


Article

# An Inter-Frame Forgery Detection Algorithm for Surveillance Video

Qian Li <sup>1,2,\*</sup> , Rangding Wang <sup>2,\*</sup> and Dawen Xu <sup>3</sup>

<sup>1</sup> College of Information Engineering, Ningbo Dahongying University, Ningbo 315175, China

<sup>2</sup> CKC Software Laboratory, Ningbo University, Ningbo 315211, China

<sup>3</sup> School of Electronics and Information Engineering, Ningbo University of Technology, Ningbo 315211, China; dawenxu@126.com

\* Correspondence: liqian\_mine@126.com (Q.L.); wangrangding@nbu.edu.cn (R.W.); Tel.: +86-0574-8804-6826 (Q.L.)

Received: 16 August 2018; Accepted: 20 November 2018; Published: 28 November 2018



**Abstract:** Surveillance systems are ubiquitous in our lives, and surveillance videos are often used as significant evidence for judicial forensics. However, the authenticity of surveillance videos is difficult to guarantee. Ascertaining the authenticity of surveillance video is an urgent problem. Inter-frame forgery is one of the most common ways for video tampering. The forgery will reduce the correlation between adjacent frames at tampering position. Therefore, the correlation can be used to detect tamper operation. The algorithm is composed of feature extraction and abnormal point localization. During feature extraction, we extract the 2-D phase congruency of each frame, since it is a good image characteristic. Then calculate the correlation between the adjacent frames. In the second phase, the abnormal points were detected by using *k*-means clustering algorithm. The normal and abnormal points were clustered into two categories. Experimental results demonstrate that the scheme has high detection and localization accuracy.

**Keywords:** surveillance video; video forensics; inter-frame forgery; 2-D phase congruency; *k*-means clustering

## 1. Introduction

Video sequences are often believed to provide stronger forensic evidence than still images. Thus, surveillance video, as important evidence, is often used in the case investigation. However, the digitization feature makes surveillance video easy to be manipulated. Tampering with a digital video without leaving visible clues is easily accomplished by using a video editing software, such as Adobe Premiere. Therefore, digital video forensics, which is designed to verify the trustworthiness of digital video, has become an important and exciting field for recent research. Katsaounidou et al. [1] introduced a framework of cross-media authentication and verification, as well as the values of cross-media authentication in journalism and judicial. Arab et al. [2] proposed a detection method for surveillance systems by embedding robust watermark in video. The algorithm was proven efficiency at detecting a wider range of tampering. However, embedding watermarks is not feasible sometimes in the tested video. Therefore, the detection algorithm which does not depend on prior information, for instance, detecting traces of forgery, has caught much attention in recent research.

Forgery detection for surveillance video can be divided into source authenticity and content authenticity. Source authenticity [3,4] is actually the analysis of the video “where it came from” and “how it came from”. Many approaches have been developed to investigate each of steps of the acquisition process [5]. While content authenticity studies whether the video has experienced some kinds of tamper operation. Forgery detection of video content includes double compression

detection [6–8], intra-frame forgery detection [9–12] and inter-frame forgery detection, because the videos may be tampered with in various ways, including spatial tampering, temporal tampering and spatio-temporal tampering [13]. Mizher et al. [14] made a detailed classification and introduction to video falsifying techniques and video forgery detection techniques. In this paper, we focus on detecting inter-frame forgery for surveillance video, which is one of the most common ways for video tampering. The purpose is to imitate or to conceal a specific event by inserting or deleting certain frames.

The existing inter-frame forgery detection methods can be divided into two categories [15].

(1) Methods based on the periodic effect of double compression

Su et al. [16] indicated that the power of high frequency region of DCT coefficients block in the inter-frame forgeries shows a clear periodic artifact. The weakness of the algorithm is that it may only apply to MPEG-2. Dong et al. [17] proposed a motion-compensated edge artifact (MCEA) scheme to detect frame-based video manipulation, by judging spikes in the Fourier transform domain after double MPEG compression. Due to the fact that frame deletion or insertion would result in the frames moving from one GOP to another, and gives rise to relatively larger motion estimation errors. A machine learning approach to detect frame deletion is put forward [18]. A number of discriminative features, such as prediction residuals, percentage of intra-coded macroblocks, quantization scales and reconstruction quality, are extracted from the video bit stream and its reconstructed images. Then, machine learning techniques were used to detect frame deletion. But the method cannot provide the exact localization of the deleted frames. Feng et al. [19] proposed a method which is applicable to video sequences with variable motion strengths. They analyzed the statistical characteristics of the most common interfering frames, then exploit a new fluctuation feature based on frame motion residuals to identify frame deletion points. However, the disadvantage of such methods is that they depend on the encoding parameters of the tested video. In addition, the forger can achieve anti-forensics by correcting the prediction error [20].

(2) Methods based on the discontinuity of content at tampering position. These methods are insensitive to encoding parameters and have more advantages in practical applications

Chao et al. [21] utilized optical flow consistency between adjacent frames to detect frame forgery, since inter-frame forgery will disturb the optical flow consistency. For frame insertion and frame deletion forgery, the authors select different detection methods. However, we could not know in advance what kind of forgery was involved. The methods have similar ideas in [22,23], which velocity field consistency and motion vector pyramid (MVP) consistency were used respectively. In [24], a method based on quotients of correlation coefficients between local binary patterns (LBPs) coded frames is proposed. The abnormal point detection is achieved by using chebyshev inequality twice. The weakness is that it fails to discuss the selection of multiple parameters while different parameters have different detection precision. Zhang et al. [25] also used chebyshev inequality to locate the tampering position. They used a three-dimensional tensor to describe the video features, then the tensor was factorized by Tucker non-negative decomposition method. Finally, they extracted time dimension matrix to calculate correlation to determine whether there is a frame insertion or deletion forgery. Zhao et al. [26] proposed an algorithm to detect the frame-deleting forgery. The feature extraction based on the normalized mutual information feature, and make use of generalized ESD test to localize the tampering point. However, the method assumes that there is only one discontinuity point and could not detect multiple tampering points.

Since content-based detection methods do not rely on encoding standards, the approach have a more widely application, and has attracted the attention of scholars in recent years. However, as mentioned above, there are still some shortcomings in the current methods. In this paper, we propose a novel scheme for inter-frame forgery detection based on 2D phase congruency and  $k$ -means clustering. By means of  $k$ -means clustering analysis, the abnormal points can be accurately located. There is no need to select multiple thresholds to avoid the impact of threshold selection on the detection

results. And it is also effective for multiple tampering. Because the surveillance video has static background, inter-frame forgery operation will reduce the correlation between adjacent frames at tampering position. Then, the consistency of the consecutive correlation coefficients is disturbed. Tampering localization could be achieved by detecting these discontinuous points, i.e., abnormal points. When calculating the correlation of adjacent frames, we use 2D phase congruency as the feature of frame since it is a good image characteristic. Furthermore, we employ  $k$ -means clustering analysis to cluster the normal and abnormal points into two categories.

