Comprehensive Feature-Based Robust Video Fingerprinting Using Tensor Model

Xiushan Nie, *Member, IEEE*, Yilong Yin, Jiande Sun, *Member, IEEE*, Ju Liu, *Senior Member, IEEE*, and Chaoran Cui

***Abstract*—Content-based near-duplicate video detection (NDVD) is essential for effective search and retrieval, and robust video fingerprinting is a good solution for NDVD. Most existing video fingerprinting methods use a single feature or concatenate different features to generate video fingerprints, and show good performance under single-mode modifications such as noise addition and blurring. However, when they suffer combined modifications, the performance is degraded to a certain extent because such features cannot characterize the video content completely. By contrast, the assistance and consensus among different features can improve the performance of video fingerprinting. Therefore, in the present study, we mine the assistance and consensus among different features based on a tensor model, and we present a new comprehensive feature to fully use them in the proposed video fingerprinting framework. We also analyze what the comprehensive feature really is for representing the original video. In this framework, the video is initially set as a high-order tensor that consists of different features, and the video tensor is decomposed via the Tucker model with a solution that determines the number of components. Subsequently, the comprehensive feature is generated by the low-order tensor obtained from tensor decomposition. Finally, the video fingerprint is computed using this feature. A matching strategy used for narrowing the search is also proposed based on the core tensor. The robust video fingerprinting framework is resistant not only to single-mode modifications but also to their combination.**

***Index Terms*—Comprehensive feature, robust video fingerprinting, near-duplicate video detection (NDVD), multiple features.**

Manuscript received February 17, 2016; revised July 10, 2016 and August 24, 2016; accepted November 11, 2016. Date of publication November 9, 2016; date of current version March 15, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61671274 and Grant 61573219, in part by the NSFC Joint Fund with Guangdong under Key Project U1201258, in part by Shandong Natural Science Funds for Distinguished Young Scholar under Grant JQ201316, in part by the Natural Science Foundation of Shandong Province under Grant ZR2014FM012, and in part by the Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Winston Hsu. *(Corresponding author: Yilong Yin.)*

X. S. Nie and C. Cui are with the School of Computer Science and Technology, Shandong University of Finance and Economics, Shandong 250014, China (e-mail: niexsh@sdufe.edu.cn; bruincui@foxmail.com).

Y. Yin and J. Liu are with the School of Computer Science and Technology, Shandong University, Shandong 250100, China, and also with the School of Information Science and Engineering, Shandong University, Shandong 250100, China (e-mail: ylyin@sdu.edu.cn; juliu@sdu.edu.cn).

J. D. Sun is with the School of Information Science and Engineering, Shandong Normal University, Shandong 250014, China, and also with the Institute of Data Science and Technology, Shandong Normal University, Shandong 250014, China (e-mail: jiandesun@hotmail.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/TMM.2016.2629758

I. Introduction

With the development of information technology, the number of digital videos available on the Web has increased explosively, and the openness of networks has made access to video content considerably easier and cheaper. Therefore, many illegal and useless video copies or near-duplicates appear on the Web. These copies are generated by simple reformatting, transformations, and editing. Illegal and useless copies result in user inconvenience when surfing the Internet. A Web user may want to search for an interesting video but may end up with many near-duplicate videos with low-quality images, which is disappointing and time-consuming. In addition, most of these copies are pirated and infringe on the video producers copyright. Therefore, the presence of massive numbers of copies imposes a strong demand for effective near-duplicate video detection (NDVD) in many applications, such as copyright enforcement, online video usage monitoring, and video database cleansing. NDVD is a broad topic that has several goals such as finding copies of brief excerpts, partial-frame copies, and near-duplicates of entire clips. In this study, we focus on the near-duplicates of entire clips.

Watermarking is a traditional technology used to detect copies of images or videos. It embeds imperceptible watermarks into the media to prove its authenticity. However, the watermarks embedded into the media may cause distortion. By contrast, robust fingerprinting techniques extract the most important features of the media to calculate compact digests that allow for efficient content identification without modifying the media. Therefore, robust video fingerprinting has drawn increasing attention in the field of NDVD.

Many video fingerprinting methods have been developed in recent years; global [1]–[6] and local feature-based [8]–[11] methods are two primary types. In global feature-based methods, the video is represented as a compact global feature, such as color space [2], histograms [3], and block ordinal ranking [4]. In the temporal [5] and transformation-based methods, 3D-discrete cosine transform [6] and nonnegative matrix factorization [7], for example, can also be considered global feature-based approaches [8]. The local feature-based methods use the descriptors of the local region around interest points, such as the Harris interest point detector [9], scale-invariant feature transform (SIFT) [10], centroid of gradient orientations (CGO) [11], and the speed-up robust feature (SURF) [12]. Global features can achieve fast retrieval speed but are less effective in handling video copies with layers of editing, such as caption/logo insertion, letter-box, and shifting. Local features

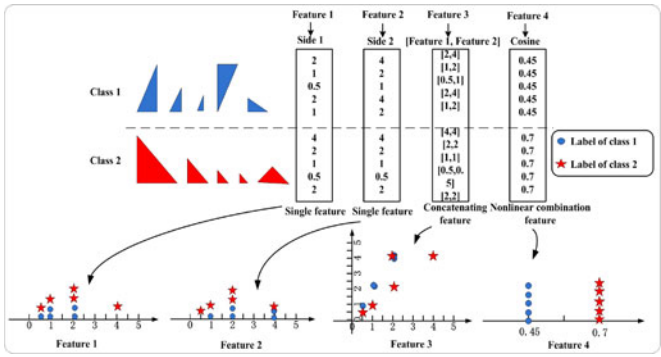

Fig. 1. Illustration of single feature, concatenating feature and nonlinear combined feature.

are more effective for most video editing; however, they usually require more computation. Recently, Li *et al.* [13] used graph model embedding to generate video fingerprints. Almeida *et al.* [14] calculated horizontal and vertical coefficients of DCT from video frames as the video fingerprint. Sun *et al.* [15] proposed a temporally visual weighting method based on visual attention to produce a video fingerprint. Nie *et al.* [16] used graph and video tomography [17] to compute a video fingerprint. Although these methods used different features, they can be roughly arranged into global or local feature-based methods.

Generally, many state-of-the-art methods only use a single type of feature, global or local, to represent the corresponding video, which causes difficulty in resisting various video attacks. Therefore, The algorithms that combine multiple features have been presented in recent years because of the limitation of individual features. Song *et al.* [18] proposed a video hashing learning method by weighted multiple feature fusion. Jiang *et al.* [19] proposed a video copy detection system based on temporal pyramid matching based on two layers. Li *et al.* [20], [21] used dense color SIFT, DCT coefficients, and weighted ASF to detect video copies, and made the final decision by fusing the detection results from different detectors. Although these methods combine multiple features to a certain extent, they do not fully use the assistance and consensus among features, which is important in multiple feature fusion.

Each type of feature reflects specific information on the original video, and can be taken as one view of the original video. According to the multi-viewing theory [36], the mutual information of different views (features) can improve the performance of the task. Therefore, a fusion of different features is beneficial to improve the performance of video fingerprinting. Intuitively, concatenating different feature vectors one by one is a directed form of feature fusion; combining multiple features with different weights is also a solution. However, the concatenated or weighted vector representation of features weakens the power of the propagation among the multiple features and even ignores their relations to a certain extent. For example, Fig. 1 illustrates a simple scenario of this issue. In this figure, two classes of right triangles exist and are similar to each other in each class (we can call them near-duplicates) because only affine transformations (scaling and rotation) are performed. We take the two sides of each right triangle as Feature 1 and Feature 2, respectively. Then, the concatenating feature [Feature 1, Feature 2] is labeled Feature 3. We use a nonlinear combination cosine as the fourth feature which is $\text{cosine} = (Feature1)/(\sqrt{(Feature\ \ 1)^2 + (Feature\ \ 2)^2})$. Obviously, the fourth feature cosine shows the best classification performance, proving that correlation and consensus can lead to performance improvement among different features.

Moreover, the concatenated or weighted feature vectors are difficult to handle with different scales. When imaging a scenario of people similarity evaluation, we use two different features of people, namely, *height* and *weight*. If we concatenate these two features to a new feature [*height, weight*], we may not evaluate the similarity of people correctly. For example, given two people with feature vector of [175 cm, 65 kg] and [160 cm, 80 kg], the first person is 15 cm taller but 15 kg lighter than the second person. Thus, evaluating these whether two people similar is difficult. Therefore, we should explore a new strategy to fuse multiple features that can capture the inherent characteristics of the original data and the assistance between different features.

