International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 26, No. 7 (2012) 1250017 (18 pages)

DOI: 10.1142/S0218001412500176

World Scientific
www.worldscientific.com

DETECTION OF FRAME DUPLICATION FORGERY IN VIDEOS BASED ON SPATIAL AND TEMPORAL ANALYSIS

GUO-SHIANG LIN* and JIE-FAN CHANG‡

Department of Computer Science and Information Engineering
Da-Yeh University, No. 168, University Rd.
Dacun, Changhua County 51591, Taiwan (R.O.C)
**khlin@mail.dyu.edu.tw*
‡*shiunyi71@gmail.com*

Received 17 April 2012
Accepted 23 October 2012
Published 9 January 2013

In this paper, we present a passive-blind scheme for detection of frame duplication forgery in videos. The scheme is a coarse-to-fine approach that is implemented in four stages: candidate segment selection, spatial similarity measurement, frame duplication classification, and post-processing. To screen and select duplicated candidates in the temporal domain, the histogram difference of two adjacent frames in the RGB color space is adopted as a feature. Then, to evaluate the similarity of two images, we use a block-based algorithm to measure the spatial correlation between the candidate segment and the corresponding frame in the query template. Based on the results of spatial and temporal analysis, we construct a classifier to detect duplicated clips. In addition, to deal with the partial detection problem, we develop a post-processing technique that examines and merges two adjacent detected candidates into a complete duplicated video clip. Our experiment results demonstrate that the proposed scheme can not only achieve detection of frame duplication forgery but also accurately detect and localize duplicated clips in different kinds of videos. The results also show that the scheme outperforms an existing method in terms of precision, recall, accuracy, and computation time.

Keywords: Video forensics; forgery detection; frame duplication.

1. Introduction

With the recent advances in low-cost multimedia devices and digital communication technologies, vast amounts of video content can now be captured, stored, and transmitted via the Internet every day. Moreover, increasingly sophisticated image/video editing tools have made it easier to copy and manipulate the content of digital videos. One way is to duplicate a sub-sequence in a video to conceal or imitate a specific event in the same video. For instance, if a car accident is recorded by a car

*Corresponding author.

camcorder, the portion of the video showing the car accident could be removed by copying and pasting another sub-sequence over it. Detecting this kind of video forgery is difficult if the copy/paste procedure is performed carefully. As a result, the field of video forensics is becoming increasingly important.

Digital watermarking technology[11,14,15] is used to protect the copyright and authenticate the content of digital media data. The technology can be thought of an active image/video forensics approach because a specific signal (i.e. a watermark) is hidden in an image/video.[13] For instance, the compressed-domain scheme in Ref. 11 provides dual protection of JPEG images based on informed embedding and a two-stage watermark extraction technique. The embedded watermark, which does not affect the visual quality of the image, can be used to verify the image's content.

However, the limitation of digital watermarking technology is that a specific signal (i.e. watermark) must be embedded in the media data. In fact, most media data, especially, videos, do not contain any digital watermarks due to lack of watermarking equipment. These reasons motive us to develop a non-watermarking (i.e. passive) scheme that does not rely on embedded information for digital video forensics.

Lin *et al.*[13] observed that several kinds of passive schemes are based on certain characteristics of digital signal processing and recording devices, such as color filter array, sensor pattern noise, JPEG quantization and blocking artifacts, and quality modification. A number of passive methods[1,2,8,9,13,17] have been proposed for image forensics. In Ref. 13, the authors introduced a forgery detection method for images that have been JPEG compressed. The method first classifies each AC DCT coefficient into a specific type; then, the corresponding quantization step size is measured adaptively from its energy density spectrum (EDS) and the EDS's Fourier transform. The inconsistency of the quantization table is used to detect tampered regions. In Ref. 1, Chen *et al.* proposed a method that performs periodicity analysis of JPEG compression artifacts in the spatial and frequency domains to detect tampering. The periodicity characteristic of JPEG compression appears in the spatial domain due to blocking artifacts resulting from block-based transformation; while in the frequency domain, the characteristic results from the quantization procedure of DCT coefficients. Tampered regions can be detected based on the inconsistency of the periodicity characteristics in the spatial and frequency domains. In Ref. 8, Kakar *et al.* proposed a forgery detection method based on the analysis of motion blur. The rationale behind the method is that it is impossible for the information about motion blur in a spliced object to be completely consistent with the blur in the rest of the image. Thus, spliced regions can be detected by the inconsistency in motion blur.