The rest of the paper is organized as follows. In Section 2, the concept of 2D phase congruency and its feasibility are introduced. Section 3 describes the  $k$ -means clustering algorithm and the detection procedure for abnormal points. Section 4 gives our experimental results and discussion. Finally conclusions are drawn in Section 5.

## 2. Feature Extraction

Digital video is composed of sequences of still images or frames, and thus, it is also referred to as motion pictures. Therefore, some descriptors for digital image are also applicable to video. Due to the limited storage space of the monitoring equipment, usually the resolution of surveillance video is low. The 2D phase congruency is very sensitive to the edge and texture of the image, which can be used to describe the content of the surveillance video.

### 2.1. 2-D Phase Congruency

The authors of [27] provide a detailed introduction of 2D phase congruency. The Local Energy Model developed by Morrone et al. [28] postulated that the sharp features are perceived at points of maximum phase congruency in an image. Phase congruency (PC) was first defined by Morrone [29] in terms of the Fourier series expansion of a signal at some location  $x$  as:

$$PC(x) = \max_{\overline{\varphi}(x) \in [0, 2\pi]} \frac{\sum_n A_n \cos(\varphi_n(x) - \overline{\varphi}(x))}{\sum_n A_n} \quad (1)$$

where  $A_n$  is the amplitude of the  $n$ th Fourier component,  $\varphi_n(x)$  is the local phase of the  $n$ th Fourier component at position  $x$ , and  $\overline{\varphi}(x)$  is the weighted mean local phase angle at position  $x$ . If  $PC$  equals to a maximal value of 1, indicating that a noticeable signal change information is detected. As in the case of step edge in square wave. Otherwise,  $PC$  takes on values between 0 and 1, which denotes that the signal of each harmonic phase begins to be inconsistent. In this case, signal changes begin to relax, even no characteristics change. The principle can be extended to a two-dimensional (2D) image signal. When each harmonic phase has a high degree consistency, the image contains sharp features, such as edge and line.

Kovesi [30] extended the 1D PC to allow for the calculation of 2D PC of image by applying 1D analysis over several orientations and combining the results in some way. To calculate 2D PC of a given image, the image is first convolved with a bank of log-Gabor filters. Let the image denoted by  $I(x, y)$ , the even-symmetric filter and odd-symmetric filter at scale  $s$  and orientation  $o$  denoted by  $M_{so}^e$  and  $M_{so}^o$ , respectively. The responses of each quadrature pair of filters is a vector.

$$[e_{so}(x, y), o_{so}(x, y)] = [I(x, y) * M_{so}^e, I(x, y) * M_{so}^o] \quad (2)$$

where  $*$  is the convolution operator. From Equation (2), the amplitude and phase of this response is given by Equations (3) and (4):

$$A_{so}(x, y) = \sqrt{e_{so}^2(x, y) + o_{so}^2(x, y)} \quad (3)$$

$$\varphi_{so}(x, y) = \arctan(e_{so}(x, y), o_{so}(x, y)) \quad (4)$$

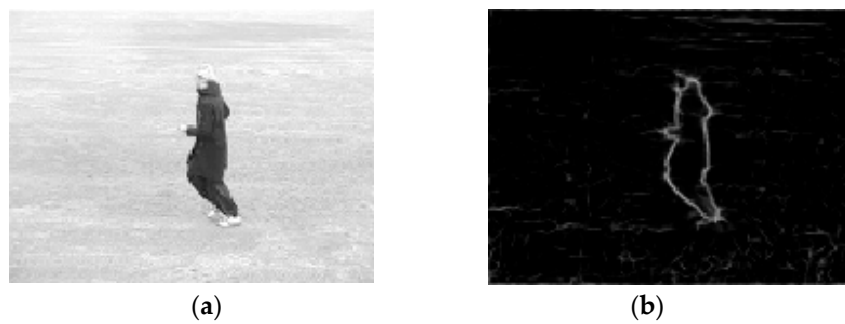
The 2D phase congruency is then calculated by:

$$PC(x, y) = \frac{\sum_o \sum_s W_o(x, y) (A_{so}(x, y) \Delta \phi_{so}(x, y) - T_o)^+}{\sum_o \sum_s A_{so}(x, y) + \varepsilon} \quad (5)$$

where  $()^+$  denotes that the enclosed quantity is equal to itself if it is positive, and equal to zero otherwise,  $W_o(x, y)$  is a measure of significance of frequency spread,  $\varepsilon$  is a small positive constant used to prevent division of zero,  $T_o$  is a quantity introduced to compensate image noise, and  $\Delta \phi_{so}(x, y)$  is a sensitive phase deviation function defined as:

$$\Delta \phi_{so}(x, y) = \cos(\phi_{so}(x, y) - \overline{\phi_o}(x, y)) - |\sin(\phi_{so}(x, y) - \overline{\phi_o}(x, y))| \quad (6)$$

The results of 2D phase consistency processing for video frame is shown in Figure 1.



**Figure 1.** The 2D phase consistency processing. (a) Video frame; (b) 2D phase congruency (PC).

## 2.2. The Correlation of Adjacent Frames

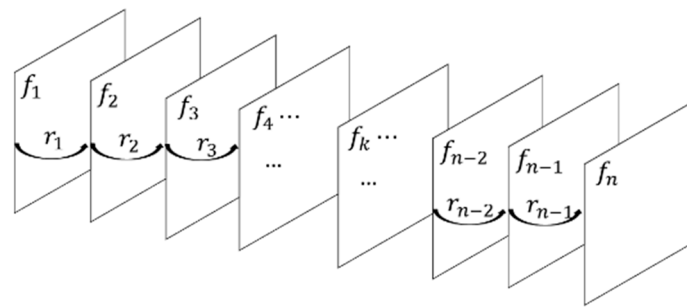
The contents of adjacent frames in video are usually very close, while the contents of distant frames may be very different. Whether the video is captured by a static camera or hand-held camera. But there should be no shot change. Therefore, we will use the correlation coefficient as a measure of the continuity of the inter frame content. The video frames were first processed through 2D PC, and the correlation coefficient between adjacent frames is defined in Equation (7):

$$r_k = \frac{\sum_i \sum_j (PC_k(i, j) - \overline{PC_k}) (PC_{k+1}(i, j) - \overline{PC_{k+1}})}{\sqrt{(\sum_i \sum_j (PC_k(i, j) - \overline{PC_k})^2) (\sum_i \sum_j (PC_{k+1}(i, j) - \overline{PC_{k+1}})^2)}} \quad k = 1, 2, \dots, n-1 \quad (7)$$

where  $r_k$  indicates the correlation coefficient between  $k$ th and  $(k+1)$ th frame,  $PC_k(i, j)$  represents the 2-D PC value of  $k$ th frame at location  $(i, j)$ ,  $n$  is the total number of video frames,  $\overline{PC_k}$  is the average of 2-D PC for the  $k$ th frame. Assuming the frame width of video is  $w$  pixels and the height is  $h$  pixels, then  $\overline{PC_k}$  can be calculated by using Equation (8):