In addition, the environment of a network is increasingly complex, and combined modifications applied to videos are increasingly common. One of the challenges in NDVD is the robustness under the combined modifications. Generally, when traditional methods such as single feature-based and concatenation-based methods suffer some combined modifications, the performances are not as good as the performance under single-mode modification. In fact, according to the boundary of multi-view analysis [37], the probability of a disagreement of two features upper bounds the error rate of either feature. Thus, by exploring the maximized agreement and consensus of different features, the error rate of each feature will be minimized. Therefore, mining the assistance and consensus among multiple features and making full use of them are necessary for NDVD.

To address the aforementioned issues, we propose a new notation called comprehensive feature to represent video content in this study. The comprehensive feature is not an video original feature, but a comprehensive, intrinsic, and transformed feature that can capture the assistance and consensus among multiple features. Compared with other types of features, the comprehensive feature has two advantages (detailed analysis is in Section II-B4): 1) It consists of the principal components and intrinsic characteristics of the original video content, which can maximize the consensus of different features. 2) It is a compact and comprehensive representation of video content that has eliminated the noises and redundant information among different features. In this study, we propose a general comprehensive feature-mining scheme based on a tensor model. Tensor is the natural generalization of vector and matrix, and the algebra of higher-order tensors can consider the contextually different features. Furthermore, the tensor representation and decomposition can express intra-feature context correlation intuitively and propagate the corresponding inter-feature context conveniently as well [22], because the tensor decomposition is a mixed staggered sampling, i.e., alternating sampling in the different components occurs rather than sampling from the components one

by one. Therefore, the tensor model is one of the best tools for comprehensive feature generation.

Security and binarization are also considered in some existing video fingerprinting systems. Randomization strategies [23], [24], [29] via secret keys are popularly used in the fingerprinting system to enhance security, and the final fingerprint sequence can also be quantified to binary values via secret keys. These common strategies can be definitely applied in the proposed method. Since these strategies and analysis have been described in detail in the existing methods, we do not discuss these issues in the present study.

In summary, the main contributions of this study are the following:

1) We present a comprehensive feature-based scheme to capture the assistance and consensus among different features using a tensor model, and it is feasible for fusing multiple features using the proposed scheme. In this scheme, the intrinsic and latent characteristics of multiple features are expressed completely through the comprehensive feature. We also show what we mined in the comprehensive feature, and give a theoretical analysis of its robustness.
2) We propose an auxiliary matching strategy based on the tensor model. In this strategy, the core tensor is used to narrow down the search range. Then, the existing matching algorithm can be implemented in the smaller obtained dataset to find a match. This matching strategy can accelerate fingerprint matching to a certain extent, especially for a large-scale video fingerprint database.

II. Proposed Method

In this section, we will describe the proposed video fingerprinting scheme, and give a theoretical analysis of the comprehensive feature mining and its robustness. To make the scheme more understandable, we first list certain notations regarding the tensor [25].

A. Related Notations and Technologies

Tensor: A tensor is a multidimensional array. Formally, an N_{th}-order tensor is an element of the tensor product of N vector spaces, each with its own coordinate system. A vector and matrix are the first- and second-order tensors, respectively. Tensors of order three or higher are called higher-order tensors.

High-order tensors can be approximated by the sum of low-rank tensors (the definition of tensor rank is described in [25]). The CANDECOMP/PARAFAC (CP) and Tucker model decomposition are two popular methods for approximation. They factorize a tensor into a sum of component low-rank tensors.

CP model: The CP model factorizes a tensor into a sum of component rank-one tensors. Given a third-order tensor $\chi \in \Re^{I_1 \times I_2 \times I_3}$, it can be concisely expressed as

$$\chi \approx \left[\left[\lambda;\ \mathbf{A}^{(1)},\ \mathbf{A}^{(2)},\ \mathbf{A}^{(3)}\right]\right] = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \mathbf{a}_r^{(3)} \tag{1}$$

where R is a positive integer, and $\mathbf{a}_r^{(n)} \in \Re^{I_n}$. λ_r is constant for $r = 1, \cdots, R$. $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \ldots, \mathbf{a}_R^{(n)}]$ $(1 \le n \le 3)$ are factor matrices of the tensor. The symbol "$\circ$" represents the vector outer product.

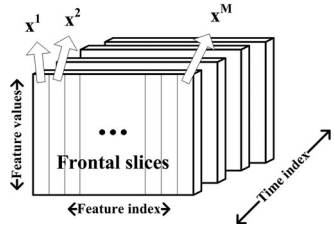

Fig. 2. Diagram of the video tensor.

Tucker model: The Tucker model decomposes a tensor into a core tensor multiplied by matrices along each mode. For example, a third-order tensor χ, can be decomposed as follows:

$$\begin{aligned}\chi \approx\ & \left[\left[\kappa; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}\right]\right] = \kappa \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} \\ =\ & \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \sum_{j_3=1}^{J_3} k_{j_1 j_2 j_3} \mathbf{a}_{j_1}^{(1)} \circ \mathbf{a}_{j_2}^{(2)} \circ \mathbf{a}_{j_3}^{(3)} \\ \text{where}\quad & \mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \ldots, \mathbf{a}_{J_n}^{(n)}] \in \Re^{I_n \times J_n}, \\ & 1 \le n \le 3 \end{aligned} \tag{2}$$

where I_n and J_n are positive integers, and κ is the core tensor. $\mathbf{A}_{(n)}$ is the factor matrix.

Tensor has been applied to many fields, such as computer vision and signal processing. In this study, the video is considered a third-order tensor in the tensor space constructed by different features.

B. Comprehensive Feature Mining-Based Robust Video Fingerprinting

In the proposed scheme, we first use different features to generate a video tensor. Then, the comprehensive feature is mined to generate the video fingerprint using the Tucker model. In addition, a matching strategy is presented by the core tensor to accelerate fingerprint matching.

1) Video Tensor Construction: We let M be the number of features and $\mathbf{x}^m \in \Re^{d_m \times 1}$ the m_{th} feature, where $m = 1, 2, \ldots, M$ and d_m is the dimensionality of the m_{th} video feature. The video is considered a third-order tensor constructed by the M features shown in Fig. 2. In the video tensor, the frontal slice is the multiple feature values, whereas mode-3 is the time sequence. The length of each feature vector and the total number of features determine the size of the frontal slice. One feasible strategy is to take the mean length of the entire feature vector as the standard length, and then concatenate all feature vectors column by column in the front slice with a standard length for the column. Then, we use tensor decomposition to generate the comprehensive feature.

2) Tensor Model Selection: The CP and Tucker models are two main tensor decomposition models. We use the Tucker model in the proposed framework. The third-order Tucker model represents the data spanning the three modes by the vectors given by columns $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$ and $\mathbf{A}^{(3)}$, as shown in (2). As a

result, the Tucker model encompasses all possible linear interactions between vectors pertaining to the various modes of the data [28]. The CP model is a special case of the Tucker model where the size of each mode is the same, i.e. $J_1 = J_2 = J_3$ in (2). In this study, we used the Tucker model instead of the CP model. The first reason for this choice is that the Tucker model is more flexible and scalable during decomposition, where users can optimally select the number of factors along each mode. The CP decomposition can only provide the same number of components in all modes, i.e., the results of CP decomposition are mathematical artifacts that may not be physically meaningful. There are strong effects among the various components decomposed by the CP model, which make the CP model unstable, slow in convergence, and difficult to interpret [30]. More importantly, a core tensor that considers all possible linear interactions between the components of each mode, can be obtained by the Tucker model, and is stable [25], [30] to a certain extent. The core tensor is used for matching in the proposed framework.

Choosing the number of components for each mode [i.e., the values of J_1, J_2 and J_3 in (2)] is a particular challenge in the Tucker model. In the present study, we use a Bayesian approach called automatic relevance determination (ARD) [28] to determine the number of components in each mode. ARD alternately optimizes the parameters to discover which components are relevant, and the parameters are modeled as either Gaussian prior or Laplace prior. We will shortly describe how we use ARD in this study, and more details can be seen in [28].

We let χ be the third-order video tensor. Then, for the Tucker decomposition

$$\chi \approx \mathbf{\Gamma} = \kappa \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)}. \tag{3}$$

The model can also be written as

$$\begin{aligned} \chi_{i_1,i_2,i_3} &\approx \Gamma_{i_1,i_2,i_3} \\ &= \sum_{j_1,j_2,j_3} \kappa_{j_1,j_2,j_3} \mathbf{A}^{(1)}_{i_1,j_1} \mathbf{A}^{(2)}_{i_2,j_2} \mathbf{A}^{(3)}_{i_3,j_3} \end{aligned} \tag{4}$$

where $\kappa \in \mathbf{R}^{J_1 \times J_2 \times J_3}$ and $\mathbf{A}^{(n)} \in \mathbf{R}^{I_n \times J_n}$. If we denote this Tucker model as $T(J_1, J_2, J_3)$, our goal is to find the optimal values of J_1, J_2 and J_3.

