Multidim Syst Sign Process
DOI 10.1007/s11045-017-0496-6

A novel passive forgery detection algorithm for video region duplication

Lichao Su[1] · **Cuihua Li**[1]

Received: 10 November 2015 / Revised: 17 January 2017 / Accepted: 2 May 2017

Abstract Forgery involving region duplication is one of the most common types of video tampering. However, few algorithms have been suggested for detecting this type of forgery effectively, especially for videos to which a mirroring operation was applied. In this paper, we summarize the properties of duplication forgery of video regions and propose a novel algorithm to detect this forgery. First, the algorithm extracts the feature points in the current frame. The tampered areas in the current frame are then searched, which is implemented in three steps. Finally, our algorithm detects the tampered areas in the remaining frames using spatio-temporal context learning and outputs the detection results. The experimental results demonstrate the satisfactory performance of our algorithm for detecting videos subjected to mirror operations and its higher efficiency than previous algorithms.

Keywords Video forgery · Region duplication · Mirror invariant · Passive forensics

1 Introduction

With the rapid development of multimedia technology and user-friendly editing software (e.g., Photoshop and Premiere by Adobe, and Mokey by Imagineer Systems), it is now simple to collect and tamper with videos. This tampering may greatly alter the original video and mislead audiences. Therefore, technology that can determine the authenticity of a video is needed. Consequently, the field of multimedia forensics has emerged to authenticate the veracity and integrity of images or videos (Sencar and Memon 2008; Yin et al. 2012).

Video forensics activity can be divided into main two categories: active forensics and passive forensics. In active forensics, the validation information used for authentication is entered when generating the videos; this technique may be limited in applicability. In passive forensics, the veracity and integrity of a video can be authenticated without any validation

✉ Lichao Su
651424071@qq.com

[1] School of Information Science and Engineering, Xiamen University, Fujian, China

Published online: 09 May 2017

information; this is a more effective technique in practice (Milani et al. 2012; Al-Qershi and Khoo 2013). In recent years, passive forensics research on digital images has become a popular topic in the multimedia security community (Li et al. 2009; Cao et al. 2012), but video passive forensics has largely been ignored. Kobayashi et al. (2009) propose a method to detect forged regions in a video based on inconsistencies of noise characteristics in the forged areas. Hsu et al. (2008) suggest a method to detect video tampering using noise correlation properties between spatially collocated blocks. Other forgery detection techniques are also proposed in Wang and Farid (2009) and Chen and Shi (2009). The main concept of these techniques is to use double quantization coefficients to detect a forgery. In Yang et al. (2014), an effective similarity-analysis-based method is proposed, and the results show that it demonstrates outstanding performance in terms of time efficiency.

Video processing software is often used to copy and paste certain existing contents from one region to another disjoint region in the same frame, which is one of the most common methods of video forgery (an example is shown in Fig. 1). However, few algorithms have been suggested for detecting this tampering operation. Wang and Farid (2007) propose a divide-and-conquer approach: the entire video is split into subparts, and different types of correlation coefficients are computed to highlight similarities to detect duplication. However, the detection efficiency is not acceptable when the forged region is small. In Subramanyam and Emmanuel (2012), a detection algorithm for region-duplication forgery is proposed based on Histogram of Oriented Gradients (HOG) feature-matching and video compression properties. It is effective and robust against various signal processing manipulations. However, the algorithm is not suitable for long videos because of its large time consumption. In Pun et al. (2015), a novel copy–move forgery detection scheme is proposed using adaptive over-segmentation and feature point matching. The author first segments the image into non-overlapping and irregular blocks. Feature points are then extracted from each block, and the features are matched with one another to locate the forged regions. This algorithm combines a block-based method with a feature points-based method and performs well. To reduce the computational complexity, the video was processed by the down-sampling proposed in Pun et al. (2015). However, the down-sampling process makes the extraction of features more difficult and impairs the accuracy.

The issue of computational cost is central to the design of the algorithms because a video of even modest length can extend to tens of thousands of frames. This paper proposes a new framework to the forgery detection of region duplication in intra-frame and it improves efficiency greatly. The algorithm improves on MISIFT (Ma et al. 2010) to extract features

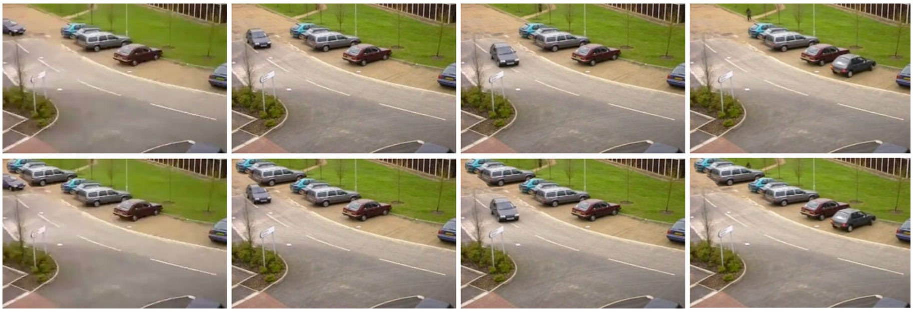

Fig. 1 An example of video intra-frame region duplication: the authentic video (*top*) and the tampered video (*bottom*)

that can be used as evidence to demonstrate the tampered areas under mirroring. Spatio-temporal context learning is first employed to locate the tampered areas in the remaining frames, avoiding the deficiency of detecting videos frame by frame. Finally, we conducted rigorous experiments to demonstrate the accuracy and efficiency of the proposed algorithm. The proposed algorithm is able to precisely locate duplicated regions without being affected by common post-processing attacks, such as rotation, scaling and mirroring and delivers outstanding performance in terms of time efficiency.

2 Preliminaries

2.1 Problem formulation

The tracking problem is formulated by computing a confidence map, which estimates the object location likelihood in (1):

$$m\left(\mathbf{x}\right) = P\left(\mathbf{x}\left|o\right.\right) \tag{1}$$

where $\mathbf{x} \in R^2$ is an object location and o denotes the object present in the scene. The context feature set is $X^c = \{c\left(z\right) = \left(I\left(z\right), z\right)\left|z \in \Omega_c\left(\mathrm{x}^*\right)\right.\}$, where $I\left(z\right)$ denotes the image intensity at location z and $\Omega_c\left(\mathrm{x}^*\right)$ is the neighborhood of location $\mathbf{x}^*$, which is twice the size of the target object. Therefore,

$$\begin{aligned} m\left(\mathbf{x}\right) &= P\left(\mathbf{x}\left|o\right.\right) \\ &= \sum\nolimits_{\mathbf{c}(\mathbf{z})\in X^c} P\left(\mathbf{x}, \mathbf{c}\left(\mathbf{z}\right)\left|o\right.\right) \\ &= \sum\nolimits_{\mathbf{c}(\mathbf{z})\in X^c} P\big(\mathbf{x}\left|\mathbf{c}\left(\mathbf{z}\right), o\right.\big) P\big(\mathbf{c}\left(\mathbf{z}\right)\left|o\right.\big) \end{aligned} \tag{2}$$

$P\left(\mathbf{x}\left|\mathbf{c}\left(\mathbf{z}\right), o\right.\right)$ is the conditional probability function, which is defined as

$$P\left(\mathbf{x}\left|\mathbf{c}\left(\mathbf{z}\right), o\right.\right) = h^{sc}\left(\mathbf{x}\text{-}\mathbf{z}\right) \tag{3}$$

where $h^{sc}\left(\cdot\right)$ is the non-radially symmetric function with respect to the relative distance and direction between an object's location and its local context location, which helps to resolve ambiguities effectively.

