Hindawi
Security and Communication Networks
Volume 2021, Article ID 9930673, 11 pages
https://doi.org/10.1155/2021/9930673

Research Article

Robust Video Hashing Based on Multidimensional Scaling and Ordinal Measures

Zhenjun Tang, Shaopeng Zhang, Zhenhai Chen, and Xianquan Zhang

Guangxi Key Lab of Multi-Source Information Mining & Security, and Department of Computer Science, Guangxi Normal University, Guilin 541004, China

Correspondence should be addressed to Zhenjun Tang; tangzj230@163.com

Received 10 March 2021; Revised 11 April 2021; Accepted 17 April 2021; Published 3 May 2021

Academic Editor: Zhili Zhou

Multimedia hashing is a useful technology of multimedia management, e.g., multimedia search and multimedia security. This paper proposes a robust multimedia hashing for processing videos. The proposed video hashing constructs a high-dimensional matrix via gradient features in the discrete wavelet transform (DWT) domain of preprocessed video, learns low-dimensional features from high-dimensional matrix via multidimensional scaling, and calculates video hash by ordinal measures of the learned low-dimensional features. Extensive experiments on 8300 videos are performed to examine the proposed video hashing. Performance comparisons reveal that the proposed scheme is better than several state-of-the-art schemes in balancing the performances of robustness and discrimination.

1. Introduction

In the digital era, multimedia (e.g., image and video) is easily captured via smart devices, such as smart phone and iPad. Many people would like to use multimedia to record their lives and share them with friends on the Internet. Consequently, there are a large number of multimedia data in the cloud server. Efficient technologies of multimedia management, e.g., multimedia search and multimedia security [1–3], are thus in demand. To address these issues, robust multimedia hashing, such as audio hashing [4, 5], image hashing [6, 7], and video hashing [8, 9], are proposed to process different kinds of multimedia data in recent years. Robust multimedia hashing can map multimedia data to a content-based short sequence called hash and finds a lot of applications [10–14], such as copy detection, authentication, tampering detection, forensics, and retrieval. In this work, we propose a novel multimedia hashing based on multidimensional scaling (MDS) and ordinal measures for processing videos.

Generally, multimedia hashing for videos should identify visually similar videos which are generated by manipulating videos with normal digital operations, such as compression and filtering. This is the property of video hashing called robustness. As there are many different videos in practical applications, video hashing should meet the property called discrimination. This property can ensure that video hashing can efficiently distinguish different videos from massive videos. Note that the discrimination and robustness are two basic properties of video hashing. For specific applications, video hashing should satisfy additional property. For example, it should be key-dependent for video authentication and forensics.

Many scholars have designed diverse video hashing schemes in the past years. As discrete cosine transform (DCT) has been widely used in some compression techniques, such as JPEG compression and MPEG-2 compression, it is extensively investigated in video hashing design. A well-known robust video hashing was introduced by De Roover et al. [15]. They computed key frames from video sequence and derived hash vector from every frame by compressed radial luminance projections with DCT. This key frame-based scheme can resist slight geometric deformation and temporal subsampling, but it is time-consuming due to high computational cost of radial luminance projection. Coskun et al. [16] investigated the use of DCT in

video hashing and presented two effective hashing schemes by using the classical basis set and a randomized basis set. Their schemes can both withstand blurring and MPEG-4 compression. In another study, Li [17] calculated random pixel cubes, applied 3D DCT to these cubes, and exploited energy relationships of DCT coefficients to make video hash. This scheme is secure and robust to MPEG-2 compression. Mao et al. [18] jointly exploited 3D DCT and the classical locality-sensitive hashing (LSH) to design video hashing for copy detection. This DCT-LSH scheme has a high precision rate. Esmaeili et al. [19] generated temporally informative representative image (TIRI) from video frames by calculating a weighted sum of frames, applied 2D DCT to overlapping blocks of every TIRI and selected the first vertical and the first horizontal coefficients to construct hash. The TIRI-DCT scheme is robust to noise and frame loss. Setyawan and Timotius [20] calculated video hash by using edge orientation histogram (EOH) and DCT. The EOH-DCT scheme is robust against luminance modification and MPEG compression.

Besides DCT, other useful techniques are also used in video hashing. For example, Mucedero et al. [21] calculated a standard video by filtering and resampling input video and extracted robust hash by computing the minimum block values from the matrix of block-based pixel variances. This scheme can identify those videos compressed by the technique of MPEG-2 or MPEG-4. Xiang et al. [22] exploited Gaussian filtering and luminance histogram to design video hashing. Their scheme can resist geometric attacks. Sun et al. [23] jointly used the TIRI [19] and visual attention model to make a novel video hashing scheme with weighted matching. This scheme demonstrates good performances in terms of recall and precision rates. Li and Monga [24] viewed videos as tensors and exploited the subspace projections of tensors, i.e., low-rank tensor approximations (LRTA), to calculate video hash. The LRTA hashing is resilient to blurring, compression, and rotation. In another work, Li and Monga [25] proposed to represent videos by graphs and used structural graphical models to derive video hash. This scheme can generate a compact hash without losing detection performance. As TIRI based video hashing schemes receive much attentions, Liu et al. [26] exploited dynamic and static attention models to develop a novel temporally weighting method for TIRI generation. The method helps to improve hash performance.

Recently, motivated by the ring partition reported in [27], Nie et al. [28] exploited spherical torus (ST) to conduct video partition and used nonnegative matrix factorization (NMF) to extract hash. The ST-NMF hashing is robust to noise and blurring. Sun et al. [29] extracted attention features via a visual attention model and combined them with visual-appearance features via a deep belief network to generate hash. This hashing scheme can resist Gaussian noise, Gaussian blurring, and median filtering. Rameshnath and Bora [30] utilized temporal wavelet transform (TWT) to generate TIRIs and conducted random projection with the Achlioptas's Random Matrix (ARM). The TWT-ARM hashing shows good robustness against MPEG-4 compression, watermark insertion, and Gaussian blurring. In another work, Tang et al. [31] used DCT to construct feature matrices and learned video hash from the matrices via NMF. This hashing is resistant to frame scaling and frame rotation with small angle. Chen et al. [32] used low-rank and spare decomposition (LRSD) to calculate feature matrix of each frame and exploited 2D DWT to perform feature compression. The LRSD-DWT hashing can resist MPEG-4 compression, Gaussian low-pass filtering, and frame rotation with small angle.

