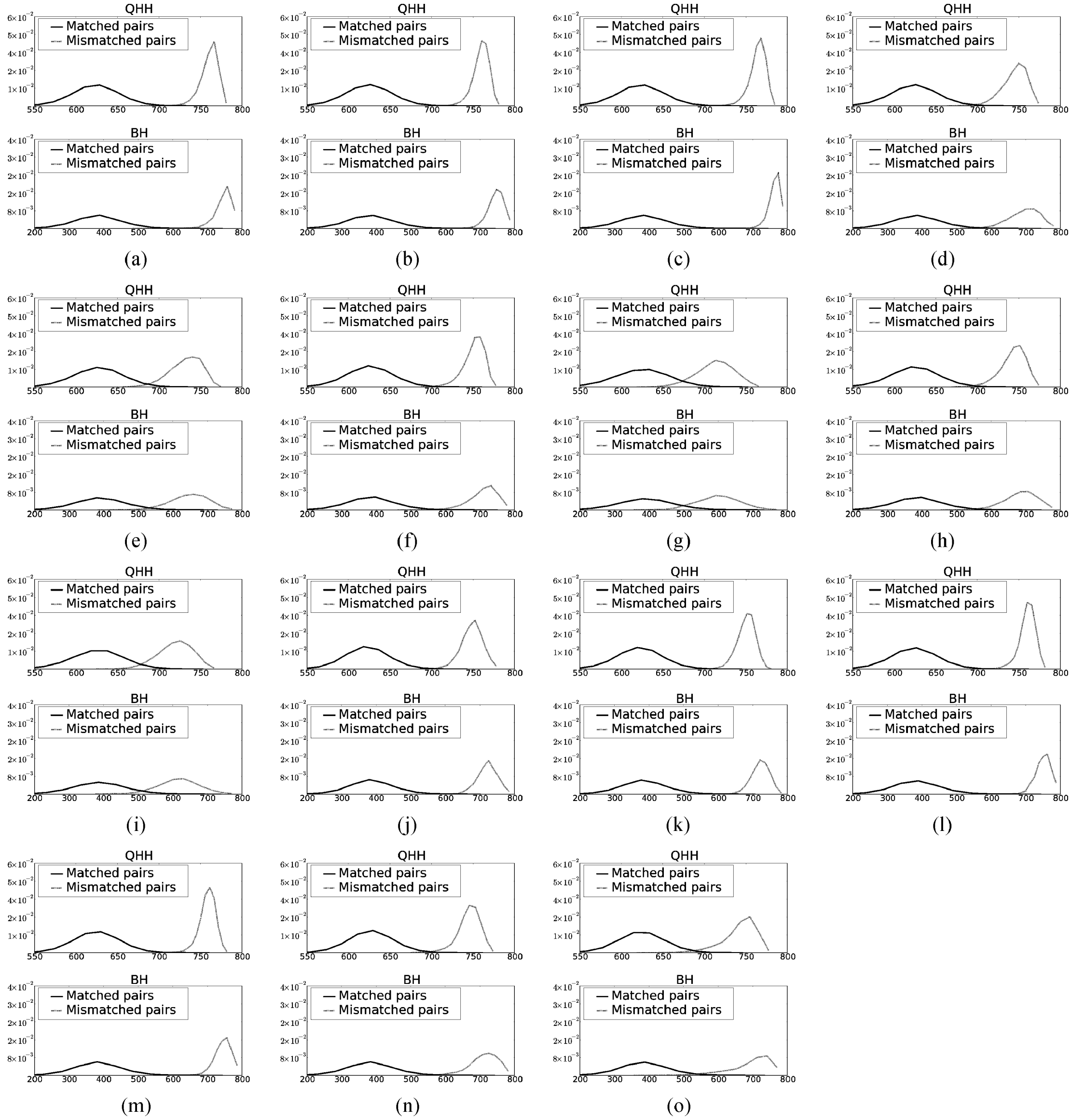


Fig. 10. Quantum video hashing: the histograms of the dissimilarity $\Gamma_l(\tilde{\mathbf{q}}_K, \tilde{\mathbf{c}}_{K,j})$ in (17) computed using matched and mismatched pairs. (a) *BR*. (b) *FPS*. (c) *CIF*. (d) *CAM*. (e) *ROT*. (f) *Composite Set 0*. (g) *Composite Set 1*. (h) *Composite Set 2*. (i) *Composite Set 3*. (j) *Substitution*. (k) *Fading*. (l) *FD50*. (m) *FD70*. (n) *FD90*. (o) *Mixed Query*.

experiments. Fig. 12 illustrates the EERs versus l in experiments using mixed query. In both audio and video experiments, the minimum EER was achieved when $l = 0.8$. In our experiments, $l = 0.8$ outperformed other l values, and we have found that the performance is relatively insensitive to l. Depending on the type of intermediate hash, the free-parameter l should be tuned to different intermediate hash.