$P\left(\mathbf{c}\left(\mathbf{z}\right)\left|o\right.\right)$, which is related to the context appearance, is the context prior probability modeled as

$$P\left(\mathbf{c}\left(\mathbf{z}\right)\left|o\right)\right. = I\left(z\right)\omega_\sigma\left(z - \mathrm{x}^*\right) \tag{4}$$

where $I\left(\cdot\right)$ is the image intensity that represents the appearance of context, $\mathbf{z}$ is the context location, x^* is the currently tracked target location, and ω is a Gaussian weighted function defined as

$$\omega_\sigma\left(z - \mathrm{x}^*\right) = ae^{-\frac{|\mathrm{Z}-\mathrm{X}^*|^2}{\sigma^2}} \tag{5}$$

where a is a restrictive normalization constant. Equation (4) models the focus of attention that is motivated by the biological visual system that concentrates on certain image regions requiring detailed analysis (Torralba 2003).

The confidence map of an object location is modeled as

$$m\left(\mathbf{x}\right) = P\left(\mathbf{x}\left|o\right.\right) = be^{-\left|\frac{\mathbf{x}-\mathbf{x}^*}{\alpha}\right|^{\beta}} \tag{6}$$

where b is a normalization constant, a is a scale parameter, and β is a shape parameter.

Zhang et al. (2014) resolve the location-ambiguity problem by choosing a proper shape parameter β. A large β (e.g., $\beta = 2$) results in an over-smoothing effect for the confidence value at locations near the object center, failing to effectively reduce location ambiguities. A small β (e.g., $\beta = 0.5$) yields a sharp peak near the object center and activates fewer positions when learning the spatial context model. This result may lead to over-fitting in searching for the object location in the coming frame. Robust results have been obtained with $\beta = 1$ in our experiments.

2.2 Fast-learning spatial context model

Based on the confidence map function and the context prior model, our objective is to learn the spatial context model. Combining (6), (4) and (3), we formulate (2) as

$$\begin{aligned} m(\mathbf{x}) &= be^{-\left|\frac{\mathbf{x}-\mathbf{x}^*}{\alpha}\right|^{\beta}} \\ &= \sum_{\mathbf{z}\in\Omega_c(\mathbf{x}^*)} h^{sc}(\mathbf{x}-\mathbf{z})\, I(\mathbf{z})\, \omega_\sigma\left(\mathbf{z}-\mathbf{x}^*\right) \\ &= h^{sc}(\mathbf{x}) \otimes \left(I(\mathbf{z})\, \omega_\sigma\left(x-x^*\right)\right) \end{aligned} \tag{7}$$

where $\otimes$ denotes the convolution operator. Formula (7) can be transformed into the frequency domain by adopting the fast Fourier transform (FFT) algorithm (Oppenheim et al. 1983). That is,

$$F\left(be^{-\left|\frac{\mathbf{x}-\mathbf{x}^*}{\alpha}\right|^{\beta}}\right) = F\left(h^{sc}(\mathbf{x})\right) \otimes F\left(I(\mathbf{x})\, \omega_\sigma\left(\mathbf{x}-\mathbf{x}^*\right)\right) \tag{8}$$

where F denotes the FFT function and $\otimes$ is the element-wise product. Therefore, we obtain formula (9):

$$h^{sc}(\mathbf{x}) = F^{-1}\left(\frac{F\left(be^{-\left|\frac{\mathbf{x}-\mathbf{x}^*}{\alpha}\right|^{\beta}}\right)}{F\left(I(\mathbf{x})\, \omega_\sigma(\mathbf{x}-\mathbf{x}^*)\right)}\right) \tag{9}$$

where F^{-1} denotes the inverse FFT function. The spatial context model learns the relative spatial relations between different pixels in a Bayesian framework (Zhang et al. 2014).

3 Proposed method

The algorithm proposed in this paper consists of three parts: (1) feature extraction, (2) locating the tampered area in the current frame, and (3) locating the tampered area in the remaining frames. Figure 2 shows the specific process of the algorithm.

The first part of the algorithm extracts the feature points of the current frame and computes the similarities of all feature points. The second part of the algorithm summarizes the three properties of the intra-frame region duplication forgery in videos and designs a coarse-to-fine approach to confirm the tampered areas in the current frame. The third part of the algorithm studies the tampered areas obtained from the previous step, then locates the tampered areas in subsequent frames and outputs the final results.

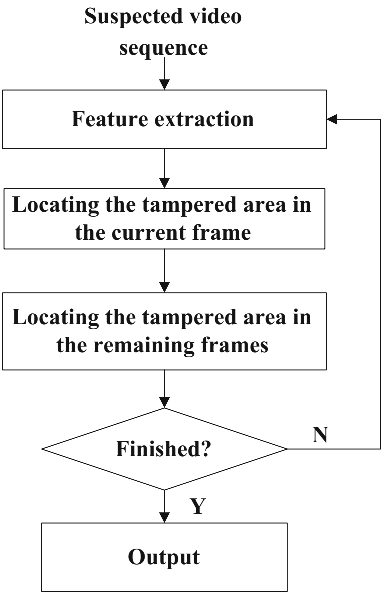
Suspected video sequence
Feature extraction
Locating the tampered area in the current frame
Locating the tampered area in the remaining frames
Finished?
N
Y
Output

Fig. 2 Flow chart of the proposed approach

3.1 Feature extraction

In an intra-frame region duplication forgery, the source and tampered regions are highly similar. However, impractical applications, attackers do not simply copy and paste one area to another; instead, to make the tampering inconspicuous, they often conduct some transformation in the tampered area such as rotation, scaling, mirroring, or mixing a variety of geometric transformations (post-processing) (Gao and Jin 2010). Therefore, Amerini et al. (2011) and Subramanyam and Emmanuel (2012) present algorithms to detect intra-frame forgery using SIFT/SURF (Lowe 2004) or HOG features (Dalal and Triggs 2005). However, these algorithms are not sufficiently capable for the following reasons:

(a) These algorithms cannot detect tampered areas subjected to mirror operation.
(b) SIFT or SURF is a feature with high dimensionality and a high computational cost, especially in the analysis of videos with high resolution.

To address these problems, we have improved the MISIFT algorithm to retain mirror invariance, which causes a significant beneficial decrease in feature dimensionality based on retaining the properties of rotation and scale invariance to significantly improve the efficiency.

After the video of interest is loaded, the current frame is denoted by $I_{current}$. Through a large number of experiments, we have found that the algorithm can still achieve satisfactory results even if the frames are scaled down between 0.7 and 1. Therefore, the current frame in the experiments is scaled down to 80% of the original size before extracting feature points, and feature points ($X = x_1, \ldots, x_n$) of $I_{current}$ are extracted. Each feature is described using a 128-D vector $\{f_1, f_2, \ldots, f_m\}$ ($m = 128$). Thus, the feature points of $I_{current}$ are described as follows:

$$X = \{x_1, x_2, \ldots, x_n\} = \left\{ \begin{matrix} f_{11}, f_{12}, \ldots, f_{1m} \\ f_{21}, f_{22}, \ldots, f_{2m} \\ \ddots \\ f_{n1}, f_{n2}, \ldots, f_{nm} \end{matrix} \right\}, \quad (m = 128) \tag{10}$$

where f_{nm} is the m-th component of the n-th feature point x_n. Because of the high dimensionality of X, we use the PCA to reduce the dimensionality as follows.

