Multimed Tools Appl (2018) 77:5303–5321
DOI 10.1007/s11042-017-4434-2

Multi-granularity geometrically robust video hashing for tampering detection

Haichao Chen[1] **· Yan Wo**[1] **· Guoqiang Han**[1]

Received: 2 July 2016 / Revised: 4 January 2017 / Accepted: 20 January 2017 /
Published online: 6 March 2017

Abstract The wide-spread video editing tools make it much easier to tamper a video, which raises a huge need for authentication techniques that can prove the originality of video content and locate the tampered regions on the video sequences. In this paper, a multi-granularity geometrically robust video hashing method is proposed for tampering detection and localization. In order to balance the robustness and sensitiveness, we describe a video from three levels of granularity: frame sequence level, block level and pixel level, and then hashes are generated at these three levels. Polar Complex Exponential Transform (PCET) moments are calculated on the low-pass sub-band of 3D Discrete Wavelet Transform (3D–DWT) on frame sequence to extract geometric invariant spatio-temporal hash, which is used for video authentication. Local PCET moments are calculated on annular and angular blocks, which are used for geometric correction and coarse tampering localization. Position information of salient objects is obtained from saliency map for fine tampering localization. Experimental results show that the proposed method is robust against temporal de-synchronization and geometrical transformation, and has high tampering localization accuracy even when the video is rotated. Compared with state-of-the-art methods, it is more robust against content-preserving operations and more sensitive to malicious manipulations.

Keywords Video hashing · Multi-granularity · Tampering detection · Geometric invariant · Temporal de-synchronization

1 Introduction

With the widespread use of powerful video processing tools, it has become easy to tamper digital videos. An attacker may forge a video and use it as evidence during digital

✉ Yan Wo
woyan@scut.edu.cn

[1] College of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China

investigation. Developing automatic techniques for video tampering detection has become a big challenge for researchers.

Perceptual video hashing is one of the most popular video tampering detection techniques. A video can be mapped to a short binary string based on its content. If two videos have the same semantic content, they should be mapped to the same hash value, which means that the hash should be robust against content-preserving operations, such as compressing, blurring, geometrical transformation and temporal de-synchronization. On the other hand, the hash should be sensitive to malicious attacks which affect video's semantic content. As for tampering localization, the hash should give an estimate of the position where the content was manipulated. There are two reasons why tampering detection is generally preferred: on one hand, one can determine whether or not the video content is still acceptable for some applications; on the other hand, in some circumstances, tampered content needs to be restored to its original semantic. To sum up, the hash should be robust against content-preserving operations and sensitive to significant content modifications. Furthermore, if a tampering event happened, it should be able to locate the tampered regions.

In recent years, a number of video hashing methods have been proposed for video authentication. These methods fall into two categories: frame-based hashing and spatio-temporal based hashing.

Frame-based hashing treats a video as a set of image frames and a video hash is extracted from each frame using an image hashing technique. Lee et al. [7, 8] proposed a video hashing method based on Centroid of Gradient Orientations (CGO), which is robust against compression and noise degradation. However, its robustness against geometrical transformation is limited. Roover et al. [4] generated a rotation-invariant hash based on Radial projection (RASH). Lee [9] analyzed the affine covariance to construct a video hash, which is robust against both geometrical and non-geometrical transformations. Nie et al. [14] proposed an isometric feature mapping based video hashing method. Frame-based hashing generates a video hash without considering the temporal property, which makes it fragile to minor temporal de-synchronization.

Spatio-temporal based hashing regards a video as a whole rather than a trivial combination of frames. Coskun et al. [3] used the low frequency coefficients derived from 3D Discrete Cosine Transform (3D–DCT) to form a video hash. Using the temporal information of 3D–DCT, Malekesmaeili et al. [13] generated a video hash, which is robust against noise degradation and contrast adjustment. Willems et al. [20] used spatio-temporal interest points as robust video hash for video authentication. Li et al. [11] proposed a Low Rank Tensor Analysis (LRTA) based method, which applied multi-linear subspace projections on 3D–cubes to extract a robust video hash. Saikia et al. [17, 18] computed 1D–DCT on the low-pass sub-band of 3D Discrete Wavelet Transform (3D–DWT) to form a video hash. Spatio-temporal based hashing is robust against minor temporal de-synchronization, such as frame rate change [10]. However, these methods are sensitive to geometrical transformation. In fact, a video still has the same visual content after geometrical transformation such as scaling or rotation.

In some circumstances, if a tampering event happened, it is necessary to locate the tampering in the video. Unfortunately, few video hashing methods concern about tampering localization. Researchers proposed some methods for image tampering detection. Although these methods are fragile to minor temporal de-synchronization, they can be generalized to be frame-based hashing methods for video tampering detection. Zhao et al. [25, 26] used Zernike moments as global features to authenticate images, and used the position and texture information as local features to detect the tampered regions. OuYang et al. [15] proposed a Scale Invariant Feature Transform (SIFT) based hashing method. A rectangle bounding nonmatched key-points is considered as a forged region. Yan et al. [23] used SIFT features extracted on multi-scale round bins and angular bins to locate forged regions. Wang et al. [19] generated a hash from the whole saliency map and thus located the tampered regions precisely. But its hash length is tens of thousands of digits.

All video hashing methods mentioned above cannot simultaneously satisfy the following requirements: (1) be robust against allowable temporal and spatial operations, especially geometrical transformation; (2) be sensitive to illegal manipulations and (3) be able to locate the tampering precisely. Fulfilling these requirements is difficult because they are contradictory [16]. For example, being sensitive to small range of tampering requires the hash to describe the detail of the video, i.e. sensitive to small change of video content. But in this situation the hash would be sensitive to tolerant operations and thus fail to meet the robustness requirement. Being robust against temporal de-synchronization requires the hash to describe the global perceptual content of the video, while being sensitive to local malicious manipulations requires the hash to describe the local detail of the video. Being able to locate the tampering requires the hash to contain the position information. Based on previous analysis, in this paper, we propose a multi-granularity geometrically robust video hashing method for tampering detection and localization. We describe a video from three levels of granularity: frame sequence level, block level and pixel level. 3D–DWT and Polar Complex Exponential Transform (PCET) are applied to extract geometrically robust frame-sequence-level hash, which is used for video authentication. Local PCET moments are extracted on annular and angular blocks to form rotation-invariant block-level hash, which is used for geometric correction and coarse tampering localization. In this paper, we focus on the object-based tampering, such as adding object, deleting object, etc. Thus position information of salient objects is extracted as pixel-level hash for fine tampering localization.

Our contributions are fourfold: (1) extract video hashes from multiple granularities to balance the robustness and sensitiveness; (2) incorporate 3D–DWT and PCET to extract geometric invariant spatio-temporal features; (3) a geometric correction technique based on the hash of angular blocks is proposed to locate the tampered regions in the rotated frame; (4) a coarse-to-fine tampering localization strategy is proposed to improve the detection accuracy.

The rest of this paper is organized as follows. 3D–DWT and PCET are introduced in Section 2, and the proposed video hashing method is described in Section 3. Section 4 demonstrates the experiments. Finally, conclusion is presented in Section 5.

2 3D discrete wavelet transform and polar complex exponential transform

2.1 3D discrete wavelet transform

3D–DWT can capture the video's temporal property [5, 6]. One level 3D–DWT is obtained by applying three separate 1D transforms along the coordinate axes of a video. Let LA↓ denote the low-pass filtering and down sampling operations, HA↓ denote the high-pass filtering and down sampling operations, a single level 3D–DWT is illustrated in Fig.1.