From the above survey, it can be found that many reported video hashing schemes can make good robustness against some digital operations. But they do not reach a desirable balance between the performances of robustness and discrimination yet. To address this issue, we jointly exploit DWT, gradient information, MDS, and ordinal measures to develop a novel video hashing, which can make a good balance between the two performances. Compared with the existing video hashing schemes, the main contributions of the proposed video hashing are presented as follows.

(1) High-dimensional matrix is constructed by using gradient features in the DWT domain of preprocessed video. Since gradient information can measure structural image features which are almost kept unchanged after digital operations, gradient features can effectively capture visual content of video frame. Therefore, the gradient features based high-dimensional matrix can guarantee good robustness against digital operations and distinguish videos with different contents.

(2) Low-dimensional features are learned from the high-dimensional matrix via MDS. The MDS is an efficient technique of data dimensionality reduction. It can effectively learn discriminative low-dimensional features from the high-dimensional data by preserving original relationship of the data. So the learned low-dimensional features can make discriminative and compact video hash.

(3) Video hash is generated by using ordinal measures of the low-dimensional features. The ordinal measures are robust and discriminative features, and their elements are all integers. Therefore, the use of the ordinal measures can contribute to a robust and discriminative video hash with short length.

Extensive experiments are performed to test the proposed video hashing. The results reveal that the proposed video hashing has a good robustness and desirable discrimination. Comparisons with several state-of-the-art hashing schemes demonstrate that the proposed video hashing is better than the compared schemes in balancing the performances of robustness and discrimination. The rest of the paper is structured as follows. The proposed video hashing is explained in Section 2. The experimental results and comparisons are discussed in Section 3 and Section 4, respectively. Conclusions are presented in Section 5.

2. Proposed Video Hashing

The proposed video hashing can be decomposed into four components. Figure 1 shows the four components of the

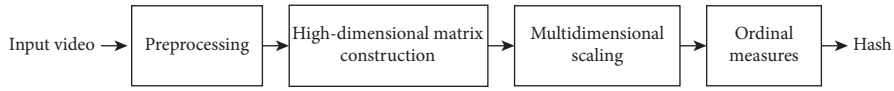

Figure 1: Components of the proposed scheme.

proposed scheme. The first component is the preprocessing which is used to make a normalized video. The second component is the high-dimensional matrix construction by using gradient features in the DWT domain. The third component is to learn low-dimensional features from the high-dimensional matrix via MDS. The final component is to calculate a compact hash by using ordinal measures of the learned low-dimensional features. The details of these components are introduced below.

2.1. Preprocessing. Temporal-spatial resampling is applied to the input video. Specifically, the temporal resampling is firstly used to map different videos to the same frame number. To do so, the pixels in the same positions of all frames are orderly picked to form a pixel tube. Every pixel tube is then mapped to a fixed length M by the linear interpolation. After this operation, a video with M frames is generated. Next, the spatial resampling is used to convert the frame resolution to a standard size $N \times N$ by bicubic interpolation. Consequently, a video with M frames sized $N \times N$ is generated after the temporal-spatial resampling.

If the input video is an RGB color video, the resized video is converted to the well-known color space called HSI space, and the intensity color component "I" in the HSI space is selected to denote the resized video. The HSI space is generally described by a conical color space. Conversion from the RGB space to the HSI space can be implemented by the following rules.

$$I = \frac{1}{3}(R + G + B), \tag{1}$$

$$S = 1 - \frac{3 \times \min(R, G, B)}{(R + G + B)}, \tag{2}$$

$$H = \begin{cases} \theta, & \text{if } B \le G, \\ 360 - \theta, & \text{if } B > G, \end{cases} \tag{3}$$

$$\theta = \cos^{-1}\left(\frac{[(R - G) + (R - B)]}{2\left[(R - G)^2 + (R - B)(G - B)\right]^{1/2}}\right), \tag{4}$$

in which R is the red channel of pixel, G is the green channel, B is the blue channel, I is the intensity component, S is the saturation component, and H is the hue component.

2.2. High-Dimensional Matrix Construction. Structural features are important image features, which can effectively describe the visual content of video frame and are almost kept unchanged after digital operations. Since image gradient [33] can measure structural features, gradient features are used to construct a high-dimensional matrix. To extract local video features, the intensity component of the resized video is divided into m groups of video frames. For each group, a secondary frame is firstly calculated. For simplicity, let M be an integer multiple of m. Thus, there are $b = M/m$ frames in every video group. Let $Q_l(i, j, k)$ be the pixel in the i-th row and the j-th column of the k-th frame in the l-th video group and $\mathbf{F}_l$ be the secondary frame of the l-th video group, whose element in the i-th row and the j-th column is denoted by $F_l(i, j)$. Therefore, the secondary frame $\mathbf{F}_l$ can be determined by

$$F_l(i, j) = \frac{1}{b}\sum_{k=1}^{b} Q_l(i, j, k). \tag{5}$$

After these operations, there are m secondary frames in total.

Next, one-level 2D DWT is applied to each secondary frame. Note that four sub-bands are obtained after decomposition, i.e., LL sub-band, LH sub-band, HL sub-band, and HH sub-band. As DWT coefficients in the LL sub-band contain approximation information of secondary frame, the LL sub-band is used to construct high-dimensional matrix. Moreover, the DWT coefficients in the LL sub-band are slightly influenced by compression and noise. Consequently, features extracted from LL sub-band can make a robust high-dimensional matrix. Suppose that the size of the LL sub-band of one-level 2D DWT is $s \times s$. Therefore, $s = \lceil N/2 \rceil$, where $\lceil . \rceil$ is the upward rounding operation.

Let $f(x, y, l)$ be the element in the coordinates (x, y) of the LL sub-band of the secondary frame of the l-th video group. Thus, its gradient can be determined by the vector defined in

$$\nabla f = \begin{bmatrix} G_x & G_y \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^{\mathrm{T}}, \tag{6}$$

where G_x and G_y are the partial derivatives approximately determined as follows:

$$G_x = f(x + 1, y, l) - f(x - 1, y, l), \tag{7}$$

$$G_y = f(x, y + 1, l) - f(x, y - 1, l), \tag{8}$$

In general, gradient feature can be denoted by its magnitude $r(x, y, l)$ or its orientation $\phi(x, y, l)$, whose definitions can be found in equation (9) and equation (10), respectively:

$$r(x, y, l) = \sqrt{G_x^2 + G_y^2}, \tag{9}$$

$$\phi(x, y, l) = \tan^{-1}\left(\frac{G_y}{G_x}\right). \tag{10}$$

As the orientation of image structure is changed after rotation, the gradient magnitude is selected as the feature instead of the orientation.