According to (3), the tensor χ can also be rewritten as

$$\chi = \mathbf{\Gamma} + \xi \tag{5}$$

where ξ is an error parameter, and its distribution can be taken as an independent identical distribution (i.i.d.) with Gaussian noise. i.e.,

$$\begin{aligned} P(\xi) &= P(\chi|\mathbf{\Gamma}, \sigma) \\ &= \prod_{i_1,i_2,i_3} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\chi_{i_1,i_2,i_3} - \Gamma_{i_1,i_2,i_3})^2}{2\sigma^2}\right). \end{aligned} \tag{6}$$

The Gaussian prior P_G and Laplace prior P_L on the parameter θ_d used in ARD are the following:

$$P_G(\theta_d|\alpha_d) = \prod_j \left(\frac{\alpha_d}{2\pi}\right)^{1/2} \exp\left(-\frac{\alpha_d}{2}\theta^2_{j,d}\right) \tag{7}$$

$$P_L(\theta_d|\alpha_d) = \prod_j \frac{\alpha_d}{2} \exp(-\alpha_d|\theta_{j,d}|_1). \tag{8}$$

According to (7) and (8), we can easily obtain $P_G(\mathbf{A}^{(n)}|\alpha^{(n)})$, $P_L(\mathbf{A}^{(n)}|\alpha^{(n)})$, $P_G(\kappa|\alpha^\kappa)$ and $P_L(\kappa|\alpha^\kappa)$ by Gaussian or Laplace prior, respectively, where $1 \le n \le 3$. The parameters $\alpha^{(n)}$ and α^κ are also given as uniform priors for simplification in ARD.

Therefore, we can explore optimal J_1, J_2 and J_3 by minimizing the least squares objective $||\chi - \mathbf{\Gamma}||^2_F$, which corresponds to minimizing the negative log-likelihood in the Bayesian framework.

The likelihood function can be written as

$$\begin{aligned} L = \; & P(\kappa, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}|\chi, \sigma, \alpha^\kappa, \alpha^{(1)}, \alpha^{(2)}, \alpha^{(3)}) \propto \\ & P(\chi|\mathbf{\Gamma}, \sigma^2) P(\kappa|\alpha^\kappa) P(\mathbf{A}^{(1)}|\alpha^{(1)}) P(\mathbf{A}^{(2)}|\alpha^{(2)}) P(\mathbf{A}^{(3)}|\alpha^{(3)}). \end{aligned} \tag{9}$$

Subsequently, the negative log likelihood ($-\mathrm{Log}(L)$) using Gaussian and Laplace priors are proportional to (10) and (11), respectively.

$$\begin{aligned} C + \frac{1}{2\sigma^2}||\chi - \Gamma||^2_F + 0.5 * \sum_n \sum_d \alpha^{(n)}_d ||\mathbf{A}^{(n)}_d||^2_F + \alpha^\kappa ||\kappa||^2_F \\ + 0.5 * I_1 I_2 I_3 \log \sigma^2 - 0.5 * \sum_n \sum_d I_n \log \alpha^{(n)}_d \\ - 0.5 * J_1 J_2 J_3 \log \alpha^\kappa \end{aligned} \tag{10}$$

$$\begin{aligned} C + \frac{1}{2\sigma^2}||\chi - \Gamma||^2_F + \sum_n \sum_d \alpha^{(n)}_d ||\mathbf{A}^{(n)}_d||_1 + \alpha^\kappa ||\kappa||_1 \\ + 0.5 * I_1 I_2 I_3 \log \sigma^2 - \sum_n \sum_d I_n \log \alpha^{(n)}_d \\ - J_1 J_2 J_3 \log \alpha^\kappa \end{aligned} \tag{11}$$

where C, $||\bullet||_F$ and $||\bullet||_1$ are a constant value, F-norm and 1-norm, respectively.

We equate the derivatives of (10) with respect to σ^2, $\alpha^{(n)}_d$ and α^κ to zero, respectively. Then, we obtain the following parameters by Gaussian prior

$$\sigma^2 = \frac{||\chi - \Gamma||^2_F}{I_1 I_2 I_3}, \quad \alpha^{(n)}_d = \frac{I_n}{||\mathbf{A}^{(n)}_d||^2_F}, \quad \alpha^\kappa = \frac{J_1 J_2 J_3}{||\kappa||^2_F}. \tag{12}$$

The parameters by Laplace prior can be obtained in the same way by equating the derivatives of (11) to zero, which are as follows:

$$\sigma^2 = \frac{||\chi - \Gamma||^2_F}{I_1 I_2 I_3}, \quad \alpha^{(n)}_d = \frac{I_n}{||\mathbf{A}^{(n)}_d||_1}, \quad \alpha^\kappa = \frac{J_1 J_2 J_3}{||\kappa||_1}. \tag{13}$$

Generally, the first row expressions of (10) and (11) can be considered l_2-regularized and l_1-regularized problems, respectively, while the second and third rows are the normalization constants in the likelihood terms. The l_2-regularized and l_1-regularized problems are equivalent to the regular ridge regression and sparse regression problems, respectively, which can solved by existing algorithms.

In (12) and (13), σ^2 can be learned from data or estimated by a signal-to-noise rate (SNR)-based method [28]. Given the

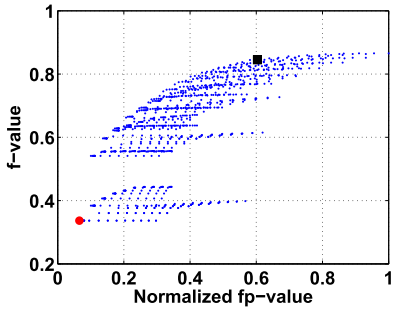

Fig. 3. f-value versus normalized fp-value.

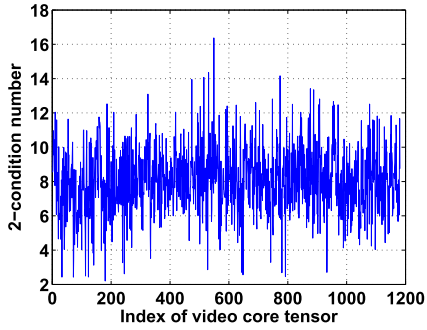

Fig. 4. Condition numbers of the transposition of the video core tensor.

initialization of the parameters and $\mathbf{A}^{(n)}$, the optimal values of J_1, J_2 and J_3 are obtained by alternately updating $\alpha_d^{(n)}$, α^κ, κ and $\mathbf{A}^{(n)}$ based on (12) and (13) with the solutions of the l_2-regularized and l_1-regularized problems until convergence. Which prior assumption (Gaussian or Laplace) is used to update the parameters corresponds to setting the parameters of the priors such that they match the posteriors distribution [38].

To evaluate the efficiency of the ARD-based Tucker model, we validate the obtained number of components using two metrics called goodness-of-fit value (f-value) and free parameters number (fp-value) [27], respectively, which are defined as

$$f = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} r_{ijk}^2}{\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} x_{ijk}^2} \tag{14}$$

$$fp = IP + JQ + KR + PQR - P^2 - Q^2 - R^2 \tag{15}$$

where x_{ijk} and r_{ijk} are the elements of the original and reconstructed tensor using the obtained components with size $I \times J \times K$, respectively. P, Q and R are the size of each mode after tensor decomposition. f-value and fp-value are widely used to evaluate the goodness-of-fit of the model and the complexity during model decomposition [27]. Generally, higher f-values means higher fit with the original data, and higher fp-values means more computation during tensor decomposition. A tradeoff between fitting and computation should exist. Fig. 3 shows the f-value versus normalized fp-value (true fp-values are divided by the maximum fp-value) for a data set under various combinations of P, Q and R. The point obtained by the ARD-based algorithm is marked with a black square in Fig. 3, which shows a better fitness of the original data. Compared with the tensor model used in [23], [24] which is marked with a red circle in Fig. 4, the proposed ARD model has a much better fitness and an acceptable computation complexity.