The lowest frequency wavelet sub-band LLL can be used as the input of the next level 3D–DWT.

2.2 Polar complex exponential transform

PCET is one of the Polar Harmonic Transforms (PHT) proposed by Yap [24], whose kernel function is simpler than Zernike moments. In our former research [22], we found that PCET

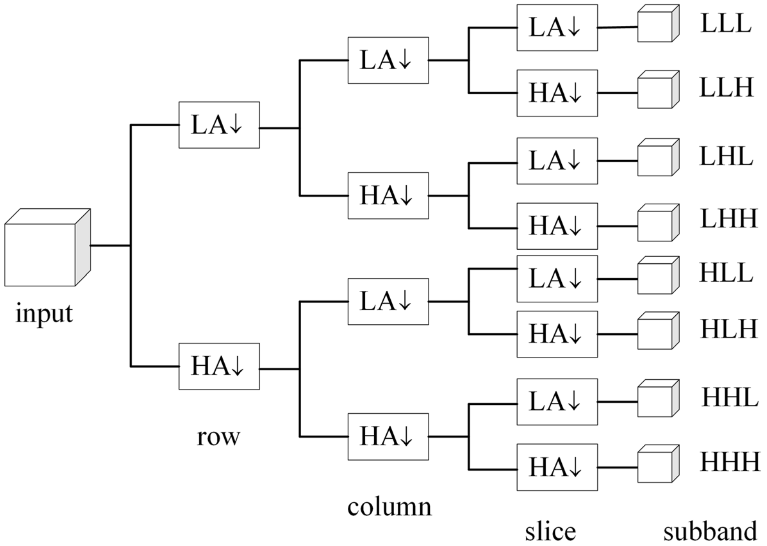


Fig. 1 Single level 3D–DWT

has the best reconstruction performance and rotational invariance. PCET is a 2D transform defined over the unit circle in the polar coordinate system. It can be expressed as:

$$M_{nl} = \Omega_n \int_0^{2\pi} \int_0^1 [H_{nl}(r,\theta)]^* f(r,\theta) \mathrm{rdrd}\theta \tag{1}$$

$$\Omega_n = \begin{cases} 1\Big/\pi, n = 0 \\ 2\Big/\pi, n > 0 \end{cases} \tag{2}$$

where $[\cdot]^*$ is the complex conjugate; n and l are order and repeatability of PHT respectively, with $n, |l| = 0, 1, \cdots \cdots, +\infty$. $H_{nl}(r,\theta) = R_n(r)e^{il\theta}$ and $R_n(r) = e^{i2\pi nr^2}$. If the image is rotated by angle φ, PCET moments of the rotated image are $\{M_{nl}^{\varphi}\}$. According to (1), $\{M_{nl}\}$ and $\{M_{nl}^{\varphi}\}$ has the relation:

$$M_{nl}^{\varphi} = M_{nl} e^{-il\varphi} \tag{3}$$

According to (3), PCET is rotation-invariant, i.e. $|M_{nl}^{\varphi}| = |M_{nl}|$. Scaling invariance can be obtained by image normalization.

3 Proposed method

3D–DWT captures the spatio-temporal property of a video, while with image normalization, PCET extracts scaling and rotation invariant features. Base on this, we propose a multi-granularity video hashing method for tampering detection and localization. Fig.2 shows the process of video hash generation. First of all, a video V is divided into several frame sequences. 3D–DWT is performed on each frame sequence to obtain the sub-band LLL, on which PCET

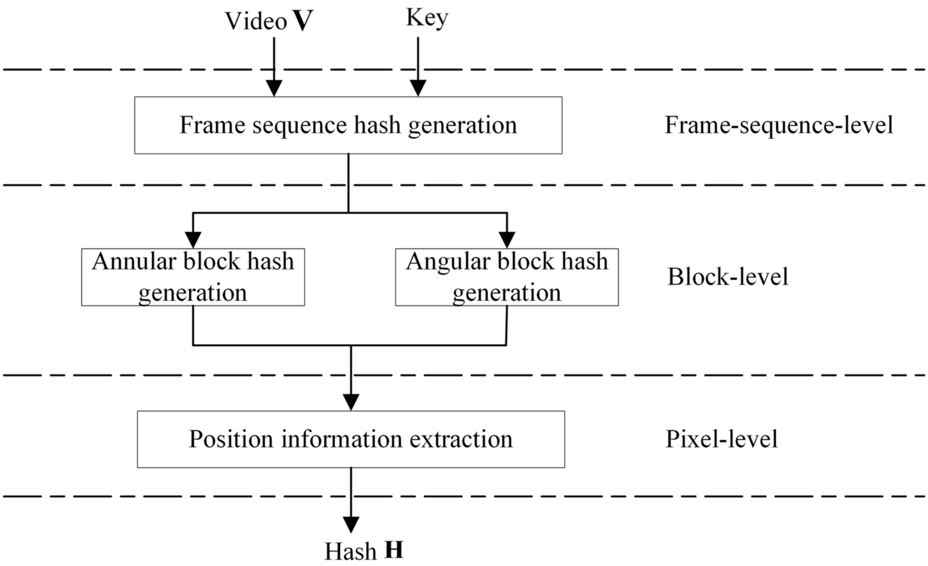

Video V
Key
Frame sequence hash generation
Frame-sequence-level
Annular block hash generation
Angular block hash generation
Block-level
Position information extraction
Pixel-level
Hash H

Fig. 2 Multi-granularity video hash generation

moments are calculated. Then a key-based random matrix projection is performed on these PCET moments to get the geometrically robust frame-sequence-level hash, which is used for video authentication. Secondly, to detect and locate small range of tampering, we need hash derived from a smaller granularity. Thus each frame is further divided into annular and angular blocks. Local PCET moments are calculated on these blocks and then randomly projected to form the block-level hash, which is used for geometric correction and coarse tampering localization. Thirdly, position information of salient objects is extracted as pixel-level hash, which is used for fine tampering localization. Finally, hashes of three granularity levels are concatenated to form the final video hash.

3.1 Frame-sequence-level hash generation

Frame-sequence-level hash is used for video authentication, and it can be generated as follows.

(1) **Video preprocessing.** Sample the video V to K frames, resize each frame to $I \times J$, thus $V \in \mathbb{R}^{I \times J \times K}$. Divide V into N frame sequences fs_i, $i = 1 , 2 , \ . \ . \ . N$. Each frame sequence fs_i has $s = \lfloor K/N \rfloor$ frames (see Fig.3).
(2) **Feature extraction.** Perform three-level 3D–DWT on each frame sequence fs_i $(1 \le i \le N)$ to get the lowest frequency sub-band LLL. Extract PCET moments on each frame of LLL and select L robust moments to form feature vector $\mathtt{FV}$ of fs_i.
(3) **Frame-sequence-level hash generation.** With a Gaussian random matrix $\mathtt{W}_1 \in \mathbb{R}^{m_f \cdot L}$ generated by a secrete key $\mathtt{Key}$, a key-based pseudorandom matrix projection [12] is performed on $\mathtt{FV}$ to generate m_f dimensions hash $\mathtt{H}_\mathtt{f}^\mathtt{i}$. Concatenate these hashes to obtain the frame-sequence-level hash of video V.

$$\mathtt{H}_\mathtt{f}^\mathtt{V} = \left\{ \mathtt{H}_\mathtt{f}^\mathtt{i} | i = 1, 2, ...N \right\} \tag{4}$$