After the calculation of gradient magnitude, a gradient feature matrix sized $s \times s$ is obtained. To extract local gradient feature, the matrix of gradient magnitudes is divided into nonoverlapping blocks sized $t \times t$. For simplicity, let s be an integer multiple of t. Thus, there are both $n = s/t$ blocks in the horizontal direction and the vertical direction. Let $\mathbf{B}_{i,j,l}$ be the block of the gradient feature matrix of the l-th secondary frame, where $1 \le i \le n$ and $1 \le j \le n$. Thus, the mean of the block $\mathbf{B}_{i,j,l}$ can be calculated by the following equation:

$$c_{i,j,l} = \frac{1}{t^2} \sum_{k=1}^{t^2} B_{i,j,l}(k), \tag{11}$$

in which $B_{i,j,l}(k)$ is the k-th element of $\mathbf{B}_{i,j,l}$. Therefore, a gradient feature sequence of the secondary frame $\mathbf{F}_l$ can be generated by arranging these means as follows:

$$\mathbf{c}_l = \left[c_{1,1,l}, c_{1,2,l}, \ldots, c_{1,n,l}, c_{2,1,l}, c_{2,2,l}, \ldots, c_{2,n,l}, \ldots, c_{n,1,l}, c_{n,2,l}, \ldots, c_{n,n,l}\right]. \tag{12}$$

Finally, a high-dimensional matrix can be constructed by stacking these m feature sequences as follows:

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_m \end{bmatrix}. \tag{13}$$

Note that the size of the high-dimensional feature matrix is $m \times q$, where $q = n^2$.

2.3. Multidimensional Scaling. In order to find low-dimensional data from the high-dimensional feature matrix $\mathbf{C}$, a well-known technique of data dimension reduction called MDS is exploited in this work. The reasons of selecting MDS to learn low-dimensional data are as follows. (1) MDS has shown good performances in many applications, including object retrieval [34], localization [35], data visualization [36], and image hashing [37]. (2) MDS can effectively learn discriminative low-dimensional features from high-dimensional data by preserving original relationship of the data. In general, the classical MDS consists of three steps [38], namely, distance matrix computation, inner product matrix calculation, and low-dimensional matrix extraction, which are described as follows.

(1) *Distance Matrix Computation.* For each row $\mathbf{c}_i$ $(1 \le i \le m)$, the Euclidean distance $d_{i,j}$ between $\mathbf{c}_i$ and $\mathbf{c}_j$ $(1 \le j \le m)$ is computed. Thus, the distance matrix $\mathbf{D} = (d_{i,j})_{m \times m}$ is available.

$$\mathbf{D} = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,m} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,m} \\ \cdots & \cdots & \cdots & \cdots \\ d_{m,1} & d_{m,2} & \ldots & d_{m,m} \end{bmatrix}, \tag{14}$$

where $d_{i,j}$ is calculated as follows:

$$d_{i,j} = \sum_{k=1}^{q} \left[c_i(k) - c_j(k)\right]^2, \tag{15}$$

in which $c_i(k)$ and $c_j(k)$ are the k-th elements of $\mathbf{c}_i$ and $\mathbf{c}_j$ $(1 \le k \le q)$, respectively.

(2) *Inner Product Matrix Calculation.* The distance matrix $\mathbf{D}$ is transformed into the inner product matrix $\mathbf{T}$ by the following equation:

$$\mathbf{T} = -\frac{1}{2}\mathbf{PDP}, \tag{16}$$

in which $\mathbf{P}$ is a centralizable matrix determined by

$$\mathbf{P} = \mathbf{E} - \frac{1}{m}\mathbf{e}\mathbf{e}^{\mathrm{T}}, \tag{17}$$

where $\mathbf{E}$ is a unit matrix sized $m \times m$ and $\mathbf{e}$ denotes a unit vector sized $m \times 1$. Thus, $\mathbf{Pe} = 0$ and $\mathbf{P}^{\mathrm{T}} = \mathbf{P}$.

(3) *Low-Dimensional Matrix Extraction.* Since $\mathbf{T}$ is symmetric and semipositive definite, it can be decomposed as follows:

$$\mathbf{T} = \mathbf{RAR}^{\mathrm{T}}, \tag{18}$$

in which $\mathbf{R}$ is an orthogonal matrix of eigenvectors and $\mathbf{A}$ is the diagonalized matrix of eigenvalues of $\mathbf{T}$. Therefore, a low-dimensional matrix $\mathbf{U}$ can be obtained by selecting the first p columns of $\mathbf{X}$.

$$\mathbf{X} = \mathbf{RA}^{(1/2)}. \tag{19}$$

The size of the low-dimensional matrix $\mathbf{U}$ is $m \times p$ $(p < q)$, and p is the selected dimension of MDS.

To generate a short and discriminative video feature sequence, the variance of each row of the low-dimensional matrix $\mathbf{U}$ is calculated. Suppose that $\mathbf{u}_i = [u_i(1), u_i(2), \ldots, u_i(p)]$ is the i-th row of the matrix $\mathbf{U}$ $(1 \le i \le m)$. Thus, its variance v_i can be determined by

$$v_i = \frac{1}{p-1} \sum_{j=1}^{p} \left[u_i(j) - \mu_i\right]^2. \tag{20}$$

where μ_i is the mean of $\mathbf{u}_i$ defined by

$$\mu_i = \frac{1}{p} \sum_{j=1}^{p} u_i(j). \tag{21}$$

After the variance calculation, a video feature sequence is obtained as follows:

$$\mathbf{v} = \left[v_1, v_2, \ldots, v_m\right]. \tag{22}$$

Note that all elements of the feature sequence are floating-point numbers.

2.4. Ordinal Measures. In practice, storage of a floating-point number requires many bits in a computer system. For example, 32 bits are needed in terms of the IEEE standard [39]. Therefore, the length of the feature sequence is $32m$ bits, which is a large number when the m value is big. To reduce the cost of the feature sequence, a technique called ordinal measures (OM) [40] is used to conduct quantization. The OM technique can extract short and robust feature codes and has been used in many applications of image and video processing [41–45]. Generally, the feature codes of the OM technique can be calculated by sorting a data sequence

in ascending order and selecting the positions in the sorted sequence. To better understand the OM technique, an example is presented in Table 1. In this table, eight numbers of an original data sequence are listed in the second row, their sorted versions in ascending order are presented in the third row, and their feature codes of the OM technique are shown in the fourth row. For example, the 1st element of the original data sequence is 20, which is ranked at the 6th position in the sorted sequence. Therefore, its feature code of the OM technique is 6. The OM codes of other elements can be determined by similar calculation.