Although a number of image forensics methods have been proposed, relatively few methods[6,10,19–22] have been developed for video forensics. In Ref. 22, Wang and Farid proposed two techniques for measuring traces resulting from tampering in de-interlaced and interlaced videos. For a de-interlaced video, the correlations derived by de-interlacing algorithms are analyzed and measured to detect tampering; while

for an interlaced video, the motion information between the fields of a single frame and the motion across the fields of neighboring frames is used to identify tampered regions. Hsu *et al.* 6, introduced a video forensics technique that explores the characteristics of sensor pattern noise to detect forgeries. In Ref. 10, Kobayashi *et al.* also analyze a noise characteristics model of an image and then estimate the noise level function. Inconsistencies in the model indicate possible tampered regions in the image.

In one kind of video forgery called frame duplication mentioned in Ref. 21, some frames selected from a video are duplicated to extend or replace a specific object/event in the same video. It is fairly easy to duplicate frames with video processing tools.[a] However, any modifications made to a video should not be detectable by the naked eye. Relatively few works[21] have addressed the frame duplication problem. In Ref. 21, the correlation of the Y component (i.e. luminance) between two adjacent frames in a video is calculated for finding possible duplicate sub-sequences. The further process is that each frame is divided into several blocks to extract the spatial information. Then, the spatial similarity of each frame between two sub-sequences is computed to determine their correlation. If the correlation is high, frame duplication forgery exists. However, efficiently detecting and localizing duplicates in videos is difficult because of the restricted portability and accessibility of original videos. Hence, blind detection of frame duplication forgery is more attractive and feasible in many cases. These reasons motivate us to develop a passive-blind scheme for detection of frame duplication forgery, in which detecting and localizing intended replicates without prior information about the original video can be achieved.

The remainder of this paper is organized as follows. In Sec. 2, we introduce the system model; and in Sec. 3, we describe the proposed frame duplication detection scheme. Section 4 details the results of experiments conducted to evaluate the scheme and Sec. 5 contains some concluding remarks.

2. System Description

If a video frame has been duplicated, the processed video will contain one or more duplicated segments. Since there is a high correlation between the duplicated segments of a manipulated video, we can measure the similarity of two segments in the spatial and temporal domains to determine whether they are duplicates. This means that determining whether a video segment has been manipulated becomes a binary classification problem. A generic paradigm for detection of frame duplication forgery can be formulated as follows. Given a video template, frame duplication is confirmed if the similarity between the template and the detected sub-sequence is higher than a threshold in a video. Therefore, to achieve detection of frame duplication forgery, a video segment randomly selected from a test video can be regarded as a query template and used to determine whether there are duplicates of it in the video.

[a] Avidemux 2.5, http://avidemux.berlios.de/index.html.

Let an original video sequence be $\boldsymbol{I^O} = \{I(x, y, t)|x \in [0, N_W - 1], y \in [0, N_H - 1], t \in [0, N_O - 1]\}$, where N_W and N_H represent the dimensions of a frame; and N_O denotes the length of the video sequence. A fake video sequence $\boldsymbol{I^F}$ with the same frame dimensions as $\boldsymbol{I^O}$ and whose length is N_F is created when an original frame is duplicated. For a query template $\boldsymbol{S^q} \in \boldsymbol{I^F}$ whose length is N_q, if there is a replica $\boldsymbol{S^r}$, which is a copy of $\boldsymbol{S^q}$, in $\boldsymbol{I^F}$, the search space for detecting a duplicate of $\boldsymbol{S^q}$ is theoretically $(N_F - N_q - 1)!$. For $N_F = 128$ and $N_q = 7$, the size of the search space is more than $8.1\,\mathrm{E} + 200$. When N_F is large, an exhaustive search method will probably examine a large number of possible solutions to find out the location of frame duplication. Apart from the size of the search space, the similarity measurement between the query template and a test video segment also affects the total computational cost. This is because the longer the query template sequence and the larger the amount of spatial information, the higher will be the computation cost of similarity measurement. In addition, since $\boldsymbol{I^O}$ is often processed and re-compressed to derive $\boldsymbol{I^F}$, $\boldsymbol{S^r}$ may not be completely equal to $\boldsymbol{S^q}$. Therefore, it is necessary to devise an effective scheme for detecting duplicated frames.

To find duplicated segments, we utilize a coarse-to-fine approach because it is efficient; however, we must still determine how to measure the similarity between $\boldsymbol{S^q}$ and $\boldsymbol{S^r}$. Similar to Ref. 16, we extract a set of features from $\boldsymbol{S^q}$ and another set from $\boldsymbol{S^r}$. Then, the similarity between $\boldsymbol{S^q}$ and $\boldsymbol{S^r}$ is measured by computing the distance between their features. As shown by the block diagram in Fig. 1, the proposed scheme is implemented in four phases: candidate segment selection, spatial similarity measurement, frame duplication classification, and post-processing.