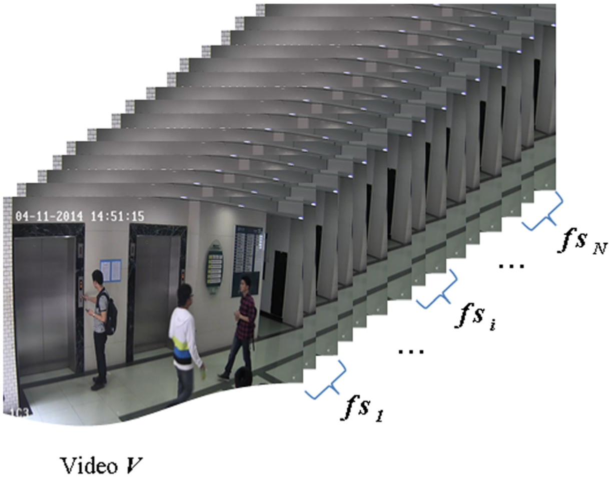


Fig. 3 Illustration of frame sequences dividing

3.2 Block-level hash generation

In order to generate hash which is robust against rotation and sensitive to small range of tampering, we generate block-level hash as follows.

(1) **Blocks division.** Divide each frame fs_i^j $(1 \leq i \leq N, 1 \leq j \leq s)$ of the i th frame sequence into N_{ann} annular blocks and N_{ang} angular blocks, as shown in Fig.4.

(2) **Annular block hash generation.** Let C_{ann} $(C_{ann} = 1/N_{ann})$ denote the width of each annular block and $R_x = (x-1)C_{ann}$, for each annular block block_x^{ann} of fs_i^j, calculate its PCET moments according to Eq.(5).

$$\mathtt{M}_{\mathtt{nl}}^{\mathtt{ann}}\left(\text{block}_x^{ann}\right) = \Omega_n \int_0^{2\pi} \int_{R_x}^{R_{x+1}} \left[H_{nl}(r,\theta)\right]^* f(r,\theta) \text{rdrd}\theta \qquad \text{where} \quad n, |l| \leq Q_{ann} \tag{5}$$

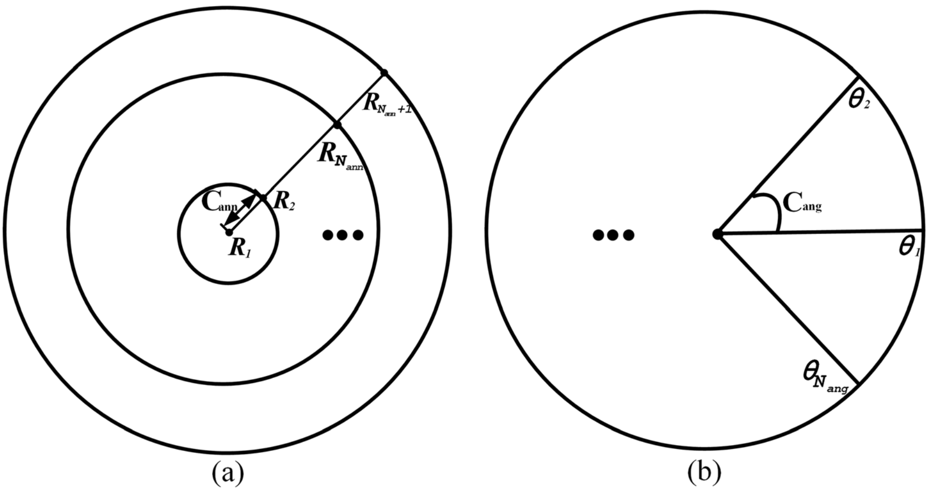


Fig. 4 Two blocking styles **a** Annular blocking **b** Angular blocking

Similar to frame-sequence-level hash generation, perform pseudorandom matrix projection on magnitude of PCET moments $\left\{\left|\mathtt{M}_{\mathtt{nl}}^{\mathtt{ann}}\left(\mathrm{block}_x^{ann}\right)\right| \quad | \; n,|l|\le Q_{ann}\right\}$ to generate a m_{ann} dimensions hash $\mathbf{H}_{\mathbf{x}}^{\mathbf{ann}}$.

(3) **Angular block hash generation**. Similarly, let C_{ang} ($C_{ang} = 2\pi/N_{ang}$) denote the size of each angular block and $\theta_y = (y-1)C_{ang}$, for each angular block block_y^{ang} of fs_i^j, calculate its PCET moments according to Eq.(6).

$$\mathrm{M}_{\mathrm{nl}}^{\mathrm{ang}}\left(\mathrm{block}_y^{ang}\right) = \Omega_n\int_{\theta_y}^{\theta_{y+1}}\int_0^1\left[H_{nl}(r,\theta)\right]^* f(r,\theta)\mathrm{rdrd}\theta \qquad \text{where} \quad n,|l|\le Q_{ang} \tag{6}$$

Perform key-based pseudorandom matrix projection on $\left\{\left|\mathtt{M}_{\mathtt{nl}}^{\mathtt{ang}}\left(\mathrm{block}_x^{ang}\right)\right| \quad | \; n,|l|\le Q_{ang}\right\}$ to generate m_{ang} dimensions hash $\mathbf{H}_{\mathbf{y}}^{\mathbf{ang}}$.

(4) **Block-level hash generation.** Concatenate all annular block hashes and angular block hashes of frame fs_i^j to form$\mathbf{H}_{\mathrm{b}}\left(fs_i^j\right)$.

$$\mathbf{H}_{\mathrm{b}}\left(fs_i^j\right) = \left\{\left\{\mathbf{H}_{\mathbf{x}}^{\mathbf{ann}}|x=1,\ldots N_{ann}\right\}, \quad \left\{\mathbf{H}_{\mathbf{y}}^{\mathbf{ang}}|y=1,\ldots N_{ang}\right\}\right\} \tag{7}$$

From (7), we can obtain the block-level hash of video V.

$$\mathrm{H}_{\mathrm{b}}^{\mathrm{V}} = \left\{\mathrm{H}_{\mathrm{b}}\left(fs_i^j\right)|i=1,\ldots N, j=1,\ldots s\right\} \tag{8}$$

3.3 Pixel-level hash generation

Salient objects, as critical video content, are more preferred to be tampered. Motivated by this fact, we keep the position information of salient objects as pixel-level hash, which is used for fine tampering localization.

We adopt our former method proposed in [21] to obtain a saliency map of each frame, which is then post-processed with binarization and morphological operations to obtain salient objects. In order to shorten the hash length, a bounding polygon with N_{vertex} vertexes is used to represent a salient object, as shown in Fig. 5. Adjusting the parameter

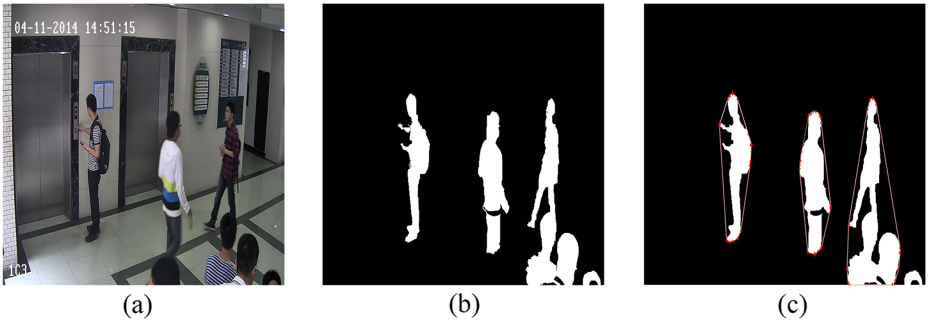


Fig. 5 Extraction of pixel-level hash **a** Tested frame **b** Salient objects **c** Bounding polygons

N_{vertex} can balance detection accuracy and hash length. The hash of a frame fs_i^j is computed as follows.

$$\boldsymbol{H_p}\left(fs_i^j\right) = \left\{p_t^{ver} | ver = 1, \dots N_{\mathrm{vertex}}, t = 1, \dots N_{\mathrm{poly}}\right\} \tag{9}$$

where p_t^{ver} denotes the position of the *ver* th vertex in the *t* th polygon, N_{poly} is the number of polygons.