Let us make assumptions that A, B are two variables; $\rho(A, B)$ is the correlation coefficient of A and B; $Cov(A, B)$ is the covariance of A and B; $Var(A)$ and $Var(B)$ are the standard deviations of A and B, respectively. According to (11), the correlation coefficient is, in essence, the "normalized" covariance. Therefore, we use the correlation coefficient matrix in the PCA algorithm instead of the covariance matrix.

$$\rho(A, B) = \frac{Cov(A, B)}{\sqrt{Var(A)\,Var(B)}} \tag{11}$$

First, we define a correlation coefficient matrix R as an $m \times m$ symmetric matrix calculated according to (12) and (13), where x_i is a 128 dimensional feature vector, r_{ij} $(i, j = 1, 2, \ldots, p, p = 128)$ is the correlation coefficient of x_i and x_j, and $r_{ij} = r_{ji}$:

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1p} \\ r_{21} & r_{22} & \cdots & r_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ r_{p1} & r_{p2} & \cdots & r_{pp} \end{bmatrix} \tag{12}$$

$$r_{ij} = \frac{\sum_{k=1}^{n}(x_{ki} - \bar{x}_i)\left(x_{kj} - \bar{x}_j\right)}{\sqrt{\sum_{k=1}^{n}(x_{ki} - \bar{x}_i)^2 \sum_{k=1}^{n}\left(x_{kj} - \bar{x}_j\right)^2}} \tag{13}$$

Then, we solve the equation $|\lambda I - R| = 0$ and calculate the eigenvalues λ and eigenvectors L of R.

Finally, we sort the eigenvectors and eigenvalues in order of decreasing eigenvalue $(\lambda_1 \geq \lambda_2 \geq \cdots, \geq \lambda_p \geq 0)$. Based on the eigenvalues, the contribution of each component calculated as follows:

$$\frac{\lambda_i}{\sum_{k=1}^{p} \lambda_k} \qquad (i = 1, 2, \ldots, p) \tag{14}$$

The cumulative contributions are calculated as follows:

$$\frac{\sum_{t=1}^{i} \lambda_t}{\sum_{k=1}^{p} \lambda_k} \qquad (i = 1, 2, \ldots, p) \tag{15}$$

We select eigenvalues $\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_k$ and their eigenvectors $l_1, l_2, l_3, \ldots, l_k$ (l_i is the eigenvector corresponding to λ_i) whose cumulative contribution is greater than 90% and calculate the principal components according to the selected eigenvectors. With this approach, the dimensionality of each feature is reduced more than 75%, thus improving the efficiency of our algorithm (Jolliffe 2002).

The similarities between any two features points are then calculated using the Euclidean distance, and the matching points are sought. x_{ik} refers to the k-th component of the ith- feature point. Similarly, x_{jk} refers to the k-th component of the j-th feature point. The calculation of similarity between the i-th and j-th feature points is then expressed as

$$dis_{ij} = \left[\sum_{k}^{k=m}\left(x_{ik} - x_{jk}\right)^2\right]^{\frac{1}{2}} \tag{16}$$

By applying the above steps, the amount of data is reduced considerably. We extract the feature points of the same image in separate methods using SIFT, TIF (proposed in Kakar

Table 1 Execution time of SIFT, TIF and our algorithm

Resolution	SIFT (s)	TIF (s)	Our algorithm (s)
640 × 480	3.5	4.4	2.4
1280 × 720	16.5	20.9	12.3

and Sudha 2012) and our algorithm and then calculate their similarities. Table 1 compares the running times.

3.2 Locating the tampered area in the current frame

Following the steps in this section confirms the tampered areas in the current frame. When tampering with a video, the attackers are generally attempting to keep the tampered contents undetected under the premise that the tampering is meaningful. Thus, the properties of the intra-frame region duplication forgery can be summarized as follows.

1. The tampered content is a meaningful connected region of reasonable size, that is, the tampered areas in a forged frame cannot be overly small.
2. To deceive the viewers, in most cases, the source areas and tampered areas often have a distance. If the source areas were too close to the tampered areas, the viewer could visually discover the forgery through shadow or lighting conflicts.
3. The forgery will occur in successive frames of a video, and the tampered regions between the adjacent frames will change only slightly.

In previous methods, the optimal candidate matching point for each feature point was found by identifying its nearest neighbor among all of the other feature points; this was the feature point with the minimum Euclidean distance. Through experimentation, we determined that simply evaluating the distance between two descriptors with respect to a global threshold to determine the optimal matching points is not satisfactory because of the high-dimensionality of the feature space, in which some descriptors are much more discriminative than others (Amerini et al. 2011). In this paper, we use the ratio of the distance of the closest neighbor to that of the second-closest neighbor and compare it with a threshold (fixed to 0.6) to determine the optimal matching points.

We have a set of feature points $X = \{x_1, \ldots, x_m\}$ and $x_i = \{f_{i1}, \ldots, f_{i20}\}$. Then, we define a similarity vector that represents the Euclidean distances with respect to the other descriptors:

$$D_n = \{d_{n1}, d_{n2}, \ldots, d_{ni}, \ldots\} \qquad 0 < n < m - 1, \quad 1 < i < m \tag{17}$$

where D_n is sorted in ascending order and d_{ni} is the distance of x_n to a certain feature point. It is assumed that d_{n1} is the distance of the n-th feature point to the k-th feature point. The n-th feature point and k-th feature point are matched only if the following constraint is satisfied:

$$d_{n1}/d_{n2} < T_r \qquad where \quad T_r \in (0, 1) \tag{18}$$

where d_{n2} is the distance of n-th feature point from the t-th feature point.

In this paper, we set $T_{\mathrm{r}} = 0.6$ and repeat the process until Eqs. (17) and (18) traverse all feature points. Finally, we obtain a set of matching points of the frame.

Now, we obtain the matching point sets Q and W of $I_{current}$. Q and W have the same number of elements, and q_i and w_i are a pair of matching points. However, experiments indicate that Q and W still contain some mismatched points.

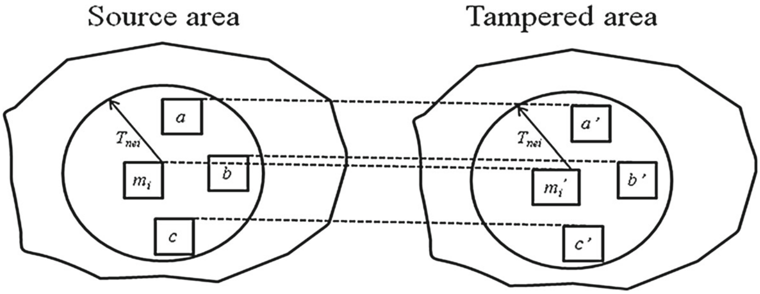

Fig. 3 The process (*b*) of removing the mismatching points

To reduce the number of mismatched points and improve the detection accuracy, we designed a three-step method to confirm the tampered region of the current frame based on the properties of the intra-frame region duplication forgery.

For a pair of matching feature points q_i and w_i and their corresponding coordinates $Loc(q_i)$ and $Loc(w_i)$ on the frame:

(a) The source areas and tampered areas often have a distance. That is, q_i and w_i will be retained only if the following formula is satisfied; otherwise, they should be removed.

$$dis > T_{dis} \quad where \quad dis = \sqrt{Loc^2(q_i) - Loc^2(w_i)} \tag{19}$$

(b) The matching points should concentrate in the region, while mismatched points are rule-less, individual noise points that are generally scattered. For any point m_i ($m_i \in Q \bigcup W$), there must be at least three other matching points within the radius T_{nei} (a graphic illustration is provided in Fig. 3). m_i and its corresponding matching point m_i' will be retained if the number of matching points is greater than 3; otherwise, they will be removed.