The objective of the first phase, candidate segmentation selection, is to reduce the size of the search space and thereby ensure that the scheme is very efficient. Initially, we conduct a coarse search to find some candidates resulting from frame duplication, i.e. the coarse search identifies a set of similar video segments. In this step, a global feature extracted from a short video sequence can be used. However, although the global feature is compact and can be extracted easily, it may not work well in some situations, such as when the lighting changes and in complex scenes. In addition, because of information loss (i.e. video content), different video clips with a similar global feature may be identified during the coarse search. To resolve this problem, a local feature is extracted from frames and used in a fine search to filter out possible video segments that were identified incorrectly by the coarse search. Finally, after the coarse-to-fine search, a classifier is used to determine whether each detected

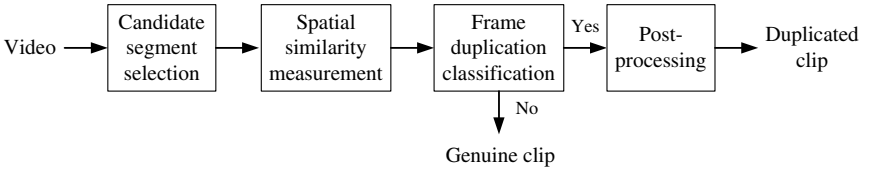

Fig. 1. Block diagram of the proposed frame duplication detection scheme.

candidate is actually a duplicate. However, after frame duplication, an image sequence is often re-compressed to generate a fake version. The re-compression process causes distortions between $\boldsymbol{I^O}$ and $\boldsymbol{I^F}$ such that only part of a replica may be detected. Thus, in the post-processing phase, consecutive partial replicas are combined to detect the location of a complete duplicate.

To summarize, the proposed scheme conducts a coarse search to find a small number of potentially similar video candidates, and then applies a matching algorithm to refine the candidates in the fine search. The coarse-to-fine search approach improves the detection performance and reduces the overall computational cost. A classifier is exploited to examine whether each detected candidate is actually a duplicate and a post-processing procedure is devised to detect the location of a complete duplicate.

3. The Proposed Frame Duplication Detection Scheme

Since a video has both spatial and temporal dimensions, the similarity between duplicates should be high in the spatial and temporal domains. Therefore, the video content in the spatial and temporal domains can be analyzed and utilized to develop the proposed frame duplication detection scheme. In the following subsections we describe the phases shown in Fig. 1.

3.1. *Candidate segment selection*

Duplicated segments may occur in any part of a video, which means that each subsequence may be a duplicate. As mentioned in Sec. 2, the size of the search space for detecting $\boldsymbol{S^r}$ is very large; therefore, in this phase, the objective is to choose a small number of candidate video segments from a test video.

Recall that the similarity between $\boldsymbol{S^q}$ and $\boldsymbol{S^r}$ is measured by the distance between their features. To measure the similarity efficiently, only temporal information is adopted as a global feature to represent a video in this phase. Color is a low-level perceptive feature that is used extensively in content-based image/video retrieval tasks. In Ref. 21, only the luminance component (Y component) of each two adjacent frames in a video is measured and used to find duplicate candidates. In addition to the advantages of the color histogram, such as scale invariance and a high degree of immunity to noise, the histogram difference of a whole frame can be calculated easily and it is less sensitive to zooming and camera motion.[3,5,12] Therefore, in contrast to the approach in Ref. 21, we utilize the color histogram difference (CHD) in this paper.

For the ith template S^{q_i}, the CHD of two adjacent frames forms a unit that is used to create a global feature to represent S^{q_i}. Since the global feature is measured in the time domain, it is also a temporal feature. In addition, similar to Ref. 21, we reduce the computational complexity of the proposed scheme by shrinking the frame size. Specifically, our scheme down-samples the frame size of a test video by a factor of eight.

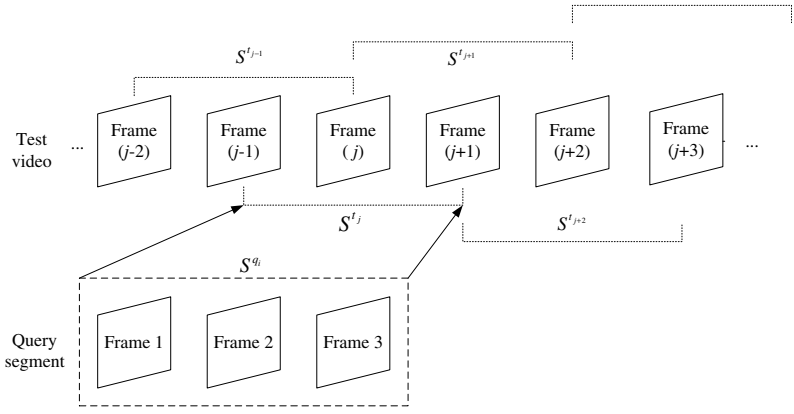

Fig. 2. An illustration of candidate segment selection.

