An improved fingerprinting algorithm for detection of video frame duplication forgery

Hu, Yongjian, Li, Chang-Tsun, Wang, Yufei and Liu, Bei-Bei. 2012. An improved fingerprinting algorithm for detection of video frame duplication forgery, *International journal of digital crime and forensics*, vol. 4, no. 3, Jul-Sep, pp. 20-32.

DOI: 10.4018/jdcf.2012070102

An Improved Fingerprinting Algorithm for Detection of Video Frame Duplication Forgery

Yongjian Hu, University of Warwick, UK, & South China University of Technology, China

Chang-Tsun Li, University of Warwick, UK

Yufei Wang, South China University of Technology, China

Bei-bei Liu, South China University of Technology, China

ABSTRACT

Frame duplication is a common way of digital video forgeries. State-of-the-art approaches of duplication detection usually suffer from heavy computational load. In this paper, the authors propose a new algorithm to detect duplicated frames based on video sub-sequence fingerprints. The fingerprints employed are extracted from the DCT coefficients of the temporally informative representative images (TIRIs) of the sub-sequences. Compared with other similar algorithms, this study focuses on improving fingerprints representing video sub-sequences and introducing a simple metric for the matching of video sub-sequences. Experimental results show that the proposed algorithm overall outperforms three related duplication forgery detection algorithms in terms of computational efficiency, detection accuracy and robustness against common video operations like compression and brightness change.

Keywords: *Algorithm, Digital Fingerprint, Forgery Detection, Frame Duplication, Temporally Informative Representative Images- Discrete Cosine Transform (TIRI-DCT)*

INTRODUCTION

Due to the popularity of camcorder and multimedia cell phone, digital video is more and more widely used in our everyday life and work. The rising of video sharing sites on Internet makes the spread of digital video easy and fast. The forgery of digital video is also facilitated by a variety of video editing software, which may cause serious forensic problems if the tampered videos are used in legal evidence, news reports or security monitoring tapes. Since the detection of video forgery is challenged by the huge amount of digital video data, the research of accurate and rapid forgery detection algorithms is of paramount significance.

There are various ways of tampering with digital videos, inspiring a wide range of detection approaches, such as the algorithm based on compression and quantization (Wang et al., 2006, 2009; Su et al., 2011), effect of interlac-

DOI: 10.4018/jdcf.2012070102

ing (Wang et al., 2007a), characteristic of noise (Hsu et al., 2008; Kobayashi et al., 2010), fusion of features (Chetty et al., 2010) and so on. Some algorithms detect the forgery based on the artifacts brought by tampering, such as the motion-compensated edge artifacts (Su et al., 2009), the ghost shadow artifacts (Zhang et al., 2009) and so on. Among various tampering approaches, frame duplication is a simple but the most widely used one, so the detection of frame duplication has attracted lots of attention from researchers. Wang et al. (2007b) proposed a frame duplication detection algorithm based on correlation coefficient matrix. While achieving satisfactory detection accuracy, the algorithm requires heavy computational load due to the large amount of correlation calculation. To reduce the computational cost, Lin et al. (2011) proposed to use histogram difference (HD) instead of correlation coefficients as the detection features. However, the HD features do not show strong robustness against common video operations or attacks. In practical applications, both computational efficiency and robustness must be taken into account. To satisfy these requirements, we aim at designing a fast and robust duplication detection algorithm. This work focuses on improving fingerprints representing video sub-sequences and introducing a simple metric to judge whether two video sub-sequences are matched.

The remainder of this paper is organized as follows: First, we will review two related frame duplication detection algorithms. Our proposed detection algorithm will be elaborated afterwards. Then we present the results of comparative experiment and discussion. Finally we will conclude the paper in the last section.