Here, the OM codes of the elements of the feature sequence $\mathbf{v}$ are selected as the elements of our video hash. Suppose that h_i is the OM code of v_i $(1 \le i \le m)$. Thus, the video hash can be denoted by $\mathbf{h}$ as follows:

$$\mathbf{h} = [h_1, h_2, \ldots, h_m]. \tag{23}$$

Therefore, the length of the video hash is m integers. Since each integer needs $\lceil \log_2 m \rceil$ bits at most, the hash length is $m\lceil \log_2 m \rceil$ bits, where $\lceil \cdot \rceil$ is upward rounding operation.

3. Experimental Results

To measure hash similarity, the distance metric called L2 norm is taken in the experiments. Let $\mathbf{h}^{(1)} = [h_1^{(1)}, h_2^{(1)}, \ldots, h_m^{(1)}]$ and $\mathbf{h}^{(2)} = [h_1^{(2)}, h_2^{(2)}, \ldots, h_m^{(2)}]$ be two video hashes. So the L2 norm can be calculated by

$$D\left(\mathbf{h}^{(1)}, \mathbf{h}^{(2)}\right) = \sqrt{\sum_{i=1}^{m} \left[h_i^{(1)} - h_i^{(2)}\right]^2}, \tag{24}$$

in which $h_i^{(1)}$ and $h_i^{(2)}$ are the i-th element of $\mathbf{h}^{(1)}$ and $\mathbf{h}^{(2)}$, respectively. In general, similar videos are expected to have similar video hashes and their L2 norm should be small. On the contrary, different videos should have different hashes and the corresponding L2 norm should be large. Therefore, two videos are considered as similar videos if the L2 norm of their hashes is smaller than a threshold. Otherwise, they are identified as different videos.In the following experiments, the used parameters of the proposed hashing scheme are set as follows. The input video is converted to $256 \times 256 \times 256$, the number of video groups is 32, the block size is 16×16, and the selected dimension of MDS is 30. In other words, $M = 256$, $N = 256$, $m = 32$, $s = \lceil N/2 \rceil = 128$, $t = 16$, and $p = 30$. Therefore, the hash length of video hash is 160 bits. The robustness analysis, discriminative performance, dimension discussion, and group number selection are presented in the sections below.

3.1. Robustness Analysis. To examine robustness of the proposed hashing scheme, 100 different videos are selected from an open video database [46]. These videos are taken from different topics, including "algae," "anemones," "ascidians," "bioerosion," "black coral," "bryozoans," "caves," "cleaning station," "crustaceans," "hurricane impacts," "jellyfish," "plankton," and "seagrass." Their frame resolutions range from 360×288 to 512×288. Some typical samples are shown in Figure 2. To produce similar videos of these 100 videos, eleven video operations are utilized to perform robustness attacks. For each operation, different parameters are selected. The used operations include brightness adjustment (8 parameters), contrast adjustment (4 parameters), 3×3 Gaussian low-pass filtering (10 parameters), salt and pepper noise (10 parameters), additive white Gaussian noise (AWGN) (6 parameters), MPEG-2 compression (10 parameters), MPEG-4 compression (10 parameters), random frame dropping (6 parameters), random frame insertion (6 parameters), frame scaling (8 parameters), and frame rotation (4 parameters). Table 2 presents the settings of the eleven operations. After robustness attacks, every original video has 82 similar videos. Therefore, the number of similar videos reaches $100 \times 82 = 8200$ and the number of total videos used in the experiment is $8200 + 100 = 8300$.

Table 1: An example of the OM technique.

Position	1	2	3	4	5	6	7	8
Original sequence	20	18	22	10	7	15	25	12
Sorted sequence	7	10	12	15	18	20	22	25
Ordinal measures	6	5	7	2	1	4	8	3

Hash distances under different kinds of operations are calculated. Figure 3 illustrates the mean L2 norm of different operations under specific parameter settings, where the x-axis represents the parameter value of the corresponding operation and the y-axis is the mean L2 norm. Table 3 presents the statistical results of hash distances. From these results, it can be seen that the mean distances of all video operations are smaller than 30 and most maximum distances are also smaller than 60. This means that 60 can be selected as a candidate threshold. In this case, the proposed hashing scheme can correctly identify 99.27% similar videos.

3.2. Discriminative Performance. The dataset with 8300 videos mentioned in Section 3.1 is exploited to analyze discriminative performance of the proposed hashing scheme. Specifically, for every original video, the distances between its video hash and the hashes of other $99 \times 82 = 8118$ attacked videos are calculated. Note that every original video is only compared with the attacked videos of other 99 different videos. In other words, every original video is not compared with its attacked videos in the experiment. Thus, there are $100 \times 8118 = 811800$ L2 norms in total. Figure 4 shows the L2 norm distribution of different video hashes, where the x-axis is the L2 norm and the y-axis is the frequency of the corresponding L2 norm. It can be observed that the mean distance is 68.55, and most L2 norms are bigger than 40. If the threshold is selected as 40, 0.69% different videos are falsely detected as similar videos. If the threshold decreases to 30, the false detection rate of different videos is 0.07%. Note that a low threshold helps to improve discriminative performance, but it will also decrease robustness performance. Table 4 presents the detailed detection rates under different thresholds. In this table, the robustness performance is measured by the correct detection rate of similar videos and the discriminative performance is indicated by the false detection rate of different

Figure 2: Some sample videos.

Table 2: Settings of eleven operations.

Operation	Parameter	Value	Number
Brightness adjustment	Photoshop's scale	−20, −15,−10, −5, 5, 10,15, 20	8
Contrast adjustment	Photoshop's scale	−20, −10, 10, 20	4
3 × 3 Gaussian low-pass filtering	Standard deviation	0.1, 0.2, . . ., 1	10
Salt and pepper noise	Density	0.001, 0.002, . . ., 0.01	10
AWGN	Signal noise ratio	1, 2, 3, 4, 5, 6	6
MPEG-2 compression	Kilobit per second	100,200, . . ., 1000	10
MPEG-4 compression	Compression quality	10,20, . . ., 100	10
Random frame dropping	Frame number	1, 2, 5, 10, 15, 20	6
Random frame insertion	Frame number	1, 2, 5, 10, 15, 20	6
Frame scaling	Ratio	0.8, 0.85, 0.9, . . ., 1.2	8
Frame rotation	Angle (degree)	−1, −0.5, 0.5, 1	4
Total			82

videos. In practice, a suitable threshold can be chosen from Table 4 according to the performance requirement of the specific application.