Figure 2 illustrates the candidate segment selection procedure. The query segment S^{q_i} has three frames and the test video is divided into some overlapped segments, e.g. S^{t_j} and $S^{t_{j+1}}$. The similarity between S^{q_i} and each segment (i.e. S^{t_j}) in the test video is calculated and then the segments with the higher value of similarity are chosen for further analysis. According to the above mention, the steps are as follows.

T1. Slide the query template S^{q_i} frame by frame from the beginning to the end of the test video.

T2. Calculate the CHD of two adjacent frames in S^{q_i} as $\bar{h}^{q_i}$, which is expressed as follows:

$$h_j^{q_i}(i) = \sum_{x,y} \delta\left(i - \left\lfloor \frac{I_j(x,y)}{\Delta} \right\rfloor\right), \tag{1}$$

$$\Delta = \frac{\min(I_j(x,y), 255) - \max(I_j(x,y), 0)}{N_{\text{bin}}}, \tag{2}$$

$$\bar{h}^{q_i}(m,n) = \frac{1}{N_{\text{bin}}} \sum_{j \in \{R,G,B\}} |h_j^{q_i}(m) - h_j^{q_i}(n)|, \quad 0 \le m,n < N_q, \tag{3}$$

where $I_j(x,y)$, $j \in \{R,G,B\}$, denotes the pixel value at the (x,y) coordinate; $h_j^{q_i}$, $j \in \{R,G,B\}$, represents the histograms of the query template in the RGB color space; $\max\{\}$ and $\min\{\}$ denote the maximum operator and the minimum operator, respectively; $\{R,G,B\}$ denotes the three color components in the RGB color space; $\bar{h}^{q_i}(m,n)$ denotes the CHD between the mth and nth frames in S^{q_i}; $\lfloor \cdot \rfloor$ represents a floor operator; $|\cdot|$ stands for the absolute operator; $\delta(\cdot)$ denotes the impulse function ($\delta(x) = 1$ for $x = 0$ and $\delta(x) = 0$ for $x \neq 0$); and N_{bin} denotes the number of bins in the color histogram. Here, N_{bin} is 16 for each component in the RGB color space. Then, the CHD of S^{q_i} is calculated by

$$\boldsymbol{H}^{q_i} = \{\bar{h}^{q_i}(m,n), 0 \le m,n < N_q, m \ge n\}. \tag{4}$$

Similarly, h^{t_j}, $i \in \{R, G, B\}$, represents the histograms of the jth test segment S^{t_j} in the RGB color space, and the CHD of S^{t_j} can be derived as $\boldsymbol{H}^{t_j}$. Similar to Ref. 21, to distinguish a stationary scene from a duplicate, we adopt the following rule: S^{q_i} is classified as a stationary scene if $\max\{\boldsymbol{H}^{q_i}\} < Thd$ (0.02). This means that all pairs of frames in S^{q_i} are nearly identical; hence, the scene is static.

T3. Calculate the correlation coefficient $C(\boldsymbol{H}^{q_i}, \boldsymbol{H}^{t_j})$ between $\boldsymbol{H}^{q_i}$ and $\boldsymbol{H}^{t_j}$ as follows:

$$C(\boldsymbol{H}^{q_i}, \boldsymbol{H}^{t_j}) = \frac{\sum(\boldsymbol{H}^{q_i} - \mu^{q_i})(\boldsymbol{H}^{t_j} - \mu^{t_j})}{\sqrt{\sum(\boldsymbol{H}^{q_i} - \mu^{q_i})^2}\sqrt{\sum(\boldsymbol{H}^{t_j} - \mu^{t_j})^2}}, \tag{5}$$

where μ^{q_i} and μ^{t_j} represent the means of $\boldsymbol{H}^{q_i}$ and $\boldsymbol{H}^{t_j}$ respectively. The higher the value of the correlation coefficient, the greater will be the similarity between $\boldsymbol{H}^{q_i}$ and $\boldsymbol{H}^{t_j}$. Next, the decision rule for identifying a candidate segment can be formulated as follows:
If $C(\boldsymbol{H}^{q_i}, \boldsymbol{H}^{t_j})$ is larger than T^t (0.96), S^{t_j} is a duplicate candidate of S^{q_i} and is selected for further analysis; otherwise, S^{t_j} is genuine.

T4. Repeat Steps T1 to T3 for each segment in the test video.

