Ma and Xing
EURASIP Journal on Image and Video Processing (2021) 2021:36
https://doi.org/10.1186/s13640-021-00577-z

EURASIP Journal on Image
and Video Processing

RESEARCH

Open Access

Perceptual hashing method for video content authentication with maximized robustness

Qiang Ma[1] and Ling Xing[2*]

*Correspondence:
xingling_my@163.com
[2] School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China
Full list of author information is available at the end of the article

Abstract

Perceptual video hashing represents video perceptual content by compact hash. The binary hash is sensitive to content distortion manipulations, but robust to perceptual content preserving operations. Currently, boundary between sensitivity and robustness is often ambiguous and it is decided by an empirically defined threshold. This may result in large false positive rates when received video is to be judged similar or dissimilar in some circumstances, e.g., video content authentication. In this paper, we propose a novel perceptual hashing method for video content authentication based on maximized robustness. The developed idea of maximized robustness means that robustness is maximized on condition that security requirement of hash is first met. We formulate the video hashing as a constrained optimization problem, in which coefficients of features offset and robustness are to be learned. Then we adopt a stochastic optimization method to solve the optimization. Experimental results show that the proposed hashing is quite suitable for video content authentication in terms of security and robustness.

Keywords: Video authentication, Perceptual hashing, Maximized robustness

1 Introduction

Thanks to the openness of Internet and the easiness of advanced multimedia tools, multimedia contents (e.g., image, video) may undergo certain operations when they are shared within the network. Operations include non-malicious actions and malicious actions. The non-malicious operations preserve the perceptual content and keep intact the understanding of the content, while the malicious operations deliberately distort the integrity. Therefore, the former is usually allowable and the latter is unacceptable. Traditional integrity authentication methods (e.g., Message Digest, Secure Hashing Algorithm) fail to check perceptual content, if both the allowable and unacceptable operations are taken into consideration. Hashes generated by those methods are bit-wise sensitive and leave no room for the perceptual content preserving actions. Perceptual hashing for image or video is an effective way to afford authentication of perceptual content, allowing the malicious-free operations happening [1]. It denotes the video by a compact string and discerns the secure content from the attacked content.

A number of perceptual video hashing methods have been proposed until now. Because video can be seen as a sequence of images, it is quite meaningful to look at methods for image hashing when we study the video hashing. Generally, image hashing can be roughly categorized into three types: (1) Image descriptor based methods. These methods extract perceptually important features from images and quantize those features into final hashes, e.g., histogram [2], Canny descriptors [3]; (2) Image matrix transformation based methods, which decompose image matrix into various components and select the most important part to form hashes. Examples include Discrete Cosine Transformation (DCT) [4], Radon transformation [5], etc.; (3) Machine learning based methods, which try to find correlation between features within high dimensional space, e.g., locally linear embedding [6] and core alignment [7].

Compared to image perceptual hashing methods, video hashing methods could incorporate the temporal property between sequential frames. Methods can be divided into two sections, i.e., spatial domain based and temporal-spatial domain based. The former usually chooses or generates representative/key frames for the video. The final hash is a concatenation of hashes of representative/key frames. Yang et al. [8] developed a video hashing based on speed up robust feature (SURF) descriptor. However, it was for video copy detection and was quite sensitive for frame content operations. Xiang et al. [9] used mean value of luminance histogram to infer hash value. It was catered for geometric distortion tolerance and showed strong robustness against content preserving operations.

Perceptual video hashing methods based on temporal-spatial domain generate hashes from information of both inter- and intra-frames. Pioneer work of this kind was the 3D-DCT method [10]. It exhibited good robustness against noise adding, luminance enhancement, etc. But its discrimination for content changing manipulations was not satisfactory. As for the representative frame research, Esmaeili et al. [11] proposed TIRI (Temporally Informative Representative Images) frame construction, which effectively fuses a series of consecutive frames. Value of a pixel on the TIRI frame is actually a combination of values of pixels on corresponding positions of those frames. Compared to the 3D-DCT method, TIRI method is less time consuming and is able to capture more semantic information. Many works have utilized TIRI for video hashing, for instance, the saliency video hashing [12] and the visual attention based hashing [13].

Some perceptual video hashing adopted matrix decomposition or machine learning methods. For example, Song et al. [14] chose a quantization based hashing, where the least quantization error is optimized using iterative methods. It was for video retrieval and was paid more attention to robustness. Another type of video hashing is based on deep learning method, e.g., multi-model stochastic recurrent neural networks for video hashing [15], binary encoder to decoder architecture for self-supervised video hashing [16], unsupervised hashing based on semantic structure [17] and cross-modal based deep hashing for video [18, 19]. Yang et al. [20] addressed the security issue of hamming space search, which improved robustness against the vulnerability of deep learning. However, these methods are mainly used for the video retrieval, video captioning and visual recognition [21–23]. They are more suitable to handle video semantics processing and our work is primarily focused on video perceptual representation.

Perceptual video hashing is widely accepted for its two main characteristics, i.e., robustness and sensitivity. Let $\boldsymbol{V}$ denote the video to be hashed, and let $H(\cdot)$ denote the hash function. Symbol $\boldsymbol{V}_{\mathrm{sim}}$ represents video which are the results of perceptual content preserving operations on $\boldsymbol{V}$. Likewise, symbol $\boldsymbol{V}_{\mathrm{dif}}$ represents those which are the results of perceptual content distorting operations. The robustness can be explained as

$$\mathrm{pr}(\|H(\boldsymbol{V}) - H(\boldsymbol{V}_{\mathrm{sim}})\| < \tau) > 1 - \theta_1 \, 0 < \theta_1 < 1 \tag{1}$$

where pr represents probability, parameters τ and θ_1 are small numbers near zero. Robustness requires that hash distance between $\boldsymbol{V}$ and $\boldsymbol{V}_{\mathrm{sim}}$ should be as small as possible. Similarly, sensitivity characteristic can be shown as

$$\mathrm{pr}(\|H(\boldsymbol{V}) - H(\boldsymbol{V}_{\mathrm{dif}})\| \geq \tau) > 1 - \theta_2 \, 0 < \theta_2 < 1 \tag{2}$$

where parameter θ_2 should be close to zero as much as possible. Sensitivity needs that large hash distance between $\boldsymbol{V}$ and $\boldsymbol{V}_{\mathrm{dif}}$ exists. Thus video with content changing can be easily detected by comparing its hash to that of the original video.

Different applications have different requirements for the strength of robustness and sensitivity. For example, when perceptual hashing is designed for video content retrieval, it is better to allow much more robustness than sensitivity. Because video retrieval should return as many similar video sets as possible. On the contrary, if video hashing is used for content authentication, sensitivity should be preferred. Since video authentication is to decide whether video content is deliberately manipulated, hash should be sensitive to those operations. As it can be observed from (1) and (2), distance between H($\boldsymbol{V}$) and H($\boldsymbol{V}_{\mathrm{sim}}$) or H($\boldsymbol{V}$) and H($\boldsymbol{V}_{dif}$) is compared to a threshold τ. Thus the threshold serves as a boundary between robustness and sensitivity. However, to use only a scalar value τ to determine the result is not enough, because it neglects the semantics and priori information for a specific circumstance.

