Chapter 7
Video Copy-Move Forgery Detection and Localization Based on Structural Similarity

Fugui Li and Tianqiang Huang

Abstract Copy-move forgery is one of the most common types of video forgeries. To detect such forgery, a new algorithm based on structural similarity is proposed. In this algorithm, we extend structural similarity to measure the similarity between two frames of a video. Since the value of similarity between duplicated frames is higher than that between the normal inter-frames, a temporal similarity measurement strategy between short sub-sequences is put forward to detect copy-move forgery. In addition, we can obtain an accurate forgery localization. Extensive experimental results evaluated on 15 videos captured by the digital camera and mobile camera in stationary and moving mode show that the precision of this algorithm can reach 99.7 % which is higher than a previous relevant study.

Keywords Video forgery · Copy-move detection · Copy-move localization · Structural similarity

7.1 Introduction

With the wide use of a variety of digital multimedia devices as well as the development of powerful video editing tools (such as Adobe Premiere Pro and Adobe After Effects, etc.), it is becoming easy for common users to edit and process videos without leaving any visual clues. When a large number of edited and forged videos appear on the video sharing sites, the news, scientific discovery

F. Li (✉) · T. Huang
School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China
e-mail: leaf304@163.com

T. Huang
e-mail: fjhtq@fjnu.edu.cn

A. A. Farag et al. (eds.), *Proceedings of the 3rd International Conference on Multimedia Technology (ICMT 2013)*, Lecture Notes in Electrical Engineering 278,
DOI: 10.1007/978-3-642-41407-7_7,

and court exhibits, there is no doubt that they will have a significant adverse effects on the stability of society and the state. Therefore, digital video forensics has become a very important research issue [1].

Video forensics can be classified into two different categories: active forensics and passive forensics. For active forensics, some pre-embedded specific information which could not be perceived in the video is needed, such as digital watermark and digital signature. In this case, one can determine whether the video is tampered or not by detecting the integrity of the information. While there is no requirement on specific information for passive forensics just by analyzing some inherent properties of videos. Recently, more attention was drawn to passive forensics. For an MPEG video, it is usually resaved in MPEG format after tampering operations. In the literature, there are already different kinds of methods for detecting video forgeries in MPEG format. In [2, 3], the authors proposed methods to detect video forgeries based on double compression and double quantization. The authors of [4] proposed a feature curve to reveal the compression history of an MPEG video file with a given GOP structure, and used the temporal patterns of block artifacts as evidence to detect tampering, Su et al. [5] utilized the motion-compensated edge artifacts (MCEA) for detecting of video forgery with the type of frame-deletion. Meanwhile, Dong et al. [6] exploited the MCEA difference between adjacent P frames, and judged whether there are any spikes in the Fourier transform domain after double MPEG compression to detect video forgery. A scheme of tampering detection using statistics of motion vectors produced by inter-frame prediction was proposed in [7]. Huang et al. [8] employed the contents continuity between frames and bidirectional motion vectors for the frame deletion and insertion tampering. These detection methods are based on analyzing coding theory of the MPEG format video. However, for the unity of the video format, these detection methods are very limited. Recently, antiforensic techniques have also been reported in [9, 10] against some of the existing forensic techniques.

In addition to the type of frame insertion and deletion in video tampering, copy-move tampering is also a common type of video tampering, containing two types: spatial tampering and temporal tampering. In spatial tampering, a region may be pasted to a different location on the same frame or other frames. In this way, the tampering aims to replace or hide the undesired object will be achieved. While in temporal tampering, multiframe is replaced by the copy of previous ones, having the scenes replaced without affecting the continuity of the video, or pasted to a different location having the scenes occurred ahead or delay. Now, different approaches are developed to detect video copy-move forgery, and all of them are based on the same concept that a copy-move forgery brings a correlation between the original frames and the duplicated ones. In allusion to the existing detecting approaches, high calculating complexity and high false alarm rate still exist. Wang et al. [11] used the similarity in the temporal and spatial correlation matrices, embodying the correlations of short sub-sequences, as evidence of detect duplicated frames in a full-length video. In [12], the authors divided the video frames into different areas, by calculating pattern noise and correlation in the temporal adjacent and spatial overlapping blocks, proposed a method to detecting tampered