D. Computational Time

In the MH system, the DB search system retrieves candidate binary hash vectors in the DB, and an accept/reject decision is made for each binary hash vector. Fig. 13 illustrates the average computational times of the BH and QHH systems versus the number of accept/reject decision makings which is equivalent to the number of 800-bit candidate binary hash vectors: for each query, the number of candidate binary hash vectors is varied from 0 to 30 000. The average computational times in Fig. 13 were obtained from 2000 queries using Intel Core2Quad 2.50 GHz CPU with 2 GB RAM. As shown in Fig. 13, the increment in computational time of the QHH over the BH does not increase as the number of decision makings. The marginal increment in computational time is due to 1) estimating the probability in (26) and 2) computing $|\psi_d^+[k]|^l$ and $|\psi_d^-[k]|^l$ for $k = 0, 1, \ldots, K-1$ and $d = 0, 1, \ldots, D-1$ as mentioned in Section II-B2.

E. Performance With Nonattack Scenario

The proposed QH algorithm was developed assuming certain distortion in the query. When the query is not distorted, the Hamming distance between the BH vectors from the query and its true-underlying content in the DB equals 0. Thus, using the BH system, the accept/reject decision can be made by comparing the Hamming distance to a zero threshold ($\tau = 0$). In the audio hashing experiment with no distortion, EER=0% was obtained for the BH, QHH, and QHN: both the BH and QH performed equally well. However, in the video hashing experiment

TABLE II
EERs (%) From Video Hashing Experiments

	Proposed (Normal)	Proposed (Histogram)	Proposed (Parzen)	Binary Hashing
BR	9.22E+00	2.99E-01	2.07E-01	4.36E-01
FPS	9.54E+00	1.75E-01	1.20E-01	1.53E-01
CIF	9.12E+00	2.22E-01	1.20E-01	1.53E-01
CAM	1.14E+01	9.81E-01	9.37E-01	1.32E+00
ROT	1.12E+01	2.78E+00	3.02E+00	3.66E+00
Composite Set 0	1.09E+01	6.43E-01	5.45E-01	9.92E-01
Composite Set 1	1.38E+01	6.95E+00	6.93E+00	8.71E+00
Composite Set 2	1.15E+01	1.20E+00	1.22E+00	1.84E+00
Composite Set 3	1.32E+01	4.91E+00	4.70E+00	6.43E+00
Fading	1.09E+01	3.18E-01	2.20E-01	2.73E-01
FD50	9.83E+00	2.18E-01	1.45E-01	1.74E-01
FD70	9.70E+00	2.51E-01	1.85E-01	2.18E-01
FD90	1.16E+01	7.19E-01	9.15E-01	9.15E-01
Substitution	1.07E+01	3.17E-01	2.18E-01	2.83E-01
Mixed Query	1.10E+01	2.10E+00	2.27E+00	2.77E+00

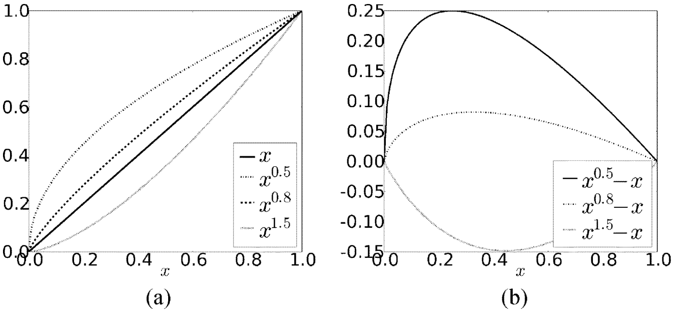


Fig. 11. x^l and $x^l - x$ for $0 \leq x \leq 1$. (a) Illustration of x versus x^l when $0 \leq x \leq 1$. (b) Illustration of $x^l - x$ when $0 \leq x \leq 1$.