In this paper we propose a novel perceptual video hashing with maximized robustness for content authentication. The idea is illustrated in Fig. 1, where red hollow triangle represents feature of original video $\boldsymbol{V}$, small circles and rectangles represent features of $\boldsymbol{V}_{\mathrm{sim}}$ and $\boldsymbol{V}_{\mathrm{dif}}$, respectively. When we design a video hashing for authentication, we keep

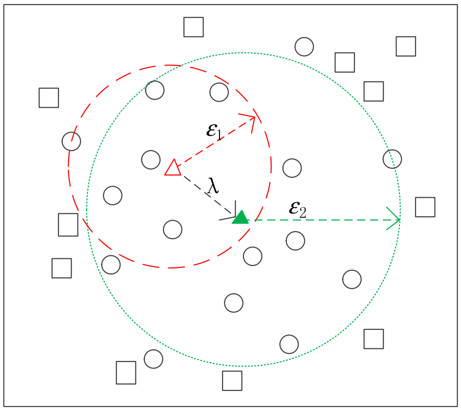


Fig. 1 Conceptual illustration of video hashing for authentication with maximized robustness. Red hollow triangle represents feature of original video. Small circles and rectangles represent features of similar and dissimilar video. The feature offset lambda can define a new boundary shown in green dotted circle, with new feature green triangle as its feature center and new radius epsilon 2 as its robustness. Compared with its former robustness epsilon 1, the robustness is maximized but without loss of sensitivity

in mind that we should detect those $\boldsymbol{V}_{\mathrm{dif}}$ as much as possible. But we also need some robustness for $\boldsymbol{V}_{\mathrm{sim}}$. In regard to video security, sensitivity has higher superiority. Therefore, we only allow robustness flexibility after we are ensured that $\boldsymbol{V}_{\mathrm{dif}}$ could be correctly spotted. If we treat the hollow triangle as the center to compare, we could find a maximized robustness length $\boldsymbol{\varepsilon}_1$, with which defines a boundary (i.e., the red dotted circle) for sensitivity and robustness. However, if we move the center to solid green triangle by a feature adjustment $\boldsymbol{\lambda}$, we could find a new robustness length $\boldsymbol{\varepsilon}_2$, which is much larger than $\boldsymbol{\varepsilon}_1$. Thus we obtain an improved robustness without loss of sensitivity. The solid green triangle is the new center to compare. The new boundary (i.e., the green dotted circle) extends the allowable features space. Therefore, the robustness is maximized but without cost of sensitivity. Remaining task is to tactically find parameters $\boldsymbol{\lambda}$ and ε_2.

In our previous paper, we proposed core alignment for image hashing [7]. Although there seem connections between these two, several main differences exist. First, motivations are quite different. In our last work, we tried to find the largest discrimination between similar and dissimilar contents by minimizing hash distances. In this paper, we endeavor to obtain the largest robustness when security is first preferred. Thus this approach is more conservative for content authentication when it comes to robustness requirement. Second, mathematical formulations and problem solving methods are quite different. Method here is to optimize two coefficients simultaneously while previous method was to find coefficients sequentially. Third, simulation and results are distinct in terms of open data sets and performances.

The novelty of this paper is that we incorporated the new idea of maximized robustness into perceptual video hashing mathematical problem forming and solving for content authentication. Instead of treating robustness and sensitivity properties as of equal importance, we make sure that sensitivity is first met before we find the robustness. The proposed video perceptual hashing method with maximized robustness is show in Fig. 2, where hashing is divided into two parts, i.e., learning part and hashing part. In the learning part, video sets are composed of original video, together with their variants under perceptual content preserving and changing operations. After preprocessing, we construct features space, including features of original and modified video. We learn

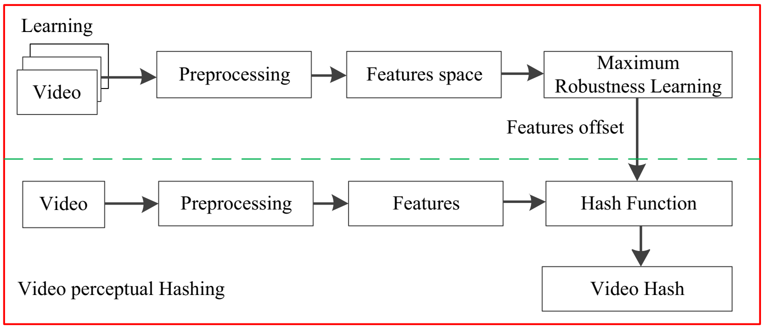


Fig. 2 Block diagram of video perceptual hashing with maximized robustness. In the learning part, video, together with their variants under perceptual content preserving and distorting operations, are preprocessed to form the feature space. The maximum feature offset is learned to be regarded as the maximized robustness. In the hashing part, a particular video is processed to obtain its feature, and the feature is adjusted by the feature offset. Hash function is performed to the feature to generate the final hash

from those features to achieve maximized robustness, which results in features offset. In the video hashing part, when a video is to be perceptually hashed, same operations of preprocessing and feature extraction are conducted. The hash function considers the learned adjustment and takes video's feature as input to produce the final hash.

The rest of this paper is organized as follows. In Sect. 2 we formulize the mathematical problem of video hashing with maximized robustness as a constrained optimization, where variables of features offset and robustness are explained. We solve the problem in Sect. 3 by Fish School Search algorithm, where two variables are learned simultaneously. In Sect. 4 we present simulation results and discussion. Finally, we conclude the paper in Sect. 5.

2 Video perceptual hashing problem formulization

Although various video formats are available, we only consider raw video for perceptual hashing method research. A raw video is a temporal sequence of frames, which do not undergo compression. If the video to be hashed is a compressed one, we adopt advanced multimedia processing tools to transform it into raw format. Considering that video may be presented in different color spaces and luminance vector carries the most significant perception information, we choose only luminance component for perceptual hashing.

We normalize the input video in terms of frame size and frame rate. Each frame is rescaled into width W and height H. Frame rate is re-sampled into F frames per second. Then we adopt method of luminance difference between adjacent frames to group frames into different sets, where each set denotes a scene [24]. Note that preprocessing operations on the video, i.e., frame size scaling, rate sampling and scene grouping, improve the robustness to certain extent.

Let Υ denote a preprocessed video with $\Upsilon = \{v_1, \cdots, v_k\}$, where v_k represents the kth group. The ith group consists of a frame set, denoted by $\{f_{i1}, \cdots, f_{il}\}$. In order to efficiently obtain hash of each group, we adopt TIRI method to represent those frames within a group. The TIRI representative frame is a linear combination of frames, which is defined as:

$$T_k(w,h) = \sum_{p=1}^{l} w_p f_{kp}(w,h) \tag{3}$$

where symbol $T_k(w,h)$ denotes value of position (w, h) on the kth TIRI frame with $1 \leq w \leq W$ and $1 \leq h \leq H$, symbol $f_{kp}(w,h)$ denotes luminance value of position (w, h) on the pth frame within kth group. Coefficient w_p is a weight for the pth frame and it is determined by γ^p. According to the empirical study, TIRI frame shows the best representative performance when γ is set 0.6 [11]. In our method, video hash is made up of representative frames' hash. The hash of a video is the concatenation of hashes of all TIRI frames. Note that videos with different number of representative frames have different hash code length. But length of each representative frame is the same.