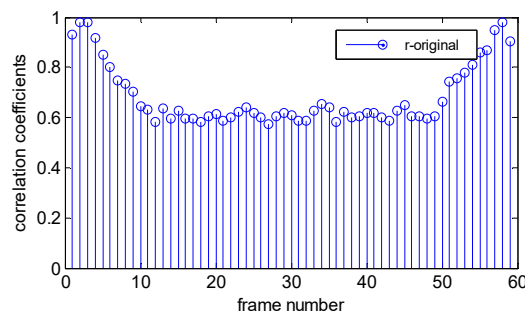
$$\overline{PC_k} = \frac{1}{w \times h} \sum_{i,j} PC_k(i, j) \quad (8)$$

As shown in Figure 2, we can get a sequence of inter-frame correlation coefficients. If the total number of video frame is  $n$ , the length of the sequence would be  $n - 1$ .

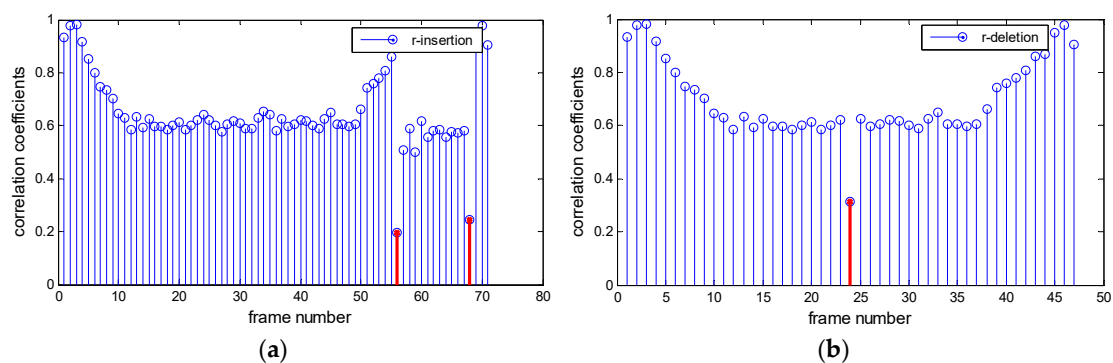


**Figure 2.** Inter-frame correlation.

In the original video, the value of correlation coefficients are close to each other, which means the curve of  $r$  is consistent. Figure 3 shows an illustration of frames 143–202 of test video “person15\_jogging”. When the original video is subjected to frame insertion tamper, the value of the correlation coefficient will decrease at the tamper position. The two frames calculating the correlation coefficient are not the original adjacent relation, regardless of whether the inserted frame is from the same video or another video. The same is true of frame deletion. Figure 4 shows the correlation coefficients of forged video. In Figure 4a, ten frames from the same video were inserted into original video. And the result of frame deletion (delete thirteen frames) was shown in Figure 4b. The red line indicates the tampering position, we can see that the correlation coefficient at tamper position is much lower than others.



**Figure 3.** Correlation coefficients of original video (person15\_jogging, frames 143–202).



**Figure 4.** Correlation coefficients of forged video (person15\_jogging\_forged) (a) Frame insertion; (b) Frame deletion.

### 2.3. The Variation of Consecutive Correlation Coefficients

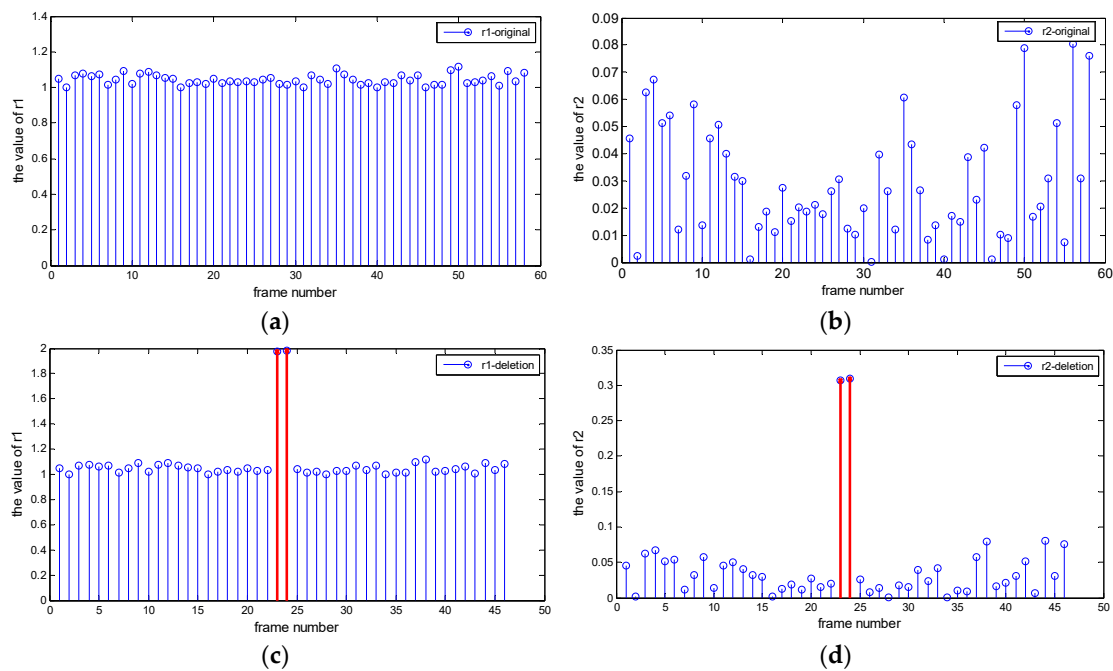
Some characteristics of video, such as the complexity of the texture or the speed of movement, will affect the value of the correlation coefficient. For instance, in Figure 3, the correlation coefficient is larger when the video content change slowly. Therefore, the detection is not accurate if we simply

use the value of correlation coefficient. To restrain this phenomenon caused by the diversification of video content, some scholars have put forward the solution of calculate the variation of consecutive correlation coefficients. Such as the quotients [24] or absolute difference [31] of consecutive correlation coefficients. The definitions are given in Equations (9) and (10):

$$\Delta r1_k = \begin{cases} \frac{r_k}{r_{k+1}}, & \text{if } r_k \geq r_{k+1} \\ \frac{r_{k+1}}{r_k}, & \text{if } r_k < r_{k+1} \end{cases}, k = 1, 2, \dots, n-2 \quad (9)$$