video with regional copy-move. Meanwhile, Kobayashi et al. [13] proposed an approach to detect suspicious regions in video captured with a static scene by using noise characteristics, but for the regions from the video itself, the algorithm would be constrained, in addition, the noise characteristics may not be estimated correctly under low compression rates. In [14], the authors utilized the Histogram of Oriented Gradients (HOG) feature matching and video compression properties for the detecting of temporal copy-move tampering in videos, but the high dimensional features of HOG lead to a higher complexity of the algorithm. Lin et al. [15] presented a coarse-to-fine approach for detecting frame duplication forgery in the temporal, but many candidates are selected for the videos, which makes the computation time significantly longer in the fine search.

In this paper, a video copy-move forgery detection algorithm based on structural similarity is proposed. In this algorithm, a full-length video sequence is divided into short overlapping sub-sequences, and then the structural similarity is extended to measure the similarity between two frames of a video. Finally, similarities between the sub-sequences in the temporal domain are measured to find out pairs of sub-sequence where replication relationship exists. Moreover, those pairs of sub-sequence are combined into a complete duplicated sequence and the location of the duplicates is located. Such an algorithm allows us to see whether a copy-move attack has occurred or not and furthermore obtain an accurate forgery localization. Extensive experimental results evaluated on 15 videos captured by the digital camera and mobile camera in stationary and moving mode show that the precision of this algorithm can reach 99.7 % which is higher than a previous relevant study.

The rest of the paper is organized as follows. Section 7.2 gives a brief introduction to the structural similarity. Section 7.3 shows the details of the proposed detection method. Experimental results and analysis are presented in Sect. 7.4. The conclusions are finally drawn in Sect. 7.5.

7.2 Structural Similarity

Considering the perceptual features of the human visual system, Wang et al. [16] introduced a structural similarity (SSIM)-based quality metric. The SSIM metric measures the similarity with three statistical components, which are luminance comparison, contrast comparison, and structural comparison. Let Y be the distorted image of X, for any two pixels $x \in X$ and $y \in Y$, the SSIM metric is as follows,

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{7.1}$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{7.2}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3} \tag{7.3}$$

Combining the three comparison functions of Eqs. (7.1)–(7.3) produces a general form of the SSIM index:

$$\mathrm{SSIM}(x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma} \tag{7.4}$$

Here, parameters α, β and γ adjust the relative importance of three components. Usually, $\alpha = \beta = \gamma = 1$, $C_3 = C_2/2$, producing a specific form of the SSIM index:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{7.5}$$

where μ_x and μ_y are the means of the local windows, which are with a size of 11×11, centered at x and y, respectively, σ_x and σ_y are the standard variance, σ_{xy} is the covariance of the two windows, C_1, C_2, C_3 are small constants to make sure the denominator not being zero.

Then the mean SSIM (MSSIM) is used to evaluate the overall image quality,

$$\mathrm{MSSIM}(x,y) = \frac{1}{M}\sum_{i=1}^{M} \mathrm{SSIM}(x_i, y_i) \tag{7.6}$$

where, M is the number of local windows in the image, the higher the MSSIM is, the better quality of the distorted image will be.

7.3 Proposed Method

In temporal tampering, multiframe is replaced by a copy of previous ones, having the scenes replaced without affecting the continuity of the video, or pasted to a different location having the scenes occurred ahead or delay. In this section, the proposed method for detecting copy-move forgery and locating the duplicated frames is presented in detail. Therefore, our method includes three parts: (1) inter-frame similarity measurement; (2) forgery detection; and (3) forgery localization.

7.3.1 Inter-Frame Similarity Measurement

As described in Sect. 7.2, structural similarity can be used to evaluate image quality, the higher the MSSIM is, the better the quality of the distorted image will be, that is, the image X is more similar to image Y. For a video sequence, it is just a successive images in the temporal domain. Thus, we extend structural similarity to measure the similarity between two images. If a video has been tampered by

copy-move in the temporal domain, duplicated frames will exist in it which makes the value of similarity between duplicated frames higher than that between the normal inter-frames. Here, we definite a threshold to judge whether a video has duplicated frames. When the MSSIM is higher than threshold $\tau = 0.994$, we consider the two images have the relationship of replication, where τ is experiment threshold.