RELATED WORKS

Wang et al. (2007b) proposed a frame duplication detection algorithm. The video is first divided into overlapping sub-sequences, with only one different frame between adjacent sub-sequences. For each sub-sequence, they computed the correlation coefficient between each pair of frames, composing a correlation coefficient matrix that carries the temporal information of this sub-sequence. To judge whether two sub-sequences are duplicated, the correlation coefficient between the matrixes of these two sub-sequences are calculated and compared with a threshold. If the coefficient exceeds the threshold, the two sub-sequences may be duplicated. To confirm the duplication, spatial information is used for further detection. Specifically, they divided the two frames into non-overlapping blocks and calculated the correlation coefficient between each pair of blocks in corresponding positions. They recorded the number of block pairs having large correlation coefficients. If the number exceeds a predefined threshold, they considered the two frames were duplicates of each other, which indicated that the video had undergone duplication forgery. However, the calculation of correlation coefficient is known to be time consuming; and moreover, the correlation coefficient was employed twice to represent the temporal and spatial information, respectively. As a result, their algorithm requires a heavy computation load.

The algorithm proposed by Lin et al. (2011) has similar pre-processing operation to that of Wang's (2007b). The video is also divided into overlapping sub-sequences with only one different frame between adjacent sub-sequences. For each sub-sequence, the histogram difference (HD) rather than the correlation coefficient between each two adjacent frames is calculated:

$$HD = \frac{1}{N_{bin}} \sum_{i=1}^{N_{bin}} \left| h_R^q(i) - h_R^t(i) \right| + \left| h_G^q(i) - h_G^t(i) \right| + \left| h_B^q(i) - h_B^t(i) \right| \quad (1)$$

where h_R^q, h_G^q and h_B^q represent the histograms of R, G and B channels of a frame while h_R^t, h_G^t and h_B^t represent the histograms of its adjacent frame. N_{bin} denotes the number of bins in the color histograms.

For a sub-sequence of length N, a feature vector with $N-1$ elements is obtained by

calculating the HD between each two adjacent frames. This vector contains the temporal information of the sub-sequence. If the correlation coefficient between two HD vectors exceeds a threshold, the corresponding two sub-sequences are likely to be duplicated. Further investigation based on spatial information is then required. The authors divided the two suspicious frames into non-overlapping blocks and computed the HD between each pair of corresponding blocks. They recorded the number of block pairs having small HD values. If the number exceeds a predefined threshold, they judged that the two frames were duplicates of each other and the video had undergone duplication forgery. Compared to the correlation coefficient calculation in Wang et al. (2007b), the small complexity of HD calculating makes this approach more efficient.

THE PROPOSED ALGORITHM

Temporal features can reflect the characteristics of a video sequence over time while spatial features can reflect the characteristics of a frame. The algorithms of Wang et al. (2007b) and Lin et al. (2011) both adopt a two-step detection scheme in which temporal and spatial information of each sub-sequence is dealt with separately. However, we think that the detection time can be shortened when the temporal information and the spatial information are integrated into one feature. A possible way of integrating the temporal information and the spatial information is to consider a video as a three-dimensional matrix and to perform three-dimensional discrete cosine transform (3D-DCT) (Coskun et al., 2006) on it. But the complexity of computing 3D-DCT is known to be very high. We find that the DCT of temporally informative representative images (TIRI-DCT) proposed in Esmaeili et al. (2011) is an appropriate feature for each sub-sequence. TIRI-DCT was originally used for a digital fingerprint for video copy detection. In this paper, we borrow the idea and improve fingerprints representing video sub-sequences. We then apply these fingerprints to frame duplication detection.

The construction steps of the fingerprint are described as follows. Firstly, the video is divided into overlapping sub-sequence with only one different frame between adjacent sub-sequences. Each sub-sequence is downsampled in time domain to reduce the number of frames to be computed. Downsampling often causes the loss of temporal information, but the detection accuracy would not be affected as long as the sampling interval is smaller than the length of the duplicated video sequence. The TIRI of each sub-sequence is calculated as the weighted sum of the frames:

$$l'_{m,n} = \sum_{k=1}^{J} \omega_k l_{m,n,k} \tag{2}$$

where $l_{m,n,k}$ is the luminance value of the (m,n)th pixel of the kth frame, and ω_k is the weight of the kth frame, J denotes the frame number of the sub-sequence after downsampling. Apparently the TIRI contains the temporal information of each sub-sequence.