3) Comprehensive Feature Mining: After the video tensor decomposition by the Tucker model, three low-rank tensor matrices $\mathbf{A}^{(n)}(n = 1, 2, 3)$ are obtained that fuse the different video features. To make the final fingerprint vector more compact, we average the absolute values of elements in component matrix $\mathbf{A}^{(n)}$ by row to obtain a component vector, and then concatenate the three component vectors to obtain the comprehensive feature $\mathbf{y}$, which is considered the final video fingerprint vector.

4) Analysis of Comprehensive Feature: What we mined in the comprehensive feature is important in evaluating the performance of the proposed method. As mentioned, the Tucker model is used in comprehensive feature mining. We first review the process of Tucker decomposition.

Given a video tensor $\chi \in \Re^{I_1 \times I_2 \times I_3}$, it can be decomposed as

$$\begin{aligned}\chi &\approx \kappa \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_N \mathbf{A}^{(3)} \\ &= \left[\left[\kappa; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}\right]\right]\end{aligned} \tag{16}$$

where each $\mathbf{A}^{(n)}$ $(1 \le n \le 3)$ is an orthogonal matrix, and κ is the core tensor. For distinct modes in a series of multiplications, the order of the multiplication is irrelevant [25]. Therefore, the core tensor κ is

$$\kappa \approx \chi \times_1 \mathbf{A}^{(1)T} \times_2 \mathbf{A}^{(2)T} \times_N \mathbf{A}^{(3)T} \tag{17}$$

where $\mathbf{A}^{(n)T}$ is the transposition of $\mathbf{A}^{(n)}$. We can use the following optimization problem to obtain κ and $\mathbf{A}^{(n)}$:

$$\begin{aligned}\min_{\kappa, \mathbf{A}^{(1)}\mathbf{A}^{(2)}\mathbf{A}^{(3)}} & ||\chi - \left[\left[\kappa; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}\right]\right]||^2 \\ \text{s.t.} \quad & \kappa \in \Re^{J_1 \times J_2 \times J_3}, \\ & \mathbf{A}^{(n)} \in \Re^{I_n \times J_n} \text{ and column-wise orthogonal.}\end{aligned} \tag{18}$$

Consequently, the objective function of (18) can be rewritten as (19)

$$\begin{aligned}&||\chi - \left[\left[\kappa; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}\right]\right]||^2 \\ &= ||\chi||^2 - 2 < \chi, \left[\left[\kappa; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}\right]\right] > \\ &\quad + || \left[\left[\kappa; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}\right]\right] ||^2 \\ &= ||\chi||^2 - 2 < \chi \times_1 \mathbf{A}^{(1)T} \times_2 \mathbf{A}^{(2)T} \times_3 \mathbf{A}^{(3)T}, \kappa > + ||\kappa||^2 \\ &= ||\chi||^2 - 2 < \kappa, \kappa > + ||\kappa||^2 \\ &= ||\chi||^2 - ||\kappa||^2 \\ &= ||\chi||^2 - ||\chi \times_1 \mathbf{A}^{(1)T} \times_2 \mathbf{A}^{(2)T} \times_3 \mathbf{A}^{(3)T}||^2.\end{aligned} \tag{19}$$

Furthermore, the optimization problem (19) is equal to the following maximization problem because $||\chi||^2$ is constant:

$$\begin{aligned}&\max_{\mathbf{A}^{(n)}} || \chi \times_1 \mathbf{A}^{(1)T} \times_2 \mathbf{A}^{(1)T} \times_3 \mathbf{A}^{(3)T} ||^2 \\ &\quad \Leftrightarrow \max_{\mathbf{A}^{(n)}} ||\mathbf{A}^{(n)T}\mathbf{W}|| \\ &\text{s.t.} \quad \mathbf{W} = \mathbf{X}_{(n)}(\mathbf{A}^{(3)} \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \mathbf{A}^{(1)}) \\ &1 \le n \le 3 \end{aligned} \tag{20}$$

where "$\otimes$" is the Kronecker product, and setting $\mathbf{A}^{(4)} = \mathbf{A}^{(0)} = \mathbf{I}$. This maximization can be solved using the ALS algorithm by setting $\mathbf{A}^{(n)}$ as the J_n leading left singular vectors of $\mathbf{W}$, and it will converge to a solution [35].

The final low-rank tensor matrix $\mathbf{A}^{(n)}$ is computed from the left singular vectors of $\mathbf{W}$, which consists of the original video tensor χ. According to the property of Singular Value Decomposition (SVD), the left matrix obtained from the SVD contains the intrinsic characteristics of the original matrix in the column space which consists of different features.

From another perspective, each column of $\mathbf{W}$ is an approximation of the feature vector. If each feature vector is taken as a point in original space, the principal components that represent the main information among the point distribution can be computed by Principal Component Analysis (PCA), and they are computed from the left singular matrix of $\mathbf{W}$ according to the relation between PCA and SVD. Therefore, the comprehensive feature mined from $\mathbf{A}^{(n)}$ can be considered the intrinsic and principal information of original data, where the noise and uselessness among different features have been eliminated. Generally, the different features should reach a consensus in representing the video content [36]. The intrinsic and principal information of the original video exploited during the comprehensive feature mining are only the consensus and assistance of different features, which lead to improved performance in representing the video content.

Considering the preceding analysis, we can conclude that the information exploited during the comprehensive feature mining is composed of the principal components and intrinsic characteristics of the original video tensor. However, the comprehensive feature mining is different from SVD or PCA because the extraction of the principal components and intrinsic characteristics in comprehensive feature mining is iterative until converge while it is one-off in SVD or PCA. The iterative process and alternative optimization capture more intrinsic information and consensus, which is the reason we use the tensor model during comprehensive feature mining.

5) Robustness of Comprehensive Feature: Robustness is the most important issue in video fingerprinting system. In this section, we present a qualitative analysis of the proposed framework. The experimental result in the next section proves the robustness of the comprehensive feature.

The video fingerprint vector of the proposed method is the comprehensive feature that consists of three modes of video tensors. Therefore, without loss of generality, the mode-3 factor matrix $\mathbf{A}^{(3)}$ is taken as an example to analyze the robustness. Given that χ is the tensor of a video, the approximation of this tensor through the Tucker model is

$$\begin{aligned}\chi &\approx \kappa \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} \\ &= \gamma \times_3 \mathbf{A}^{(3)}\end{aligned} \tag{21}$$

where $\gamma = \kappa \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)}$. According to the tensor multiplication and matricization, (21) can be written by matrices

$$\mathbf{X} \approx \mathbf{A}^{(3)}\mathbf{P} \tag{22}$$

where $\mathbf{X}$ and $\mathbf{P}$ are the matricizations of χ and γ in mode-3, respectively. $\mathbf{X}$ is the multiple feature matrix in the proposed method. Furthermore, we rewrite (22) as

$$\mathbf{P}^T\mathbf{A}^{(3)T} \approx \mathbf{X}^T. \tag{23}$$

Obviously, we can approximately consider (23) as a linear equation (24), where $\mathbf{P}^T$, $\mathbf{A}^{(3)T}$ and $\mathbf{X}^T$ are the coefficient matrix, variable and constant term, respectively.

$$\mathbf{P}^T\mathbf{A}^{(3)T} = \mathbf{X}^T \tag{24}$$

Now, we can consider the robustness of comprehensive feature $\mathbf{A}^{(3)}$ as the stability of the solution in this linear equation. Therefore, if the solution of (24) is stable, $\mathbf{A}^{(3)}$ becomes robust. The metric that measures the solution stability of a linear equation is the condition number of the coefficient matrix. Smaller condition numbers mean a stronger solution stability. Then, the condition number of the coefficient matrix in (24) is

$$\text{condition}_2(\mathbf{P}^T) \approx \text{condition}_2((\kappa \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)})^T) \tag{25}$$

where $\text{condition}_2(\bullet)$ is the 2-norm condition number of $\bullet$, and it is defined as $\text{condition}_2(\bullet) = ||\bullet||_2||\bullet^{-1}||_2$. $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ are the factor matrices after the Tucker decomposition, which are column-wise orthogonal [25]. According to the property of condition (i.e., $\text{condition}_2(\mathbf{UM}) = \text{condition}_2(\mathbf{M})$, if $\mathbf{U}$ is an orthogonal matrix), (25) is rewritten as

$$\begin{aligned}&\text{condition}_2((\kappa \times \mathbf{A}^{(1)} \times \mathbf{A}^{(2)})^T) \\ &= \text{condition}_2((\mathbf{A}^{(2)})^T \times (\mathbf{A}^{(1)})^T \times \kappa^T) = \text{condition}_2(\kappa^T).\end{aligned} \tag{26}$$

As mentioned, κ^T is the transposition of the core tensor, and its matricization is a non-singular matrix. Generally, its condition is small. Fig. 4 shows the distribution of their condition numbers. Most of the condition numbers are between 2 and 18, which are acceptable values for a well-conditioned problem. Therefore, (24) is a well-conditioned equation and has a stable solution when the coefficient matrix or constant term changes. In other words, $\mathbf{A}^{(3)}$, which is used to generate the final video fingerprint, is robust.