Figure 7.1 shows the procedure of similarity measure between two images. For any of the two images I_1 and I_2 in the video sequence, first, we convert images from color to grayscale for reducing computation time, and the luminance information of gray-scale images are extracted. Then we remove the luminance information of the images to calculate the contrast information. Finally, we divide by the contrast information to calculate structural information. As described in Sect. 7.2, the MSSIM will be obtained to measure the similarity between two images.

For all video sequence with different content and captured by different equipment, it is difficult or impossible to obtain an ideal threshold τ. Therefore, through measuring the similarities between adjacent frames of the normal videos, we can estimate the maximum possible value of similarity in the video sequence.

Generally, video frame rates vary with different capturing equipment. For instance, videos taken by mobile cameras are usually at 15 or 20 fps, and those taken by digital cameras are at 24 or 30 fps. The higher the video frame rate, the more the frames will be per second, thus it will make the higher similarities between adjacent frames. Figure 7.2 shows the similarities between adjacent frames of the normal videos. In Fig. 7.2a, videos are taken by mobile camera at 15 fps and digital camera at 30 fps in the same scene with stationary cameras. For the difference of frame rate, we can see that the similarities between adjacent frames in digital camera are relatively flat and the values are nearly to 1, and the maximum value is 0.9931. It means that if the frame rate is higher, the similarities will also be higher. Similarly, In Fig. 7.2b, videos are taken by digital camera at