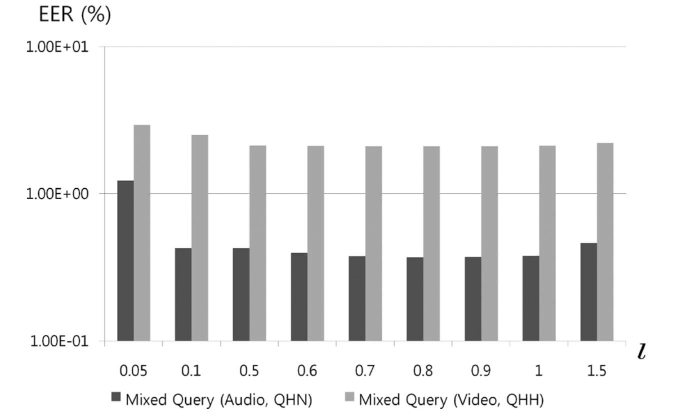


Fig. 12. EERs of the mixed query experiments versus l: QHN and QHH are used for audio and video experiments, respectively.

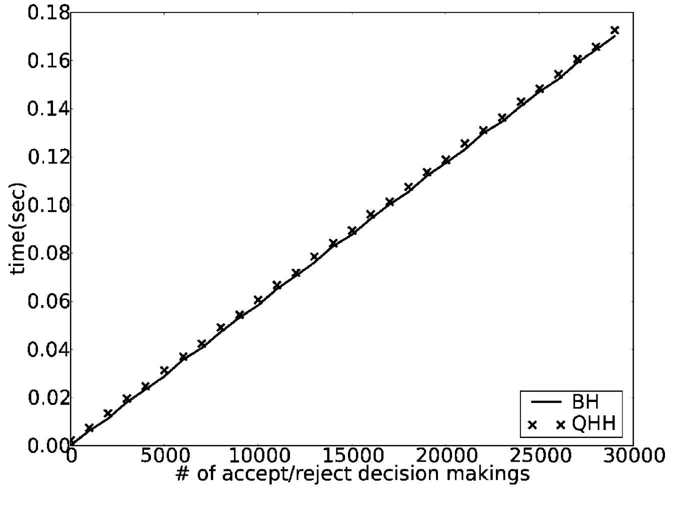


Fig. 13. The average computational times of the BH and QHH systems versus the number of make accept/rejection decision makings which is equivalent to the number of candidate binary hash vectors: the 800-bit candidate binary hash vectors are used to measure the computational time.

with no distortion, the EER of the BH system was 3.48E-06% while that of the QHH system was 1.31E-01%. In the nonattack scenario, the true PDF of the distortion $p_{e,d}(x)$ is a delta function since $e_d[k] = 0$ for all d and k. When $p_{e,d}(x)$ is a delta function, the dissimilarity $\gamma_l(q_d[k], c_d[j])$ becomes the Hamming distance as follows:

$$\begin{aligned}\gamma_l(q_d[k], c_d[j]) &= \begin{cases} \left(1 - P(e_d[k] > \mathrm{v} - \kappa)\right)^{l/2}, & \text{if } c_d[j] = 0 \\ P(e_d[k] > \mathrm{v} - \kappa)^{l/2}, & \text{otherwise} \end{cases} \\ &= \begin{cases} \left.\begin{matrix} 0 & \text{if } \mathrm{v} < \kappa \\ 1 & \text{if otherwise} \end{matrix}\right\}, & \text{if } c_d[j] = 0 \\ \left.\begin{matrix} 1 & \text{if } \mathrm{v} < \kappa \\ 0 & \text{if otherwise} \end{matrix}\right\}, & \text{if } c_d[j] = 1 \end{cases} \\ &= \begin{cases} \left.\begin{matrix} 0 & \text{if } b_d[k] = 0 \\ 1 & \text{if } b_d[k] = 1 \end{matrix}\right\}, & \text{if } c_d[j] = 0 \\ \left.\begin{matrix} 1 & \text{if } b_d[k] = 0 \\ 0 & \text{if } b_d[k] = 1 \end{matrix}\right\}, & \text{if } c_d[j] = 1 \end{cases} \end{aligned} \tag{27}$$