(c) We use the k-means clustering algorithm to classify all retained points into two categories (i.e., source region Q' and tampered region W'). Any pair of matching feature points q_i and w_i will be removed if they meet either of the two following conditions:

$$\begin{array}{ccc} q_i \in Q' & and & w_i \in Q' \\ & or & \\ q_i \in W' & and & w_i \in W' \end{array} \tag{20}$$

After the removal of mismatching points, the number of retained matching point-pairs will be calculated. If the number is more than 5, the frame is considered to be tampered. If the number is 5 or less, the frame can be considered authentic, and there is no need to process the next frame using the above method. Therefore, to improve the efficiency of the algorithm, an inspection interval parameter t is specified, and $current = current+t$; that is, frame $current+t$ should be taken as the current frame $I_{current}$ in the next detection, and the algorithm must be resumed at part one to continue detection. Figure 5 shows a comparison between removing and not removing mismatching points with the method proposed in this paper.

3.3 Locating the tampered area in the remaining frames

After completing the previous section, a tampered area in the first tampered frame is discovered. The next frame would likely be subjected to intra-frame region duplication forgery, and

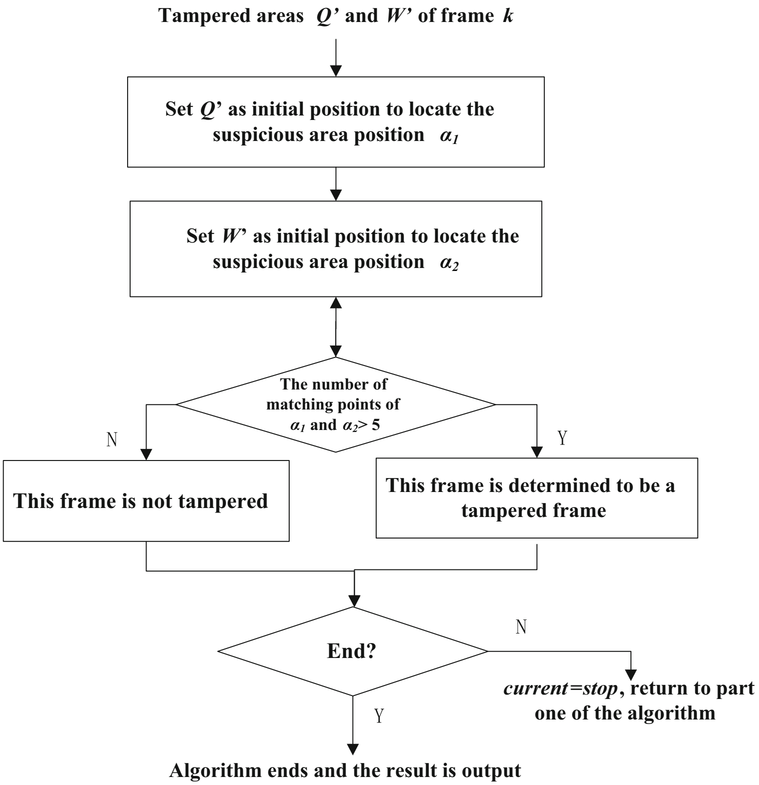
Tampered areas Q' and W' of frame k
Set Q' as initial position to locate the suspicious area position α₁
Set W' as initial position to locate the suspicious area position α₂
The number of matching points of α₁ and α₂> 5
N
Y
This frame is not tampered
This frame is determined to be a tampered frame
End?
N
current=stop, return to part one of the algorithm
Y
Algorithm ends and the result is output

Fig. 4 Flow chart of locating the tampered area in the remaining frames

the tampered regions in adjacent frames would likely change only slightly. In this paper, to avoid feature extraction and matching frame-by-frame, spatio-temporal context learning is employed to locate the tampered region in the subsequent frames of the video; this technique improves the detection efficiency considerably.

Step 1 The Q' obtained from the previous step in the k-th frame is set as the initial position of the target, and we calculate the spatial context model with the following formula deduced in Sect. 2.

$$h_k^{sc}(\mathbf{x}) = F^{-1}\left(\frac{F\left(be^{-\left|\frac{\mathbf{x}-\mathbf{x}^*}{\alpha}\right|^{\beta}}\right)}{F\left(I(\mathbf{x})\,\omega_\sigma(\mathbf{x}-\mathbf{x}^*)\right)}\right) \tag{21}$$

Step 2 In the k-th frame, the spatial context model calculated by step 1 is used to update the spatio-temporal context model H_{k+1}^{stc} to reduce noise as follows. H_{k+1}^{stc} is then applied to detect the object location in the $(k+1)$-th frame:

$$H_{k+1}^{stc} = (1-\rho)\,H_k^{stc} + \rho h_k^{sc} \tag{22}$$

where ρ is a learning parameter and h_k^{sc} is the spatial context model computed by step 1. In the first frame, initialize the spatio-temporal context model H_1^{stc} to be equivalent to the spatial context model h_1^{sc}.

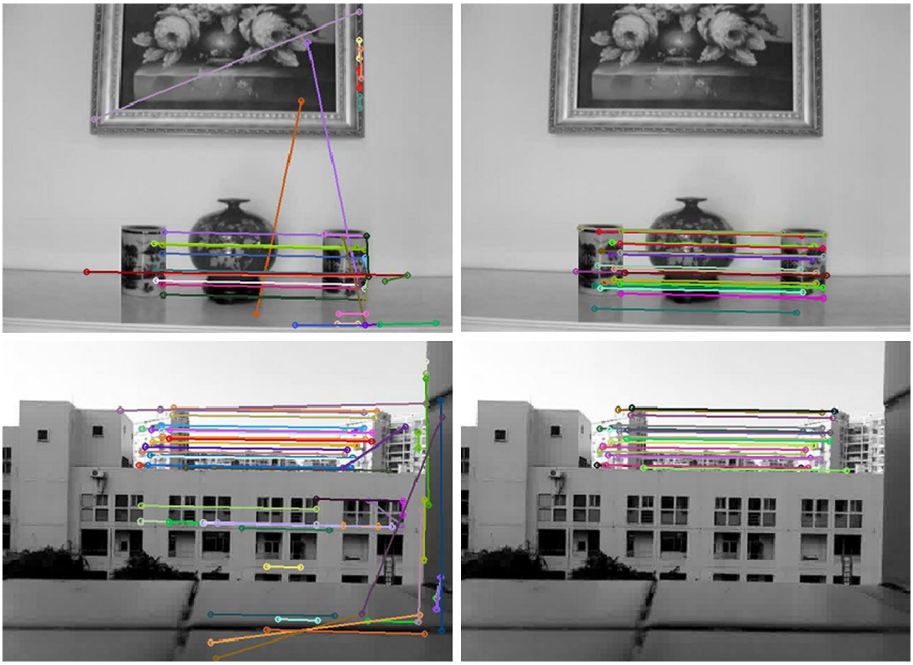

Fig. 5 Comparison between not removing mismatching points (*left*) and removing mismatching points (*right*)

Step 3 Compute a confidence map as follows:

$$c_{k+1}\left(x\right) = F^{-1}\left(F(H_{k+1}^{stc}(x)) \otimes F\left(I_{k+1}(x)\omega_{\sigma}\left(x-x^{*}\right)\right)\right) \tag{23}$$

Step 4 Obtain the maximum value, which is the location of the object; that is,

$$x_{k+1}^{*} = \underset{x\in\Omega_c(x_k^{*})}{\arg\max}\, c_{k+1}(\mathrm{x}) \tag{24}$$

where $\Omega_c(x_k^*)$ is the local context region based on the tracked location x_k^* in the k-th frame.