3.2. *Spatial similarity measurement*

In the first phase, some spatial information is lost by shrinking the frame size to make the selection of candidate segments more efficient. In addition, because the color histogram of the image difference of two adjacent frames is used as a temporal feature, the spatial layout and structure of the image content are not utilized in the first phase. As a result, the coarse search may be inaccurate, since some candidate segments may be mis-detected after candidate segment selection. Therefore, further analysis is necessary to determine whether each candidate segment is similar to the query template.

Since only temporal information is used for the selection of candidate segments, spatial and local information can be adopted to filter out possible video segments that were identified incorrectly by the coarse search. To achieve the goal, we consider three factors to perform the fine search. First, as a fake video may be re-compressed with a different compression ratio after frame duplication, the image content should be changed after lossy re-compression. To reduce the impact of lossy re-compression on measuring the similarity in the spatial domain, a smoothing operation is performed at the beginning of the second phase. Second, to consider efficiency, only the Y component in the YC_bC_r color space is adopted in the second phase. Finally, as noted in Ref. 23, the luminance part of a video contains more information about the image content, e.g. the structure, and the perception of structured patterns is more acute. Similar to Ref. 21, we divide the Y component of a frame into several blocks to extract local information as a spatial feature.

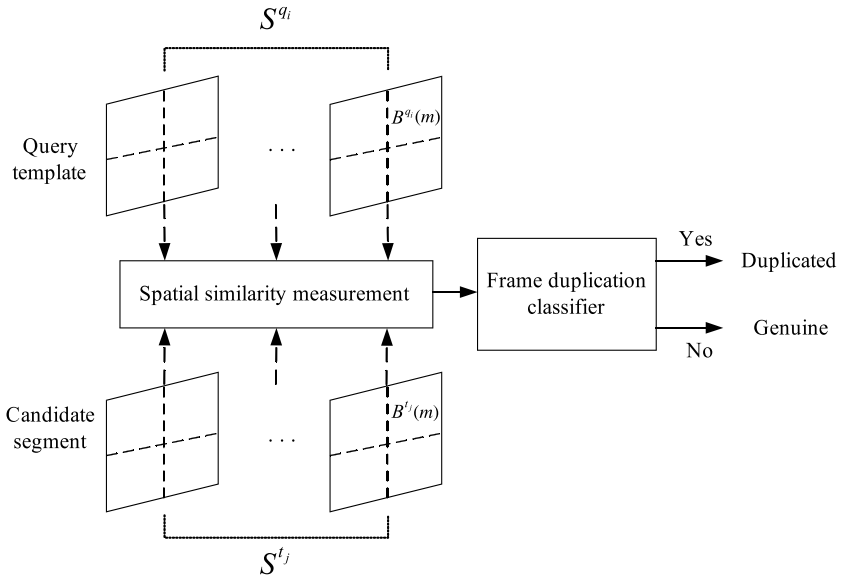

Fig. 3. The block-based spatial similarity measurement and frame duplication classification procedure.

According to the above three factors, we design a block-based algorithm to measure the spatial similarity to further analyze the relation between the query segment and each candidate. In contrast to Ref. 21, we measure the block similarity of each corresponding frame of the query template and the candidate segment directly in the spatial domain. Figure 3 shows the block-based spatial similarity measurement procedure. Each frame of S^{q_i} as well as S^{t_j} is divided into some blocks. The block correlation of each frame between S^{q_i} and S^{t_j} is calculated to measure the content similarity of each frame between S^{q_i} and S^{t_j} in the spatial domain. According to the above mention, the steps are as follows.

S1. Smooth and divide each frame of S^{q_i} into some blocks.

S2. Compute the histogram of the Y component for each block in a frame of S^{q_i} as well as the candidate segment.

S3. Calculate the histogram difference of each block of a frame between the query template and candidate segment as $B(m) = \{B^{q_i}(m) - B^{t_j}(m), 0 \leq m < N_B\}$, where $B^{q_i}(m)$ and $B^{t_j}(m)$ denote the histogram difference of the mth block of a frame in the query template and test video candidate segment respectively; and N_B denotes the number of blocks in a frame ($N_B = 15 \times 15$). If $B(m)$ is less than a pre-defined threshold (0.07), the block of the frame in S^{q_i} is similar to that in S^{t_j}; otherwise, the blocks are dissimilar.

S4. Count the number of dissimilar blocks N_{dis} between the query and candidate frames. N_{dis} can be expressed as $N_{\text{dis}} = \frac{1}{N_B} \sum_{m=1}^{N_B} U(B(m) > T_B)$, where $U(\cdot)$ denotes the unit step function ($U(x) = 1$ for $x > 0$ and $U(x) = 0$ for $x \leq 0$), and T_B is the pre-defined threshold (0.07). It is expected that if the value of N_{dis} is

small, then the frame in S^{q_i} has a high correlation with the corresponding frame in S^{t_j}.