6) Matching Strategy: Fingerprint vector matching is one of the important steps in a video fingerprinting system. In a real application, the video website manages a video fingerprint database. When a user uploads a new video, the management first generates its fingerprint and then matches the fingerprints in the database. Many matching schemes exist in the literature, such as exhaustive matching, the tree-based strategy and inverted files. In this study, we provide an auxiliary strategy for

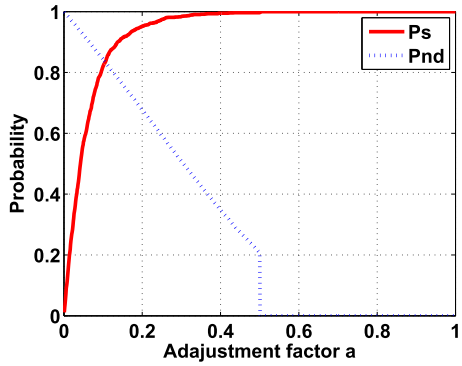

Fig. 5. Values of P_s , P_{nd} under different adjustment factors.

the existing matching schemes. The purpose of this strategy is to narrow down the matching range in advance. Then, the existing matching schemes can be executed in the obtained dataset with smaller sizes compared with the original. The proposed matching strategy is based on the core tensor κ, which represents the level of interaction between the different components.

In this strategy, we use the sum of elements in the core tensor as a match tag, and the pre-matching is conducted in the database using the tag. Given a query video fingerprint with match tag T_q, the fingerprint, whose match tags are between $T_q - a * T_q$ and $T_q + a * T_q$, are obtained in the database after the pre-matching. Because the length of the match tag is one, the pre-matching is rapid. The parameter $a \in [0, 1]$ is called the adjustment factor. We show how to select the adjustment factor in the next paragraph.

Given that S is the fingerprint dataset obtained after the pre-matching, P_s is the rate of falling into S for the visually similar videos (which are the true near-duplicates we want to find), while P_{nd} is the rate of not falling into S for the visually different videos (which are not the near duplicates). Intuitively, we could expect that both P_s and P_{nd} are high. However, the larger a causes a higher P_s but lower P_{nd}. Conversely, a smaller a leads to a high P_{nd} but lower P_s. Therefore, a tradeoff should be considered between P_s and P_{nd} by choosing the adjustment factor a. Given a video dataset, we can use the curves of P_s and P_{nd} versus a to choose the adjustment factor. Fig. 5 shows P_s and P_{nd} versus the adjustment factor a for a video database. The intersection of the two curves is a good choice for the adjustment factor. However, we cannot use that point because we should first consider avoiding a mismatch in the pre-matching, i.e., the fingerprint vectors similar to the query should be involved in S as much as possible. P_s is not sufficiently high (less than 90%) on the intersection, which leads to a mismatch. Therefore, we choose a, while giving priority to P_S. Taking Fig. 5 as an example, we choose $a = 0.3$, where the P_s is nearly 1, and P_{nd} is more than 0.6. In other words, almost all of the fingerprint vectors of visually similar videos compared with the query are involved in S after pre-matching, and those of visually different videos are not involved in S with a rate more than 0.6. Less than 40% of the visually different video fingerprint vectors fall into S. Therefore, the size of the dataset after pre-matching is reduced. Further matching can be performed by any existing search technology. Generally, we assume that N videos exist in the original database, and the complex of the matching is reduced to $O(K)$, where K is the number of fingerprint vectors after the pre-matching, and $K < N$.

III. Experimental Results and Analysis

A. Experimental Setting

In the experiment, as mentioned, we take the global feature (normalized 64-bin histograms), local feature (SURF points) and temporal feature (the differences of 64-bin histograms between adjacent frames) as the inputs of the scheme. The purpose of video fingerprinting is to detecting near-duplicate videos or video copies. It must be kept in mind that the modifications performed on near-duplicate videos or video copies, such as noise addition, blurring and geometrical transformation, always influence the visual features, so we concentrate more on the low-level visual feature fusion in this study. However, among different feature levels, high-level semantic features are an important factor for narrowing down the semantic gap in computer vision. This may also be an assistance for low-level hand-crafted features, but fusing them with low-level features directly under the proposed strategy may not be a good choice. The existing works [39], [40] have investigated the effectiveness of semantic deep features in computer vision, and the work in [39] makes initial attempts in a complementary way with low-level features and high-level semantic deep features in the field of action recognition. Inspired by this work, we will investigate semantic deep features in video fingerprinting in the future study.

Extensive experiments were conducted to evaluate the performance of the proposed scheme. Two types of database are used in the experiments. One is a combined database downloaded from the CC_WEB_VIDEO Dataset (vireo.cs.cityu.edu.hk/webvideo) and the OV Dataset (www.open-video.org); another is a standard database collected from the TRECVID CCD task [31]. In the two databases, we compared the proposed scheme with LRTA [23], [24], CGO [11], 3-D DCT [6] and MFH [18] methods, respectively.

1) Combined Database: In this database, the original videos were downloaded from the CC_WEB_VIDEO Dataset and OV Dataset. We then applied different modifications to each video. Thus, almost 20,000 videos are used in our database for the experiments. A total of 11 single-mode modifications/attacks were applied on the test videos: 1) rotation; 2) Additive Gaussian White Noise (AGWN); 3) blurring; 4) contrast enhancement; 5) letter-box; 6) logo/caption insertion; 7) cropping; 8) flipping; 9) picture in picture; 10) affine transformation; and 11) re-sampling. The characterization or parameters of various modifications are provided in Table I. Moreover, to evaluate the performance of the proposed scheme under combined modifications, we applied four types of combinations of more than one modification: 1) decrease in quality 1 (contrast + change of gamma + AGWN); 2) decrease in quality 2 (contrast + change of gamma + AGWN + blur + frame dropping); 3) post-production 1 (crop + flip + insertion of patterns); and 4) post-production 2 (crop + flip + insertion of patterns + picture in picture + shift).

TABLE I
DESCRIPTION OF VIDEO MODIFICATIONS

Attacks	Parameter Setting
Rotation	5 deg counterclockwise
AGWN	$\sigma_N = 110$
Blurring	motion blur, 10 pixels
Contrast	1% is saturated at low and high intensities
Letter-box	10% of pixels are replaced by black box
caption insertion	insert a line of text at the bottom
Picture in picture	insert a different picture, size 100 x 100
Frame cropping	about 25% cropping
Frame re-sampling	about 5% frames changing
Affine transformation	transformation matrix [1 0 0; 0.5 1 0; 0 0 1]

TABLE II
TRANSFORMATIONS USED IN THE TRECVID CBCD TASK

Label	Description
T1	Simulated camcording
T2	Picture in picture Type 1
T3	Insertion of pattern
T4	Strong re-encoding
T5	Change of gamma
T6	Decrease in quality
T8	Post production (randomly combined 3 transformations)
T10	Combination of 3 categories from T1-T8

2) Standard Database: The main goal of the TREC Video Retrieval Evaluation (TRECVID) is to promote progress in the content-based analysis of and retrieval from digital video via an open metrics-based evaluation. The performance of the proposed method is primarily evaluated using the benchmarking CBCD (also abbreviated as CCD) task of TRECVID 2010. In TRECVID 2010, the reference database has 425-hour videos composed of 11,524 videos collected from the Internet. A query dataset includes 10,976 videos that are an average of 70 seconds long for each query. Eight video transformations that are defined by TRECVID are applied to the videos that are shown in Table II.

B. Performance Evaluation

1) Results on the Combined Database: To evaluate the performance of the proposed scheme in the combined database, the receiver operating characteristic (ROC) curve and F-score are used, respectively.

The miss and false alarm probability are considered in the ROC curve. The miss probability is defined as the probability of true copies without detection, whereas the false alarm probability is defined as the probability of false positive copies that are actually negative cases. Fig. 6 shows the ROC (log) curves under different attacks. The performance of the proposed scheme is better than the one of the other methods under almost all modifications, especially under some popular manipulations in the user-generated videos and video post-production, such as caption insertion and picture-in-picture. In addition, the performance has significant improvement under the combined modifications, as can be observed in Fig. 6(l)–(o). However, the performance under AWGN is not as good as the other performances. The main reason is that some noisy points are incorrectly considered interested points in the SURF detector.