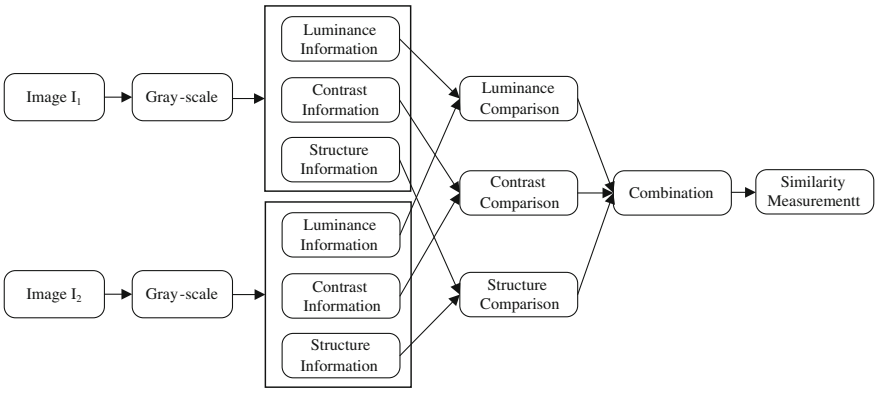

Fig. 7.1 Similarity measurement between two images

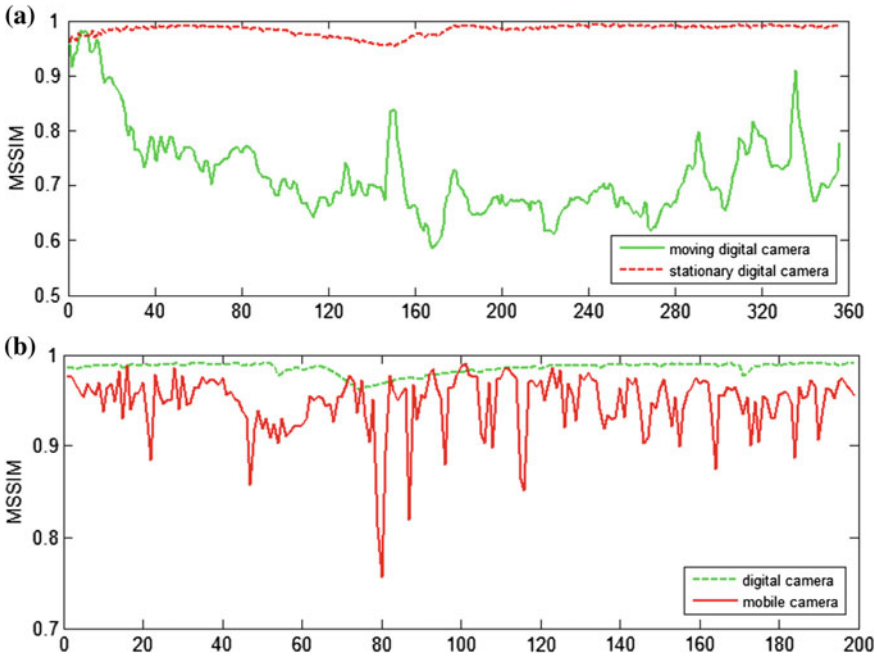

Fig. 7.2 Similarities between adjacent frames of normal videos. **a** similarities between adjacent frames in videos taken by different cameras, **b** similarities between adjacent frames in videos taken by digital camera

30 fps in the same scene with stationary camera and moving camera. For the reason that the content are changing slowly in video taken by stationary camera, the similarities between adjacent frames are higher and the maximum value is 0.9934.

For a copy-move tampered video, it will make similarities between duplicated frames higher than that between normal frames. Meanwhile, we divide the video sequences into short overlapping sub-sequences to complete the temporal similarity measurement. Based on the previous analysis, we set threshold $\tau = 0.994$, which can be used to distinguish normal frames and duplicated frames.

7.3.2 Forgery Detection

For a copy-move tampered video in the temporal domain, to achieve better tamper results, it will be usually duplicated with consecutive frames, rather than a single frame. Therefore, we divide the video sequences into short overlapping subsequences to complete the temporal similarity measurement.

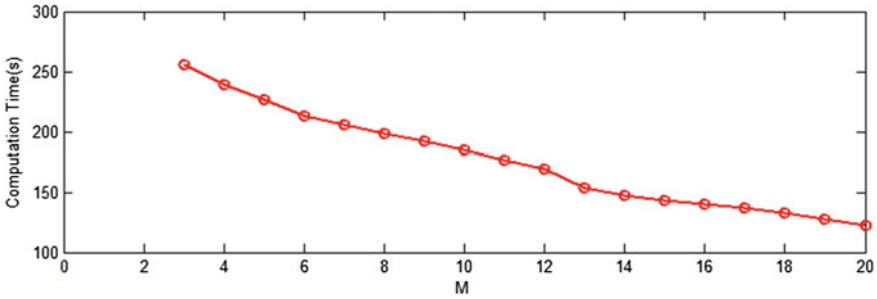

Fig. 7.3 The computation time of a test video with different length of sub-sequence

7.3.2.1 Sub-sequence Partition

A video sequence consists of many continuous images, which can be expressed as $F = I(x, y, t), x \in [0, W-1], y \in [0, H-1], t \in [0, N-1]$, where W and H represent the size of a frame, N is the length of video sequence. First, converting each frame from color to grayscale. Then, dividing a full-length grayscale video sequence into short overlapping sub-sequences, we suppose M is the length of sub-sequence.

For the unknown of whether the video has be tampered, as well as the unknown length of the duplicated sequence. Therefore, when the M is larger than the length of duplicated sequence, it will increase false detection rate of the algorithm, while the M is low, it will increase the times of similarity measurement in each sub-sequence and lead to high time complexity of the algorithm. Figure 7.3 illustrations the computation time of a test video with different length of sub-sequence. With the increasing of M, the computation time shows a decreasing trend.

In [11], the authors selected 30 frames as the length of sub-sequence, but for the number of duplicated frames less than 30 frames, the method will be failed. Considering frame rate of the current digital products and that few numbers of tampered frames have few effects on understanding the content of the video, we set $M = 15$ as the length of sub-sequence.

7.3.2.2 Temporal Similarity Measurement Strategy

In this subsection, a temporal similarity measurement strategy between short sub-sequences is put forward. We set $M = 15$ as the length of sub-sequence, by sliding a frame to get a new sub-sequence, therefore, there is $N-1$ sub-sequence in total. Supposing the first sub-sequence is Seq^1, so the last sequence is Seq^{N-1}, where the superscript number of the sub-sequence equals to the first frame's number in each sub-sequence. An example of detailed temporal similarity measurement strategy is described in Fig. 7.4.