$$\Delta r2_k = |r_{k+1} - r_k|, k = 1, 2, \dots, n-2 \quad (10)$$

where  $\Delta r1$  and  $\Delta r2$  represent the quotients and absolute difference of consecutive correlation coefficients, respectively. By construction,  $\Delta r1 \geq 1$  and  $\Delta r2 \geq 0$ . We selected 60 frames from video “person15\_jogging” for testing. The curves of  $\Delta r1$  and  $\Delta r2$  of the test video are shown in Figure 5, the left is  $\Delta r1$  and the right is  $\Delta r2$ . The red line represents the tamper position. Moreover, there appears a pair of peaks, which are called the abnormal points. Comparing to  $r$  and  $\Delta r2$ ,  $\Delta r1$  is more credible as video changes in content have little impact on it. Therefore, the quotients of consecutive correlation coefficients ( $\Delta r1$ ) are more suitable for inter-frame forgery detection. We set  $\Delta r1$  as the measure of the variation of consecutive correlation coefficients.



**Figure 5.** The curve of  $\Delta r1$  and  $\Delta r2$ . (a,b) Original video; (c,d) Forged video (frame deletion-15 frames were deleted).

### 3. Detection Scheme for Abnormal Points

From Section 2.3, we can conclude that frame insertion and deletion will influence the consistency of the variation of consecutive correlation coefficients. The peaks at the tampering position are the abnormal points that we need to detect. From Figure 5, we can find that the normal value of  $\Delta r1$  is near 1, this feature will have good clustering effect.

We hope to use clustering algorithm to detect outliers in sample points. *K*-means clustering is sensitive to outliers and may converge to a local minimum. It may help us to cluster the outliers into one category accurately.

In this section, we will describe the  $k$ -means clustering algorithm and the detection procedure for abnormal points.

### 3.1. The $k$ -Means Clustering Algorithm

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). The  $k$ -means (KM) clustering [32] is the most widely used clustering algorithm due to its simplicity and efficiency. The objective of the algorithm is to minimize an objective function in order to assign a group of data to its centroid.

Given a set of observations  $X(x_1, x_2, x_3, \dots, x_n)$ , where each observation is a  $d$ -dimensional real vector,  $k$ -means clustering aims to partition the  $n$  observations into  $k$  ( $k \leq n$ ) sets  $S = \{S_1, S_2, S_3, \dots, S_k\}$ , so as to minimize the within-cluster sum of squares. In other words, its objective is to find:

$$\operatorname{argmin}_s \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2, j = 1, 2, \dots, n \quad (11)$$

where  $\mu_i$  is the mean of points in  $S_i$ .

Combined with the characteristics of this paper, we introduce the step of  $k$ -means clustering algorithm as below.

Step 1. Load  $\Delta r1$  as observations  $X$ .

The variation of consecutive correlation coefficients  $\Delta r1$  can be obtained by applying Equation (9). Where inter-frame correlation coefficients  $r$  and 2-D PC for each frame were calculated according to Sections 2.1 and 2.2, respectively.

Step 2. Initialize clustering parameters.

Since our purpose is to classify the sample points into two categories, one cluster is normal points and another is abnormal, so we set cluster number  $k = 2$ .  $K$ -means algorithm intensively depends on the selection of initial clustering centers. While the outliers to be detected are the several largest values in the sample points, so we select the two largest values of  $X$  as cluster  $S_1$ , and the centroid is the mean of  $S_1$ . While the other values of  $X$  as cluster  $S_2$ , the centroid is the minimum of  $X$ . The storage in cluster is the location of value.

Step 3. Assign each observation  $x_p$  to the cluster whose mean yields the minimum distance between  $x_p$  and centroid of cluster.

Because  $\Delta r1$  is one-dimensional, the distance is defined as the absolute difference between  $x_p$  and centroid  $m$ :

$$S_i^{(t)} = \left\{ x_p : \left| x_p - m_i^{(t)} \right| \leq \left| x_p - m_j^{(t)} \right| \right\}, 1 \leq i, j \leq k \quad (12)$$

where  $(t)$  represents iterations. Each  $x_p$  is assigned to exactly one  $S^{(t)}$ , even if it could be assigned to two or more of them.

Step 4. Calculate the new means to be the centroids of the observations in the new clusters.

$$m_i^{(t+1)} = \frac{1}{N} \sum_{j=1}^N x_j, \forall x_j \in S_i^{(t)} \quad (13)$$

Step 5. Steps 3 and 4 are repeated until convergence has been reached.

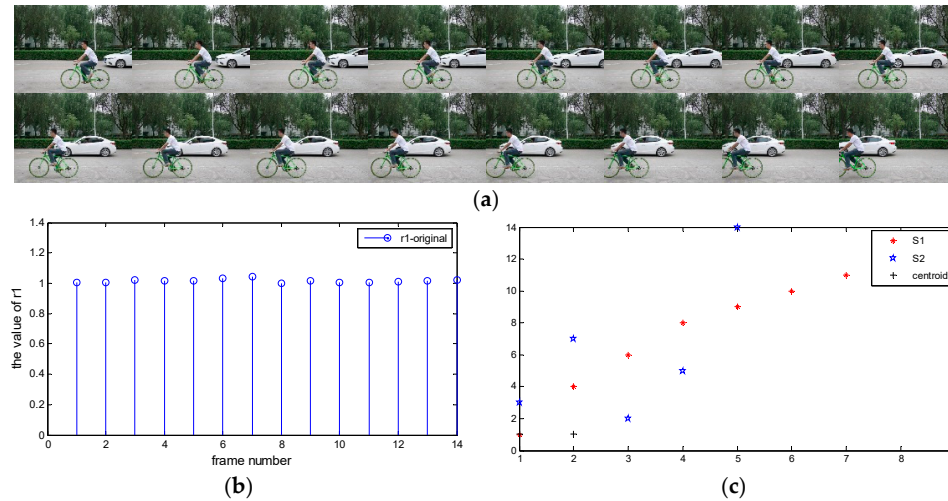
### 3.2. Abnormal Points Detection Based on KM

In this section, we will illustrate the effectiveness of the feature and the feasibility of the detection method in detail.