Concatenate hashes of all frames to form the pixel-level hash of the video *V*.

$$\mathrm{H}_\mathrm{p}^\mathrm{V} = \left\{\mathrm{H}_\mathrm{p}\left(fs_i^j\right) | i = 1, \dots N, j = 1, \dots s\right\} \tag{10}$$

The final video hash H_V is expressed as follows.

$$\mathrm{H}_\mathrm{V} = \left[\mathrm{H}_\mathtt{f}^\mathrm{V} \quad \mathrm{H}_\mathrm{b}^\mathrm{V} \quad \mathrm{H}_\mathrm{p}^\mathrm{V}\right] \tag{11}$$

3.4 Video tampering detection scheme

For a received video *V'* and its hash $\mathrm{H}_v^{'}$, we compare $\mathrm{H}_v^{'}$ with H_v for video authentication and tampering localization. Frame-sequence-level hash $\mathrm{H}_f^{'}$ is used for video authentication, while block-level hash $\mathrm{H}_b^{'}$ and pixel-level hash $\mathrm{H}_p^{'}$ are used for coarse-to-fine tampering localization.

3.4.1 Video authentication

Video authentication determines whether a frame sequence of the received video is forged or not. For each frame sequence fs_i' of the received video *V'*, if the distance between $\mathrm{H}_\mathtt{f}^\mathtt{i}$ and $\mathrm{H}_\mathtt{f}^\mathtt{i}$ is greater than threshold τ_f, the received frame sequence fs_i' is classified as a forged frame sequence.

$$\mathrm{forged}(fs_i{}') = \begin{cases} 1 & \text{if } \mathrm{Dh}\left(\mathrm{H}_\mathtt{f}^\mathtt{i}, \mathrm{H}_\mathtt{f}^\mathtt{i}\right) \geq \tau_f, \\ 0 & \text{else} \end{cases}, \quad 1 \leq i \leq N \tag{12}$$

where Dh(·) indicates the 2-norm distance.

3.4.2 Tampering localization

Once fs_i' is judged as a forged frame sequence, we need to know the exact position of the tampering. The procedure of tampering localization is described as follows.

(1) Geometric correction

In order to locate the tampering in a rotated frame, geometric correction is needed to be performed before spatial tampering localization. It detects the block offset δ by minimizing the objective function.

$$\arg\min_{\delta} \sum_{q'=\delta}^{N_{ang}+\delta} \sum_{q=1}^{N_{ang}} \mathrm{Dh}\left(\mathrm{H}_q^\mathtt{ang},\ \mathrm{H}_{\mathbf{mod}\left(q', N_{ang}\right)}^\mathtt{ang}\right) \tag{13}$$

where $\mathrm{H}_{\mathrm{q}}^{\mathrm{ang}}$ is the q th angular block hash of the original video V, $\mathrm{H}_{\mathbf{mod}(q',N_{ang})}^{\mathrm{ang}}$ is the mod(q', N_{ang}) th angular block hash of the received video V'. mod(q', N_{ang}) denotes the modular function on q' and N_{ang}.

After solving out δ, with known angular block size C_{ang}, we obtain the estimated rotated angle φ according to Eq.(14).

$$\varphi = (\delta - 1)C_{ang} \tag{14}$$

(2) Coarse-to-fine tampering localization

After geometric correction, we then determine whether the block of each suspected frame is forged or not. For a received annular block block_x^{ann}, $1 \le x \le N_{ann}$ (or angular block block_y^{ang}, $1 \le y \le N_{ang}$), if the distance between $\mathrm{H}_{\mathrm{x}}^{\mathrm{ann}}$ ($\mathrm{H}_{\mathrm{y}}^{\mathrm{ang}}$) and $\mathrm{H}_{\mathrm{x}}^{\mathrm{ann}}$ ($\mathrm{H}_{\mathrm{y}}^{\mathrm{ang}}$) is greater than a given threshold τ_{ann} (τ_{ang}), the received block is classified as a forged block. If two forged blocks block_x^{ann} and block_y^{ang} satisfy the requirement $\mathrm{block}_x^{ann} \cap \mathrm{block}_y^{ang} \ne \varnothing$, we obtain the tampered sector block $\mathrm{block}_k = \mathrm{block}_x^{ann} \cap \mathrm{block}_y^{ang}$, as shown in Fig. 6. All $block_k$ form the forged block set BS.

To fine tune BS, pixel-level hashes $\mathrm{H}_{\mathrm{p}}^{\mathrm{V}}$ and $\mathrm{H}_{\mathrm{p}}^{\mathrm{V}}$ are compared to determine whether a polygon (salient object) is forged or not. Let PS denote all the pixels in the polygons represented by $\mathrm{H}_{\mathrm{p}}^{\mathrm{V}}$, PS' denote all the pixels in the polygons represented by $\mathrm{H}_{\mathrm{p}}^{\mathrm{V}}$, a pixel p is forged if it locates in the intersection of BS and the union regions of PS and PS'. All forged pixels form the final detection result DR.

$$DR = \{\ p | p\ \in BS \cap\ (PS \cup PS')\} \tag{15}$$

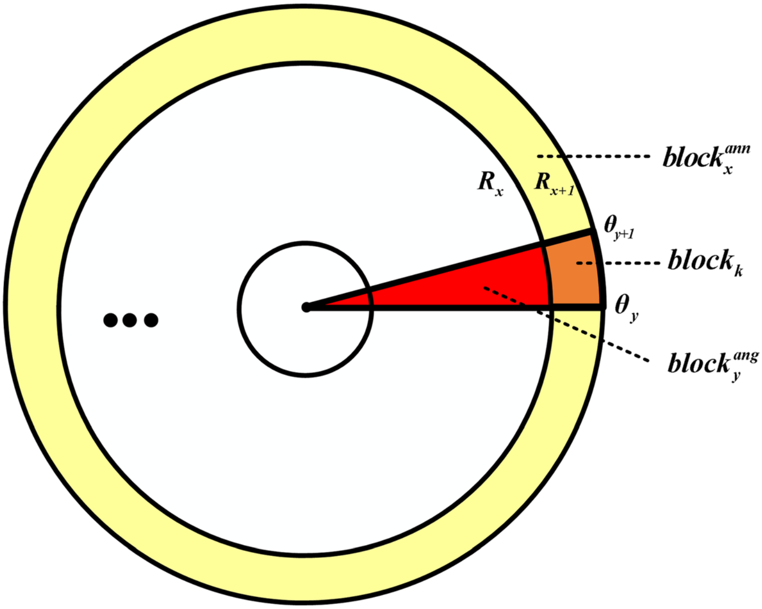


Fig. 6 Illustration of block $block_k$ intersected by block_x^{ann} and block_y^{ang}

4 Experimental results

Proposed hashing method is evaluated from two aspects: video authentication and tampering localization. We compare our method with CGO [8], Rash [4], 3D–DCT [3], LRTA [11], 3D–DWT [17], Zernike method [26] and SIFT method [23]. Since only Zernike method and SIFT method can locate the tampered regions, we compare our method with these two methods in tampering localization test.