In regard to find the best features offset for all video, we consider all TIRI frames of whole training video. We denote all representative frames by a set $\mathbf{T}$,$\boldsymbol{T} = \{\boldsymbol{T}_1, \cdots, \boldsymbol{T}_i, \cdots, \boldsymbol{T}_n\}$, which means that total n TIRI frames are generated for n groups of original video in the training part. For each frame, we extract its feature and denote it by symbol $\boldsymbol{\varphi}$. Original feature sets is denoted by $\boldsymbol{\Phi}^O$, with $\boldsymbol{\Phi}^{\mathrm{O}} = \{\boldsymbol{\varphi}_1^O, \cdots, \boldsymbol{\varphi}_i^O, \cdots, \boldsymbol{\varphi}_n^O\}$. Then, for each video, we conduct content preserving and distortion operations. A number of versions for each video are obtained. In order to describe the notations more clearly, we allow the number of operations on each group to be the same. Assume numbers of content preserving and distorting operations on video are P and Q, respectively, then for each original representative frame's feature, there are P and Q features for those operations. We denote features under content preserving operations by $\boldsymbol{\Phi}^A$, $\boldsymbol{\Phi}^A = \{\boldsymbol{\varphi}_1^A, \cdots, \boldsymbol{\varphi}_i^A, \cdots, \boldsymbol{\varphi}_n^A\}$. Similarly, we denote features under content distorting operations by $\boldsymbol{\Phi}^D$, $\boldsymbol{\Phi}^D = \{\boldsymbol{\varphi}_1^D, \cdots, \boldsymbol{\varphi}_i^D, \cdots, \boldsymbol{\varphi}_n^D\}$. Note that we have $\left|\boldsymbol{\varphi}_i^A\right| = P$ and $\left|\boldsymbol{\varphi}_i^D\right| = Q$, where symbol $|\cdot|$ means cardinality.

Now we have the features space of $\{\boldsymbol{\Phi}^O \cup \boldsymbol{\Phi}^A \cup \boldsymbol{\Phi}^D\}$ in the training part. We denote the variables of features offset and robustness by $\boldsymbol{\lambda}$ and $\boldsymbol{\varepsilon}$, respectively. Then the question to find the maximized robustness can be written as follows:

$$\begin{aligned} &\max \ \|\boldsymbol{\varepsilon}\| \\ &\text{s.t.} \left\|\boldsymbol{\varphi} - (\boldsymbol{\varphi}_i^O + \boldsymbol{\lambda})\right\| > \|\boldsymbol{\varepsilon}\| \ \forall \boldsymbol{\varphi} \in \boldsymbol{\varphi}_i^D \\ &\ 1 \le i \le n \end{aligned} \tag{4}$$

where the objective is to maximize the robustness. Variables $\boldsymbol{\lambda}$ and $\boldsymbol{\varepsilon}$ define a feature space, which provides allowable perceptual content preserving operations' consequences. The first constraint requires that for every feature belonging to the content distorting video, it should maintain a distance larger than the robustness. The distance is measured between the feature $\boldsymbol{\lambda}$ and the improved feature. Thus by comparing the distance, we are able to detect those content distorting video. The second constraints needs that the two variables should be valid for every video group, which states that the result of robustness is the consensus of all circumstances.

An interesting point to look at is that we do not include in the constraints the distance between the feature of content preserving video and the improved feature. Two cases of the distance exist, which are as follows:

$$\left\|\boldsymbol{\varphi} - (\boldsymbol{\varphi}_i^O + \boldsymbol{\lambda})\right\| > \|\boldsymbol{\varepsilon}\| \ \exists \boldsymbol{\varphi} \in \boldsymbol{\varphi}_i^A \tag{5}$$

$$\left\|\boldsymbol{\varphi} - (\boldsymbol{\varphi}_i^O + \boldsymbol{\lambda})\right\| \le \|\boldsymbol{\varepsilon}\| \ \exists \boldsymbol{\varphi} \in \boldsymbol{\varphi}_i^A \tag{6}$$

The first one implies that for certain feature of space $\boldsymbol{\Phi}^A$, its distance to the improved feature is larger than the robustness. This scenario means that although the feature is indeed an allowable feature from content preserving operations, it still would be judged dissimilar because security requirement is more enhanced by a newly defined boundary. We call circumstance of (5) unachievable robustness. On the contrary, circumstance of (6) says that some feature of space $\boldsymbol{\Phi}^A$ remains in the allowable feature space. This is called achievable robustness. Therefore, we claim that our perceptual video hashing is quite conservative and it emphasizes much more on the security, which is the requirement for our purpose to authenticate video content.

3 Optimization problem methods

Recall that in the constrained optimization problem of (4), two variables are to be found, i.e., features offset and robustness. The robustness variable exist both in the objective function and the first constraint, while features offset only appears in the constraint. The maximized robustness is obtained once we find the appropriate feature offset. As for our constrained problem solution, traditional deterministic methods are not applicable, since derivatives of the objective and inequality constraint are hard to find. We look at these two variables from a stochastic perspective. The features offset is brought about by human eyes' adjustment to different types of allowable operations with different strengths upon the original video content. To some extent the adjustment could be thought of the result of various noises on the original feature. A noise represents a kind of operation on the content. Therefore, we use a stochastic way to solve the problem.

Considering the specific characteristics of the problem, we adopt fish school search algorithm to tackle the optimization. The fish school search algorithm imitates food finding behavior of fish schools [25]. The population based algorithm has been widely used to find the best solution in various optimization applications. It feeds the fish and fish acts both individually and collectively. The fish swim toward the food according to the positive gradient of fitness function. Each fish in the school is treated as a potential solution to the problem. Fish swims based on the local and collective information. Because we need to find two variables, we define two kinds of fish school according to problem. However, these two schools are correlated by the first constraint. We define individual fish $\boldsymbol{\lambda}_j$ and $\boldsymbol{\varepsilon}_j$ for features offset and robustness, respectively. Sizes of both schools are equal to M. During each of iteration, fish firstly moves as follows:

$$\boldsymbol{\lambda}_j(t+1) = \boldsymbol{\lambda}_j(t) + \text{rand}(-1,1) \times \text{step}_{\text{ind1}} \tag{7}$$

$$\boldsymbol{\varepsilon}_j(t+1) = \boldsymbol{\varepsilon}_j(t) + \text{rand}(-1,1) \times \text{step}_{\text{ind2}} \tag{8}$$

where rand (− 1, 1) generates a random number within range (− 1, 1), $\text{step}_{\text{ind1}}$ and $\text{step}_{\text{ind2}}$ are two coefficients used to control displacement of the movement. Symbols $t+1$ and t represent the count after and before the individual movement, respectively. The movement is valid on condition that the first constraint is achieved. Otherwise, fish needs to remain the position of iteration t.