Robustness and discrimination are important for a video fingerprinting system. Generally, robustness and discrimination are expressed by recall and precision [32], respectively. The F-score which is a single combined metric is taken as a quantitative measure in this study, and is defined as follows [32]:

$$F_\beta = (1+\beta^2)\frac{P_p * P_r}{\beta^2 P_p + P_r} \tag{27}$$

where P_p and P_r are precision and recall rate. β is a parameter that defines how much weight should be given to recall versus precision. We used $\beta = 0.5$ in the experiments. A larger F-score means better algorithm performance. The F-score results under various modifications are shown in Table III. In contrast to other methods, the proposed method is better under most modifications. The improvement under combined modification is greater than those under single-mode modification.

2) Results on Standard Database: In the standard database TRECVID, the receiver operating characteristic (ROC) curve are also used in the performance evaluation. Fig. 7 shows the results under the modifications of T2, T5, T8 and T10, and we can see that the proposed method has an apparent improvement compared with other methods.

C. Threshold Analysis

In a real application, the video fingerprinting system should decide whether the query video is a modified copy. The common method is to set a threshold τ in advance. Based on the assumption that the original and query fingerprints are $\mathbf{F}(V)$ and $\mathbf{F}(V_A)$, respectively, a decision is made as follows:

$$\begin{cases} ||\mathbf{F}(V) - \mathbf{F}(V_A)||_2 \le \tau & V_A \text{ is } a \text{ copy of } V \\ ||\mathbf{F}(V) - \mathbf{F}(V_A)||_2 > \tau & V_A \text{ and } V \text{ are different} \end{cases}. \tag{28}$$

The threshold is important in a real video fingerprinting system. A smaller threshold can improve the true positive probability but negatively affect the miss probability. By contrast, a larger threshold causes a lower miss probability, but the false alarm probability may be higher as a result. Therefore, the choice of threshold should be considered in a real system. To analyze the choice of this threshold, we first list some symbol notations in Table IV.

We suppose that w and v yield i.i.d. with Gaussian noise because the distances between pairs of fingerprint vectors of video contents are independent of each other. To prove the preceding distribution model assumption, we conducted an experiment in the video database under different modifications. Fig. 8(a) and 8(b) show the bars of v and w, respectively, which indicate that the assumption of normal distribution approximately fits the real data.

According to the model assumptions, we plot the fitting curves in Fig. 9. Obviously, the intersection of these two fitting curves located between 0.3 and 0.4 is a good choice for the threshold. In these experiments, we set $\tau = 0.32$.

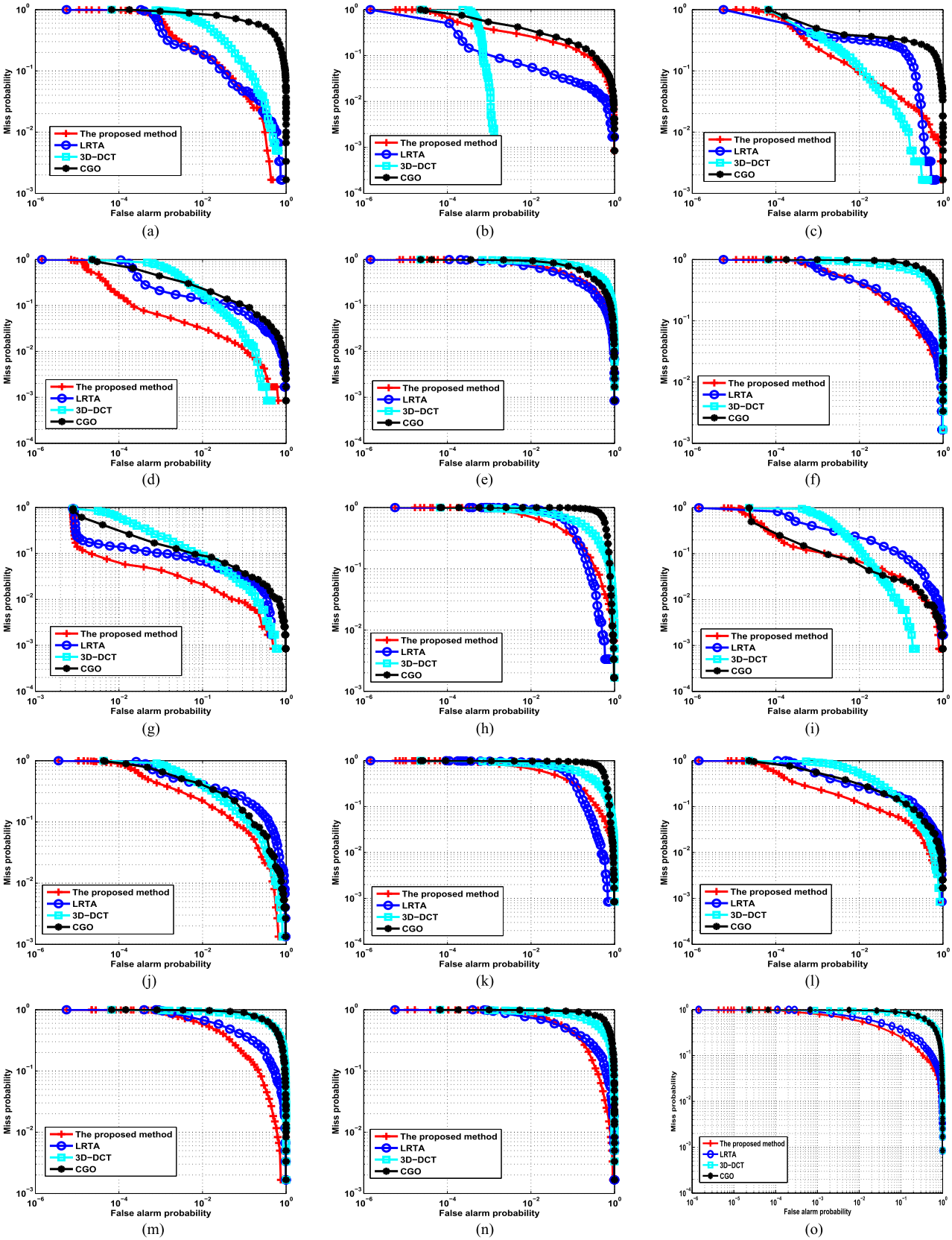

Fig. 6. Performance of the combined database under different modifications : (a) rotation; (b) AWGN; (c) letter-box; (d) caption insertion; (e) cropping; (f) flipping; (g) picture in picture; (h) affine transformation; (i) contrast enhancement; (j) logo substitution; (k) temporal resampling; (l) contrast + change gamma + AGWN; (m) contrast + change of gamma + AGWN + blur + frame dropping; (n) crop + flip + insertion of patterns; and (o) crop + flip + insertion of patterns + picture in picture + shift.

TABLE III
F-SCORE RESULTS

Modifications	Proposed	LRTA	3D-DCT	CGO
Letter-box	**0.9715**	0.9253	0.9708	0.9112
Logo insertion	**0.9476**	0.8908	0.9234	0.9046
Noise	0.9294	0.9820	**0.9989**	0.9083
Caption insertion	**0.9883**	0.9618	0.9627	0.9411
Contrast	**0.9788**	0.9403	0.9731	0.9784
Picture in picture	**0.9786**	0.9612	0.9251	0.9411
Cropping	0.8015	**0.8226**	0.6527	0.7036
Flipping	**0.9063**	0.9051	0.6403	0.5464
Frame changing	**0.8428**	0.8423	0.7148	0.4680
Combined modification 1	**0.9655**	0.9279	0.9097	0.9263
Combined modification 2	**0.8725**	0.8292	0.5745	0.5556

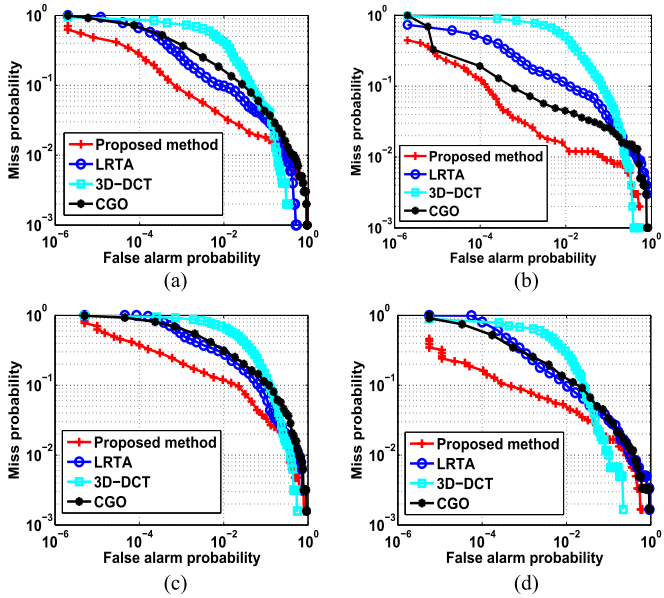

Fig. 7. Performance in the standard database under different modifications: (a) T2 modification; (b) T5 modification; (c) T8 modification; and (d) T10 modification.