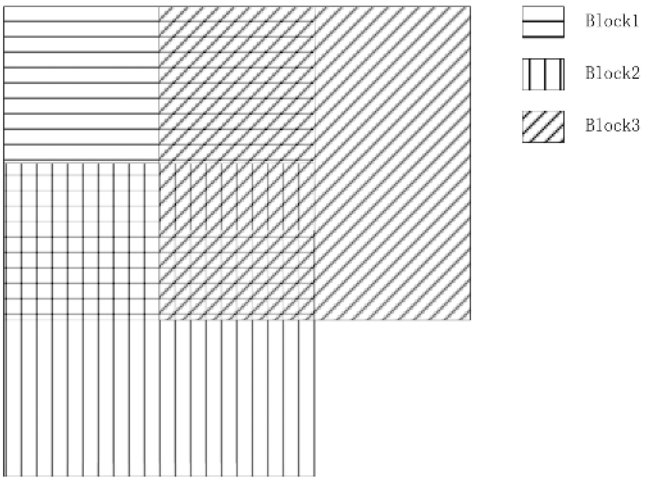
Secondly, the TIRI is divided into overlapping blocks of size $2w \times 2w$, each having 50% of the area overlapped with the vertical and horizontal adjacent blocks, respectively. Figure 1 shows the partition of each block. The reason for overlapping the blocks is to get adequate spatial information for accurate detection.

Thirdly, for each block, we perform two-dimensional DCT and extract the two AC coefficients closest to the DC coefficient as the features. The two AC coefficients represent the vertical and horizontal texture features of the block. The coefficients can be calculated through the following two equations:

$$\alpha_i = v^T B_i E \tag{3}$$

$$\beta_i = E^T B_i V \tag{4}$$

Figure 1. The block partitioning of the TIRI



where α_i and β_i denote the first vertical and horizontal AC coefficients of the ith block B_i, respectively. V is a column vector with $2w$ elements and

$$V = \left[\cos\left(0.5\pi/2w\right), \cos\left(1.5\pi/2w\right), \cdots, \cos\left(\pi - 0.5\pi/2w\right)\right],$$

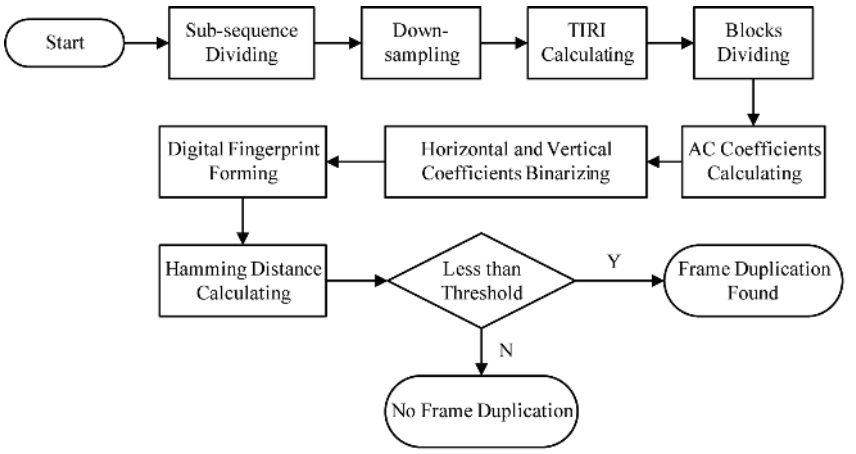
E is an all ones column vector of length $2w$. With formulas (3) and (4), we can obtain the two AC coefficients without fully performing the DCT, which greatly reduces the computational cost.

Finally, the AC coefficients obtained are binarized to act as the digital fingerprint of each sub-sequence. We binarize the vertical and horizontal coefficients separately so as to reduce the loss of directional information of each block. Specifically, suppose each TIRI is divided into N_B overlapping blocks, we calculate the medians m_α and m_β of the AC coefficients as follows:

$$\begin{cases} m_\alpha = median(\alpha_1, ..., \alpha_{N_B}) \\ m_\beta = median(\beta_1, ..., \beta_{N_B}) \end{cases} \quad (5)$$