Equation (21) is deduced from (7), and an FFT is used for fast convolution (F denotes the FFT function). H_{k+1}^{stc} in (23) is derived from step 2.

The algorithm in this section stops when it encounters one of the following conditions.

(a) The areas obtained go beyond the frames;
(b) The size of the region is less than T_{nei};
(c) The minimum distance between two regions is less than T_{dis};
(d) It is the last frame of the video.

After stopping, we set W' as the initial position and repeat the process described above to obtain the suspicious area position α_2. At this time, $\alpha_1(i)$ and $\alpha_2(i)$correspond to the two suspicious areas of the i-th frame. The frame number is recorded as *stop* when the algorithm in this section stops. After seeking the matching points in two suspicious areas of each frame recorded in α_1 and α_2, if the number of matching points is greater than 5, this frame is considered to have been tampered with; otherwise, this frame is considered to be authentic.

Finally, the algorithm stops if the matching target in the last frame is completed; otherwise, let *current*=*stop* and return to part one of the algorithm to continue the detection. Figure 4 shows the flow chart corresponding to the procedures in this section.

3.4 The complete algorithm

Input the video, parameter t, the number *first* of the first frame of the video and the number *last* of the last frame.

Step-1 Let *current*=*first*;

Step-2 If *current* is greater than *last*, the whole algorithm stops; otherwise, extract features in $I_{current}$ as described in Sect. 3.1;

Step-3 Obtain the candidate matching set by adopting the three-step method described in Sect. 3.2;

Step-4 If $I_{current}$ is authentic, let $current = current + t$ and return to Step-2; Otherwise, continue the steps;

Step-5 Locate the tampered areas in the subsequent frames by adopting the method described in Sect. 3.3 and record the frame number *stop* where the tracking algorithm stops;

Step-6 Seek the matching points in tracking results of each frame obtained from Step-5. If the number of matching points is greater than 5, the frame is considered to have been tampered;

Step-7 Let *current*=*stop* and return to Step-2.

4 Experiments and analysis

The experiments in this chapter were conducted with videos shot by the author, videos downloaded from the Internet, and videos from the Surrey University Library for Forensic Analysis (SULFA). SULFA is a public video library which has been designed and built for the purpose of video forensics (Qadir and Yahaya 2012). All test videos used in this part were MPEG-2 encoded with frame rates of 25 fps. The computer configuration in this experiment was as follows:

CPU: Intel(R) Core(TM)2 i7-4700MQ 2.4GHz;
Memory size: 8GB;
Video card: NVIDIA GeForce GT 755M;
OS: Microsoft Windows 7;
Coding: MATLAB Version 7.12.0.635 (R2011a).

4.1 Detection result of a tampered area with no post-processing

In this section, the tampered areas of all test videos were not processed. The experimental parameters were configured as follows: parameter $t = 5$, $T_{\text{dis}} = 20$. Figure 6 shows the detection results.

The experimental results in Fig. 6 demonstrate that the tampered areas of the video processed by our algorithm were recorded and marked with red boxes, and matching points in two areas were connected with lines. More lines indicate more matching points and higher similarity. In the experiment, the algorithm accurately determined whether the video was tampered, regardless of the size of the tampered area, and located the tampered area, which demonstrates the effectiveness of our algorithm.

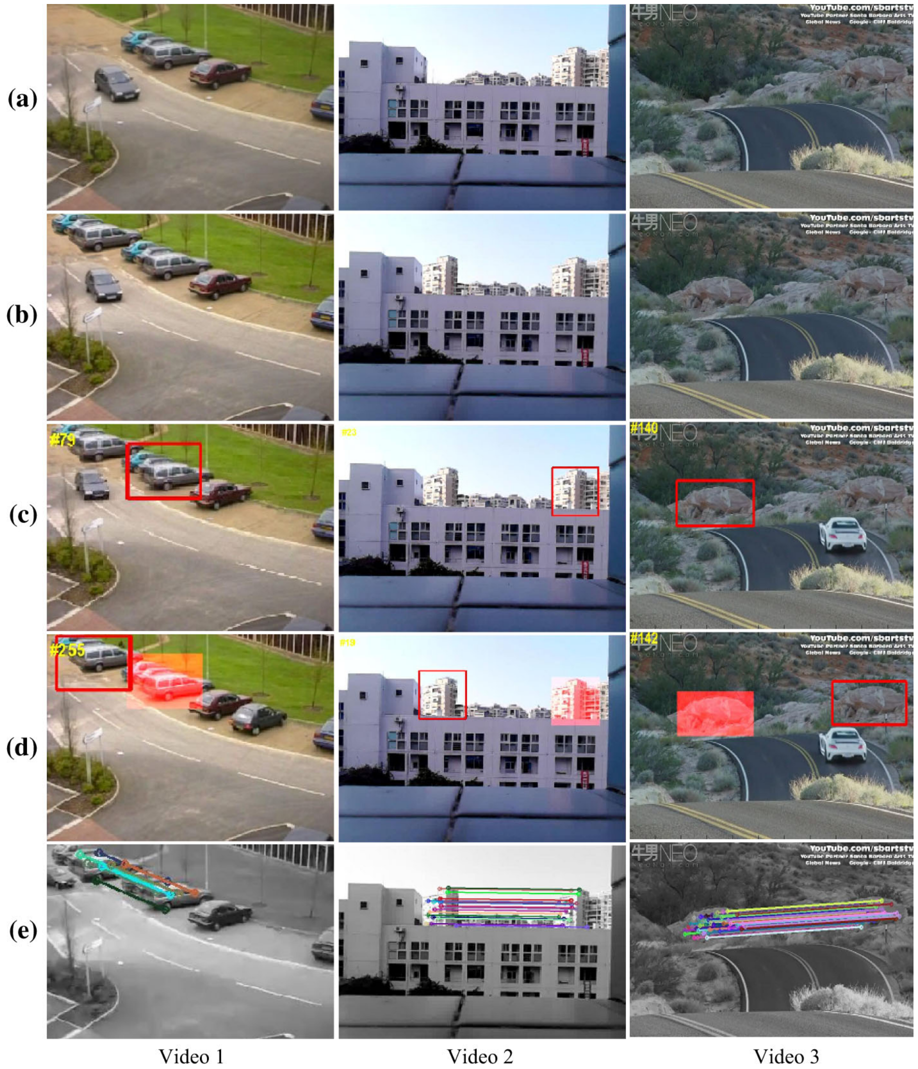

Fig. 6 Snapshots of the test videos: **a** original (untampered) video clip; **b** video clip tampered through intra-frame region duplication; **c** video clip with spatio-temporal context learning algorithm first used to locate the tampered area; **d** video clip with spatio-temporal context learning algorithm used for the second time to locate the tampered area; **e** matching for two areas

4.2 Detection result of the tampered area with post-processing

In this section, the tampered areas in videos 4–6 were subjected to a mirror operation. Furthermore, the camera was shaking slightly while recording video 5. The tampered areas in videos 7 and 8 were scaled up and scaled down, respectively. The tampered areas in video 9 were scaled-down and rotated, and the experimental parameters were configured as follows: test interval parameter $t = 5$ and $T_{\text{dis}} = 20$. Figures 7 and 8 show the detection results.