TABLE IV
NOTATIONS

Symbol	Defination
w	L2-norm of the fingerprint distance between two fingerprints of visually different video contents
v	L2-norm of the fingerprint distance between two fingerprints of visually similar video content
τ	The threshold
$P(\bullet)$	Distribution of "$\bullet$"

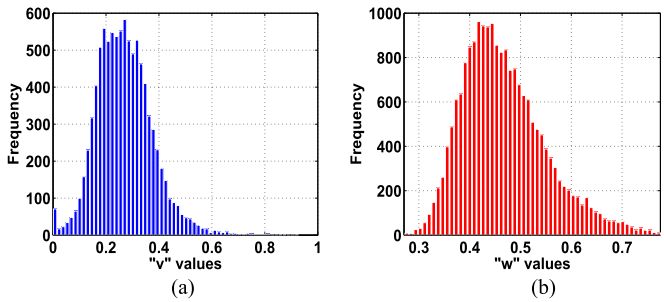

Fig. 8. Description of values w and v: (a) bar of v, and (b) bar of "w".

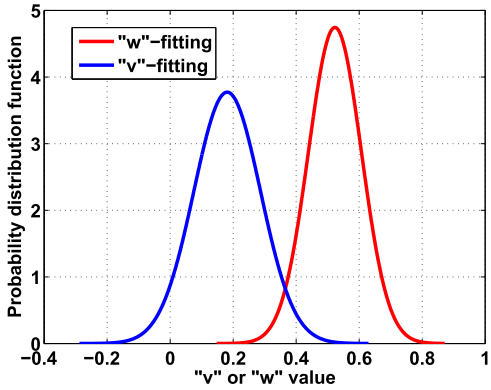

Fig. 9. Illustration of threshold selection.

D. Comprehensive Feature Versus Single and Other Fusion Features

Compared with the single-mode feature, the comprehensive feature is more effective under combined modification in a video fingerprinting system. To show the advantages of the comprehensive feature, we also generated video fingerprints using two main single-mode features (global and local features), which are normalized 64-bin histograms and SURF points, respectively. We than concatenated them to obtain a new feature. We also compared the proposed method with the work in [18], which is a multiple feature fusion-based method. They first construct an affine matrix for each feature, and then minimize the distance of hash codes using the affine matrix in each feature. Finally, they sum all minimized object functions of different features by weights to optimize the final hashes. We conducted the experiments using this idea without the learning process. The comparison of performances between the proposed method and the others under combined modifications (crop + flipping + insertion of patterns) is shown in Fig. 10. Fig. 10(a) and 10(b) show the ROC curves (linear) of the proposed method versus single features, while Fig. 10(c) shows the one between the proposed method and the other feature fusion strategies. The performance of the proposed method is better than that of the other methods. Thus, we can conclude that the comprehensive feature has improved the performance well.

E. Matching Performance

In the proposed scheme, we present a matching strategy based on the video core tensor. In this study, we designed an experiment to evaluate the effectiveness of the proposed strategy. The goal of the pre-matching in the proposed strategy is to narrow down the searching range, and the existing matching method can be applied in further matching. For ease of discussion, we just take the scheme of exhaustive matching as an example to show the improvement of the proposed strategy. In the experiments, we take the time consumed by the exhaustive search as the benchmark in each type of modification. Relative

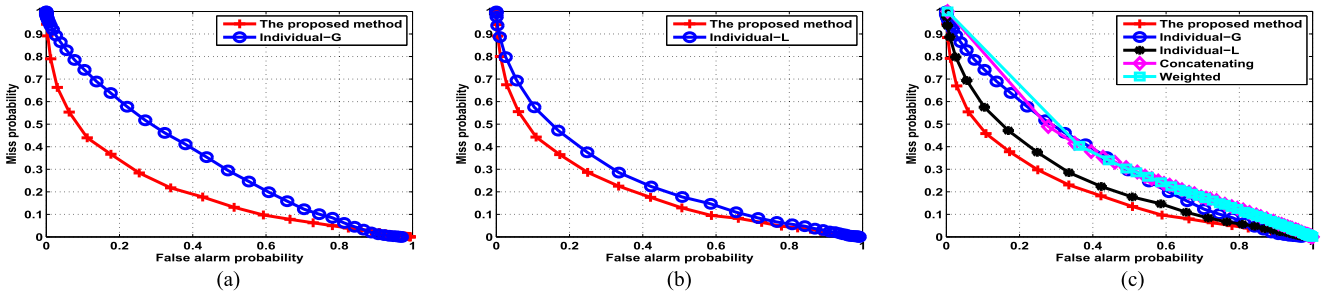

Fig. 10. ROC curve (linear) of comprehensive feature versus the individual feature: (a) the proposed method versus Global feature; (b) the proposed method versus Local feature; (c) the proposed method versus Global/Local/Concatenating feature/Weighted feature.

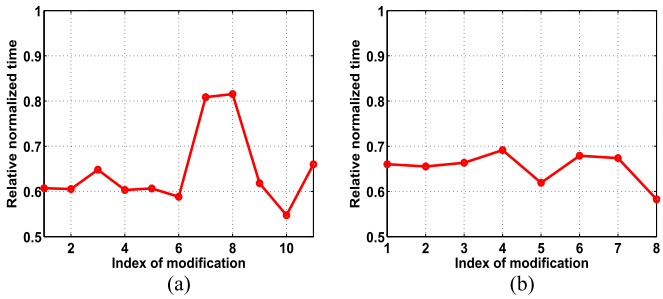

Fig. 11. Performance of matching: (a) combined database and (b) standard database.

normalized time is used for evaluation, which is defined as the time consumed by the proposed strategy divided by the time consumed by exhaustive search. Different modifications were first applied to the video database, and each video in the database was taken as a query. We then tried to find a match in the database with and without the proposed strategy in the exhaustive searching. Fig. 11 shows the relative normalized time of querying one video during matching in the combined and standard database (using a computer with an AMD FX-8300 8-core processor, 8G RAM). The adjustment factor and threshold used in the experiment were 0.3 and 0.32, respectively. It can be seen that the matching time consumed using the proposed strategy ranges from approximately 60% to 70% of the timings of the exhaustive search without the proposed strategy.

IV. Conclusion

In this study, a novel robust video fingerprinting scheme was proposed to make full use of multiple features by the comprehensive feature, which contains the assistance and consensus among different features. Compared with the state-of-the-art methods, the proposed method not only fuses multiple features but also captures the intra- and inter-feature consensus intuitively. The proposed method has good robustness and discrimination, especially under the combined modifications.

In this study, we focus on how to mine comprehensive feature by a tensor model and its robustness. What features need to be selected and how to select features are important topics but are beyond the scope of this study. Moreover, semantic features have been used in some fields of computer vision, such as action recognition [39]. However, the influence of sematic features in video fingerprinting is more complicated because most near-duplicates or video copies differ only in low-level visual features. Even so, the assistance of sematic features should be investigated in the future. Therefore, the input feature selection and how to fuse semantic features with low-level feature in video fingerprinting are important issues that we will investigate in the future.

Acknowledgment

The authors would like to gratefully acknowledge the support of the NVIDIA Corporation with the donation of the TITAN X GPU used for this research.

References

[1] H. Shen, X. Zhou, Z. Huang, J. Shao, and X. Zhou, "UQLIPS: Areal-time near-duplicate video clip detection system," in *Proc. 33rd Int. Conf. Very Large Data Bases*, 2007, pp. 1374–1377.

[2] X. Wu, A. G. Hauptmann, and C. W. Ngo, "Practical elimination of near-duplicates from web video search," in *Proc. ACM Int. Conf. Multimedia*, 2007, pp. 218–227.

[3] A. Hampapur, K. H. Hyun, and R. M. Bolle, "Comparison of sequence matching techniques for video copy detection," in *Proc. SPIE, Storage Retr. Media Databases*, 2002, pp. 194–201.

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

[5] C. Li and F.W.M. Stentiford, "Video sequence matching based on temporal ordinal measurement," *Pattern Recog. Lett.*, vol. 41, no. 29, pp. 1824–1831, 2008.