3.3. Dimension Discussion. The selected dimension p is the only parameter of MDS. To view effect of the selected dimension on the performances of robustness and discrimination, the Receiver Operating Characteristic (ROC) graph [47] is utilized to analyze experimental results. In the graph, false positive rate (FPR) and true positive rate (TPR) are both calculated under the control of a given threshold. Therefore, some points with coordinates (FPR, TPR) can be generated by using a set of thresholds. Consequently, the ROC curve is obtained by orderly connecting these points. Detailed definitions of FPR and TPR are found in

$$\text{FPR} = \frac{\text{number of different videos detected as similar ones}}{\text{number of different videos}}, \tag{25}$$

$$\text{TPR} = \frac{\text{number of similar videos detected as similar ones}}{\text{number of similar videos}}. \tag{26}$$

In practice, the area under the ROC curve (AUC) is calculated to make quantitative comparison. The minimum AUC is 0 and the maximum AUC is 1. A curve with large AUC outperforms the curve with small AUC.

In this experiment, the used parameters are $p = 10$, $p = 20$, $p = 30$, $p = 40$, and $p = 50$. The curves of different p values are shown in Figure 5, and the AUCs of different p values are then calculated. It is found that the AUC of $p = 10$ is 0.99371, the AUC of $p = 20$ is 0.99490, the AUC of $p = 30$ is 0.99508, the AUC of $p = 40$ is 0.99508, and the AUC of $p = 50$ is 0.99508. The proposed hashing scheme reaches the biggest AUC when $p = 30$. This means that the proposed hashing scheme with $p = 30$ is better than the proposed hashing scheme with other p values in terms of the performances of discrimination and robustness.

3.4. Group Number Selection. Effect of the group number on the performances of robustness and discrimination is also discussed. The selected group numbers are $m = 8$, $m = 16$, $m = 32$, and $m = 64$. Similarly, the ROC curves of group numbers are also calculated to make visual comparison. The ROC results are shown in Figure 6. It is found that the AUC of $m = 8$ is 0.96944, the AUC of $m = 16$ is 0.99468, the AUC of $m = 32$ is 0.99508, and the AUC of $m = 64$ is 0.99419. The proposed hashing scheme reaches the biggest AUC when $m = 32$. This means that the proposed hashing scheme with $m = 32$ is better than the proposed hashing scheme with

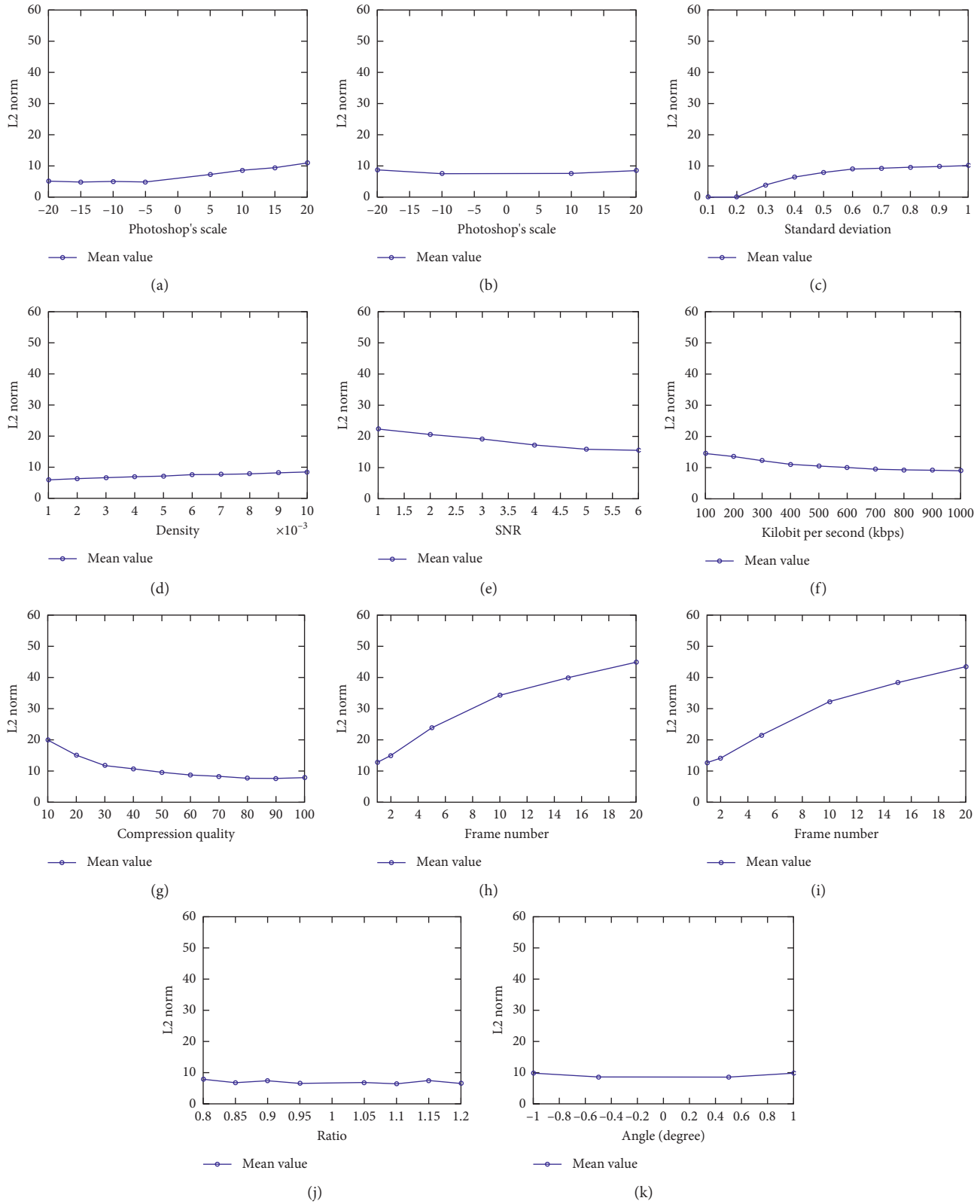

Figure 3: Means of L2 norms under different operations. (a) Brightness adjustment. (b) Contrast adjustment. (c) 3×3 Gaussian low-pass filtering. (d) Salt and pepper noise. (e) AWGN. (f) MPEG-2 compression. (g) MPEG-4 compression. (h) Random frame dropping. (i) Random frame insertion. (j) Frame scaling. (k) Frame rotation.