### 3.2.1. Clustering Results of Original Video

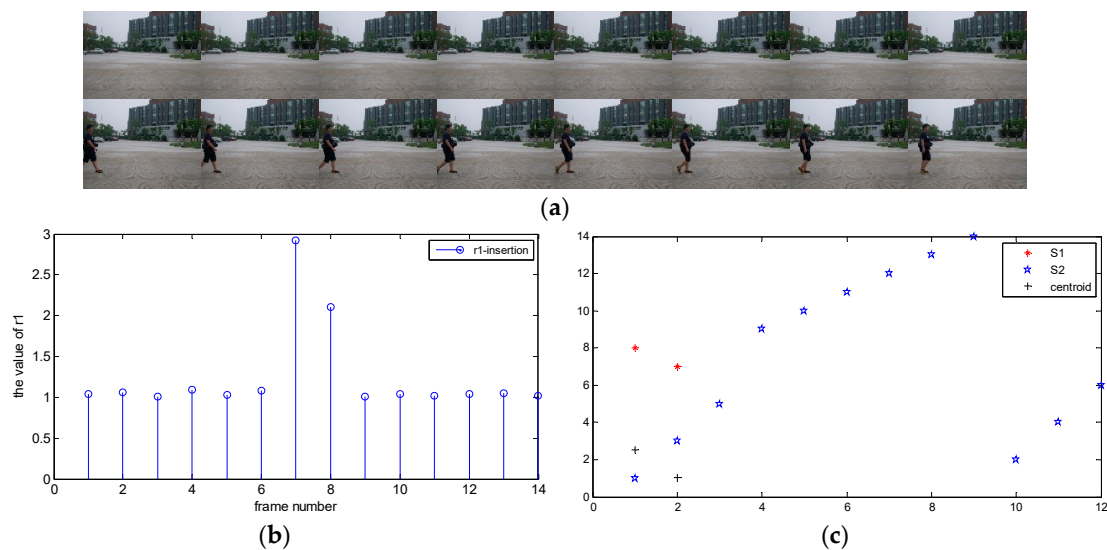
Figure 6a shows sixteen sequential frames of an original video, the resolution is  $640 \times 480$ .  $\Delta r1$  was calculated using Equation (9), and the clustering results of  $\Delta r1$  were shown in Figure 6c. ‘\*’ indicates the cluster  $S_1$ , ‘☆’ represents the cluster  $S_2$ , ‘+’ denotes the centroids of the clusters. The centroids of two clusters were 1.0389 and 1.0118, respectively. We find that both of the values of centroids were very close to 1, if the video is not forged.



**Figure 6.** Original video frames and clustering results. (a) Sixteen frames from an original video; (b) The curve of  $\Delta r1$ ; (c) clustering results.

### 3.2.2. Clustering Results of Forged Video by Frame Insertion

In Figure 7a, the last eight adjacent frames are the insertion frames from the same video with the first eight frames. In Figure 7b, there is a pair of peaks at the 7th and 8th frames because of the low correlation between the 8th and 9th frame. If we detect the abnormal points, we can prove that the video has been forged. The location of abnormal point is where tampering happens. In clustering results, cluster  $S_1$  represents the detected outliers. The centroids of two clusters were 2.5144 and 1.0413, respectively. We find that the centroid of  $S_1$  is deviate from 1, due to the tampered operation.

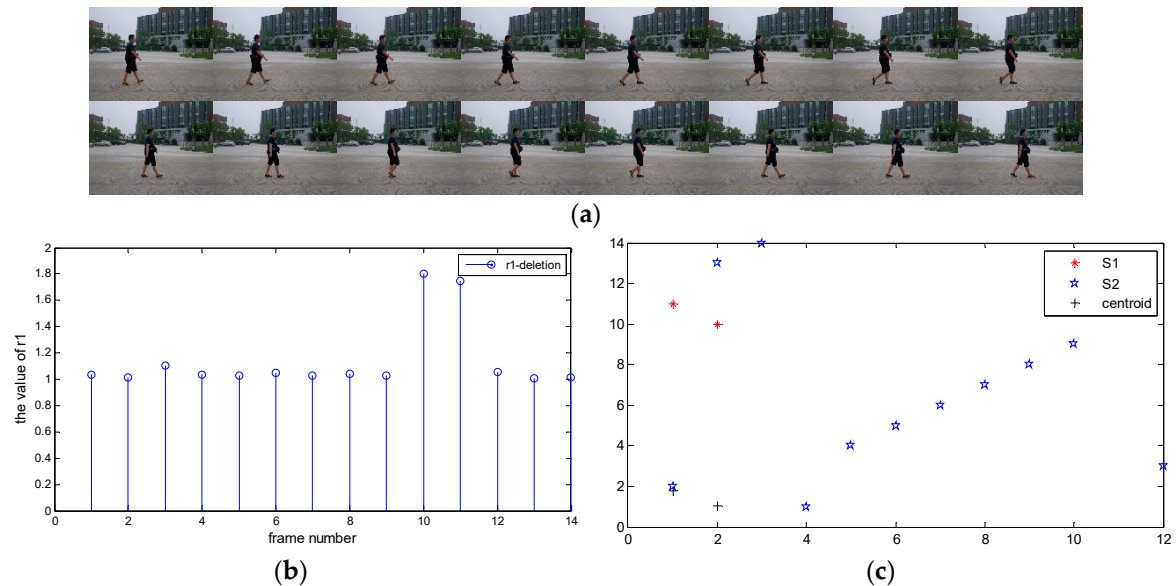


**Figure 7.** Frame insertion and clustering results. (a) Sixteen frames from a video tampered by frame insertion; (b) The curve of  $\Delta r1$ ; (c) clustering results.



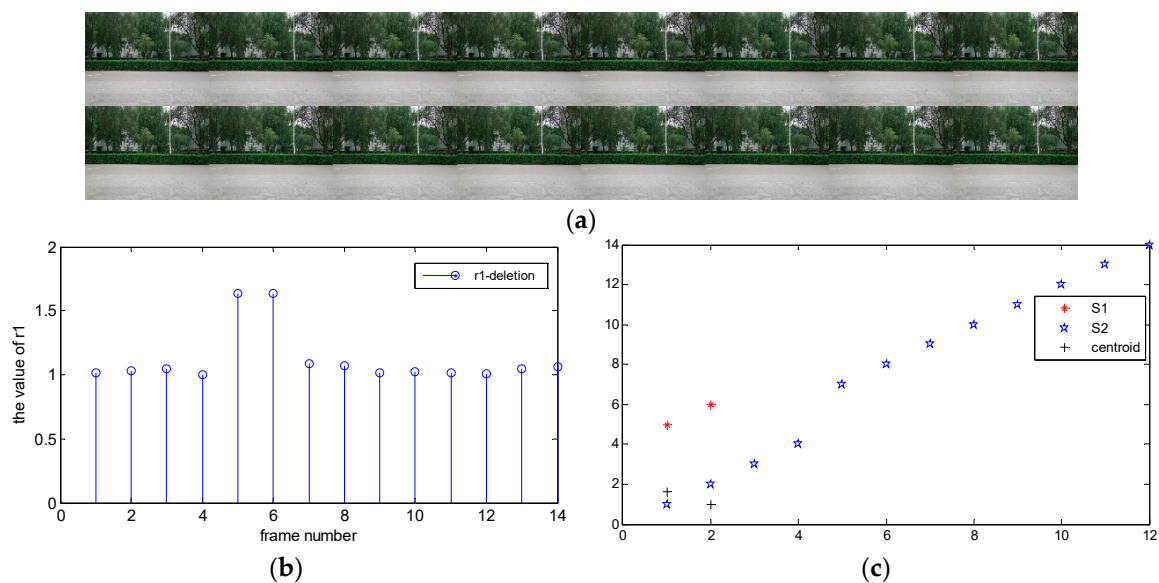
### 3.2.3. Clustering Results of Forged Video by Frame Deletion

In Figure 8a, the 11th and the 12th frame are not adjacent, because we delete several frames between them. The consistency of  $\Delta r1$  was destroyed at the delete position, and the frame number 10, 11 were clustered in  $S_1$ . The centroids of  $S_1$  and  $S_2$  were 1.7757 and 1.0359, respectively.