[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] O. Cirakman, B. Gunsel, N. Serap, and S. Kutluk, "Content-based copy detection by a subspace learning based video fingerprinting scheme," *Multimedia Tools Appl.*, vol. 71, no. 3, pp. 1381–1409, 2014.

[8] Z. K. Wei, Y. Zhao, and C. Zhu, C. Xu, and Z. Zhu, "Frame fusion for video copy detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 1, pp. 15–28, Jan. 2011.

[9] A. Joly, O. Buisson, and C. Frelicot, "Content-based copy detection using distortion-based probabilistic similarity search," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 293–306, Feb. 2007.

[10] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[11] 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.

[12] G. Yang, N. Chen and Q. Jiang, "A robust hashing algorithm based on SURF for video copy detection," *Comput. Secur.*, vol. 31, no. 1, pp. 33–39, 2012.

[13] M. Li and V. Monga, "Compact video fingerprinting via structural graphical models," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 11, pp. 1709–1721, Nov. 2013.
[14] A. S. Divya and D. G. Wiselin, "Video copy detection using spatio-temporal and texture features," *Int. J. Emerging Technol. Advanced Eng.*, vol. 2, pp. 293–296, 2012.
[15] X. C. Liu, J. D. Sun, and J. Liu, "Visual attention based temporally weighting method for video hashing," *IEEE Signal Process. Lett.*, vol. 20, no. 12, pp. 1253–1256, Dec. 2013.
[16] X. S. Nie, J. D. Sun, Z. H. Xing, and X. Liu, "Video fingerprinting based on graph model," *Multimedia Tools Appl.*, vol. 69, no. 2, pp. 429–442, Mar. 2014.
[17] X. S. Nie, J. Liu, J. D. Sun, L. Q. Wang, and X. Yang, "Robust video hashing based on representative-dispersive frames," *Sci. China: Inf. Sci.*, vol. 56, no. 6, pp. 1–11, 2013.
[18] J. K. Song, Y. Yang, Z. Huang, H. T. Shen, and J. Luo, "Effective multiple feature hashing for large-scale near-duplicate video retrieval," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 1997–2008, Dec. 2013.
[19] M. L. Jiang, Y. H. Tian, and T. J. Huang, "Video copy detection using a soft cascade of multimodal features," in *Proc. Int. Conf. Multimedia Expo*. 2012, pp. 374–379.
[20] Y. N. Li *et al.*, "Copy detection with visual-audio feature fusion and sequential pyramid matching," PKU-INM@ TRECVid, 2010. [Online]. Available: http://www-nlpir.nist.gov/projects/tvpubs/tv10.papers/pku-idm-ccd.pdf
[21] L. T. Mou, T. J. Huang, Y. Tian, M. L. Jiang, and W. Gao, "Content-based copy detection through multimodal feature representation and temporal pyramid matching," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 10, no. 1, 2011, Art. no. 5.
[22] F. Wu, Y. N. Liu, and Y. T. Zhuang, "Tensor-based transductive learning for multimodality video semantic concept detection," *IEEE Trans. Multimedia*, vol. 11, no. 5, pp. 868–878, Aug. 2009.
[23] M. Li and V. Monga, "Desynchronization resilient video fingerprinting via randomized low-rank tensor approximations," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Oct. 2011, pp. 1–6.
[24] M. Li and V. Monga, "Robust video hashing via multilinear subspace projections," *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 4397–4409, Oct. 2012.
[25] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3 pp. 455–500, 2009.
[26] B. H. Tuytelaars and T. G. LV, "SURF: Speeded up robust features," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.
[27] E. Ceulemans and H. A. L. Kiers, "Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method," *Brit. J. Math. Stat. Psychology*, vol. 59, no. 5, pp. 113–150, 2006.
[28] M. Mørup and L. K. Hansen, "Automatic relevance determination for multi-way models," in *J. Chemometr. Special Issue: Honor Professor Richard A. Harshman*, vol. 23, no. 7/8, pp. 352–363, 2009.
[29] M. Mørup, "Decomposition methods for unsupervised learning," Ph.D. dissertation, Tech. Univ. Denmark, Kongens Lyngby, Denmark, 2008.
[30] M. Morten, "Applications of tensor (multiway array) factorizations and decompositions in data mining," *Wiley Interdisciplinary Rev.: Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 24–40, 2011.
[31] G. Awad, P. Over, and W. Kraaij, "Content-based video copy detection benchmarking at TRECVID," *ACM Trans. Inf. Syst.*, vol. 32, no. 3, 2014, Art. no. 14.
[32] M. M. Esmaeili, M. Fatourechi, and R. K. Ward, "A robust and fast video copy detection system using content-based fingerprinting," *IEEE Trans. Inf. Forensics Secur.*, vol. 6, no. 1, pp. 213–226, Mar. 2011.
[33] F. Comon, X. Luciani, and A. L. F. de Almeida, "Tensor decompositions alternating least square and other tales," *J. Chemometrics*, vol. 23, no. 7, pp. 393–405, 2009.
[34] B. De Lathauwer, D. Moor, and J. Vandewalle, "On the best rank-1 and rank-(R1,R2,..., RN) approximation of higher-order tensors," *SIAM J. Matrix Anal. Appl.*, vol. 21, pp. 1324–1342, 2000.
[35] P. M. Kroonenberg and J. De Leeuw, "Principal component analysis of three-mode data by means of alternating least squares algorithms," *Psychometrika*, vol. 45, pp. 69–97, 1980.
[36] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *CoRR*, 2013. [Online]. Available: http://arxiv.org/abs/1304.5634
[37] S. Dasgupta, M. L. Littman, and D. McAllester, "PAC generalization bounds for co-training," *Advances Neural Inf. Process. Syst.*, vol. 1, pp. 375–382, 2002.
[38] L. K. Hansen, K. H. Madsen, and S. T. Lehn, "Adaptive regularization of noisy linear inverse problems," in *Proc. 14th Eur. Signal Process. Conf.*, 2006, pp. 1–5.
[39] J. Cai, M. Merler, S. Pankani, and Q. Tian, "Heterogeneous semantic level features fusion for action recognition," in *Proc. ACM Int. Conf. Multimedia Retr.*, 2015, pp. 307–314.
[40] J. Cai, R. Hong, M. Wang, and Qi Tian, "Exploring feature space with semantic attributes," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jun.-Jul. 2015, pp. 1–6.
[41] "CBCD evaluation plan TRECVID," 2010. [Online]. Available: http://www-nlpir.nist.gov/projects/tv2010/Evaluation-cbcd-v1.3.htm#eval

Xiushan Nie (M'12) received the Ph.D. degree from Shandong University, Shandong, China, in 2011.

He is currently a Professor with Shandong University of Finance and Economics, Shandong, China. From 2013 to 2014, he was a Visiting Scholar with the University of Missouri-Columbia, Columbia, MO, USA. His research interests include multimedia retrieval and indexing, multimedia security, and computer vision.

Yilong Yin received the Ph.D. degree from Jilin University, Changchun, China, in 2000.

He is currently the Director of the Machine Learning and Applications Group and a Professor with Shandong University, Jinan, China. From 2000 to 2002, he was a Postdoctoral Fellow with the Department of Electronic Science and Engineering, Nanjing University, Nanjing, China. His research interests include machine learning, data mining, computational medicine, and biometrics.

Jiande Sun (M'06) received the Ph.D. degree in information and communication engineering from Shandong University, Shandong, China, in 2005.

He is currently an Associate Professor with the School of Information Science and Engineering, Shandong Normal University, Shandong, China, and a Research Fellow with the Institute of Data Science and Technology, Shandong Normal University. His research interests include multimedia information mask and digital watermark, and content-based multimedia analysis.

Ju Liu (M'01–SM'09) received the B.S. and M.S. degrees in electronic engineering from Shandong University (SDU), Shandong, China, in 1986 and 1989, respectively, and the Ph.D. degree in signal processing from the Southeast University, Nanjing, China, in 2000.

Since July 1989, he has been a Professor with the Department of Electronic Engineering, SDU. His current research interests include space-time processing in wireless communication, blind signal separation, and multimedia communications.

Chaoran Cui received the B.S. degree in software engineering and the Ph.D. degree from Shandong University, Shandong, China, in 2010 and 2015, respectively.

He was a Research Fellow with Singapore Management University, Singapore, from 2015 to 2016. He is currently a Professor with the School of Computer Science and Technology, Shandong University of Finance and Economics, Shandong, China. His research interests include information retrieval, analysis and understanding on multimedia information, and computer vision.