Table 3: Statistical results of hash distances.

Operation	Max	Min	Mean	Standard deviation
Brightness adjustment	29.39	0.00	7.01	0.58
Contrast adjustment	26.57	0.00	8.08	0.21
3×3 Gaussian low-pass filtering	21.73	0.00	6.60	1.41
Salt and pepper noise	55.75	0.00	7.28	0.85
AWGN	80.42	4.00	18.46	0.75
MPEG-2 compression	57.34	2.00	10.89	1.61
MPEG-4 compression	87.90	1.41	10.73	3.68
Random frame dropping	73.93	4.69	28.45	3.42
Random frame insertion	74.66	4.00	27.07	3.22
Frame scaling	20.00	0.00	6.99	0.25
Frame rotation	23.15	2.45	9.21	0.39

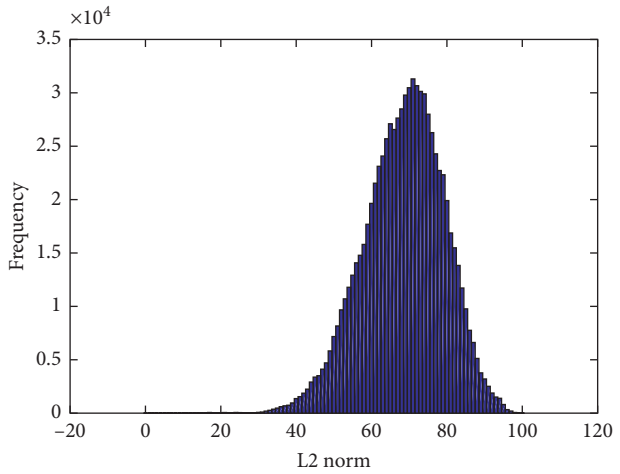

Figure 4: L2 norm distribution of different video hashes.

Table 4: Detection rates under different thresholds.

Threshold	Correct detection rate of similar videos (%)	False detection rate of different videos (%)
80	99.98	86.14
70	99.82	52.70
60	99.27	20.79
50	98.01	4.93
40	95.96	0.69
30	92.79	0.07

other m values in terms of the performances of discrimination and robustness.

4. Performance Comparisons

To illustrate superiority of the proposed hashing scheme, it is compared with some state-of-the-art hashing schemes. The selected hashing schemes are EOH-DCT hashing scheme [20], ST-NMF hashing scheme [28], TWT-ARM hashing scheme [30], and LRSD-DWT hashing scheme [32]. All these selected schemes are designed for videos and recently reported in the literature. In the comparisons, the videos used in Section 3.1 and Section 3.2 are exploited to test

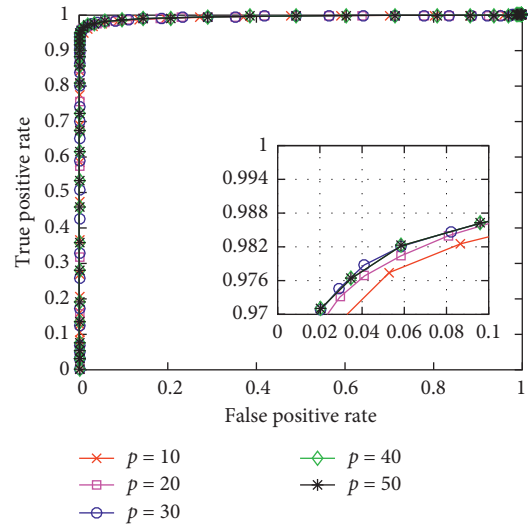

Figure 5: Curves of different dimensions.

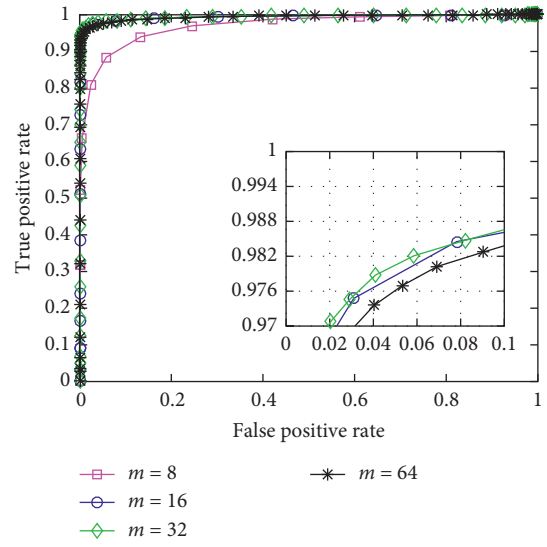

Figure 6: Curves of different group numbers.

robustness and discrimination, respectively, and all videos are converted to a standard size $256 \times 256 \times 256$ before hash calculation. Distance metrics of the compared schemes for measuring hash similarity are selected the same as the original papers. Specifically, the EOH-DCT hashing and the TWT-ARM hashing schemes select the normalized Hamming distance, the ST-NMF hashing scheme uses the Euclidean distance, and the LRSD-DWT hashing scheme chooses the Hamming distance. For the proposed hashing scheme, the experimental results under the parameters of $p = 30$ and $m = 32$ are taken for comparisons.

As different hashing schemes use different distance metrics to measuring hash similarity, it is impossible to directly present their calculated similarity results of robustness/discrimination in the same figure using a single distance metric. From the calculation of ROC curve described in Section 3.3, it can be seen that the ROC curve is a statistical result determined by a set of thresholds. It is

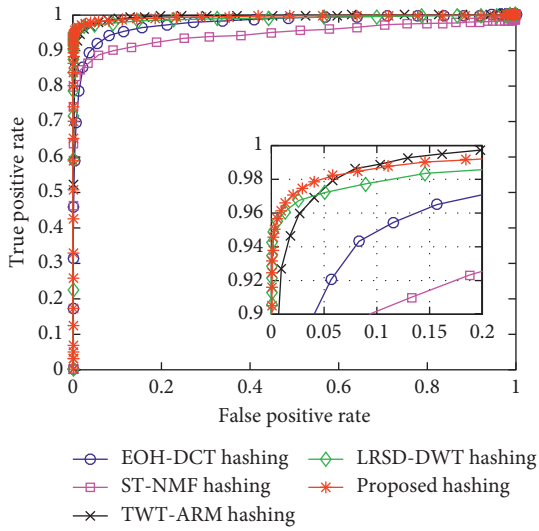

Figure 7: Curves of different schemes.