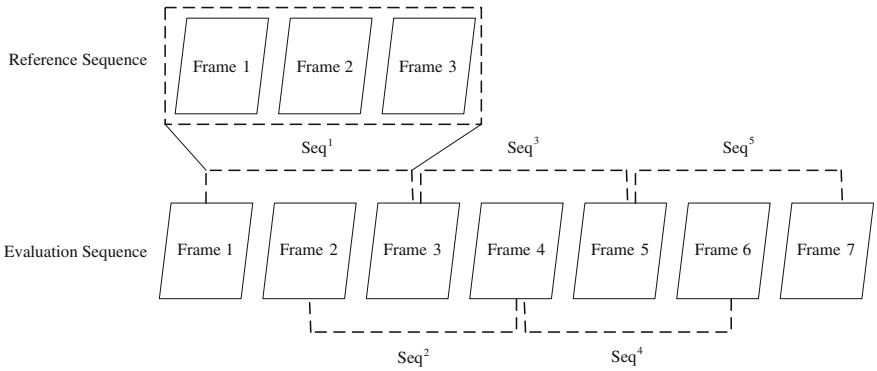

Fig. 7.4 An example of temporal similarity measurement between sub-sequences

Due to the length of sub-sequence is M, we need to measure similarity between each reference sequence and evaluation sequence for M times. If the two sub-sequences exist replication relationship, it means that the value of M times similarity measurement is higher than threshold τ. When the value of the pervious measurement is higher than threshold τ, we only conduct the next measurement, thus it will decrease the times of measurement and improve the efficiency of the algorithm. Conversely, it will keep jumping to next measurement of another two sub-sequences until the measurements of the entire video sequence are completed.

The steps are as follows. First define a value K and initialize it with zero, which is used to record the number of inter-frame similarity value higher than the predefined threshold τ between two sub-sequences. Then, if getting a match, the value will be updated as $K = K+1$. When K equals to M, we consider the current evaluation sequence as a replica of the reference sequence. Finally, we record the superscript number of the two sub-sequences in matrix A. If the matrix A is an empty matrix, we consider that it is a normal video.

7.3.2.3 Merging Duplicated Sub-Sequences

When a duplicated sequence is divided into sub-sequences for detecting copy-move forgery, we need to merge several duplicated sequences to form a complete duplicated sequence. Moreover, we need to remove the false duplicated sequences, therefore, a simple and effective merging strategy is designed.

For a copy-move tampered video, the differences are equal between the superscript of the reference sequence and evaluation sequence. However, due to the influence of adjacent sub-sequences, a reference sequence may be matched with two or more evaluation sequences. Therefore, we select the maximum frequency of distance as copy-move tamper distance to merge these sub-sequences to form a new sequence (see Fig. 7.6).

7.3.3 Forgery Localization

Through the above steps, we can get the merged two copy-move sequences, but we cannot distinguish which sequence is the original sequence, and which sequence is the duplicated sequence. For an original sequence, based on the continuity of the content in a video, both the first and last frame of the sequence are highly similar to the adjacent frames, but the duplicated sequence destroyed the continuity of video, so the value of similarity between them will be relatively low. We suppose the pair of forgery frames are $i-j$ and $i+m-j+m$, by calculating the similarity between the first and last frame of two sequences with the adjacent frame respectively, to distinguish the position of the original sequence and the duplicated sequence. The calculation formulas are as follows:

$$\mathrm{MSSIM}_i = \mathrm{SSIM}(i, i-1) \tag{7.7}$$

$$\mathrm{MSSIM}_j = \mathrm{SSIM}(j, j+1) \tag{7.8}$$

$$\mathrm{MSSIM}_{i+m} = \mathrm{SSIM}(i+m, i+m-1) \tag{7.9}$$

$$\mathrm{MSSIM}_{j+m} = \mathrm{SSIM}(j+m, j+m+1) \tag{7.10}$$

If $\mathrm{MSSIM}_i + \mathrm{MSSIM}_j > \mathrm{MSSIM}_{i+m} + \mathrm{MSSIM}_{j+m}$, we think that the original sequence is $i-j$ and the duplicated sequence is $i+m-j+m$, otherwise, the original sequence is $i+m-j+m$ and the duplicated sequence is $i-j$, thus we have located the location of duplicated sequence.