Then fish is updated through collective-instinctive movement. Individual fish is moved by an average movement, which is calculated as follows:

$$I_1 = \frac{\sum_{j=1}^{M} \Delta\boldsymbol{\varepsilon}_j \Delta\varepsilon_j^{\circ}}{\sum_{j=1}^{M} \Delta\varepsilon_j^{\circ}} \tag{9}$$

where symbol $\Delta\varepsilon_j^{\circ}$ stands for fitness enhancement achieved and $\Delta\varepsilon_j^{\circ} = \left\|\boldsymbol{\varepsilon}_j(t)\right\| - \left\|\boldsymbol{\varepsilon}_j(t-1)\right\|$. Symbol $\Delta\varepsilon_j$ stands for movement displacement for the jth fish and $\Delta\boldsymbol{\varepsilon}_j = \boldsymbol{\varepsilon}_j(t) - \boldsymbol{\varepsilon}_j(t-1)$. Each fish is updated by the average movement as follows:

$$\boldsymbol{\varepsilon}_j(t+1) = \boldsymbol{\varepsilon}_j(t) + I_1 \tag{10}$$

Accordingly, we define the average movement for features offset fish school as follows:

$$I_2 = \frac{\sum_{j=1}^{M} \Delta\boldsymbol{\lambda}_j \Delta\boldsymbol{\varepsilon}_j^{\circ}}{\sum_{j=1}^{M} \Delta\boldsymbol{\varepsilon}_j^{\circ}} \tag{11}$$

Each fish λ_j is updated as:

$$\boldsymbol{\lambda}_j(t+1) = \boldsymbol{\lambda}_j(t) + I_2 \tag{12}$$

Also we need to find the collective-volitive movement for both fishes. Barycenters are calculated as follows:

$$B_1 = \frac{\sum_{j=1}^{M} \boldsymbol{\varepsilon}_j(t) w_j(t)}{\sum_{j=1}^{M} w_j(t)} \tag{13}$$

where B_1 stands for the barycenter of robustness fish school and $w_j(t)$ stands for feeding weight of the *j*th fish. The weight is calculated as follows:

$$w_j(t+1) = w_j(t) + \frac{\Delta\varepsilon_j^{\circ}}{\max(\|\varepsilon_j\|)} \tag{14}$$

where $\max(\|\varepsilon_j\|)$ represents the maximized value of fitness function variation. Note that the weight is bounded by W_{scale} and it varies from 1 to W_{scale}. Initial values of all weights are set to W_{scale}. If total robustness fish weight has improved from last iteration, each fish is moved towards the barycenter according to (15). Otherwise, each fish is moved away from the barycenter according to (16).

$$\boldsymbol{\varepsilon}_j(t+1) = \boldsymbol{\varepsilon}_j(t) - \text{step}_{\text{vol1}} \text{rand}(0,1) \frac{\boldsymbol{\varepsilon}_j(t) - B_1(t)}{\|\boldsymbol{\varepsilon}_j(t) - B_1(t)\|} \tag{15}$$

$$\boldsymbol{\varepsilon}_j(t+1) = \boldsymbol{\varepsilon}_j(t) + \text{step}_{\text{vol1}} \text{rand}(0,1) \frac{\boldsymbol{\varepsilon}_j(t) - B_1(t)}{\|\boldsymbol{\varepsilon}_j(t) - B_1(t)\|} \tag{16}$$

where coefficient $\text{step}_{\text{vol1}}$ is used to control the displacement movement like $\text{step}_{\text{ind1}}$. Similarly, we calculate the barycenter for features offset fish school as follows:

$$B_2 = \frac{\sum_{j=1}^{M} \boldsymbol{\lambda}_j(t) w_j(t)}{\sum_{j=1}^{M} \boldsymbol{\lambda}_j(t)} \tag{17}$$

The features offset fish is moved towards or far away from the barycenter on the same condition of robustness fish. The movement displacement is as follows:

$$\boldsymbol{\lambda}_j(t+1) = \boldsymbol{\lambda}_j(t) - \mathrm{step}_{\mathrm{vol2}}\mathrm{rand}(0,1)\frac{\boldsymbol{\lambda}_j(t) - B_2(t)}{\left\|\boldsymbol{\lambda}_j(t) - B_2(t)\right\|} \tag{18}$$

$$\boldsymbol{\lambda}_j(t+1) = \boldsymbol{\lambda}_j(t) + \mathrm{step}_{\mathrm{vol2}}\mathrm{rand}(0,1)\frac{\boldsymbol{\lambda}_j(t) - B_2(t)}{\left\|\boldsymbol{\lambda}_j(t) - B_2(t)\right\|} \tag{19}$$

After certain number of iterations, the algorithm stops and features offset with the maximized robustness is returned as the optimal solution. Note that in our hashing optimization, features of $\boldsymbol{\Phi}^A$ are not included. However, they provide a valuable clue to initialization of fishes. In practice, we choose the minimal difference between the original feature and the allowable feature to be the initial value of feature offset fish $\boldsymbol{\lambda}_j$. The minimum distance is used to set the the initial value of robustness fish $\boldsymbol{\varepsilon}_j$. They are calculated as follows:

$$\tilde{i} = \arg\min_i \left\|\varphi_i^O - \varphi_i^A\right\| \tag{20}$$

$$\boldsymbol{\lambda}_j = \varphi_{\tilde{i}}^O - \varphi_{\tilde{i}}^A \tag{21}$$

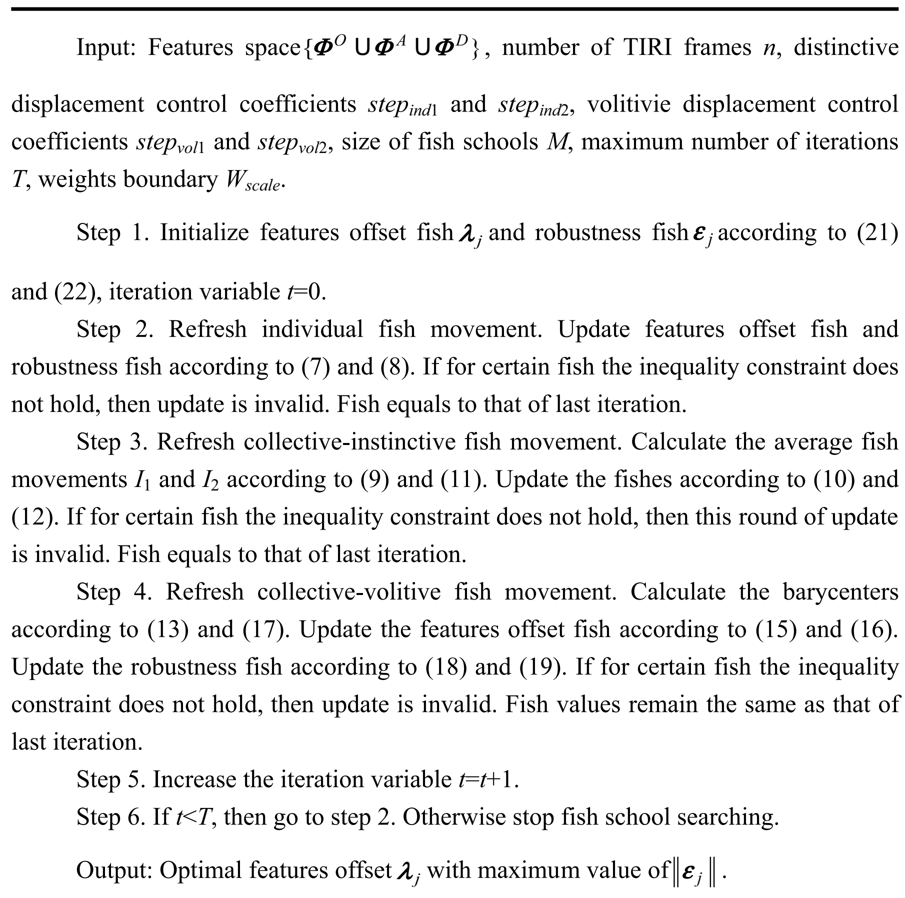
$$\boldsymbol{\varepsilon}_j = \left\|\varphi_{\tilde{i}}^O - \varphi_{\tilde{i}}^A\right\| \tag{22}$$