S5. Repeat Steps S1 to S4 to measure the similarity between each frame in S^{q_i} and the corresponding frame in S^{t_j}.

3.3. *Frame duplication classification and post-processing*

As mentioned in Sec. 2, determining whether a video segment has been duplicated is a binary classification problem. Therefore, after performing temporal and spatial analysis, we need construct a classifier to select the true duplicates among the candidates.

3.3.1. *Frame duplication classifier*

As shown in Fig. 3, after measuring the spatial correlation of each frame in S^{q_i} and the corresponding frame in S^{t_j}, we evaluate the similarity of S^{q_i} and S^{t_j} as a whole. Given the spatial similarity value of each frame, a classifier based on majority voting is used to determine whether a video segment is a duplicate.

Based on the spatial similarity measurement discussed in Sec. 3.2, the similarity between S^{q_i} and S^{t_j} is computed as follows:

$$\Phi(S^{q_i}, S^{t_j}) = \frac{1}{N_q} \sum_{i=1}^{N_q} 1 - U(N_{\text{dis}}(i) > T_{\text{dis}}), \tag{6}$$

where $\Phi(S^{q_i}, S^{t_j})$ denotes the similarity between S^{q_i} and S^{t_j} and T_{dis} (0.01) is a pre-defined threshold. If a candidate segment is classified as a duplicate, the spatial correlation of each frame in S^{q_i} and the corresponding frame in S^{t_j} should be high (i.e. $1 - U(N_{\text{dis}}(i) > T_{\text{dis}})$ is large), and the value of $\Phi(S^{q_i}, S^{t_j})$ should be larger. Conversely, if a segment is deemed to be genuine, the spatial correlation between each frame S^{q_i} and the corresponding frame in S^{t_j} should be low and the value of $\Phi(S^{q_i}, S^{t_j})$ should be smaller. It can be observed that the rationale behind Eq. (6) is majority voting. Therefore, based on Eq. (6), we can determine whether the candidate segment is a replica of the query template by the following rule:

If the value of $\Phi(S^{q_i}, S^{t_j})$ is higher than the pre-defined threshold ($T_{\text{clip}} = 0.9$), S^{t_j} is a duplicate of S^{q_i}; otherwise, it is genuine.

Moreover, if S^{t_j} is identified as a replica of S^{q_i}, detection of frame duplication forgery is achieved in the test video.

3.3.2. *Post-processing*

Due to the effect of video re-compression, a whole duplicate may not be completely detected. That is, part of the duplicate clip may not be detected correctly. In addition, as a fixed-size query segment is used to find duplicates, it may not find the exact region of a duplicate clip. The partial detection problem is not addressed in Ref. 21; however, in this paper, we investigate how to locate the whole duplicated clip

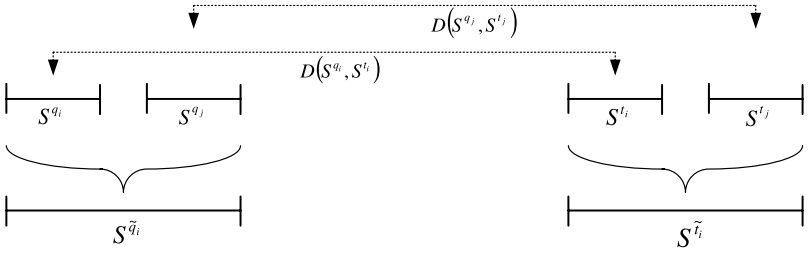

Fig. 4. The post-processing phase.

precisely, given the detected segments. This means that we must combine several detected segments to form a complete duplicated clip.

When a duplicate is divided into segments for detecting frame duplication forgery, some original segments should match the detected segments. Therefore, to combine several duplicated segments and form a complete duplicated clip, we devise a merge-and-refine algorithm as a post-processing application. The rationale behind the algorithm is that two adjacent detected duplicates can be merged if the distance between them is the same as that between the two adjacent query segments. Moreover, to solve the partial problem, the algorithm refines the length of each detected segment for examination. Figure 4 shows the post-processing phase in the proposed scheme. The steps of the merge-and-refine algorithm are as follows:

P1. Compute the distance $D(S^{q_i}, S^{t_i})$ between S^{q_i} and S^{t_j}, where $D(\cdot, \cdot)$ represents a distance function in the time domain.
P2. Examine two adjacent candidate segments to determine if the distance between them is the same as the distance between the corresponding query segments, i.e. $D(S^{q_i}, S^{t_i}) + D(S^{t_i}, S^{t_j}) = D(S^{q_i}, S^{q_j}) + D(S^{q_j}, S^{t_j})$, and $D(S^{q_i}, S^{q_j})$ is small.
P3. Combine the two adjacent segments S^{q_i} and S^{q_j} to form one segment and measure the spatial similarity of the merged segment.
P4. If the spatial similarity measurement of the merged segment is higher than a pre-defined threshold, obtain a duplicate clip.
P5. Enlarge or shrink the length of the duplicated clip frame by frame until the spatial similarity measurement is lower than the threshold. Then the output duplicate clip is the final result.
P6. Repeat Steps P1 to P5 for all the duplicated segments.