since $b_d[k] = f_\kappa(v_d[k] = \mathrm{v})$ is 1 if $\mathrm{v} \geq \kappa$ and 0 otherwise. The dissimilarity in (27) is identical to the Hamming distance, which indicates that the QH system is identical to the BH system when $p_{e,d}(x)$ is a delta-function. As mentioned above, the true PDF $p_{e,d}(x)$ is a delta-function under nonattack scenario, thus the BH system outperformed the QHH system. A possible remedy for the QHH system under the nonattack scenario is to first compute the Hamming distance between the query and a candidate from the DB. Then, if the Hamming distance is 0, the system accepts the candidate as the query: otherwise, the system makes the accept/reject decision with the dissimilarity computed using the QH system. With such a remedy, we could reduce the EER of the QH system to 3.48E-06%, which is identical to the EER of the BH system.

V. Conclusion

In this paper, we considered an MH system based on quantum hash. The PDF of the intermediate hash difference is estimated from query intermediate hash, and it is used to represent the uncertainty of the binary hash value is either 0 or 1. Our experimental results have shown that the QH system improves the robustness against serious distortions.

The intermediate hash values in our experiments are distributed around the origin, thus small distortions can cause the binary hash value of the query to be different from that of the original. In the BH system, the dissimilarity between two multimedia contents are computed in a manner that all binary hash values are equally contributing to the dissimilarity based on the Hamming distance. However, in the QH system considered, the dissimilarity between two multimedia contents are computed by considering the probability that the binary hash values of its true-underlying content is either 0 or 1. By considering these probabilities, the QH system outperformed the BH system as shown in Figs. 7 and 9.

The QH system does not significantly outperform the BH system when the distortion is not severe. As the distortion becomes severe, the performance gain of the QH system over the BH system becomes significant. In our experiments, the CGO is robust against various distortions, but since it is based on the gradient *orientation*, its performance degrades when the query is distorted by rotation. Also in the composite distortion, the CGO does not perform well. In such cases, the proposed QH system, which is a coding scheme with statistical analysis on the distortion, improved the performance of the system. If a feature vector is robust against all possible distortions, then we would not need a system such as ours. Unfortunately, a feature vector that is robust against all possible distortions has not been found yet. In the meanwhile, the robustness of a BH system can be improved using the proposed QH algorithm, which is a soft decoding based on distortion modeling. As shown in our experiments, the robustness of the MH system against such distortions can be improved by the proposed QH system with properly estimated the distortion PDF. One contribution of this paper is to find the soft decoding based on the distortion modeling that can improve the robustness of the BH system. Our future work will focus on better modeling on the distortions and fast DB search using graph decoding.

Appendix

NSSM: Let $P_m[k]$ be the short-time power spectrum of an audio signal at frequency bin m of the kth frame for $k = 0, 1, \ldots, K-1$ and $m = 0, 1, \ldots M-1$, where K and M are the number of frames and the number of frequency bins, respectively. In addition, let C_d be the index of the first frequency bin in the dth critical band. The first-order moment $\xi_d[k]$ of $P_m[k]$ in the dth critical band is given by

$$\xi_d[k] = \frac{\sum_{m=C_d}^{C_{d+1}-1} m P_m[k]}{\sum_{m=C_d}^{C_{d+1}-1} P_m[k]}. \tag{28}$$

Then, $\xi_d[k]$ is normalized as follows:

$$\eta_d[k] = \frac{\xi_d[k] - \hat{\mu}_d}{\hat{\sigma}_d} \tag{29}$$

where

$$\hat{\mu}_d = \frac{1}{K} \sum_{k=0}^{K-1} \xi_d[k], \tag{30}$$

$$\hat{\sigma}_d = \sqrt{\frac{1}{K} \sum_{k=0}^{K-1} \xi_d^2[k] - \hat{\mu}_d^2}. \tag{31}$$

CGO: Lee and Yoo [20] proposed the block-wise CGO for video identification. The block-wise CGO is extracted as follows:

1) Resample the video into 10 frame per sec (fps).
2) Divide the each frame into D blocks
3) Extract the gradient orientations from all pixels, and let $r[x, y, k]$ and $\theta[x, y, k]$ be the gradient magnitude and orientation of pixel $[x, y]$ in the kth frame, respectively
4) The CGO of the dth block $\mathbb{C}_d$ in the kth frame is computed as follows:

$$\zeta_d[k] = \frac{1}{\pi} \frac{\sum_{[x,y] \in \mathbb{C}_d} r[x, y, k] \theta[x, y, k]}{\sum_{[x,y] \in \mathbb{C}_d} r[x, y, k]}. \tag{32}$$

References

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

[2] J. Oostveen, T. Kalker, and J. A. Haitsma, "Visual hashing of digital video: Applications and techniques," in *Proc. SPIE*, Nov. 2001, vol. 4518, Multimedia Syst. Appl. IV.