7.4 Experimental Results and Analysis

In our experiment, we selected 15 test videos captured by the digital camera and mobile camera in stationary and moving mode. Each frame is 640 × 480 pixels in size, and the frame rate is 30 and 15 fps. To evaluate the performance of the proposed algorithm, we created duplicated frames with different lengths ranging from 35 to 200, and used MPEG-VCR and Adobe Premiere Pro CS4 to tamper the videos. Table 7.1 shows the details of 15 test videos. The computer used in experiments is configured as 3.06 GHz Intel processor and the operating environment is MATLAB R2010b.

A simple example of video copy-move tamper is shown in Fig. 7.5, the video sequences in the first row are captured normally, but the video sequences in the second row are tampered with the type of copy-move forgery. As the example shows, the location of video frame 4 and 5 are replaced by frame 1 and 2, which make the car disappeared without leaving any visual clues.

Table 7.2 shows the detection results of the proposed method for the 15 test videos in the experiments. Obviously, the proposed method is able to detect temporal copy-move tampering correctly and locate the location of the duplicates

Table 7.1 Test videos

Test videos	Equipment	Length	Resolution	Tamper location
Video 1	Digital camera	679	640×480	No
Video 2		374		66–150 are copied to 251–335
Video 3		535		101–200 are copied to 301–400
Video 4		247		57–111 are copied to 170–224
Video 5		222		31–80 are copied to 111–160
Video 6		791		121–320 are copied to 521–720
Video 7		500		136–235 are copied to 357–456
Video 8		320		86–125 are copied to 233–272
Video 9		504		101–194 are copied to 348–441
Video 10	Mobile camera	169		31–70 are copied to 121–160
Video 11		291		No
Video 12		318		71–140 are copied to 201–270
Video 13		362		47–100 are copied to 269–322
Video 14		348		51–150 are copied to 231–330
Video 15		264		21–55 are copied to 96–130

Fig. 7.5 A simple example of a forged video sequence

exactly, and the results of videos 10–15 have exemplified it which are taken by mobile camera with the frame rate 15 fps. For video 4 and video 7, there are relatively a few miss-detected replicas. The main reason is that the first and last frame of duplicated sequences will be affected by the adjacent frames. For video 7 and video 9, the scenes are changing slowly which are taken by the stationary digital camera, therefore, the number of duplicated pairs is higher than in ideal state. Therefore, it is necessary to remove the miss-detected duplicated pairs, as described in Sect. 7.3.2.3. Compared with the method in [15], the computation time is shorter which will be more acceptable. With the increasing length of the video sequence, the computation time will be longer. For the reason that the longer the video sequence is, the more the measurement times will be needed, this is exemplified by the results for video 1, video 3, and video 6 in Table 7.2.

Figure 7.6a shows the number of duplicated pairs between reference sequence and evaluation sequence in the experiments. Generally, we should get a continuous values of duplicated pairs between reference sequence and evaluation sequence,

Table 7.2 Detection results

Test videos	Results	No. of duplicated pairs	Computation time (s/frame)
Video 1	Normal video		11.303
Video 2	Originals: 66–150 Duplicates: 251–335	87	5.921
Video 3	Originals: 101–200 Duplicates: 301–400	86	8.824
Video 4	Originals: 57–112 Duplicates: 170–226	68	3.682
Video 5	Originals: 31–80 Duplicates: 111–160	66	3.209
Video 6	Originals: 121–320 Duplicates: 521–720	186	13.464
Video 7	Originals: 136–236 Duplicates: 357–457	144	8.141
Video 8	Originals: 86–125 Duplicates: 233–272	44	4.876
Video 9	Originals: 101–194 Duplicates: 348–441	134	8.289
Video 10	Originals: 31–70 Duplicates: 121–160	26	2.318
Video 11	Normal video		4.384
Video 12	Originals: 71–140 Duplicates: 201–270	56	5.455
Video 13	Originals: 47–100 Duplicates: 269–322	40	5.676
Video 14	Originals: 51–150 Duplicates: 231–330	86	5.487
Video 15	Originals: 21–55 Duplicates: 96–130	21	3.948

which can be fitted by a straight line, just like the video 3. But the content of the videos which captured by stationary camera will be changed slowly, and even have the situation of some frames with the same content, thus making some original frames will have a higher value of similarities in the video. For video 9, it appears a reference sequence may be matched with two or more evaluation sequences. According to the merging strategy described in Sect. 7.3.2.3, we can use it to remove the false pairs, as the Fig. 7.6b shows.