These two medians are used as the thresholds, respectively. In particular, a vertical coefficient α_i is binarized as 0 if it is less than the threshold m_α; otherwise, it is binarized as 1. Likewise, a horizontal coefficient β_i is binarized. After the operation, we concatenate the two binarized coefficient vectors together to form a digital fingerprint that contains both the temporal and spatial information of each subsequence. Note that, if the number of blocks N_B is even, m_α / m_β is not any vertical/horizontal AC coefficient but the average of the two numbers in the middle of the sorted vertical/horizontal AC coefficients. As a result, the binarization processing always yields the equal number of 0's and 1's vertically as well as horizontally. So the resulting fingerprint consists of the same number of 0's and 1's. However, if the number of blocks N_B is odd, m_α / m_β is the number in the middle of the sorted vertical/horizontal AC coefficients. In this case, we have to binarize the medians themselves. If both m_α and m_β are binarized as 0 or 1, the number of 0's and that of 1's in the fingerprint are not equal. To evenly reflect vertical and horizontal information from the AC coefficients of the

Figure 2. Flowchart of the proposed algorithm



DCT, this work proposes to binarize m_α / m_β as 0 and at the same time, m_β / m_α as 1. This scheme produces a balanced fingerprint, that is, the number of 0's and 1's are equal. We have to stress that Esmaeili et al. (2011) once proposed another binarization way to create the digital fingerprint. They first calculated the median of all the vertical and horizontal AC coefficients, and then binarize those coefficients using the median as the threshold to obtain a binary sequence, i.e., the fingerprint. Because their median is calculated from all the AC coefficients without distinguishing directions, it is most likely to create a fingerprint which contains more elements from the vertical coefficients than from the horizontal coefficients or the opposite. In other words, the resulting fingerprint cannot evenly reflect the characteristics of vertical coefficients and horizontal coefficients. This is the basic difference between that method and our method in the aspect of construction of the fingerprint.

After obtaining the fingerprints of the sub-sequences, we need to compare each pair of the fingerprints to determine whether they are duplicate sub-sequences. If duplication forgery exists, the fingerprints from different sub-sequences should be similar. Hence, how to measure the similarity of two sub-sequences is crucial for the success of the detection algorithm. In this work, we propose to use the Hamming distance as a metric for fingerprint matching. The Hamming distance would be small when two video sequences are the same or very similar. As a result, if the Hamming distance between the fingerprints of two sub-sequences is less than a predefined threshold, we think that the frame duplication forgery exists. The calculation of Hamming distance is simple and is realizable using XOR binary operation. Hence the detection process of our algorithm is computationally efficient. Figure 2 gives the flowchart of our algorithm.

RESULTS AND ANALYSIS

Each video clip used in our experiments has 300 frames, with a sequence of about 100 frames intentionally duplicated. The original videos are in YUV format without compression.

Parameters Settings

There are three CIF format video clips used in experiments. We intentionally select videos with different characteristics. In particular, *coastguard_cif* has more dynamic content, where both the object and the background are moving fast. For n*ews_cif*, only very few regions

Figure 3. Sample frames from videos used in experiments: coastguard_cif

Figure 4. Sample frames from videos used in experiments: news_cif

(e.g., mouth) on its objects in front have obvious movements. However, part of its background (e.g., the content on the small screen) changes very fast. Specifically, the content on the small screen is a clip of ballet and this clip is repeatedly played twice. In other words, *news_cif* is a video clip with the background containing partially duplicated content. On the other hand, in the first half of *foreman_cif*, this video has relatively static background and moving object in front. But in the last half, the object in front disappears and only the relatively static background is left. The sample frames of these three videos are showed in Figure 3 through Figure 5. We tampered with these video clips and the detailed duplication information is listed in Table 1.

Now we discuss how to set up the parameters employed in our algorithm, including the length of sub-sequence, the interval of downsampling, ω_k for (2.2), the block size used for the construction of the fingerprint, and the decision threshold for frame duplication detec-

Table 1. Videos and their duplication information

Forgery Video	Duplication Information
coastguard_1	Copying 150th to 279th frames of coastguard_cif to its 0th to 129th frames.
news_1	Copying 0th to 99th frames of news_cif to its 150th to 249th frames.
foreman_1	Copying 0th to 119th frames of foreman_cif to its 150th to 269th frames.