The overall searching algorithm is described in Fig. 3, where the inequality constraint of (4) is enforced upon every run of fish updates. The output is the optimal result of features offset and robustness. Note that for these two fishes, the robustness fish with the maximum fitness value is what we are looking for. Thus this fish is regarded as the maximized robustness and its corresponding features offset fish is chosen to be the optimal features offset. The optimal features offset obtained are then used for features adjustment in the video perceptual hashing part.

4 Results and discussion

We validate our hashing method based on video which are downloaded from an open video database, i.e., open video project. The database is maintained by interaction design laboratory of University of North Carolina Chapel Hill. It contains various types of video, in terms of contents formats and duration. We download 50 video for our simulation. Contents of downloaded video include education, history, speech and documentary. Formats include MPEG-1 and MPEG-2. Durations include one minute, one to two minutes, two to five minutes and five to ten minutes.

Video are first preprocessed before they are input to the hashing method. We normalize video frame size as 320 × 240 and frame rate as 10 frames per second. By comparing luminance difference, we divide video frames into various groups. Each group corresponds to a certain perceptual understanding for humans. Then we calculate TIRI frame for every group. We randomly choose 25 video for training and the remaining is used for testing. We implement the simulation by tool of Matlab with version R2012a on a computer with 8 GB memory and 3.9 GHz CPU.



Input: Features space$\{\boldsymbol{\Phi}^O \cup \boldsymbol{\Phi}^A \cup \boldsymbol{\Phi}^D\}$, number of TIRI frames n, distinctive displacement control coefficients $step_{ind1}$ and $step_{ind2}$, volitivie displacement control coefficients $step_{vol1}$ and $step_{vol2}$, size of fish schools M, maximum number of iterations T, weights boundary W_{scale}.

Step 1. Initialize features offset fish $\boldsymbol{\lambda}_j$ and robustness fish $\boldsymbol{\varepsilon}_j$ according to (21) and (22), iteration variable t=0.

Step 2. Refresh individual fish movement. Update features offset fish and robustness fish according to (7) and (8). If for certain fish the inequality constraint does not hold, then update is invalid. Fish equals to that of last iteration.

Step 3. Refresh collective-instinctive fish movement. Calculate the average fish movements I_1 and I_2 according to (9) and (11). Update the fishes according to (10) and (12). If for certain fish the inequality constraint does not hold, then this round of update is invalid. Fish equals to that of last iteration.

Step 4. Refresh collective-volitive fish movement. Calculate the barycenters according to (13) and (17). Update the features offset fish according to (15) and (16). Update the robustness fish according to (18) and (19). If for certain fish the inequality constraint does not hold, then update is invalid. Fish values remain the same as that of last iteration.

Step 5. Increase the iteration variable t=t+1.

Step 6. If t<T, then go to step 2. Otherwise stop fish school searching.

Output: Optimal features offset $\boldsymbol{\lambda}_j$ with maximum value of$\|\boldsymbol{\varepsilon}_j\|$.

Fig. 3 Algorithm for finding optimal features offset and robustness. Here show the main steps for optimization problem solving. Inputs are features space and some coefficients. Outputs are the optimal value for feature offset and maximum robustness. The steps include fish initialization, fish individual refresh, fish collective-instinctive movement and fish collective-volitive movement

In order to mimic the operations video may undergo over the network, we conduct content preserving operations and content distorted attacks on the video. TIRI frames are generated for the changed video. The content preserving operations are listed as follows.

- Rotation with clock-wise 2, 5, 10 and 15 degrees
- Scaling with factor of 0.6, 0.8, 1.2 and 1.5
- Translation by first scaling the frame with factor of 0.5 and then shifting it horizontally and vertically with coordinates adjustment of (− 10, 0), (10, 0), (0, 10), (0, − 10) and (10, 10)
- Gaussian noise adding with mean zero and variance of 0.01, 0.03 and 0.05
- Salt and pepper noise adding with density of 0.01, 0.02, 0.03, 0.04 and 0.05
- Average filtering with filter size of 2 × 2, 4 × 4 and 6 × 6.
- Intensity changing with value of 0.90, 0.95, 1.10 and 1.20
- Median filtering with filter size of 2 × 2, 4 × 4, 5 × 5 and 6 × 6.

Attacks that deliberately change the perceptual contents of video are as follows.

- Block overlaying on original frames by three types, i.e., white, black and Mosaic blocks, each type has two blocks of 50 × 50, two blocks of 100 × 100 and one block of 200 × 200, respectively.
- Pasting on original frames by a totally different image, with randomly chosen position and size of content being pasted to be 10%, 20%, 30% and 40%.
- Block shuffling by dividing a frame into blocks of equal size and randomly rearranging them to form a new frame, with number of blocks to be 2, 4, and 16.

We show an example of video perceptual content changing manipulations in Fig. 4, where (a) is the original frame, (b) is the result of block overlaying by one Mosaic of 200 × 200, (c) is the result of pasting by an image with 20% content being pasted and (d) is the result of block shuffling with 16 equal blocks.

In regard to frame feature representation, we choose Radon transformation to form feature vector for each TIRI frame. Radon transformation has superior advantage to describe image feature, which shows strong robustness over rotation, scaling and translation operations [5, 26]. We apply discrete Fourier transform to these coefficients and choose norm of transform as TIRI frame feature. In our experiment, angle is chosen to be 0 to 179 degrees with one degree step and order is chosen to be one to six. Length of our TIRI feature vector is 546 with each element being a real number. We also adopt median filtering method to generate binary hash value. Therefore, our hash length for TIRI frame is 546 bits.