4.1 Dataset and parameters setting

4.1.1 Dataset

Dataset SYSU-OBJFORG [2] contains 100 pristine videos and 100 corresponding tampered videos. All videos are 1280 × 720, H.264/MPEG-4 encoded with frame rate of 25 frames per second (fps). Each tampered video clip contains one or two forged segments which lasting from one to five seconds. The tampered types include object addition and object deletion. No perceptive traces can be easily found in forged frames.

Each similar version of frame sequences is processed with temporal de-synchronization, rotation or blurring as listed in Table 1 to construct the corresponding similar dataset.

4.1.2 Parameters setting

Experiments have been performed on Matlab R2014a, Windows 7 PC with Intel Core i5 CPU 3.30 GHz. Half of the dataset is used for parameters determination. By comprehensive considering both the ROC performance and hash length, the optimal parameters are found by grid search, and then the performance over the whole dataset is reported.

Each video is resized to 256 × 256 × 270, and is divided into 10 frame sequence, thus each frame sequence has 27 frames, i.e.$s = 27$.

When generate frame-sequence-level hash, set $L = 60$ and $m_f = 5$. It means that 60 features are selected, and the hash length of each frame sequence is 5.

Each frame is divided into 5 annular blocks and 24 angular blocks, i.e. $N_{ann} = 5$, $N_{ang} = 24$. When generate the block-level hash, set $Q_{ann} = 5$ and $Q_{ang} = 5$, $m_{ann} = 4$ and $m_{ang} = 2$. Thus the block-level hash length is 68.

Because the number of salient objects is not more than three in dataset, we choose $N_{poly} = 3$, each polygon has $N_{vertex} = 8$ vertexes. Pixel-level hash length is 48.

Table 1 Content-preserving operations and parameters setting

Operations	Parameters setting
Frame rate change	Subsampling by a factor of 2, 3, 4
Rotation	Degree: 5, 15, 30, 45–315 with step 45
Scaling	Scaling factor: 0.5, 0.8, 0.9, 1.1, 1.4, 1.5, 1.7, 2.0
JPEG compression	Quality factor: 10, 30, 50, 70, 90
Contrast adjustment	Low_in: 0.1,0.2,0.3,0.4 with high_in: 1
Gaussian noise	Mean: 0, variance: 0.002–0.01 with step 0.002
Salt and pepper noise	Variance: 0.02–0.1 with step 0.02
Gaussian blurring	Filter size: 3 × 3 to 9 × 9 with step 2
Median blurring	Filter size: 3 × 3 to 9 × 9 with step 2

4.2 Performance of video authentication

Video authentication determines whether a received frame sequence is forged or not. For it is a binary decision-making process, the receiver operating characteristics (ROC) curve, created by plotting the true positive rate *(TPR)* against the false positive rate *(FPR)* at various thresholds, is employed to evaluate its performance. Let $\mathrm{H}_{\mathtt{f}}^{\mathtt{V}}$, $\mathrm{H}_{\mathtt{f}}^{\mathtt{V}^*}$ and $\mathrm{H}_{\mathtt{f}}^{\mathtt{V}'}$ denote the hash of the original frame sequence, similar frame sequence and tampered frame sequence separately, we calculate *TPR* and *FPR* as follows.

$$TPR(\tau) = Pr\left(\mathrm{Dh}\left(\mathrm{H}_{\mathtt{f}}^{\mathtt{V}} - \mathrm{H}_{\mathtt{f}}^{\mathtt{V}^*}\right) < \tau\right) \tag{16}$$

$$FPR(\tau) = Pr\left(\mathrm{Dh}\left(\mathrm{H}_{\mathtt{f}}^{\mathtt{V}} - \mathrm{H}_{\mathtt{f}}^{\mathtt{V}'}\right) < \tau\right) \tag{17}$$

where the Pr(•) function indicates the classification probability.

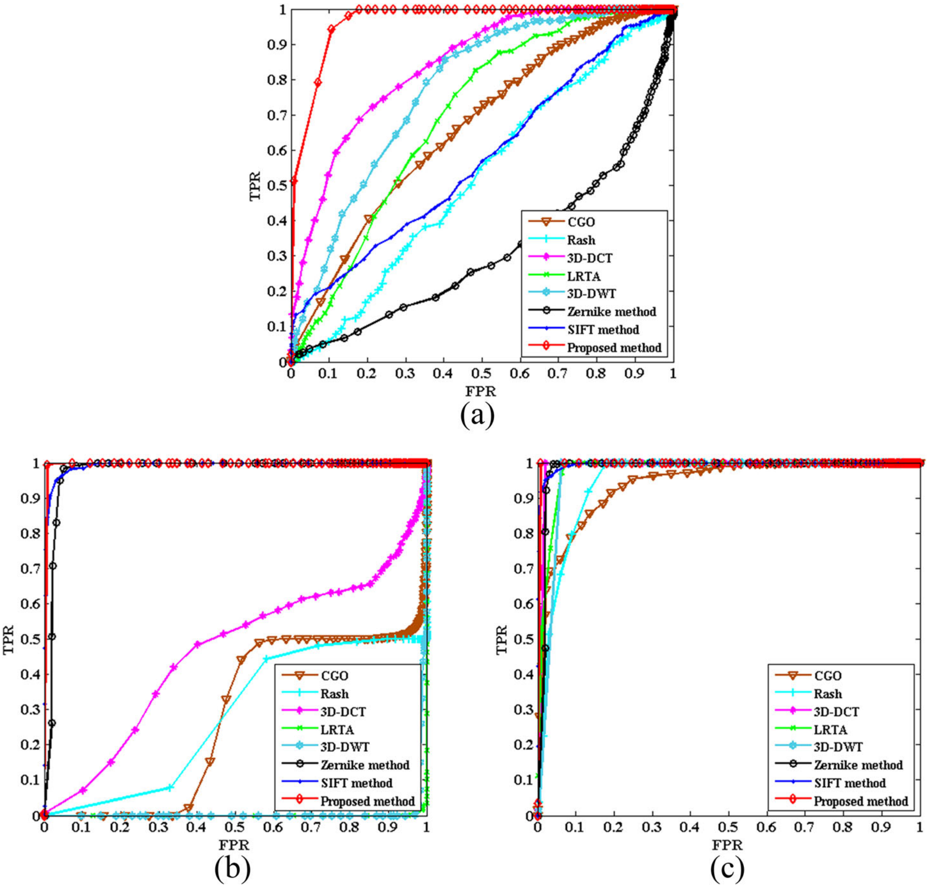
Original frame sequences and tampered frame sequences in SYSU-OBJFORG, and the corresponding similar frame sequences generated by performing tolerant operations (as shown in Table 1) on the original frame sequences are used in video authentication test. Parameters of compared methods are given by the relevant literature and are fine-tuned in our experiments.