Figure 5. Sample frames from videos used in experiments: foreman_cif

tion. The length of the sub-sequence is an important parameter in experiments. A short length would increase the calculation time while a long length would miss the detection of duplication sub-sequences shorter than this length. So the length of sub-sequence is the lower bound for duplication forgery detection. In Wang et al. (2007b), the length of the sub-sequence was set to 30. For detection of 100 to 200 long duplication frames, this setting may make a good tradeoff between computational complexity and detection capability. In this work, we set the length of the sub-sequence to 31 instead of 30 due to the effect of downsampling. When generating the TIRIs, we tend to reserve the first and last frames of the sub-sequence after downsampling in order to well represent the sub-sequence. We set the downsampling interval to 4 so that a 31-frame long sub-sequence can be downsampled into 7 frames for the calculation of TIRI. In this case, both the first and last frames are selected. For the sake of comparison, this work also sets the length of sub-sequence to 31 for the algorithms in Wang et al. (2007b) and the HD-based algorithm (Lin et al., 2011). The other parameters for these algorithms have not been altered.

For ω_k in formula (2-2), we choose the exponential weighting function $\omega_k = \gamma^k$, where $\gamma = 0.64$ according to Esmaeili et al. (2011). The block size w is determined based on the results in Table 2, where the results have been obtained with different values of w. Note that, in this paper, the TPR (True Positive Rate) is calculated by dividing the number of cor-

Table 2. Performance with different values of block size w

Video	Value of w	TPR (%)	FPR (%)	Time (ms)
coastguard_1	4	100	0	63008
	8	100	0	16014
	16	100	20	12573
news_1	4	100	0	68234
	8	100	0	16391
	16	100	100	12542
foreman_1	4	100	0	65832
	8	100	0	16462
	16	100	46.67	12682

rectly detected forgery frames by the total number of the forgery frames, and the FPR (False Positive Rate) is calculated by dividing the wrongly detected forgery frames by the total number of the no forgery frames. The use of TPR and FPR can comprehensively describe the performance of an algorithm. We can see in Table 2 that a smaller value of w often generates a longer fingerprint and obtains higher detection sensitivity but requires longer computation time, while a larger value of w makes the fingerprint shorter and the robustness stronger but the detection sensitivity lower. Apparently, the setting of $w = 8$ is a good tradeoff between time consumption and detection sensitivity.

The setting of decision threshold for the Hamming distance affects the detection performance. A low threshold would raise the sensitivity of detection, and at the same time, produce more false alarms. However, a high threshold would yield the converse result. In practice, the selection of an optimal threshold is quite complex since it involves the resolution of the video employed, the length of digital fingerprint and the block size. According to our experiments, we set the decision threshold to 3 in this work, which is a tradeoff between detection accuracy and robustness.

Test on Video Clips without Undergoing Video Operations

We first evaluate the performance of the proposed algorithm on videos that have not undergone video operations. Here video operations refer to content-preserving operations/attacks (distortions) that are made to the video intentionally or unintentionally. The commonly used operations include format changes, signal processing operations, changes in brightness/contrast, added noise, rotation, cropping, logo insertion, compression, etc. To justify our discussions, we compare our algorithm with the correlation coefficient-based algorithm (Wang et al., 2007b), the HD-based algorithm (Lin et al., 2011) and the TIRI-DCT based algorithm (Esmaeili et al., 2011). It is worth mentioning that the TIRI-DCT based algorithm (Esmaeili et al., 2011) was originally proposed for video copy detection rather than frame duplication detection. Generally, video copy means the copy

Table 3. Results about video clips without video operations

Video	Algorithm	TPR(%)	FPR (%)	Time (ms)
coastguard_1	Ours	100	0	16014
	TIRI-DCT	100	0	19734
	Correlation Coefficient	100	0	238056
	HD	100	0	11555
news_1	Ours	100	0	16391
	TIRI-DCT	100	6	20233
	Correlation Coefficient	100	0	176483
	HD	100	0	14570
foreman_1	Ours	100	0	16462
	TIRI-DCT	100	0	20077
	Correlation Coefficient	100	0	178464
	HD	100	0	10498