As shown in Fig. 4, S^{t_i} and S^{t_j} are examined and merged to form a complete duplicate clip $S^{\tilde{t}_j}$ after post-processing.

4. Experimental Results

We conducted a number of experiments to evaluate the performance of the proposed scheme for duplicate frame duplication. In this section, we (1) explain how to prepare the test videos; (2) describe the four performance measurements used in the

evaluations; (3) analyze the proposed scheme; and (4) compare our scheme's performance with that of the method proposed in Ref. 21.

As mentioned in Sec. 2, a long query template increases the computational cost of searching for duplicated clips. For the test videos used in our experiments, the length of a query template is 7.

4.1. *Preparation of test video clips*

To evaluate the performance of the proposed scheme, we selected several genuine videos and generated 15 fake clips. Four of the duplicated clips were generated from four videos captured by digital consumer video cameras with a frame size of 720 × 480 pixels. Two of the four videos were recorded by a fixed camera, and the other two were taken by a hand-held camera. The other 11 duplicate clips were derived from five movies, namely, What Happens in Vegas, Final Fantasy XIII, IpMan2, Avatar, and Yes Man. The frames in the 11 fake videos are in two sizes: 640 × 272 pixels and 608 × 256 pixels. To evaluate the performance of the proposed scheme, we created duplicates with different lengths ranging from 9 to 198 frames. There is at least one duplicate in these fake videos. The consumer software, Avidemux 2.5[b] and Nero 7 Ultra Edition,[c] are used to modify and re-encode genuine videos to generate fake versions.

We provide one example of fake video clips generated by frame duplication. Figure 5 shows an example of frame duplication for a video captured outdoors by a digital consumer video camera. In the figure, the number on each image indicates its order in the time domain. The top and bottom rows show genuine and fake sequences respectively. In the fake sequence, the scooter and rider appear four times compared to twice in the genuine sequence. Figure 5 demonstrates that it is difficult for the human eye to determine whether the image sequence is genuine. In addition, it is not easy to detect the duplicated segment because noise and small changes among frames exist.

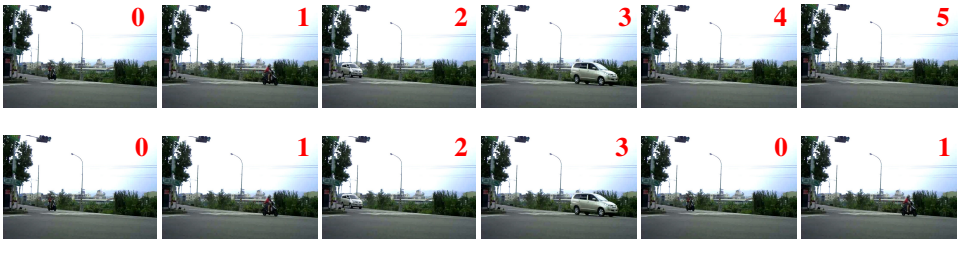

Fig. 5. An example of a fake image sequence derived from a captured video. Top: the genuine sequence; bottom: the fake sequence (color online).

[b] Avidemux 2.5, http://avidemux.berlios.de/index.html.

[c] Nero 7 Ultra Edition, http://www.nero.com/.

4.2. *Performance indices*

To evaluate the proposed scheme in terms of its capability and accuracy, we consider four performance indices: precision, recall, accuracy, and dice coefficient (DC).[4,7,12,18]

The recall and precision rates are two traditional criteria that are widely used to measure the performance of shot change detection.[4,7,12,18] The recall rate is the ratio of correct detections to the total number of true duplicates in the test videos; and the precision rate is the ratio of correct detections to total number of detected duplicates. The definitions of recall and precision are expressed as

$$\text{Recall} = \frac{N_C}{N_C + N_M} \quad \text{and} \quad \text{Precision} = \frac{N_C}{N_C + N_F}, \tag{7}$$

where N_C, N_M and N_F are the numbers of correct detections, missed detections, and false alarms, respectively; $(N_C + N_M)$ is the total number of true duplicates; and $(N_C + N_F)$ is the total number of detected duplicates. Theoretically, if a frame duplication detection scheme achieves high recall and precision rates, its performance is considered good.