independent of the selected distance metric. Based on this consideration, the ROC graph is also used to compare different schemes' performances of robustness and discrimination. Figure 7 demonstrates the curves of different hashing schemes. The AUCs of different hashing schemes are calculated for quantitative comparison. The detailed results are as follows. The AUC of the EOH-DCT hashing scheme is 0.97706, the AUC of the ST-NMF hashing scheme is 0.94710, the AUC of the TWT-ARM hashing scheme is 0.99425, the AUC of the LRSD-DWT hashing scheme is 0.99076, and the AUC of the proposed hashing scheme is 0.99508. Clearly, the AUC of the proposed hashing scheme is bigger than all compared hashing schemes. It illustrates that the proposed hashing scheme outperforms all compared hashing schemes in balancing the performances of robustness and discrimination. The proposed hashing scheme makes a better AUC performance than the compared hashing schemes. This can be understood as follows. The proposed hashing scheme constructs the gradient features based high-dimensional matrix which can guarantee good robustness against digital operations and efficiently distinguish videos with different contents. It exploits MDS to learn low-dimensional features from the high-dimensional matrix which contribute to discrimination and compactness. In addition, the use of the ordinal measures can enhance robustness and discrimination of the proposed hashing scheme.

Time of hash generation is also examined. To this end, the total time of generating the hashes of the 100 original videos is calculated to determine the average time of a video hash. The coding language is MATLAB and the used machine is a computer workstation with a 2.1 GHz CPU and 64.0 GB RAM. The average time of the EOH-DCT hashing scheme is 7.24 seconds, the average time of the ST-NMF hashing scheme is 18.45 seconds, the average time of the TWT-ARM hashing scheme is 6.72 seconds, the average time of the LRSD-DWT hashing scheme is 37.88 seconds, and the average time of the proposed hashing scheme is 7.03 seconds. Clearly, the proposed hashing scheme is faster than the EOH-DCT hashing, ST-NMF hashing, and LRSD-DWT hashing schemes, but it runs slower than the TWT-ARM hashing scheme. Hash lengths of different schemes are compared. The length of the EOH-DCT hashing scheme is 60 bits, the length of the ST-NMF hashing scheme is 2048 bits, the length of the TWT-ARM hashing scheme is 128 bits, the length of the LRSD-DWT hashing scheme is 256 bits, and the length of the proposed hashing scheme is 160 bits. As to the performance of hash length, the proposed hashing scheme is better than the ST-NMF hashing scheme and the LRSD-DWT hashing scheme, but it is worse than the EOH-DCT hashing scheme and the TWT-ARM hashing scheme. Performances of these schemes are summarized in Table 5.

Table 5: Performance summary.

Scheme	AUC	Hash length (bit)	Time (s)
EOH-DCT hashing	0.97706	60	7.24
ST-NMF hashing	0.94710	2048	18.45
TWT-ARM hashing	0.99425	128	6.72
LRSD-DWT hashing	0.99076	256	37.88
Proposed hashing	0.99508	160	7.03

5. Conclusions

A novel video hashing scheme based on MDS and OM has been proposed in this paper. In the proposed hashing scheme, a high-dimensional matrix is constructed by using gradient features in the DWT domain and then mapped to the low-dimensional features via MDS. Since MDS can preserve original relationship of high-dimensional data in the low-dimensional data, the learned low-dimensional features are discriminative and compact. In addition, the OM codes of the learned low-dimensional features are exploited to generate video hash. As the OM codes are robust and discriminative features, the use of OM can contribute to a short, robust, and discriminative video hash. Extensive experiments on 8300 videos have been performed to test the proposed scheme. The results have revealed that the proposed scheme has a good robustness and desirable discrimination. Comparisons have demonstrated that the proposed scheme is better than several state-of-the-art schemes in balancing the performances of robustness and discrimination. In the future, we will investigate video hashing schemes with other useful techniques, such as deep learning and sparse representation.

Data Availability

The dataset used to support the findings of this work can be downloaded from the public websites whose hyperlink is provided in this paper.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grant nos. 61962008, 62062013, and 61762017), Guangxi "Bagui Scholar" Team for Innovation and Research, the Guangxi Talent Highland Project of Big Data Intelligence and Application, Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, and the Innovation Project of Guangxi Graduate Education (Grant no. XYCSZ2021009).

References

[1] X. Wang, L. Feng, and H. Zhao, "Fast image encryption algorithm based on parallel computing system," *Information Sciences*, vol. 486, pp. 340–358, 2019.

[2] X. Wang and S. Gao, "Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory," *Information Sciences*, vol. 507, pp. 16–36, 2020.

[3] X. Wang and S. Gao, "Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network," *Information Sciences*, vol. 539, pp. 195–214, 2020.

[4] N. Chen and H.-d. Xiao, "Perceptual audio hashing algorithm based on Zernike moment and maximum-likelihood watermark detection," *Digital Signal Processing*, vol. 23, no. 4, pp. 1216–1227, 2013.

[5] H.-G. Kim, H.-S. Cho, and J. Y. Kim, "Robust audio fingerprinting using peak-pair-based hash of non-repeating foreground audio in a real environment," *Cluster Computing*, vol. 19, no. 1, pp. 315–323, 2016.

[6] Z. Tang, L. Chen, X. Zhang, and S. Zhang, "Robust image hashing with tensor decomposition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 549–560, 2019.

[7] Z. Tang, X. Zhang, X. Li, and S. Zhang, "Robust image hashing with ring partition and invariant vector distance," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 200–214, 2016.

[8] J. Oostveen, T. Kalker, and J. Haitsma, "Visual hashing of digital video: applications and Techniques," *Applications of Digital Image Processing*, vol. 4472, pp. 121–131, 2001.

[9] X. Nie, J. Qiao, J. Liu, J. Sun, X. Li, and W. Liu, "LLE-based video hashing for video identification," in *Proceedings Of IEEE International Conference On Signal Processing (ICSP)*, pp. 1837–1840, Beijing, China, 2010.

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

[11] F. Khelifi and A. Bouridane, "Perceptual video hashing for content identification and authentication," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 50–67, 2019.

[12] R. Anuranji and H. Srimathi, "A supervised deep convolutional based bidirectional long short term memory video hashing for large scale video retrieval applications," *Digital Signal Processing*, vol. 102, Article ID 102729, 2020.

[13] H. Chen, Y. Wo, and G. Han, "Multi-granularity geometrically robust video hashing for tampering detection," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 5303–5321, 2018.