The experimental results in Fig. 8 demonstrate that our algorithm can detect forged video in which the tampered areas were subjected to the mirror operation. It also exhibits good robust-

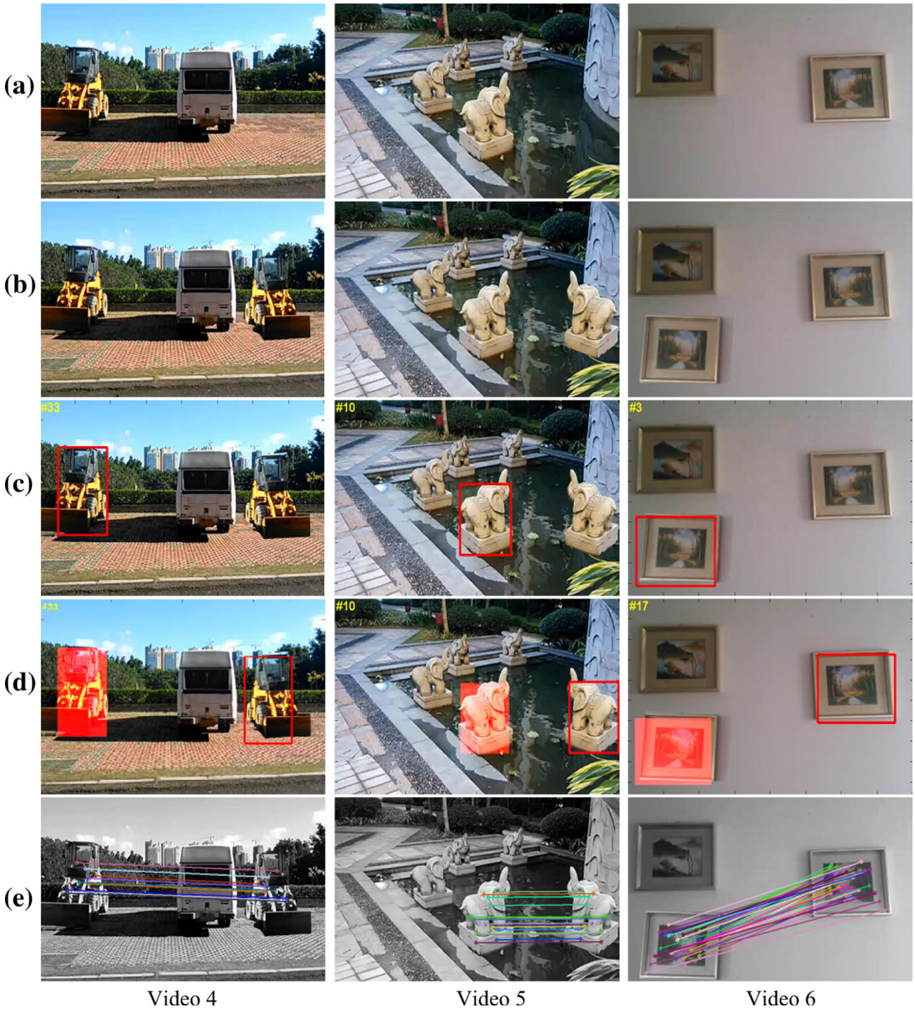

Fig. 7 Snapshots of the test videos: **a** original (untampered) video clip; **b** video clip tampered through intra-frame region duplication and mirror operation; **c** video clip with spatio-temporal context learning algorithm first used to locate the tampered area; **d** video clip with spatio-temporal context learning algorithm used for the second time to locate the tampered area; **e** matching for two areas

ness when considering rotation and scaling. Because our algorithm uses spatio-temporal context learning technology, the algorithm can accurately locate the tamper position even when the camera corresponding to the tampered video was shaking slightly during recording.

4.3 Execution time of the proposed algorithm

In this section, Table 2 shows the execution times of the proposed algorithm for selected videos in our experiments. The comparison methods include the detection method based on HOG in Subramanyam and Emmanuel (2012) (denoted by A.V.), the methods proposed by Wang and Farid (2007) (denoted by Wang) and the methods proposed by Pun et al. (2015).

For videos with resolution of 640 × 480 pixels, the runtime for our algorithm was approximately 0.7 s per frame, compared with 1.3 s per frame for the method in Wang and Farid

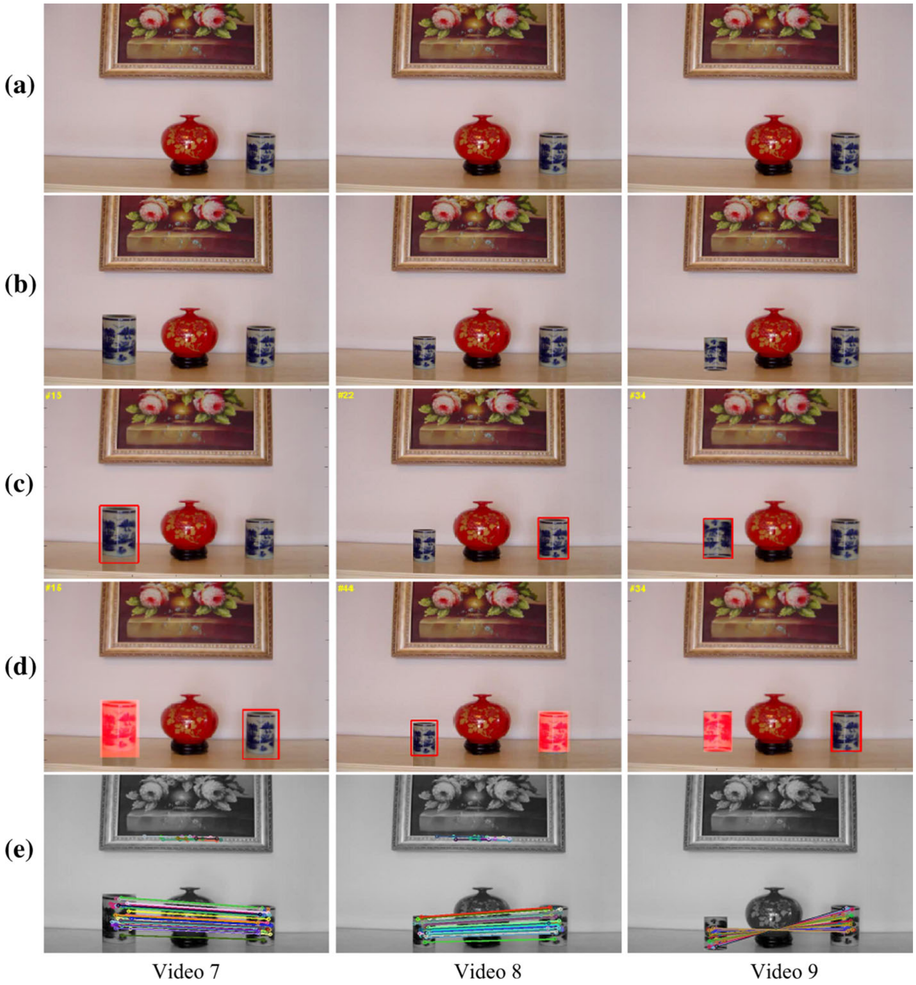

Fig. 8 Snapshots of the test videos: **a** original (untampered) video clip; **b** video clip tampered through intra-frame region duplication and geometric transformation; **c** video clip with spatio-temporal context learning algorithm first used to locate the tampered area; **d** video clip with spatio-temporal context learning algorithm used for the second time to locate the tampered area; **e** matching for two areas

(2007); the time efficiency of our algorithm provides an improvement of more than 35%. The methods proposed in Subramanyam and Emmanuel (2012) and Pun et al. (2015) extract and match features frame-by-frame, which takes much more time than the proposed algorithm.