To evaluate the proposed method, we utilize the precision and recall which can be expressed as:

$$\text{Precision} = N_c/(N_c + N_f) \tag{7.11}$$

$$\text{Recall} = N_c/(N_c + N_m) \tag{7.12}$$

where N_c is the numbers of correct detections, N_f is the numbers of false alarms, N_m is the numbers of missed detections.

We give a comparison of our results against the results reported in [15] for frame duplication. As shown in Table 7.3, the precision of our algorithm can reach 99.7 % which is higher than the method in [15], the main reason is that we extend the structural similarity as a feature and measurement tool in the temporal similarity measurement strategy. From the procedure of structural similarity, the inter-frame similarity measurement actually contains the spatial similarity measurement between two frames. In [15], although the duplicated frames which are tampered in the videos can be found by the coarse-to-fine search strategy, due to the low accuracy of the search, more frames will be miss-detected as duplicated frames. Meanwhile, this strategy will also make the computation time more complex. In

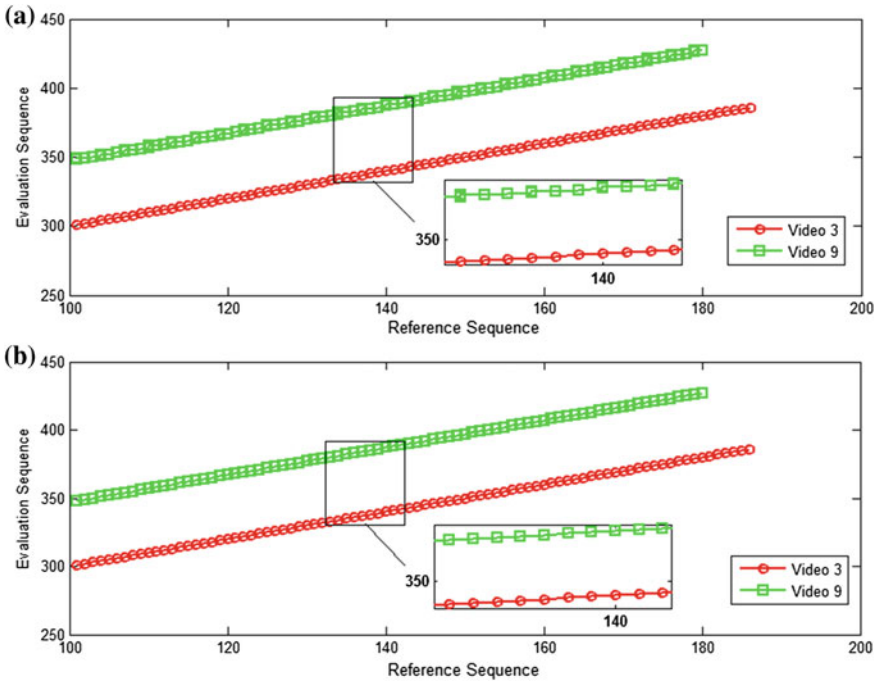

Fig. 7.6 Illustrations of pairs of duplicates. **a** Original pairs of duplicates, **b** pairs of duplicates after merging strategy

Table 7.3 Comparison with Lin et al.'s [15] method

Method	Precision	Recall	Location of duplicates
Ref. [15]	0.849	1	No
Proposed	0.997	1	Yes

addition, for different kinds of test videos, the proposed method can not only detect the duplicates but also locate the location precisely. Therefore, we can exclude the interference of the duplicated frames and understand the content of the videos correctly. This means that the proposed method shows a better performance than the method in [15].