**Figure 8.** Frame deletion and clustering results. (a) Sixteen frames from a video tampered by frame deletion; (b) The curve of  $\Delta r1$ ; (c) clustering results.

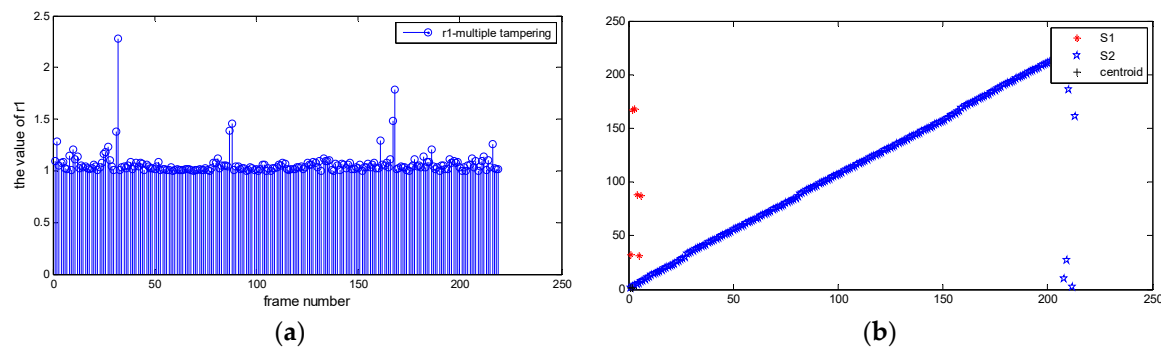
Sometimes, in order to falsify evidence, the forger will delete all frames which contain characters in the video. For example, in Figure 9a, we entirely remove a group of frames where a subject passes. There looks no discontinuities, but in fact, inconsistency appeared in the delete position. Shown in Figure 9b.



**Figure 9.** Frame deletion and clustering results. (a) Sixteen frames from a video tampered by frame deletion; (b) The curve of  $\Delta r1$ ; (c) clustering results.

### 3.2.4. Clustering Results of Forged Video by Multiple Tampering

Sometimes video suffers from multiple tampering issues. For example, the combination of frame insertion and deletion. As shown in Figure 10a, the video suffered three forgeries. Due to the difference of tampering positions and the number of frames, the values of peaks are different. However, all the tampering positions were precisely detected as shown in Figure 10b. The centroids of  $S_1$  and  $S_2$  were 1.6289 and 1.0489, respectively.



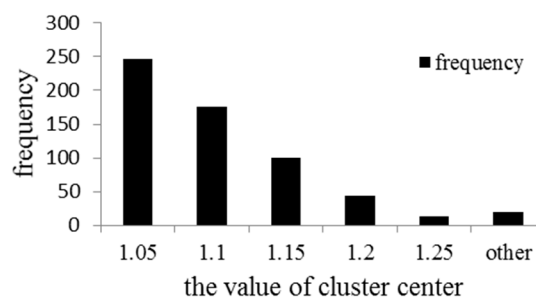
**Figure 10.** Clustering results of multiple tampering. (a) The curve of  $\Delta r1$ ; (b) clustering results.

From the above examples and analysis, we can conclude that the selected feature can effectively reflect the inter-frame manipulation. Moreover, the KM algorithm is able to locate the tampering position. From the clustering results, it is not difficult to see that if the video is forged, the value of  $S_1'$  centroid will be larger, which is deviated from 1, corresponding to the peaks of the tampering. Thus, the points in cluster  $S_1$  are the abnormal points. But when the video is original, both of the two clustering centers are very close to 1. Therefore, the value of the cluster center is the basis for judging whether the video has been tampered with or not. Moreover, generally the value of the cluster center in frame insertion is larger than that of frame deletion, but it is not true in all cases.

### 3.3. Threshold Decision

According to the description in Section 3.2, the normal and abnormal points are clustered into two categories by  $k$ -means clustering, the value of centroid is the basis for judging whether the video has been tampered with or not.

We tested the proposed method on the original sub-database, and analyzed the cluster center of  $S_1$  of 599 original videos. The histogram of the value of centroid is shown in Figure 11. The  $x$ -axis indicates the value of centroid of the cluster, the  $y$ -axis indicates the frequency of occurrence of each value.



**Figure 11.** The histogram of cluster center.

From Figure 11, we conclude that most of the clustering center values are less than 1.25. So we set  $T = 1.25$  as a threshold to distinguish original video and forged video. When centroid of  $S_1$  is greater than  $T$ , the video is detected as forged. Otherwise, the video is detected as original.

## 4. Experimental Results and Discussion

### 4.1. Dataset

In our experiments, two datasets of different sources and resolutions are selected. The first dataset selects from public Kungliga Tekniska Högskolan (KTH) [33], include one original sub-dataset, and four forgeries sub-datasets with different number of tampered frames. The videos contain six types of human actions, namely walking, jogging, running, boxing, hand clapping, and hand waving. The test videos compressed by MPEG, and were taken with a static background with 50fps frame rate, the resolution is  $180 \times 144$ . The number of tested videos of the five sub-datasets is 599, 599, 599, 599, and 598, respectively.

The second dataset has 480 videos, half of them are original videos, and the other are forged videos, include frame insertion and deletion. The number of tampered frames is more than 20. The composition of the second video dataset is shown in the Table 1.