of all the frames that constitute the video. So we directly apply this algorithm to frame duplication detection. The experimental results are given in Table 3. The TPR values for all of the four algorithms are 100%. On the other hand, the FPR values for all the algorithms except the TIRI-DCT based algorithm (Esmaeili et al., 2011) are 0. For the TIRI-DCT based algorithm (Esmaeili et al., 2011), however, it has a 6% FPR when dealing with *news_1*. This result implies that the algorithm is very sensitive to the duplicated content, and even this duplicated content (i.e., the drama shown on the small screen in the background) is truly part of the original content and has not undergone any video processing, this algorithm still regards the video clip as forgery video. Apparently, such an over sensitivity is not a good property because it would greatly increase false alarm rates in real-world applications. In terms of computational efficiency, our algorithm requires much less time than the algorithm in Wang et al. (2007b). It also has a small advantage over the algorithm in Esmaeili et al. (2011). But the HD-based algorithm requires even less time than our algorithm due to its small computational complexity. Overall, the HD-based algorithm (Lin et al., 2011) has the best performance on videos without undergoing video operations.

Test on Video Clips with Brightness Change

A fingerprint should be robust to the content-preserving distortions present in a video (Esmaeili et al., 2011). Due to various reasons (e.g., to improve video quality, save storage space or communicate on a narrow band-width network), video operations are widely used in real-world applications, which may greatly affect the performance of forgery detection algorithms. This work focuses on investigating the robustness of the detection algorithm against typical video processing such as brightness change and video compression. In this subsection, we first address the performance of the proposed algorithm on videos subject to brightness change. We increase by 1 the average luminance of the duplicated frames to generate the tampered video *coastguard_2*, *news_2* and *foreman_2* and reduce by 1 the average luminance of the duplicated frames to produce the tampered video *coastguard_3*, *news_3* and *foreman_3*. The detection results are given in Table 4.

It can be observed that the FPR values for the four algorithms remain the same as those obtained on the videos without undertaking video operations. With respect to TPR, three algorithms have obvious changes. However, the performance of the correlation-based algorithm (Wang et al., 2007b) is least affected. The main reason is that the mean subtraction operation in the normalized correlation calculation can suppress the effect of the average change to pixel values to some degree. We can observe that the algorithm in Wang et al. (2007b) outperforms either our algorithm or the algorithm in Esmaeili et al. (2011) for *coastguard_2* although it has the same good performance as our algorithm in other cases. On the other hand, the impact of brightness change on our algorithm and the algorithm in (Esmaeili et al., 2011) can be reflected by the decrease of TPR values in the cases of *coastguard_2, foreman_2* and *foreman_3*. However, our algorithm is a little better than the algorithm in Esmaeili et al. (2011) when we take into account the FPR values. The latter algorithm still suffers from a 6% FPR when handling *news_2*. We owe the similar performance between our algorithm and the algorithm in Esmaeili et al. (2011) to the fact that both of them adopt the same AC coefficients of DCT for generating the fingerprint. The average luminance increase or decrease often does not have much impact on the AC coefficients. In contrast, the HD-based algorithm in Lin et al. (2011) has the worst performance among the four algorithms. For all the test videos, the TPR values for this algorithm are 0, meaning that it can barely resist the effect of brightness change. The underlying reason is that such a video operation usually results in a significant change to the histogram difference.

Let us compare computation time (i.e., computational complexity) of these four algorithms. The correlation-based algorithm in

Table 4. Results about video clips with brightness change

Videos	Algorithm	TPR (%)	FPR (%)	Time (ms)
coastguard_2	Ours	99.23	0	15958
	TIRI-DCT	99.23	0	19453
	Correlation Coefficient	100	0	236044
	HD	0	0	1201
coastguard_3	Ours	100	0	15896
	TIRI-DCT	100	0	19593
	Correlation Coefficient	100	0	233813
	HD	0	0	1201
news_2	Ours	100	0	16494
	TIRI-DCT	100	6	20233
	Correlation Coefficient	100	0	176342
	HD	0	0	5366
news_3	Ours	100	0	16479
	TIRI-DCT	100	6	20225
	Correlation Coefficient	100	0	176358
	HD	0	0	6427
foreman_2	Ours	83.33	0	16621
	TIRI-DCT	83.33	0	20124
	Correlation Coefficient	83.33	0	181662
	HD	0	0	1419
foreman_3	Ours	83.33	0	16495
	TIRI-DCT	83.33	0	20061
	Correlation Coefficient	83.33	0	178792
	HD	0	0	1232