The time consumption of our algorithm is dominated by the feature-extraction stage. Therefore, the experimental results demonstrate that as more textural details are contained in the video, a longer amount of time is required for detection with the algorithm.

4.4 Comparison experiments with other algorithms

In this section, 6000 frames from 15 different videos downloaded from SULFA were used for experiments. In each video, a tampered area was specified and copied and pasted to another area in the same frame. We calculated the detection accuracy (*DA*) as follows:

Table 2 Execution time of the proposed algorithm

Name	Resolution	Number of frames	Proposed (s)	Wang (s)	A.V. (s)	Pun (s)
Video 1	320 × 240	100	36	70	313	697
Video 2	640 × 480	100	65	137	752	>1000
Video 3	1280 × 720	150	221	511	>1000	>1000
Video 4	640 × 480	100	69	141	790	>1000
Video 5	1024 × 768	150	142	324	>1000	>1000
Video 6	1024 × 768	100	103	228	978	>1000
Video 7	640 × 480	300	60	417	>1000	>1000
Video 10	320 × 240	200	57	150	618	>1000
Video 11	320 × 240	200	66	157	650	>1000
Video 12	320 × 240	200	58	147	623	>1000
Video 13	320 × 240	200	70	178	693	>1000
Video 14	320 × 240	200	61	163	633	>1000

Table 3 Performance evaluation for the proposed algorithm

	Positive (%)	Negative (%)
True	95.2	90.0
False	4.8	10.0

Table 4 Comparison of the detection accuracy with other algorithms

	Detection accuracy (%)	Mirror invariant
Pun	90.8	No
A.V.	89.7	No
Wang	70.0	No
Proposed	92.6	Yes

$$DA = \frac{TP + TN}{TP + TN + FP + FN} \tag{25}$$

Here, *TP* means that authentic was detected as authentic, *TN* means that forged was detected as forged, *FP* means that authentic was detected as forged, and *FN* means that forged was detected as authentic. The experimental data set of *TN* and *FN* was calculated from 6000 tampered frames. The experimental dataset of *TP* and *FP* was calculated from the authentic original 6000 frames above. The compared methods include the detection method based on HOG in Subramanyam and Emmanuel (2012) (denoted by A.V.), the methods proposed by Wang and Farid (2007) (denoted by Wang) and the methods proposed by Pun et al. (2015). Table 3 shows the performance evaluation for the proposed algorithm. Table 4 shows the comparison experiments with other algorithms. Figure 9 shows examples of tampering detection in videos downloaded from SULFA.

Tables 3 and 4 demonstrate that our algorithm offers higher detection accuracy than the other algorithms under the same conditions. The methods proposed in Wang and Farid (2007), Subramanyam and Emmanuel (2012) and Pun et al. (2015) cannot detect tampered areas subjected to mirror operations (such as videos 4–6) because these algorithms lack abilities of mirror-invariant detection. Furthermore, Fig. 10 shows some of the experimental results

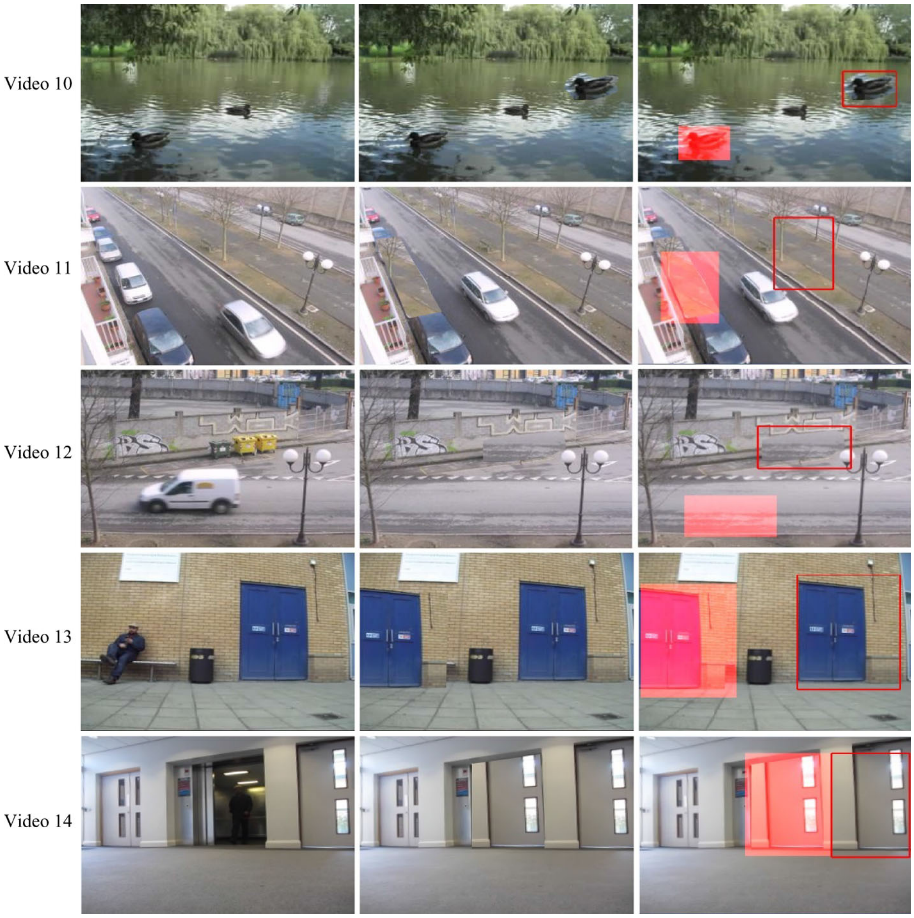

Fig. 9 Some detection results for videos downloaded from SULFA

of the method proposed by Pun et al. (2015). It is observed from Fig. 10 that the method proposed in Pun et al. (2015) is not only invalid for detecting videos under mirror but also does not perform well in areas with similar textures.

5 Conclusions and future works

In this paper, we propose a novel algorithm to detect duplication forgery in a region of a video. The experimental results demonstrate that our algorithm features higher efficiency scores than previous algorithms and exhibits satisfactory performance in terms of detection effectiveness for videos subjected to mirror operations. Our main contributions can be summarized as follows.

1. Mirror operation is the one of the most commonly used operations in video forgery, but previously reported algorithms cannot detect this type of forgery due to lack of mirror invariance. The problem can be solved by extracting the feature points with improved MISIFT.