**Table 1.** The composition of the second video dataset.

Source	Frame Rate	Resolution	Number of Original Videos	Number of Forged Videos
SULFA [34]	30fps	$320 \times 240$	120	120
Camera	30fps	$640 \times 480$	120	120

### 4.2. Evaluation Metrics and Method Assessment Procedure

In order to evaluate the validity of the scheme, we consider six performance indices: *TPR* (True Positive Rate), also known as *recall*, *TNR* (True Negative Rate), *PPV* (positive predictive value), also known as *precision*, *Accuracy*, *F1 score*, and *Location Precision*. *Accuracy* is the average detection accuracy. *F1 score* can be interpreted as the weighted average of the *precision* and *recall*. *Location Precision* is the percentage of correct localization among all the correct detected forgery videos:

$$TPR (recall) = \frac{TP}{TP + FN} \quad (14)$$

$$TNR = \frac{TN}{TN + FP} \quad (15)$$

$$PPV (precision) = \frac{TP}{TP + FP} \quad (16)$$

$$Accuracy (ACC) = \frac{TP + TN}{TP + FP + TN + FN} \quad (17)$$

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad (18)$$

$$Location Precision (LP) = \frac{TPL}{TP} \quad (19)$$

where *TP* is the number of true positive, means that the forged video was detected as forged, *TN* is the number of true negative, means that the original video was detected as original; *FP* is the number of false positive, means that the original video was detected as forged; *FN* is the number of false negative, means that the forged video was detected as original, *TPL* is the number of correct localization,  $(TP + FN)$  is the total number of forged videos,  $(TN + FP)$  is the total number of original videos,  $(TP + FP + TN + FN)$  is the total number of database.

In order to improve *LP*, the location results will be post-processed. We reject the single suspected abnormal points, since in the tamper position there is always a pair of peaks.

### 4.3. Experimental Results

We tested the proposed method on the two dataset. The statistical data of test results were given in Tables 2 and 3, where ‘-’ indicates that the data does not exist.

**Table 2.** Test results of the first dataset.

Source	TP	TN	FP	FN	TPL	TPR	TNR	Precision	ACC	F1	LP
Original	-	579	20	-	-	-	0.9666	-	-	-	-
25 frames inserted	599	-	-	0	582	1.00	-	-	-	-	0.9716
100 frames inserted	599	-	-	0	581	1.00	-	-	-	-	0.9699
25 frames deleted	550	-	-	49	500	0.9182	-	-	-	-	0.9091
100 frames deleted	586	-	-	12	560	0.9799	-	-	-	-	0.9556
Average	584	579	20	15	556	0.9750	0.9666	0.9669	0.9708	0.9724	0.9520

**Table 3.** Test results of the second dataset.

Source	TP	TN	FP	FN	TPL	TPR	TNR	Precision	ACC	F1	LP
SULFA	114	111	9	6	109	0.95	0.925	0.9268	0.9375	0.9383	0.9561
Camera	112	110	10	8	105	0.9333	0.9167	0.9180	0.925	0.9256	0.9375
All	226	221	19	14	214	0.9417	0.9208	0.9224	0.9313	0.9320	0.9469

We give a comparison of our results against the results reported in [24,25]. To ensure fairness, we tested all methods on the same database, which is a mixture of the two datasets mentioned above. And used the parameters recommended from the method. The results are shown in Table 4. We can see that the proposed method is better than reference [24,25].

**Table 4.** Comparison of three methods on mixed database.

Method	Recall	Precision	F1	LP
Reference [24]	0.8673	0.8954	0.8811	0.8896
Reference [25]	0.9272	0.9455	0.9363	0.9361
Proposed method	0.9584	0.9447	0.9522	0.9495

We also tested our algorithm on another dataset, the 5 frames-deleted and the 5 frames-inserted database, which are also generated from the original database. In the 5 frames-deleted database, the *TPR* and *LP* are 55.09% and 64.85%, respectively. In the 5 frames-inserted database, the *TPR* and *LP* are 100% and 96.83%, respectively. The reason is that, deleting 5 frames doesn’t efficiently change the video contents, and the correlation of adjacent frames changes a little after the tamper. Inserting a few frames may have a great impact on video content. Therefore, the *TPR* and *LP* are higher in frame insertion detection than frame deletion.

### 4.4. Time Complexity Analysis

The proposed method consists of feature extraction and abnormal point localization. During feature extraction, we extract the 2-D phase congruency of each frame. The computational complexity is related to the video resolution and the frame number. The higher the resolution and the frame number, the more time consuming the process. The *k*-means clustering algorithm was used to localize the abnormal points. The time complexity is  $O(n \times k \times t)$ , where *n* is the number of objects in dataset, *t* represents the number of iterations of the algorithm, and *k* is the number of clusters. We test two different resolutions with the frame number is 500, and the results are shown in Table 5.

**Table 5.** The time consuming of the algorithm.

Video Resolution	Frame Number	Time of Feature Extract (s)	Time of Clustering (s)
320 × 240	500	185.56	0.0545
640 × 480	500	373.15	0.0632

## 5. Conclusions

In this paper, an inter-frame forgery detection scheme based on 2D phase congruency and *k*-means clustering was proposed for surveillance video. We calculate 2D PC for each frame firstly. Then, the correlation coefficients of adjacent frames and the variation of consecutive correlation coefficients are obtained. Finally, the discontinuous points caused by tampering are detected by using *k*-means clustering algorithm. Experimental results show that our approach can detect and localize the tampering positions efficiently. The shortcoming is that when deleting frames appear at the beginning or the end of the video, the detection method is impossible. Moreover, in this scheme, the *TPR* and *LP* are higher in frame insertion detection than frame deletion. Therefore, in the future work, we will focus on finding a better solution to improve the precision of detecting frame deletion.

In addition, the method locates the inter-frame tampering operations without distinguishing whether the inserted frames are copied from the same video or are spliced from another video. In future work, we will try to distinguish different frame insertion forgery operations.

**Author Contributions:** Q.L., R.W. and D.X. discussed and designed the forgery detection method. Q.L. designed and implemented the detection algorithm, D.X. tested and analyzed the experimental results, R.W. thoroughly reviewed and improved the paper. All authors have discussed and contributed to the manuscript. All authors have read and approved the final manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (Grant No. U1736215, 61672302, 61771270), Zhejiang Natural Science Foundation (Grant No. LZ15F020002, LY17F020010, LY17F020013), Ningbo Natural Science Foundation (Grant No. 2017A610123), Mobile Network Application Technology Key Laboratory of Zhejiang Province (Grant No. F2018001), Ningbo University Fund (Grant No. XKXL1509, XKXL1503) and K.C. Wong Magna Fund in Ningbo University.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Katsaounidou, A.; Dimoulas, C.; Veglis, A. *Cross-Media Authentication and Verification: Emerging Research and Opportunities*; IGI Global: Hershey, PA, USA, 2018; pp. 155–188.
2. Arab, F.; Abdullah, S.M.; Hashim, S.Z.M.; Manaf, A.A.; Zamani, M. A robust video watermarking technique for the tamper detection of surveillance systems. *Multimed. Tools Appl.* **2016**, *75*, 10855–10885. [[CrossRef](#)]
3. Chen, S.; Pande, A.; Zeng, K.; Mohapatra, P. Live video forensics: Source identification in lossy wireless networks. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 28–39. [[CrossRef](#)]
4. Amerini, I.; Caldelli, R.; Del Mastio, A.; Di Fuccia, A.; Molinari, C.; Rizzo, A.P. Dealing with video source identification in social networks. *Signal Process. Image Commun.* **2017**, *57*, 1–7. [[CrossRef](#)]
5. Tao, J.J.; Jia, L.L.; You, Y. Review of passive-blind detection in digital video forgery based on sensing and imaging techniques. *Proc. SPIE* **2017**, *10244*, 102441C.
6. Li, Z.H.; Jia, R.S.; Zhang, Z.Z.; Liang, X.Y.; Wang, J.W. Double HEVC compression detection with different bitrates based on co-occurrence matrix of PU types and DCT coefficients. In Proceedings of the ITM Web of Conferences, Guangzhou, China, 26–28 May 2017; p. 01020.
7. He, P.; Jiang, X.; Sun, T.; Wang, S. Double compression detection based on local motion vector field analysis in static-background videos. *J. Vis. Commun. Image R* **2016**, *35*, 55–66. [[CrossRef](#)]
8. Zheng, J.; Sun, T.; Jiang, X.; He, P. Double H.264 compression detection scheme based on prediction residual of background regions. In *Intelligent Computing Theories and Application*; Springer: Cham, Switzerland, 2017; pp. 471–482.