Wang et al. (2007b) runs quite slow due to large amount of correlation computations. Our algorithm greatly outperforms that algorithm because it does not rely on correlation computations. Our algorithm is also better than the algorithm in Esmaeili et al. (2011) to some degree. The reason is that the medians of coefficients in our algorithm are respectively calculated in vertical direction and in horizontal direction, and thus the complexity is lower because the sequence length is halved in this situation compared with that in Esmaeili et al. (2011). Consider the HD-based algorithm fails in this situation, we do not compare its computation

time with that of other algorithms. Based on the detection performance and computational complexity, our algorithm is superior to the other three algorithms.

Test on MPEG Compressed Videos

Videos in MPEG format are very common in daily life. So the performance of detection algorithms on compressed videos becomes an important issue in practical applications. To evaluate the performance of the proposed detection algorithm, we generate six MPEG compressed videos. For generality, this work adopts the GOP (group of pictures) structure of (15, 3), which is one of the most commonly used settings. By employing the encoding bit rates of 4 Mbps and 3.2 Mbps, we obtain the videos

Table 5. Results about MPEG compressed videos

Video	Algorithm	TPR (%)	FPR (%)	Time (ms)
coastguard_4M	Ours	51.54	0	17185
	TIRI-DCT	50.77	0	20685
	Correlation Coefficient	0	0	238758
	HD	0	0	7753
coastguard_3.2M	Ours	32.31	0	17242
	TIRI-DCT	48.46	0	20592
	Correlation Coefficient	0	0	234187
	HD	0	0	7675
news_4M	Ours	37	2	17467
	TIRI-DCT	0	0	21356
	Correlation Coefficient	0	0	180242
	HD	0	0	7566
news_3.2M	Ours	50	0	17430
	TIRI-DCT	0	0	21340
	Correlation Coefficient	0	0	176358
	HD	0	0	8143
foreman_4M	Ours	71.67	0	17514
	TIRI-DCT	44.17	0	21075
	Correlation Coefficient	0	0	178885
	HD	0	0	6786
foreman_3.2M	Ours	53.33	0	17195
	TIRI-DCT	56.67	0	21013
	Correlation Coefficient	0	0	178841
	HD	0	0	6536

coastguard_4M, *coastguard_3.2M*, *news_4M*, *news_3.2M*, *foreman_4M* and *foreman_3.2M*. The detection results of the four algorithms are given in Table 5.

With respect to FPR, all the four algorithms have good performance in this aspect except a 2% FPR for our algorithm in the case of *news_4M*. On the other hand, the TPR for all of the four algorithms declines seriously. It can be seen that the correlation coefficient-based algorithm in Wang et al. (2007b) and the HD-based algorithm in Lin et al. (2011) can hardly detect any duplication. In contrast, our algorithm and the algorithm in Esmaeili et al. (2011) show a certain degree of robustness. For the encoding bit rates of 4 Mbps, the TPR values for our algorithm are apparently higher than those for the algorithm in Esmaeili et al. (2011). However, with the increase of compression, the latter may outperform our algorithm. The TPR values for the algorithm in Esmaeili et al. (2011) are higher than those for our algorithm in the case of *coastguard_3.2M* and *foreman_3.2M* when the encoding bit rates decline to 3.2 Mbps. It is a little surprising that the TPR values for the algorithm in Esmaeili et al. (2011) are 0. Perhaps it is because *news* contains less dynamic contents and the MPEG compression reduces the features of those contents. From this result, the flexibility of our algorithm to deal with videos with different contents is also exhibited.

As for computation time, our algorithms run faster than the algorithm in Esmaeili et al. (2011). Since both the algorithm in Wang et al. (2007b) and the algorithm in Lin et al. (2011) fail in this situation, we do not discuss their computation time. Obviously, our algorithm still has the best overall performance among the four algorithms.

CONCLUSION

Frame duplication is a low-cost video operation to change the meaning of a video. Such a manipulation can be used innocently and maliciously. When a video is used as evidence in court of law, we have to verify whether this video has undergone any video forgery manipulation including frame duplication. Therefore, the study on this topic is important from the perspective of forensic investigation. Although some attention has been paid to this research area, the gap between the algorithms available in literature and the real-world applications is still large. More efforts need to be made for the design of more efficient and robust algorithms.