According to [11], frame-based hashing method (CGO, Rash) used 2 dimensions per frame in video authentication, thus 2 × 27 = 54 dimensions pre frame sequence are used by CGO, Rash methods, while 128 dimensions per frame sequence are used by 3D–DCT, LRTA and 3D–DWT methods. In [26], Zernike method used 22 dimensions per frame, while in [23], SIFT method used 11 dimensions per frame. Thus 594 and 297 dimensions per frame sequence are respectively used by two hashing methods. The proposed method used only 5 dimensions per frame sequence in video authentication. Table 2 shows the hash length of each method for video authentication test. It shows that the proposed method uses less hash length than other methods.

Figure 7 shows the video authentication performance of the eight methods under frame rate change and geometrical transforms. From Fig. 7a we can see that frame-based methods, such as CGO, Rash, Zernike method and SIFT method which do not consider the temporal property of the video, perform not so well. Spatio-temporal based hashing methods such as the proposed method, 3D–DWT, 3D–DCT, LRTA are more robust against temporal de-synchronization than frame-based methods. The proposed method is particularly robust against temporal de-synchronization. Fig. 7b-c show the performance under geometrical transforms: rotation and scaling respectively. Fig. 7b shows that the proposed method, Zernike method and SIFT method show excellent robustness against rotation. 3D–DCT, CGO, Rash methods are robust against small angle rotation but sensitive to large angle rotation. Due to the image normalization, all hashing methods are robust against scaling, as shown in Fig. 7c. It can be seen from Fig. 7 that only the proposed method is robust against both temporal de-synchronization and geometrical transforms.

Table 2 Hash length for video authentication (unit: dimension per frame sequence)

	CGO [8]	Rash [4]	3D–DCT [3]	LATA [11]	3D–DWT [17]	Zernike method [26]	SIFT method [23]	Proposed method
Hash length	54	54	128	128	128	594	297	5



(a)

(b)

(c)

Fig. 7 Video authentication performance under **a** frame rate change **b** rotation and **c** scaling

Figure 8 shows the video authentication performance of the eight methods under JPEG compression, contrast adjustment, noise degradation and blurring. Noise degradation includes Gaussian noise degradation and salt and pepper noise degradation, while blurring includes Gaussian blurring and median blurring. From Fig. 8a, we can see that all hashing methods show excellent robustness against JPEG compression. But the performance of these methods degrades under contrast adjustment. From Fig. 8b it can be seen that the proposed method shows the best robustness against contrast adjustment over other methods. Since the LLL band of 3D–DWT filters out most of the high frequency noise, the extracted features are robust against noise degradation (shown in Fig. 8c). Fig. 8d shows that all hashing methods are robust against blurring.

4.3 Performance of tampering localization

Tampering localization locates the tampered regions in a frame. To analyze its performance, more quantitative indicators such as precision, recall and F1 score are adopted, which are often-used measures in the field of information retrieval [1]. Let *TP* denote the number of detected points in forgery region, *TN* denote the number of undetected points in non-forged

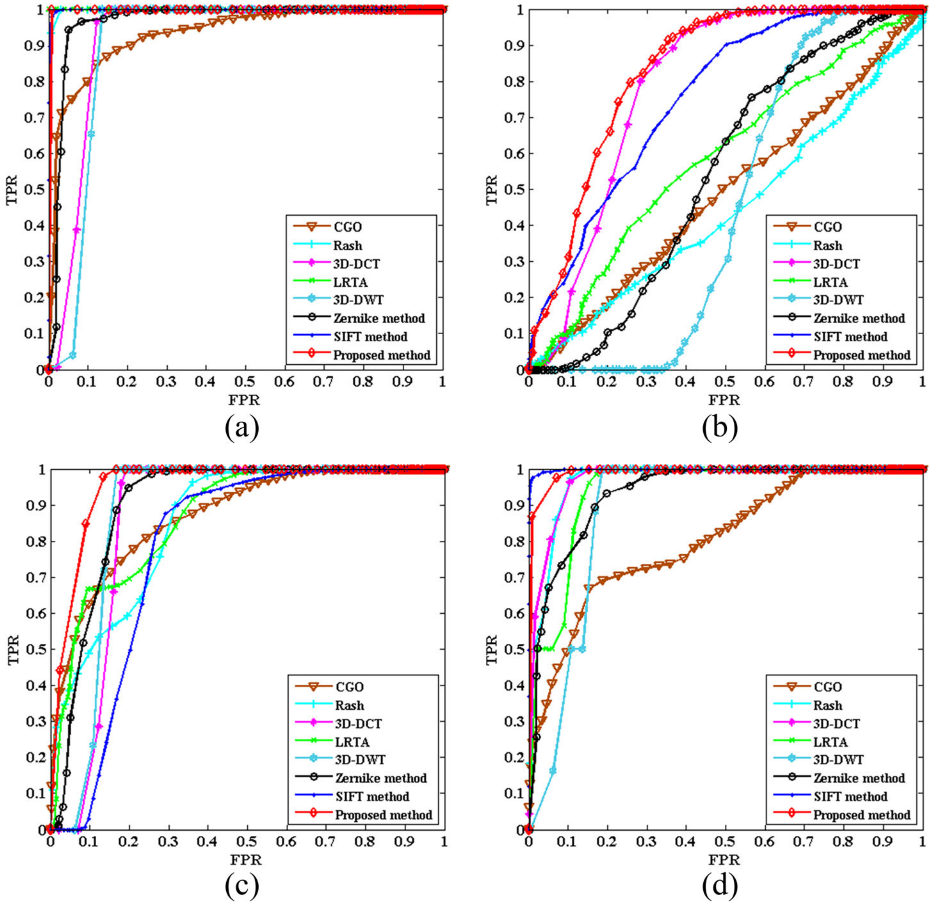


Fig. 8 Video authentication performance under different tolerant operations **a** JPEG compression **b** Contrast adjustment **c** Noise degradation **d** Blurring

region, *FP* denote the number of detected points in non-forged region and *FN* denote the number of undetected points in forgery region, we calculate the precision, recall, F1 as follows.