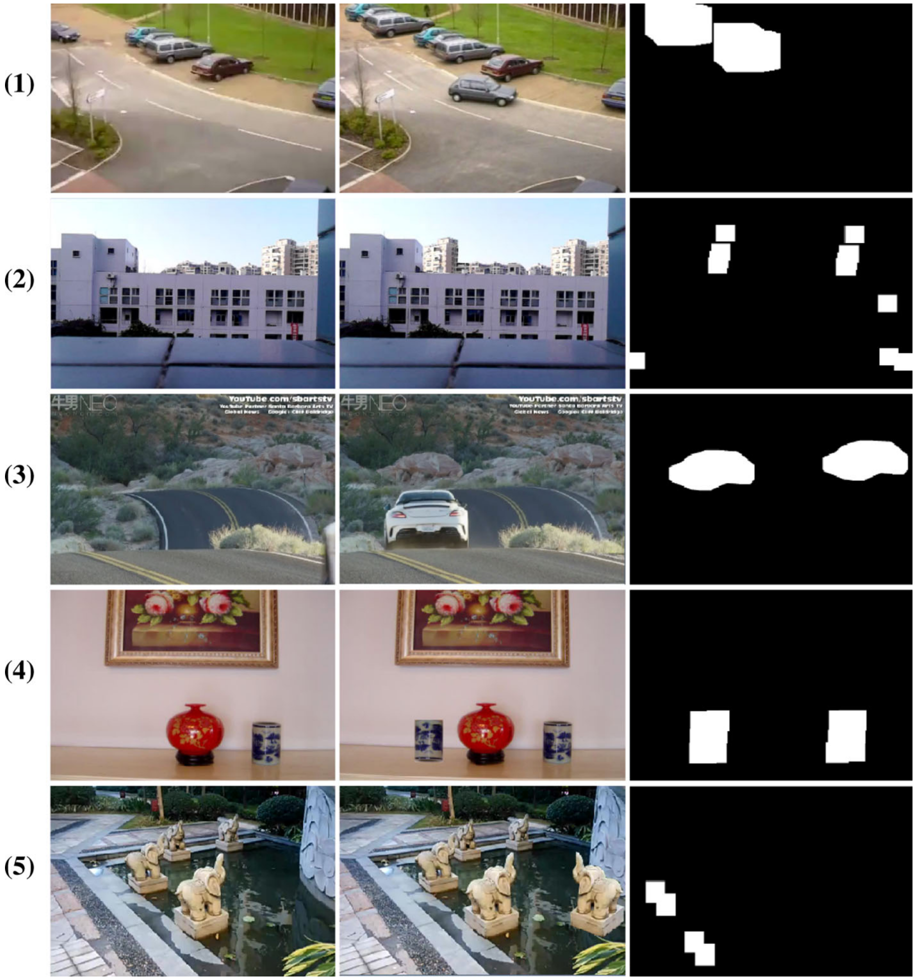

Fig. 10 Some experimental results using the method proposed in Pun et al. (2015)

2. A new framework for the detection of intra-frame region duplication forgery is proposed, and spatio-temporal context learning is used for the first time in video forgery detection, improving efficiency by avoiding feature extraction and frame-by-frame matching.

However, considerable shaking of the video during shooting or an overly high movement velocity (including rotation or movement speed) of the tampered area will cause non-ideal experimental effects. We will further explore such situations in future research. We will further explore such situations in our future research. We will also extend our method to detect other types of video tampering.

Acknowledgements This work was supported by National Natural Science Foundation of China under Grant Number 61373077, the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20110121110020) and the National Defense Basic Scientific Research Program of China.

References

Al-Qershi, O. M., & Khoo, B. E. (2013). Passive detection of copy-move forgery in digital images: State-of-the-art. *Forensic Science International*, *231*(1), 284–295.

Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., & Serra, G. (2011). A SIFT-based forensic method for copy-move attack detection and transformation recovery. *IEEE Transactions on Information Forensics and Security*, *6*(3), 1099–1110.

Cao, Y., Gao, T., Fan, L., & Yang, Q. (2012). A robust detection algorithm for copy-move forgery in digital images. *Forensic Science International*, *214*(1), 33–43.

Chen, W., & Shi, Y. (2009). Detection of double mpeg compression based on first digit statistics. *Lecture notes in computer science, Digital Watermarking, 5450*, 16–30.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In: *IEEE computer society conference on computer vision and pattern recognition, 2005*. CVPR 2005. 2005. IEEE (pp. 886–893).

Gao, B., & Jin, Y. (2010). Detection of Image copy-move tamper Using SURF in digital forensics. In: *2010 Asia-Pacific conference on information network and digital content security* (pp. 58–62).

Hsu, C.-C., Hung, T.-Y., Lin, C.-W., & Hsu, C.-T. (2008). Video forgery detection using correlation of noise residue. In: *2008 IEEE 10th Workshop on Multimedia Signal Processing, 2008* (pp. 170–174). IEEE.

Jolliffe, I. (2002). *Principal component analysis*. New York: Wiley.

Kakar, P., & Sudha, N. (2012). Exposing postprocessed copy-paste forgeries through transform-invariant features. *IEEE Transactions on Information Forensics and Security*, *7*(3), 1018–1028.

Kobayashi, M., Okabe, T., & Sato, Y. (2009). Detecting video forgeries based on noise characteristics. In: *Advances in image and video technology* (pp. 306–317). Berlin: Springer.

Li, W., Yuan, Y., & Yu, N. (2009). Passive detection of doctored JPEG image via block artifact grid extraction. *Signal Processing*, *89*(9), 1821–1829.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Ma, R., Chen, J., & Su, Z. (2010). MI-SIFT: Mirror and inversion invariant generalization for SIFT descriptor. In: *Proceedings of the ACM international conference on image and video retrieval, ACM* (pp. 228–235).

Milani, S., Fontani, M., Bestagini, P., Barni, M., Piva, A., Tagliasacchi, M., et al. (2012). An overview on video forensics. *APSIPA Transactions on Signal and Information Processing*, *1*, e2.

Oppenheim, A. V., Willsky, A. S., & Nawab, S. H. (1983). Signals and systems, vol. 2. Englewood Cliffs, NJ: Prentice-Hall. *6*(7):10.

Pun, C.-M., Yuan, X.-C., & Bi, X.-L. (2015). Image forgery detection using adaptive oversegmentation and feature point matching. *IEEE Transactions on Information Forensics and Security*, *10*(8), 1705–1716.

Qadir, G., Yahaya, S., & Ho, A. T. (2012). Surrey university library for forensic analysis (SULFA) of video content. In: IET conference on image processing (IPR 2012). IET (pp. 1–6).

Sencar, H. T., & Memon, N. (2008). Overview of state-of-the-art in digital image forensics. *Algorithms, Architectures and Information Systems Security*, *3*, 325–348.

Subramanyam, A., & Emmanuel, S. (2012). Video forgery detection using HOG features and compression properties. In: 2012 IEEE 14th International Workshop on Multimedia Signal Processing (MMSP), 2012. IEEE (pp. 89–94).

Torralba, A. (2003). Contextual priming for object detection. *International Journal of Computer Vision*, *53*(2), 169–191.

Wang, W., & Farid, H. (2007). Exposing digital forgeries in video by detecting duplication. In: *Proceedings of the 9th workshop on multimedia and security, ACM* (pp. 35–42).

Wang, W., & Farid, H. (2009). Exposing digital forgeries in video by detecting double quantization. In: *Proceedings of the 11th ACM workshop on multimedia and security, 2009* (pp. 39–48). ACM.

Yang, J., Huang, T., & Su, L. (2014). Using similarity analysis to detect frame duplication forgery in videos. *Multimedia Tools and Applications,* 1–19.

Yin, H., Hui, W., Li, H., Lin, C., & Zhu, W. (2012). A novel large-scale digital forensics service platform for Internet videos. *IEEE Transactions on Multimedia*, *14*(1), 178–186.

Zhang, K., Zhang, L., Liu, Q., Zhang, D., & Yang, M.-H. (2014). Fast visual tracking via dense spatio-temporal context learning. In: *Computer vision–ECCV 2014* (pp. 127–141). Springer.