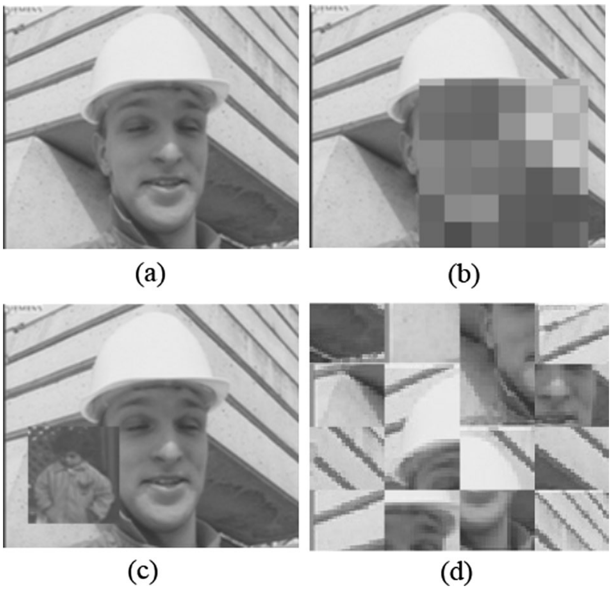


Fig. 4 Example of attacks of content distortion manipulations. **a** is original frame, **b** is block overlaying, **c** is pasted and **d** is reshuffled. Here shown are the examples of attacks on the frames. The upper left panel is the original frame; the upper right panel is the result of block overlaying attack. The lower left panel is the result of being pasted and the lower right panel is the result of reshuffling attack

We choose metrics of true positive rate (P_T) and false positive rate (P_F) to compare the performance of methods. Definitions of P_T and P_F are as follows.

$$P_T = \frac{\text{number of correctly claimed authentic TIRI frames}}{\text{number of claimed authentic TIRI frames}} \tag{23}$$

$$P_F = \frac{\text{number of incorrectly claimed unauthentic TIRI frames}}{\text{number of claimed unauthentic TIRI frames}} \tag{24}$$

The claimed authentic or unauthentic TIRI frames mean that those frames are judged secure or insecure by hashing methods. On the contrary, the correctly claimed authentic or incorrectly claimed unauthentic TIRI frames mean that those frames judged by hashing methods secure or insecure are actually secure, whose understanding is supported by a priori knowledge. P_T numerically shows the robustness to some extent, while P_F shows the security of hashing methods correspondingly. In the simulation we adopt ROC (Receiver Operation Curve) to demonstrate performances of robustness and security simultaneously.

We choose for performance comparison three related hashing methods, i.e., Radon-based [5], TIRI + DCT [11] and LSH-core method [7]. Since our hashing method adopt Radon feature as TIFI feature vector, we chose a hashing method also based on Radon feature to evaluate. The chosen Radon-based method utilized the third order moment of Radon transformation to obtain frame's statistical feature and adopted the first 15 DFT coefficients to construct final hash. Its hash length was 150 bits. We include TIRI + DCT method here because it also used TIRI representative frames to deal with large redundant video frames. It differs from ours in that DCT was conducted on TIRI frame blocks and two coefficients of each block are concatenated to form final hash with 640 bits. LSH-core method was distinguished in that its object function neglected the strict first priority of security as in the proposed maximized robustness. It adopted learning to find best feature core and applied LSH to reduce the dimensionality, resulting a hash length of 350 bit.

4.1 Experimental results

We show performance comparisons in Fig. 5, where ROC curve for all fours methods are drawn. The x-axis and y-axis denote P_F and P_T, respectively. Note that although 32 similar and 16 dissimilar versions of TIRI frames exist, we show the averaged values under all operations in Fig. 4 to understand the whole performance. Note that in ROC comparison the higher the curve, the better the performance. For example, when we set value of P_T at 0.95, values of P_F for our method, LSH-core, Radon-based and TIRI + DCT are 0.03, 0.06, 0.07 and 0.10, respectively. When we constrain P_F to be value of 0.05, we can achieve P_T value of 0.97. Values for the others are 0.93, 0.91 and 0.90, respectively. Thus it demonstrates that our method has maximized robustness when security is limited.

In order to analyze the effects of various perceptual content preserving operations upon the robustness and sensitivity, we present brief results in Table 1, where true positive rate and false positive rate performances of four methods are shown. For each type of operations, we show the averaged results across different coefficients settings. Values of P_F and P_T are obtained when optimal thresholds are used for each type. It can be

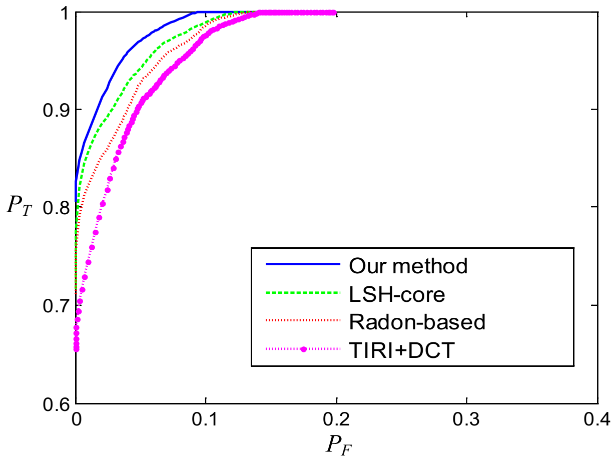


Fig. 5 ROC curves of performances for four hashing methods. Here shown are the performance comparison of four hashing methods in form of ROC curves. The false positive rates and true positive rates are the *x*-axis and *y*-axis, respectively. Besides our method, three other methods, i.e., LSH-core, Radon-based and TIRI + DCT, are included. The higher the curve is, the better the hashing method in terms of robustness and security

Table 1 Robustness and sensitivity performances of the perceptual content preserving operations

Methods	**Our method**		**LSH-core**		**Radon-based**		**TIRI + DCT**	
	P_F **(%)**	P_T **(%)**	P_F **(%)**	P_T **(%)**	P_F **(%)**	P_T **(%)**	P_F **(%)**	P_T **(%)**
Rotation	2.17	100	2.28	100	2.51	100	2.64	100
Scaling	2.31	100	2.35	100	2.47	99.72	2.49	98.17
Translation	2.25	100	2.41	99.85	2.61	98.15	2.81	97.16
Gaussian noise	2.31	99.80	2.51	99.11	2.63	98.04	2.71	97.12
Salt and pepper	2.34	99.84	2.61	98.99	2.68	97.85	2.83	97.56
Average filtering	2.51	99.15	2.71	98.14	2.78	96.62	3.12	96.10
Intensity changing	2.43	98.84	2.54	98.17	2.58	97.84	2.65	97.27
Median filtering	2.48	98.57	2.98	97.16	3.08	95.10	3.47	94.02

Here shown in this table are the false positive rates and true positive rates for all four methods under various operations. The listed each operation has the averaged results for each method at each row. The higher the true positive rate, the better the robustness; likewise, the lower the false positive rate, the better the sensitivity the method has

noted that our method has much higher true positive rate than those of the other three methods as for each operation. Moreover it has lower false positive rate. This implies that our method can achieve more robustness but still can preserve strong sensitivity.