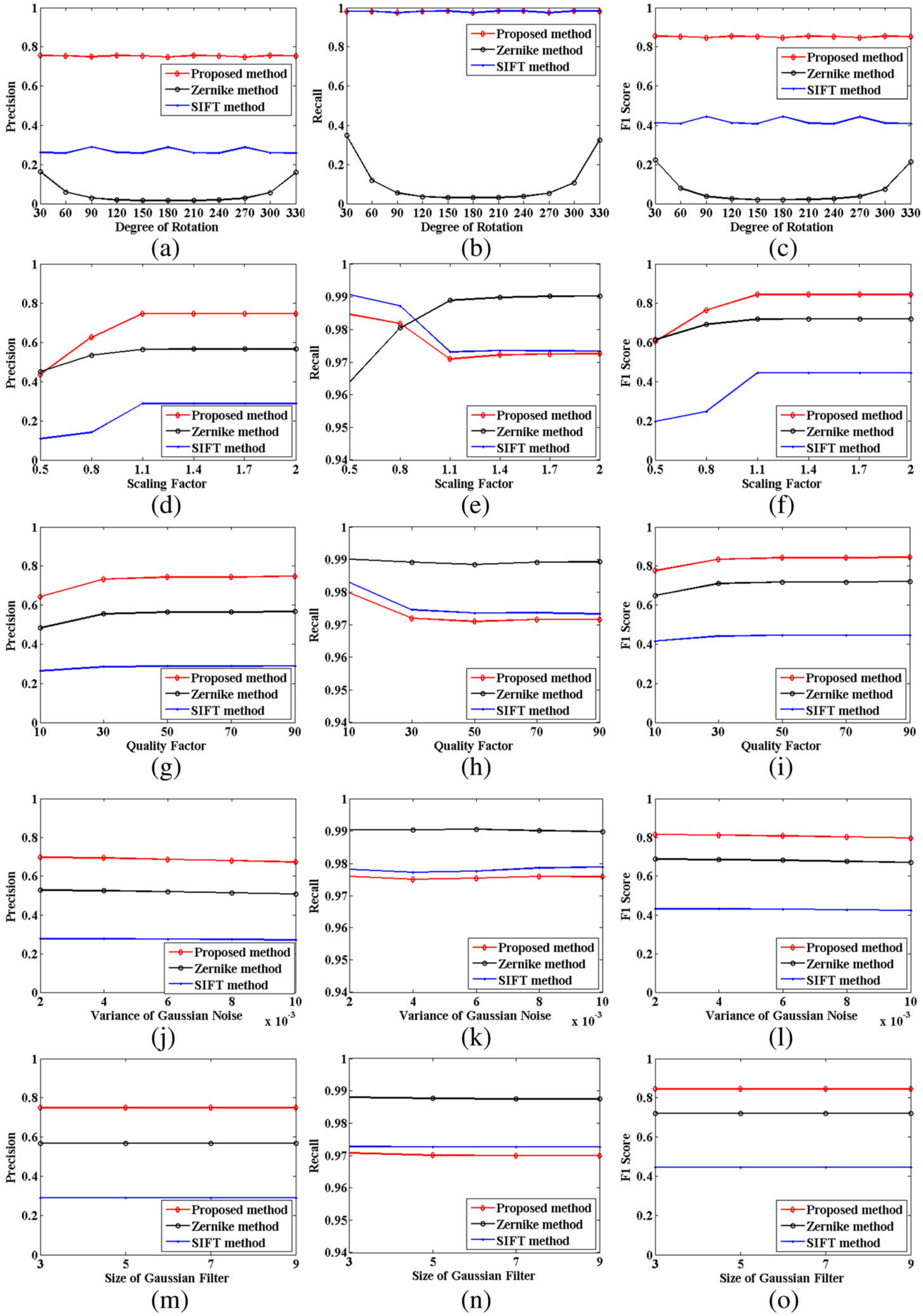
$$\text{precision} = \frac{TP}{TP + FP} \tag{18}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{19}$$

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{20}$$

The higher the precision, the fewer mismatches are detected. The higher the recall, the more complete forged regions are detected. The higher the *F*1, the better performance is achieved. All the 10,440 forged frames in dataset and their corresponding original frames are used in the tampering localization test. Each forged frame is processed with rotation, scaling or blurring as listed in Table 1. Zernike method and SIFT method are compared with the proposed method.

Figure 9 shows the tampering localization performance of three hashing methods under various tolerant operations. Rotation, scaling, JPEG compression, Gaussian noise degradation



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Fig. 9 Tampering localization performance under various tolerant operations. Rotation, scaling, JPEG compression, Gaussian noise degradation and Gaussian blurring are shown from top to bottom. Precision, recall and F1 are shown from left to right

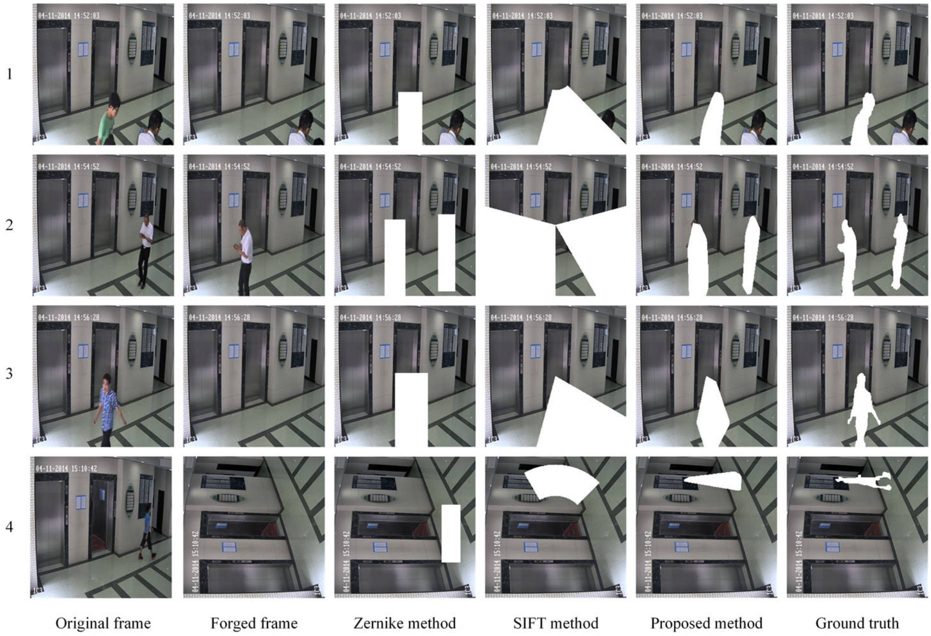


Fig. 10 The tampering localization results

and Gaussian blurring are shown from top to bottom, while precision, recall and F1 score are shown from left to right. Column 1 shows the precision rate obtained by each hashing method under content-preserving operations. It shows that the proposed method detects the tampered regions more precisely than other methods. Based on a coarse-to-fine strategy, the proposed method reduces lots of mismatches and has the highest precision. SIFT method has the worst performance since the sector blocking strategy is improper to detect rectangle-like object like human body. It causes a low precision rate because too many non-forged areas are detected as forged. Column 2 shows recall rate obtained by each hashing method, indicating that Zernike method detects the most complete forged regions. This is because the bounding box contains the most complete of the object. Column 2 also shows that more than 96% forged regions are detected by three hashing methods. Column 3 shows F1 score obtained by each hashing method. It shows that the proposed method performs better than other two methods in locating the tampered regions. Although other methods have higher recalls, their low precisions lower their F1 scores. It can be concluded from Fig.9 that the proposed method shows the best performance in tampering localization under various tolerant operations.

Figure 10 displays the detection results obtained by three hashing methods. It is obvious that the proposed method reduces most of mismatches and therefore obtains the highest precision over the other two methods. With a coarse-to-fine strategy, the proposed method retrieves

Table 3 Hash length for tampering localization (unit: dimension per frame)

	Zernike method [26]	SIFT method [23]	Proposed method
Hash length	48	291	116

Table 4 Time consumption of hash generation (unit: second per frame)

	CGO [8]	Rash [4]	3D–DCT [3]	LRTA [11]	3D–DWT [17]	Zernike method [26]	SIFT method [23]	Proposed method
Video authentication	0.11	0.22	0.03	0.03	0.04	0.03	0.03	0.02
Tampering localization	/	/	/	/	/	2.44	0.69	3.09

excellent detection accuracy. Sector-blocking strategy used by SIFT method introduces many mismatches. Because Zernike method does not restore the rotated image before tampering localization, it cannot detect the tampering in a rotated frame, as shown in Fig. 10 row 4. SIFT method and the proposed method correct the rotated angle before tampering localization, thus these two methods detect the tampered regions accurately in the rotated frame.