9. Li, L.; Wang, X.; Zhang, W.; Yang, G.; Hu, G. Detecting removed object from video with stationary background. In Proceedings of the International Workshop on Digital Forensics and Watermarking, Auckland, New Zealand, 1–4 October 2013.
10. Lin, C.S.; Tsay, J.J. A passive approach for effective detection and localization of region-level video forgery with spatio-temporal coherence analysis. *Digit. Investig.* **2014**, *11*, 120–140. [[CrossRef](#)]
11. Chen, R.C.; Yang, G.B.; Zhu, N.B. Detection of object-based manipulation by the statistical features of object contour. *Forensic Sci. Int.* **2014**, *236*, 164–169.
12. Su, L.; Huang, T.; Yang, J. A video forgery detection algorithm based on compressive sensing. *Multimed. Tools Appl.* **2015**, *74*, 6641–6656. [[CrossRef](#)]
13. Mulla, M.U.; Bevinamarad, P.R. Review of techniques for the detection of passive video forgeries. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.* **2017**, *2*, 199–203.
14. Mizher, M.A.; Ang, M.C.; Mazhar, A.A.; Mizher, M.A. A review of video falsifying techniques and video forgery detection techniques. *Int. J. Electron. Secur. Digit. Forensics* **2017**, *9*, 191–208. [[CrossRef](#)]
15. Han, Y.X.; Sun, T.F.; Jiang, X.H. Design and performance optimization of surveillance video Inter-frame forgery detection system. *Commun. Technol.* **2018**, *51*, 215–220.
16. Su, Y.T.; Ning, W.Z.; Zhang, C.Q. A frame tampering detection algorithm for MPEG videos. In Proceedings of the IEEE Joint International Information Technology and Artificial Intelligence Conference, Chongqing, China, 20–22 August 2011; pp. 461–464.
17. Dong, Q.; Yang, G.B.; Zhu, N.B. A MCEA based passive forensics scheme for detecting frame-based video tampering. *Digit. Investig.* **2012**, *9*, 151–159. [[CrossRef](#)]
18. Shanableh, T. Detection of frame deletion for digital video forensics. *Digit. Investig.* **2013**, *10*, 350–360. [[CrossRef](#)]
19. Feng, C.; Xu, Z.; Jia, S.; Zhang, W.; Xu, Y. Motion-adaptive frame deletion detection for digital video forensics. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 2543–2554. [[CrossRef](#)]
20. Kang, X.; Liu, J.; Liu, H.; Wang, Z.J. Forensics and counter anti-forensics of video inter-frame forgery. *Multimed. Tools Appl.* **2016**, *75*, 13833–13853. [[CrossRef](#)]
21. Chao, J.; Jiang, X.H.; Sun, T.F. A novel video Inter-frame forgery model detection scheme based on optical flow consistency. In *Digital Forensics and Watermarking*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 267–281.
22. Wu, Y.; Jiang, X.; Sun, T.; Wang, W. Exposing video inter-frame forgery based on velocity field consistency. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 2674–2678.
23. Zhang, Z.; Hou, J.; Li, Z.; Li, D. Inter-frame forgery detection for static-background video based on MVP consistency. *Proc. Lect. Notes Comput. Sci.* **2016**, *9569*, 94–106.
24. Zhang, Z.; Hou, J.; Ma, Q.; Li, Z. Efficient video frame insertion and deletion detection based on inconsistency of correlations between local binary pattern coded frames. *Secur. Commun. Netw.* **2015**, *8*, 311–320. [[CrossRef](#)]
25. Zhang, X.L.; Huang, T.Q.; Lin, J.; Huang, W. Video tamper detection method based on nonnegative tensor factorization. *Chin. J. Netw. Inf. Secur.* **2017**, *3*, 42–49.
26. Zhao, Y.; Pang, T.; Liang, X.; Li, Z. Frame-deletion detection for static-background video based on multi-scale mutual information. In Proceedings of the International Conference on Cloud Computing and Security (ICCCS), Nanjing, China, 16–18 June 2017; pp. 371–384.
27. Chen, W.; Shi, Y.Q.; Su, W. Image splicing detection using 2-D phase congruency and statistical moments of characteristic function. *Proc. SPIE* **2007**, *6505*, 65050–65058.
28. Morrone, M.C.; Ross, J.; Burr, D.C.; Owens, R. Mach bands are phase dependent. *Nature* **1986**, *324*, 250–253. [[CrossRef](#)]
29. Morrone, M.C.; Burr, D.C. Feature detection in human vision: A phase-dependent energy model. *Proc. R. Soc. Lond. B* **1988**, *235*, 221–245. [[CrossRef](#)] [[PubMed](#)]
30. Kovess, P. Image features from phase congruency. *J. Comput. Vis. Res.* **1999**, *1*, 1–26.
31. Huang, T.Q.; Chen, Z.W.; Su, L.C.; Zheng, Z.; Yuan, X.J. Digital video forgeries detection based on content continuity. *J. Nanjing Univ. (Nat. Sci.)* **2011**, *47*, 493–503.
32. *k*-Means Clustering. Available online: [https://en.wikipedia.org/wiki/K-means\\_clustering#cite\\_note-lloyd1957-3](https://en.wikipedia.org/wiki/K-means_clustering#cite_note-lloyd1957-3) (accessed on 22 November 2018).



33. Schuldt, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. *Pattern Recognit.* **2004**, *33*, 32–36.
34. Qadir, G.; Yahaya, S.; Ho, A.T.S. Surrey university library for forensic analysis (SULFA) of video content. In Proceedings of the IET Conference on Image Processing, London, UK, 3–4 July 2012; pp. 1–6.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).