7.5 Conclusions

In this paper, we have proposed a new algorithm for detecting copy-move tampered digital videos based on structural similarity. In this algorithm, a full-length video sequence is divided into some short overlapping sub-sequences, and then the

structural similarity is extended to measure the similarity between two frames of a video. Finally, similarities between the sub-sequences in the temporal domain are measured to find out pairs of sub-sequence where replication relationship exists. Moreover, those pairs of sub-sequence are combined into a complete duplicated sequence and the location of the replica is located.

Extensive experimental results evaluated on 15 videos captured by the digital camera and mobile camera in stationary and moving mode show that the precision of this algorithm can reach 99.7 % which is higher than a previous relevant study. The algorithm is able to detect and locate the location of the duplicates correctly. But one limitation is that it have a few duplicates miss-detected for the videos have a long time still scenes. Future work will be mainly dedicated to investigating how to reduce the computation time. In particular, integration with other forensics techniques applied onto video copy-move forgery is envisaged.

Acknowledgments This work was supported by the National Natural Science Foundation of China (Grant No. 61070062), Industry-university Cooperation Major Projects in Fujian Province (Grant No. 2012H6006), Program for New Century Excellent Talents in University in Fujian Province(Grant No. JAI1038).

References

1. Milani S, Fontani M, Bestagini P et al (2012) An overview on video forensics. APSIPA Trans Signal Inf Process 1:e2. doi:10.1017/ATSIP.2012.2
2. Wang W, Farid H (2006) Exposing digital forgeries in video by detecting double MPEG compression. In: Proceedings of the 8th workshop on multimedia and security. doi: 10.1145/1161366.1161375
3. 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. doi: 10.1145/1597817.1597826
4. Luo W, Wu M, Huang J (2008) MPEG recompression detection based on block artifacts. In: Proceedings of the SPIE on security, forensics, steganography and watermarking of multimedia imaging. doi:10.1117/12.767112
5. Su Y, Zhang J, Liu J (2009) Exposing digital video forgery by detecting motion-compensated edge artifact. In: Proceedings of international conference on computational intelligence and software engineering. doi: 10.1109/CISE.2009.5366884
6. Dong Q, Yang G, Zhu N (2012) A MCEA based passive forensics scheme for detecting frame-based video tampering. Digit Invest 9(2):151–159
7. Qin Y, Sun G, Zhang X (2009) Exposing digital forgeries in video via motion vectors. J Comput Res Dev. 46(Suppl.):227–233 (in Chinese)
8. Huang T, Chen Z (2011) Digital video forgeries detection based on bidirectional motion vectors. J Shandong Univ (Engineering Science) 41(4):13–19 (in Chinese)
9. Stamm MC, Liu KJR (2011). Anti-forensics for frame deletion/addition in MPEG video. In: Proceedings of 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP). doi: 10.1109/ICASSP.2017.5946872
10. Stamm MC, Lin WS, Liu KJR (2012) Temporal forensics and anti-Forensics for motion compensated video. IEEE Trans Inf Forensics Secur 7(4):1315–1329
11. Weihong W, Hany F (2007) Exposing digital forgeries in video by detecting duplication. In: Proceedings of the 9th workshop on multimedia and security. doi: 10.1145/1288869.1288876

12. Chih-Chung H, Tzu-Yi H, Lin C-W, Chiou-Ting H (2008) Video forgery detection using correlation of noise residue. In: Proceedings of 2008 IEEE 10th workshop on multimedia signal processing. doi: 10.1109/MMSP.2008.4665069
13. Kobayashi M, Okabe T, Sato Y (2010) Detecting forgery from static-scene video based on inconsistency in noise level functions. IEEE Trans Inf Forensics Secur 5(4):883–892
14. Subramanyam AV, Emmanuel S (2012) Video forgery detection using HOG features and compression properties. In: Proceedings of 2012 IEEE 14th international workshop on multimedia signal processing (MMSP). doi: 10.1109/MMSP.2012.6343421
15. Lin G-S, Chang J-F (2012) Detection of frame duplication forgery in videos based on spatial and temporal analysis. Int J Pattern Recognit Artif Intell 26(7):1–18
16. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612