Table 3 shows the hash lengths of different methods used for tampering detection test. In the experiments, it takes the proposed method 68 dimensions to do geometric correction, 48 dimensions for polygons to represent salient objects, which makes the proposed method generate longer hash than Zernike method. However, the geometric correction helps the proposed method gain higher detection accuracy under rotation than Zernike method, as shown in Fig.9a-c and Fig.10 (row 4); while the strategy of using polygons improves the localization accuracy no matter what the shape of the object is. Zernike method shows good localization performance for rectangle-like objects (such as human bodies), but for non-rectangle-like objects, it contains too many mismatches and therefore its localization performance degrades. In conclusion, our hashing method generates a longer video hash, but performs better in detecting the any-shape object in any-angle-rotated frame.

4.4 Time consumption

Table 4 shows that the proposed method consumes less time in generating hash for video authentication, which is mainly because we extract features at frame sequence level rather than frame level (CGO, Rash, Zernike method, SIFT method). However, the proposed method and Zernike method consume more time than SIFT method in generating hash for tampering localization. The time consumption of the proposed method focuses on: (1) Calculating PCET moments on each block of each frame; (2) Obtaining a saliency map for each frame.

5 Conclusion

In this paper, a multi-granularity video hashing method is proposed for tampering detection. To balance the robustness and sensitiveness, we generate a video hash from different levels of granularity (frame sequence level, block level, pixel level). Frame-sequence-level hash, which is generated by 3D–DWT and PCET, shows particularly robustness against temporal desynchronization and geometrical transformation. Hashes of block-level and pixel-level are used for geometric correction and coarse-to-fine tampering localization, which obtains an excellent tampering localization results. Our video hash describes the video from multiple aspects, while the hash generation time cost is higher than other methods. Our future work is to speed up our method by using GPU acceleration approach.

Acknowledgements This paper is supported by National Natural Science Foundation of Guangdong (No.2016A030313472); National Natural Science Foundation of China (Grant No. 61472145); Guangdong Province Special funds for University-Industry Cooperation (No. 2013B090500015); Science and Technology Planning Project of Guangdong Province, China (No. 2016B090918042).

References

1. Ceri S, Bozzon A, Brambilla M et al (2013) An introduction to information retrieval [M]//web information retrieval. Springer, Berlin Heidelberg, pp 3–11
2. Chen S, Tan S, Li B, et al. (2015) Automatic Detection of Object-based Forgery in Advanced Video [J]
3. Coskun B, Sankur B, Memon N (2006) Spatio–temporal transform based video hashing [J]. ieee Trans on Multimed 8(6):1190–1208
4. De Roover C, De Vleeschouwer C, Lefèbvre F et al (2005) Robust video hashing based on radial projections of key frames [J]. IEEE Trans Signal Process 53(10):4020–4037
5. Karthikeyan A, Saranya P, Jayashree N. (2013) An Efficient VLSI Architecture for 3D DWT Using Lifting Scheme [J]. International Journal of Engineering Science and Innovative Technology (IJESIT) Volume, 2
6. Kumar C A, Madhavi B K, Lalkishore K. (2016) Pipeline and parallel processor architecture for fast computation of 3D–DWT using modified lifting scheme [C]//Wireless Communications, Signal Processing and Networking (WiSPNET), International Conference on. IEEE, 2123–2128
7. Lee S, Yoo C D (2006) Video fingerprinting based on centroids of gradient orientations [C]// (2006) IEEE International Conference on Acoustics Speech and Signal Processing Proceedings. IEEE, 2: II-II
8. Lee S, Yoo CD (2008a) Robust video fingerprinting for content-based video identification [J]. IEEE Trans on Circuits Syst for Video Technol 18(7):983–988
9. Lee S, Yoo C D (2008b) Robust video fingerprinting based on affine covariant regions [C]// 2008 I.E. International Conference on Acoustics, Speech and Signal Processing. IEEE, 1237–1240
10. Li M, Monga V. (2011) De-synchronization resilient video fingerprinting via randomized, low-rank tensor approximations [C]//Multimedia Signal Processing (MMSP), 2011 I.E. 13th International Workshop on. IEEE, 1–6
11. Li M, Monga V (2012) Robust video hashing via multilinear subspace projections [J]. IEEE Trans Image Process 21(10):4397–4409
12. Lu W, Varna AL, Wu M (2014) Confidentiality-preserving image search: a comparative study between homomorphic encryption and distance-preserving randomization [J]. IEEE Access 2:125–141
13. Malekesmaeili M, Fatourechi M, Ward R K. (2009) Video copy detection using temporally informative representative images [C]//Machine Learning and Applications, 2009. ICMLA'09. International Conference on. IEEE, 69–74
14. NIE X, LIU J, SUN J et al (2011) Key-frame based robust video hashing using isometric feature mapping [J]. J Comput Inf Syst 7(6):2112–2119
15. Ouyang J, Liu Y, Shu H (2016) Robust hashing for image authentication using SIFT feature and quaternion Zernike moments [J]. Multimed Tools and Appl:1–18
16. Roy S, Sun Q (2007) Robust hash for detecting and localizing image tampering [C]//2007 I.E. International Conference on Image Processing. IEEE, 6: VI-117-VI-120.
17. Saikia N. (2015) Perceptual hashing in the 3D–DWT domain [C]//Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on. IEEE, 694–698.
18. Saikia N, Bora P K (2011) Robust video hashing using the 3D–DWT [C]//Communications (NCC), 2011 National Conference on. IEEE, 1–5.
19. Wang X, Pang K, Zhou X et al (2015) A visual model-based perceptual image hash for content authentication [J]. IEEE Trans on Inform Forensics and Sec 10(7):1336–1349
20. Willems G, Tuytelaars T, Van Gool L. (2008) Spatio-temporal features for robust content-based video copy detection [C]//Proceedings of the 1st ACM international conference on Multimedia information retrieval. ACM, 283–290
21. Wo Y, Chen X, Han G (2015) A saliency detection model using aggregation degree of color and texture [J]. Signal Process Image Commun 30:121–136
22. Yan W, Jiao X (2012) Accurate and fast harmonic transform of polar coordinates [J]. J South China Univ Technol: Nat Sci Ed 40(4):23–29
23. Yan CP, Pun CM, Yuan XC (2016) Multi-scale image hashing using adaptive local feature extraction for robust tampering detection [J]. Signal Process 121:1–16
24. Yap PT, Jiang X, Kot AC (2010) Two-dimensional polar harmonic transforms for invariant image representation [J]. IEEE Trans Pattern Anal Mach Intell 32(7):1259–1270
25. Zhao Y, Wang S, Feng G et al (2010) A robust image hashing method based on Zernike moments [J]. J Comput Inf Syst 6(3):717–725
26. Zhao Y, Wang S, Zhang X et al (2013) Robust hashing for image authentication using Zernike moments and local features [J]. IEEE Trans on Inform Forensics and Secur 8(1):55–63

Haichao Chen received the B.S. degree in Computer Science from South China University of China in 2014. He is currently pursuing the master degree in Computer Science from South China University of Technology. His research interests include image processing, multimedia content analysis and computer vision.

Yan Wo received the M.S. degree in Computer Science from Lanzhou University, in 1999, and a Ph.D. degree in Computer Science from South China University of Technology in 2004. She is now a Professor of Department of Computer Science and Engineering of South China University of Technology, China. Her current research interests are in the fields of image processing, information security, and pattern recognition.

Guoqiang Han received the M.S. degree in Computer Science from Sun Yat-sen University, in 1985, and a Ph.D. degree in Computer Science from Sun Yat-sen University in 1988. He is now a Professor of Department of Computer Science and Engineering of South China University of Technology, China. His recent work concentrates on digital signal processing, and information security of digital media.