[3] S. Kim and C. D. Yoo, "Boosted binary audio fingerprint based on spectral subband moments," in *Proc. ICASSP*, 2007, vol. 1, pp. 241–244.

[4] M. L. Miller, M. A. Rodriguez, and I. J. Cox, "Audio fingerprinting: Nearest neighbor search in high dimensional binary spaces," *J. VLSI Signal Process.*, vol. 41, no. 3, pp. 285–291, 2005.

[5] F. Kurth, M. Clausen, and A. Ribbrock, "Identification of highly distorted audio material for querying large scale data bases," in *Proc. 112th AES Convention*, Munich, Germany, 2002.

[6] B. Coskun, B. Sankur, and N. Memon, "Spatio-temporal transform based video hashing," *IEEE Trans. Multimedia*, vol. 8, no. 6, pp. 1190–1208, Dec. 2006.

[7] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 215–230, Jun. 2006.

[8] C. H. Bennett and P. W. Shor, "Quantum information theory," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2724–2742, Oct. 1998.

[9] A. Steane, "Quantum computing," *Reports on Progress in Phys.*, vol. 61, no. 2, pp. 117–173, 1998.

[10] D. Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer," in *Proc. Royal Soc. London Ser. A*, 1985, vol. A400, pp. 97–117.

[11] P. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Ann. Symp. Found. Comput. Sci.*, 1994, vol. 124.

[12] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Ann. ACM Symp. Theory of Computing (STOC)*, 1996, pp. 212–219.

[13] S. Kak, "On Quantum Neural Computing," *Inf. Sci.*, vol. 83, no. 3, pp. 143–160, 1995.

[14] F. Li and B. Zheng, "A study of quantum neural networks," in *Proc. NNSP*, 2003, vol. 1, pp. 14–17.

[15] R. Zhou, L. Qin, and N. Jiang, "A quantum perceptron network," in *Proc. ICANN*, 2006, vol. 1, pp. 651–657.

[16] Y. Eldar, "Quantum signal processing," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, 2001.

[17] C. Kim and B. Vasudev, "Spatiotemporal sequence matching for efficient video copy detection," *IEEE Trans. Circuits Syst.*, vol. 15, no. 1, pp. 127–132, Jan. 2005.

[18] P. Cano, E. Batlle, H. Mayer, and H. Neushmied, “Robust sound modeling for song detection in broadcast audio,” in *Proc. Audio Eng. Soc. 112th Int. Conv.*, 2002, pp. 1–7.

[19] 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.

[20] S. Lee and C. D. Yoo, “Robust video fingerprinting for content-based video identification,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 983–988, Jul. 2008.

[21] S. Lin, J. Daniel, and J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[22] G. R. Waissi and D. F. Rossin, “A sigmoid approximation of the standard normal integral,” *Appl. Math. Computat.*, vol. 77, no. 1, pp. 91–95, 1996.

[23] D. N. Joanes and C. A. Gill, “Comparing measures of sample skewness and kurtosis,” *Royal Statist. Soc.*, vol. 47, no. 1, pp. 183–189, 2002.

[24] S. M. Ross, *Stochastic Processes*. New York: Wiley, 1992.

[25] V. C. Raykar and R. Duraiswami, “Fast optimal bandwidth selection for kernel density estimation,” in *Proc. 6th SIAM Int. Conf. Data Mining*, 2006, pp. 524–528.

[26] C. R. Loader, “Bandwidth selection: Classical or plug-in?,” *Ann. Statist.*, vol. 27, pp. 415–438, 2007.

[27] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, “The DET curve in assessment of detection task performance,” in *Proc. Eurospeech*, 1997, pp. 1895–1898.

Minho Jin (S’06) received the B.S., M.S., and Ph.D. degrees from the Korea Advanced Institute of Science and Technology, Daejeon, in 2002, 2004, and 2009, respectively, all in electrical engineering.

His research interests are multimedia identification, speaker verification, speech recognition, multimedia retrieval, and machine learning.

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

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

Dr. 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.