Because video content authentication is the primary goal for hashing method, it is quite meaningful to compare the false positive rate performance under various attacks. We show the results in Fig. 6, where values of P_F are obtained when optimal thresholds are set for all methods. Figure 6a–c are for blocks overlapping attacks of white, black and Mosaic type, respectively. The x-axis for them denotes index of block size, i.e., No. 1, 2 and 3 representing two blocks of 50×50, two blocks of 100×100 and one block of 200×200, respectively. Figure 6d is for pasting attacks, where x-axis denotes pasting type, i.e., No. 1 ~ 4 representing pasting content of TIRI image being 10%, 20%, 30% and 40%, respectively. As for the block reshuffling attacks, all four methods can detect manipulations with value P_F of zero.

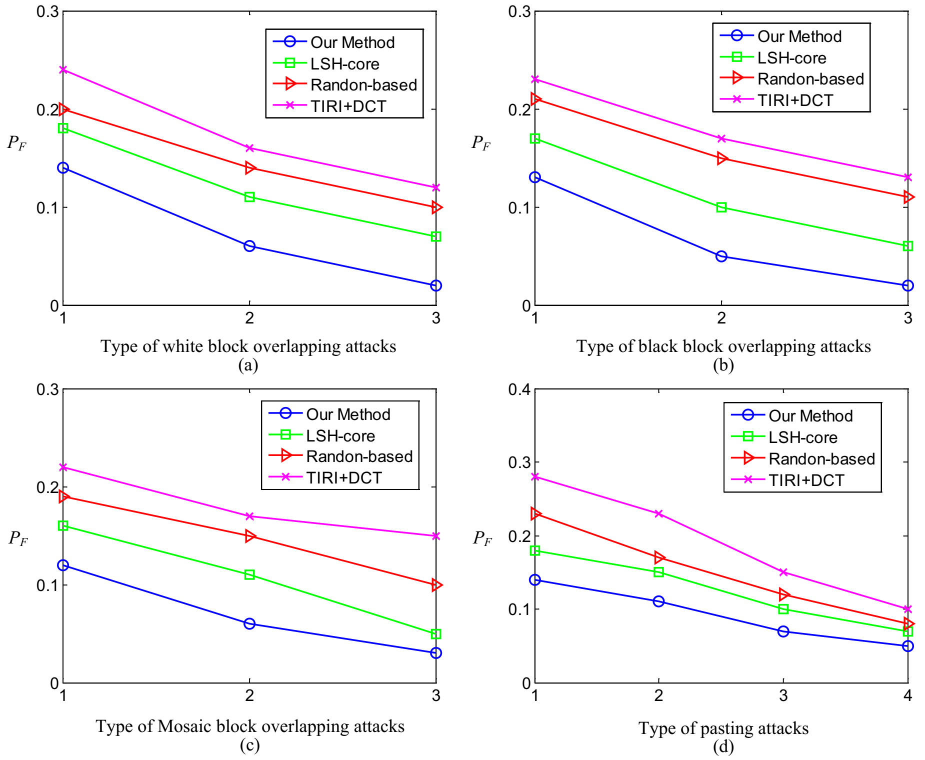


Fig. 6 False positive rate comparisons across different types of attacks. Here shown are the comparisons of false positive rate for four methods under different types of attacks. The upper left panel is for the type of white block overlapping attacks with block size growing; the upper right panel is for the black block overlapping attacks with block size growing; the lower left panel is for the white Mosaic overlapping attacks with block size growing; the lower right panel is for the pasting attacks with pasting area growing

It is observed that our method has the lowest value of P_F among all methods, which states that our method is superior to others when video content security is required. For instance, our method has false positive rate of 0.06 for two 100 × 100 white block overlapping, while LSH-core, Radon-based and TIRI + DCT need 0.11, 0.14 and 0.16, respectively. From Fig. 5a–c it is seen that curves for these three types block overlapping are not fundamentally different, which means that types of blocks do not have distinct effects on the false positive rates.

4.2 Discussion

From Fig. 5 it can be seen that our method is the most secure under same robustness requirements. In other words, robustness is maximized in our hashing method compared with other methods. Note that TIRI + DCT has the lowest ROC curve in Fig. 4, which means it has the highest false positive rates given the robustness requirement. The reason may be that this method only chooses low frequencies of DCT coefficients and those coefficients could capture quite coarse perceptual information. When this method is used for content authentication, large useful information cannot be implied in the hash value, leading to higher value of false positive rates.

It can also be observed that our method has 3% security improvement compared with LSH-core method when P_T is 0.95. This is due to that our hashing is more conservative on security criteria. As is stated in the constrained problem optimization, security requirement is the first priority. However, it can be seen from Fig. 5 when we allow no false positive rates, we could obtain nearly similar values of P_T for our method and LSH-core method. This can be understood that although we strengthen the security requirement for perceptual hashing, loss of robustness does not happen. On the contrary, since we try to achieve the best feature offset to adjust final hash, we obtain slightly improved robustness.

With regard to performances comparisons of perceptual content preserving operations in Table 1, it is meaningful to analyze different types of operations in terms of robustness and sensitivity. Overall it can be noted that operations of Rotation, Scaling and Translation exhibit much better true positive rate performances than the remaining operations for all four methods. This can be understood that these superior performances result from the Radon feature, which has strong robustness against the three operations. However, false positive rate performances across the four methods have much more obvious differences. For instance, as for Translation, our method has about 7%, 16% and 24% improvements for LSH-core, Radon-based and TIRI + DCT, respectively. It means that our method dose not loss much sensitivity compared with the other methods. As for the Gaussian noise and the Salt & Pepper, they have slightly similar effects on the performance, because these two methods cause similar consequences on pixels values of TIRI frames. Similarly, all four methods show good results as for the Intensity changing. As for the Average and Median filtering, our method and LSH-core show much better results than the other two. This is due to that the feature to be considered as center indeed needs to be adjusted in order to maintain the best differentiation degree of feature space.

From Fig. 6, it can be seen that our method exhibits more effective when it comes to content altering manipulations. It can detect more unauthentic TIRI frames than other methods. Some interesting phenomenon can be observed here. First, when sizes of blocks or pasting contents grow, false positive rates decrease for all four methods. This can be understood that when overlapping blocks become larger, more perceptual contents of frames are distorted and it is much easier for hashing methods to detect those manipulations. In other words, when attacks on frames become fiercer, effects of those attacks surpass the thresholds of hash value comparison and unauthentic frames results are triggered. Second, our method has relatively lower false positive rates. The reason is that we take into consideration the robustness and security prior knowledge when we form the perceptual content hashing problem. Moreover, we put the security as the constraint condition. Robustness is divided into achievable and unachievable ones as in (5) and (6). Thus it shows much better performance when we authenticate video content.

Note that in all four methods, only our method and LSH-core need training to obtain optimal coefficients. In our simulation, we record the training and testing time for all video of four methods. The average testing time for one TIRI frame is 1.41 s, 1.35 s, 1.14 s and 1.09 s for our method, LSH-core, Radon-based and TIRI + DCT, respectively. Average training time is 2.84 s and 2.45 s for our method and LSH-core.