[14] Z. Tang, Z. Huang, H. Yao, X. Zhang, L. Chen, and C. Yu, "Perceptual image hashing with weighted DWT features for reduced-reference image quality assessment," *The Computer Journal*, vol. 61, no. 11, pp. 1695–1709, 2018.

[15] C. De Roover, C. De Vleeschouwer, F. Lefebvre, and B. Macq, "Robust video hashing based on radial projections of key frames," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 4020–4037, 2005.

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

[17] Y. Li, "Energy based robust video hash algorithm," in *Proceedings Of IEEE International Conference on Computational Intelligence and Security (CIS)*, pp. 433–436, Xian, China, 2010.

[18] H. Mao, G. Feng, X. P. Zhang, and H. Yao, "A robust and fast video fingerprinting based on 3D-DCT and LSH," in *Proceedings Of 2011 International Conference on Multimedia Technology*, pp. 108–111, Hangzhou, China, 2011.

[19] M. M. Esmaeili, M. Fatourechi, and R. K. Ward, "A robust and fast video copy detection system using content-based fingerprinting," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 1, pp. 213–226, 2011.

[20] I. Setyawan and I. K. Timotius, "Spatio-temporal digital video hashing using edge orientation histogram and discrete cosine transform," in *Proceedings Of International Conference On Information Technology Systems and Innovation (ICITSI)*, pp. 111–115, Hangzhou, China, 2014.

[21] A. Mucedero, R. Lancini, and F. Mapelli, "A novel hashing algorithm for video sequences," in *Proceedings Of IEEE International Conference On Image Processing (ICIP 2004)*, pp. 2239–2242, Singapore, 2004.

[22] S. Xiang, J. Yang, and J. Huang, "Perceptual video hashing robust against geometric distortions," *Science China Information Sciences*, vol. 55, no. 7, pp. 1520–1527, 2012.

[23] J. Sun, J. Wang, J. Zhang, X. Nie, and J. Liu, "Video hashing algorithm with weighted matching based on visual saliency," *IEEE Signal Processing Letters*, vol. 19, no. 6, pp. 328–331, 2012.

[24] M. Li and V. Monga, "Robust video hashing via multilinear subspace projections," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4397–4409, 2012.

[25] M. Li and V. Monga, "Compact video fingerprinting via structural graphical models," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1709–1721, 2013.

[26] X. Liu, J. Sun, and J. Liu, "Visual attention based temporally weighting method for video hashing," *IEEE Signal Processing Letters*, vol. 20, no. 12, pp. 1253–1256, 2013.

[27] Z. Tang, X. Zhang, and S. Zhang, "Robust perceptual image hashing based on ring partition and NMF," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 711–724, 2014.

[28] X. Nie, Y. Chai, J. Liu, J. Sun, and Y. Yin, "Spherical torus-based video hashing for near-duplicate video detection," *Science China Information Sciences*, vol. 59, no. 5, Article ID 059101, 2016.

[29] J. Sun, X. Liu, W. Wan, J. Li, D. Zhao, and H. Zhang, "Video hashing based on appearance and attention features fusion via DBN," *Neurocomputing*, vol. 213, pp. 84–94, 2016.

[30] S. Rameshnath and P. K. Bora, "Perceptual video hashing based on temporal wavelet transform and random projections with application to indexing and retrieval of near-identical

videos," *Multimedia Tools and Applications*, vol. 78, no. 13, pp. 18055–18075, 2019.

[31] Z. Tang, L. Chen, H. Yao, X. Zhang, and C. Yu, "Video hashing with DCT and NMF," *The Computer Journal*, vol. 63, no. 7, pp. 1017–1030, 2020.

[32] L. Chen, D. Ye, and S. Jiang, "High accuracy perceptual video hashing via low-rank decomposition and DWT," in *Proceedings Of International Conference on Multimedia Modeling (MMM)*, pp. 802–812, Daejeon, South Korea, 2020.

[33] S. Lee and C. D. Yoo, "Robust video fingerprinting for content-based video identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 7, pp. 983–988, 2008.

[34] Z. Lian, A. Godil, X. Sun, and H. Zhang, "Non-rigid 3D shape retrieval using multidimensional scaling and bag-of-features," in *Proceedings Of IEEE International Conference On Image Processing (ICIP 2010)*, pp. 3181–3184, Hong Kong, China, 2010.

[35] W. Jiang, C. Xu, L. Pei, and W. Yu, "Multidimensional scaling-based TDOA localization scheme using an auxiliary line," *IEEE Signal Processing Letters*, vol. 23, no. 4, pp. 546–550, 2016.

[36] H. Klock and J. M. Buhmann, "Data visualization by multidimensional scaling: a deterministic annealing approach," *Pattern Recognition*, vol. 33, no. 4, pp. 651–669, 2000.

[37] Z. Tang, Z. Huang, X. Zhang, and H. Lao, "Robust image hashing with multidimensional scaling," *Signal Processing*, vol. 137, pp. 240–250, 2017.

[38] S. L. France and J. D. Carroll, "Two-way multidimensional scaling: a review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 5, pp. 644–661, 2011.

[39] IEEE Std754–2008, *Standard for Floating-Point Arithmetic*, pp. 1–70, IEEE, New York, NY, USA, 2008.

[40] D. N. Bhat and S. K. Nayar, "Ordinal measures for visual correspondence," in *Proceedings Of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 1996)*, pp. 351–357, San Francisco, CA, USA, 1996.

[41] Z. Chai, Z. Sun, H. Méndez-Vázquez, R. He, and T. Tan, "Gabor ordinal measures for face recognition," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 1, pp. 14–26, 2014.

[42] Z. Sun and T. Tan, "Ordinal measures for iris recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2211–2226, 2009.

[43] W. Kim, J. Lee, M. Kim, D. Oh, and C. Kim, "Human action recognition using ordinal measure of accumulated motion," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, Article ID 219190, 2010.

[44] X. Hua, X. Chen, and H. Zhang, "Robust video signature based on ordinal measure," in *Proceedings Of IEEE International Conference on Image Processing (ICIP 2004)*, pp. 685–688, Singapore, 2004.

[45] Z. Tang, H. Zhang, S. Lu, H. Yao, and X. Q. Zhang, "Robust image hashing with compressed sensing and ordinal measures," *EURASIP Journal on Image and Video Processing*, vol. 2020, 2020.

[46] ReefVid, "Free reef video clip database," 2020, http://www.reefvid.org/.

[47] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.