In this paper, we have proposed a frame duplication detection algorithm. Our fingerprint is based on the improved features derived from TIRI-DCT. Our major contributions are the construction of new fingerprints of subsequences and the use of a simple metric for video matching. Compared with current related algorithms, the proposed detection algorithm achieves a better tradeoff between computation time and detection capability. It shows stronger robustness against typical video operations like MPEG compression and brightness change. On the other hand, our experiments also exhibit some drawbacks of the proposed algorithm, for example, the 2% FPR in the case of *news_4M*. We think that these drawbacks mainly result from the structure of fingerprints, the metric employed for video matching, and the selection of decision threshold. In our future work, we will focus on investigating these issues.

ACKNOWLEDGMENT

This work was partially supported by the EU FP7 Digital Image and Video Forensics project (Grant Agreement No. 251677, Acronym: DI-VeFor) and the Fundamental Research Funds for the Central Universities, SCUT (Project No. 2012ZM0027).

REFERENCES

Chetty, G., Biswas, M., & Singh, R. (2010). Digital video tamper detection based on multimodal fusion of residue features. In *Proceedings of the 4th International Conference on Network and System Security*, Melbourne, Australia (pp. 606-613).

Coskun, B., Sankur, B., & Memon, N. (2006). Spatio–temporal transform based video hashing. *IEEE Transactions on Multimedia*, *8*(6), 1190–1208. doi:10.1109/TMM.2006.884614

Esmaeili, M., Fatourechi, M., & Ward, R. (2011). A robust and fast video copy detection system using content-based fingerprinting. *IEEE Transactions on Information Forensics and Security*, *6*(1), 213–226. doi:10.1109/TIFS.2010.2097593

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

Kobayashi, M., Okabe, T., & Sato, Y. (2010). Detecting forgery from static-scene video based on inconsistency in noise level functions. *IEEE Transactions on Information Forensics and Security*, *5*(4), 883–892. doi:10.1109/TIFS.2010.2074194

Law-To, J., Chen, L., Joly, A., Laptev, I., Buisson, O., & Gouet-Brunet, V. …Stentiford, F. (2007). Video copy detection: A comparative study. In *Proceedings of the ACM International Conference Image and Video Retrieval* (pp. 371-378).

Lin, G., Chang, J., & Chuang, C. (2011). Detecting frame duplication based on spatial and temporal analyses. In *Proceedings of the 6th International Conference on Computer Science & Education*, Singapore (pp. 1396-1399).

Su, Y., Nie, W., & Zhang, C. (2011). A frame tampering detection algorithm for MPEG videos. In *Proceedings of the 6th IEEE Joint International Information Technology and Artificial Intelligence Conference*, Chongqing, China (pp. 461-464).

Su, Y., Zhang, J., & Liu, J. (2009). Exposing digital video forgery by detecting motion-compensated edge artifact. In *Proceedings of the International Conference on Computational Intelligence and Software Engineering*, Wuhan, China (pp. 1-4).

Wang, W., & Farid, H. (2006). Exposing digital forgeries in video by detecting double MPEG compression. In *Proceedings of the ACM Multimedia and Security Workshop*, Geneva, Switzerland (pp. 37-47).

Wang, W., & Farid, H. (2007a). Exposing digital forgeries in interlaced and deinterlaced video. *IEEE Transactions on Information Forensics and Security*, *2*(3), 438–449. doi:10.1109/TIFS.2007.902661

Wang, W., & Farid, H. (2007b). Exposing digital forgeries in video by detecting duplication. In *Proceedings of the ACM Multimedia and Security Workshop*, Dallas, TX (pp. 35-42).

Wang, W., & Farid, H. (2009). Exposing digital forgeries in video by detecting double quantization. In *Proceedings of the ACM Multimedia and Security Workshop* (pp. 39-48).

Zhang, J., Su, Y., & Zhang, M. (2009). Exposing digital video forgery by ghost shadow artifact. In *Proceedings of the First ACM Workshop on Multimedia in Forensics*, Beijing, China (pp. 49-54).