Although our method performs well in terms of security and robustness, it is slightly more time consuming, especially when real-time hashing requirement is needed.

5 Conclusion

In this paper we have proposed a video perceptual hashing method to authentication video content based on maximized robustness idea. The proposed maximized robustness means that video hashing should obtain its largest robustness property only the security requirement is first met. We have addressed the ambiguous problem between security and robustness properties in video hashing. First, we formulate the mathematical problem as a constrained optimization, where two coefficients, i.e., features offset and robustness should be decided. The optimization utilizes a priori knowledge of to what extent similar or dissimilar video under content preserving or manipulating operations should be. The constraints in the optimization strictly define the security characteristics of video hashing, which tells how robustness the hashing could achieve. The optimization is learned by a population based solving method. Two coefficients are learned simultaneously. We evaluate the proposed method on a video set and comparisons are conducted in terms of robustness and security. Experimental results show the superiority of our hashing method when it comes to video content authentication. Future work would be focused on video perceptual hashing based on deep learning, which takes into consideration the relationship between low-level and high-level semantics to enhance robustness and security.

Abbreviations

DCT: Discrete cosine transformation; SURF: Speed up robust feature; 3D: Three dimension; TIRI: Temporally informative representative images; ROC: Receiver operation curve.

Acknowledgements

Authors gratefully acknowledge the valuable comments and suggestions of reviewers, which have improved the presentation.

Authors' contributions

QM and LX designed the research and performed the research; QM and LX wrote the paper. All authors read and approved the final manuscript.

Funding

This research is supported by National Natural Science Foundation of China (Grant No. 62171180, No. 62071170) and Doctoral Research Foundation (Grant No. 17zx7158), and Program for Innovative Research Team in University of Henan Province (21IRTSTHN015).

Availability of data and materials

Please contact author for data requests.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China. [2]School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China.

Received: 29 October 2018 Accepted: 23 March 2019
Published online: 21 November 2021

References

1. Z. Tang, Y. Dai, X. Zhang et al., Robust image hashing via colour vector angles and discrete wavelet transform. IET Image Proc. **8**(3), 142–149 (2013)

2. L.N. Vadlamudi, R.P.V. Vaddella, V. Devara, Robust hash generation technique for content-based image authentication using histogram. Multimed. Tools Appl. **75**(11), 6585–6604 (2016)
3. Z. Tang, L. Huang, X. Zhang et al., Robust image hashing based on color vector angle and Canny operator. Int. J. Electron. Commun. **70**(6), 833–841 (2016)
4. Z. Tang, F. Yang, L. Huang et al., Robust image hashing with dominant DCT coefficients. Optik Int. J. Light Electron Opt. **125**(18), 5102–5107 (2014)
5. Y. Lei, Y. Wang, J. Huang, Robust image hash in Radon transform domain for authentication. Signal Process. Image Commun. **26**(6), 280–288 (2011)
6. Z. Tang, L. Ruan, C. Qin et al., Robust image hashing with embedding vector variance of LLE. Digital Signal Process. **43**, 17–27 (2015)
7. Q. Ma, L. Xu, L. Xing et al., Robust image authentication via locality sensitive hashing with core alignment. Multimed. Tools Appl. **77**(6), 7131–7152 (2018)
8. G. Yang, N. Chen, Q. Jian, A robust hashing algorithm based on SURF for video copy detection. Comput. Secur. **31**, 33–39 (2012)
9. X.S. Jun, Y.J. Quan, H.J. Wu, Perceptual video hashing robust against geometric distortions. Sci. China Inf. Sci. **55**(7), 1520–1527 (2012)
10. B. Coskun, B. Sankur, N. Memon, Spatio-tempereal tranfomation based video hashing. IEEE Trans. Multimed. **8**(6), 1190–1208 (2006)
11. M.M. Esmaeili, M. Fatourechi, R. Kreidieh, A robust and fast video copy detection system using content-based fingerprinting. IEEE Trans. Inf. Forens. Secur. **6**(1), 231–243 (2011)
12. J. Sun, J. Wang, J. Zhang et al., Video hashing algorithm with weighted matching based on visual saliency. IEEE Signal Process. Lett. **19**(6), 328–331 (2012)
13. X. Liu, J. Sun, J. Liu, Visual attention based temporally weighting method for video hashing. IEEE Signal Process. Lett. **20**(12), 1253–1256 (2013)
14. J.K. Song, L.L. Gao, L. Liu et al., Quantization-based hashing: a general framework for scalable image and video retrieval. Pattern Recogn. **75**, 175–187 (2018)
15. J.K. Song, Y.Y. Guo, L.L. Gao, et al., From deterministic to generative: multi-modal stochastic RNNs for video captioning. IEEE Trans. Neural Netw. Learn. Syst. **30**(10), 3047–3058 (2019)
16. J.K. Song, H.W. Zhang, X.P. Li et al., Self-supervised video hashing with hierarchical binary auto-encoder. IEEE Trans. Image Process. **27**(7), 3210–3221 (2018)
17. E. K. Yang, C. Deng, T. Liu, et al., Semantic structure-based unsupervised deep hashing. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 2018, International Joint Conferences on Artificial Intelligence: 1064–1070.
18. E. K. Yang, C. Deng, W. Liu, et al., Pairwise relationship guided deep hashing for cross-modal retrieval. In: 31st AAAI Conference on Artificial Intelligence, AAAI 2017, San Francisco, CA, United states, 2017, AAAI press: 1618–1625.
19. C. Deng, Z.J. Chen, X.L. Liu et al., Triplet-based deep hashing network for cross-modal retrieval. IEEE Trans. Image Process. **27**(8), 3893–3903 (2018)
20. E.K. Yang, T.L. Liu, C. Deng, et al., Adversarial examples for hamming space search. IEEE Trans. Cybernet. **50**(4), 1473–1484 (2020)
21. Y. C. Gong, S. Lazebnik. Iterative quantization: a procrustean approach to learning binary codes. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, Colorado, USA, 2011, IEEE: 817–824
22. L.L. Gao, Z. Guo, H.W. Zhang et al., Videos captioning with attention-based LSTM and semantic consistency. IEEE Trans. Multimed. **19**(9), 2045–2055 (2017)
23. X.H. Wang, L.L. Gao, P. Wang et al., Two-stream 3D convNet fusion for action recognition in videos with arbitrary size and length. IEEE Trans. Multimed. **20**(3), 634–644 (2018)
24. C.D. Roover, C.D. Vleeschouwer, F. Lefebvre et al., Robust video hashing based on radial projections of key frames. IEEE Trans. Signal Process. **53**(10), 4020–4037 (2005)
25. C.J.A. Bastos-Filho, D.O. Nascimento, An enhanced fish school search algorithm. In: 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence, Ipojuca, Brazil, IEEE: 152–157 (2013)
26. B. Xiao, J.T. Cui, H.X. Qin et al., Moments and moment invariants in the Radon space. Pattern Recogn. **48**(9), 2772–2784 (2015)

Publisher's Note