We use the accuracy and dice coefficient (DC) measurements to evaluate whether the proposed scheme can localize duplicates correctly. Let Ω and Ω^* be the regions of actual and detected duplicated segments respectively. Then, the accuracy $\zeta(\Omega, \Omega^*)$ of Ω and Ω^* is defined as follows:

$$\zeta(\Omega, \Omega^*) = 1 - \delta(\Omega^* \wedge \Omega), \tag{8}$$

where the operator $\wedge$ denotes the intersection between the true and the detected regions. It is assumed that the higher the accuracy, the better will be the detection rate of the proposed scheme.

The DC measures the quality of the matches between the ranges of the detected and actual duplicates. Specifically, the DC computes the number of overlapping frames between the actual duplication and the detected one. It is defined as follows:

$$\text{DC} = \frac{2 \cdot |R_{\text{actual}} \wedge R_{\text{detected}}|}{|R_{\text{actual}}| + |R_{\text{detected}}|}, \tag{9}$$

where $|R_{\text{actual}}|$ and $|R_{\text{detected}}|$ are the numbers of frames in the actual and the detected duplicates respectively. The higher the DC value, the better will be the performance of the frame duplication detection scheme.

4.3. *Analysis of the proposed scheme*

We analyze the proposed scheme in the following.

4.3.1. *Coarse-to-fine search*

As mentioned in Sec. 3.1, the candidate segment selection phase reduces the size of the search space. Table 1 shows the number of candidates selected by the proposed scheme. For Video 5, the size of the search space is theoretically about $2.2\text{E}+407$

Table 1. The number of candidates for each test video under the proposed scheme.

	No. of Actual Replicas	No. of Candidates	
		Coarse Search	Coarse-to-Fine Search
Video 01	9	11	9
Video 02	10	76	10
Video 03	15	41	15
Video 04	35	150	35
Video 05	50	6531	50
Video 06	94	10716	94
Video 07	54	159	54
Video 08	88	674	102
Video 09	211	1002	240
Video 10	144	1821	253
Video 11	193	2015	253
Video 12	18	1228	18
Video 13	131	1188	181
Video 14	197	3661	197
Video 15	198	2186	212

when $N_F = 222$ and $N_q = 7$. However, the table shows that the number of candidates for Video 5 is reduced significantly from $2.2\text{E}+407$ to 6531. The result demonstrates that the candidate segment selection phase is effective in reducing the size of the search space.

Recall that the fine search filters out possible mis-classified segments generated in the coarse search and finds actual replicas. Table 1 shows that, for each video, the number of candidates is much higher in the coarse search than in the fine search. We observe that the number of candidates is reduced by at least 81.8% after the spatial similarity measurement step. The result shows that spatial similarity measurement can effectively rule out some candidates caused by noise and improve the performance of the proposed scheme after the coarse-to-fine search.

4.3.2. *Computation time*

Table 2 shows the computation times of the proposed scheme for the 15 video clips used in the experiments. Clearly, the coarse search requires less time than the fine search. In fact, the latter consumes at least 80.2% of the total computation time. Though the three components of the RGB color space are used to find possible duplicate segments, the global feature (i.e. CHD) effectively improve the efficiency of the coarse search. In addition, this is reasonable because the block-based spatial information in each frame is used to calculate the similarity between $\boldsymbol{S^q}$ and $\boldsymbol{S^r}$; and the larger the number of candidates, the longer will be the computation time. This is exemplified by the results for Video 6, Video 14, and Video 15 in Tables 1 and 2. Many candidates are selected for the videos and the computation time for the fine search is significantly longer than that required for the other videos. The results match our analysis in Secs. 2 and 3.2.

Table 2. Computation times of the test videos under the proposed scheme.

	Computation Time (s/frame)		
	Coarse Search	Fine Search	Total
Video 01	0.172	0.046	0.218
Video 02	0.391	0.250	0.641
Video 03	0.625	0.266	0.891
Video 04	0.688	0.484	1.172
Video 05	0.774	19.434	20.208
Video 06	2.559	160.552	163.111
Video 07	0.478	6.680	7.158
Video 08	0.830	243.520	244.350
Video 09	1.195	421.791	422.987
Video 10	1.013	378.576	379.589
Video 11	0.914	314.452	315.366
Video 12	0.605	88.313	88.917
Video 13	0.576	81.717	82.293
Video 14	1.182	1394.170	1395.352
Video 15	0.886	503.438	504.323

4.3.3. *Post-processing*

Figure 6 shows an example of post-processing. Figures 6(a) and 6(b) are the original and duplicated segments respectively. In this example, Frames 0 to 6 are copied and pasted as Frames 10 to 16. The two red rectangles (dashed lines) indicate two detected replicas. Lossy re-compression leads to partial detection of frames by the proposed scheme. The blue rectangle (solid line) in the figure represents a duplicate after post-processing, and demonstrates that the complete duplicate can be detected after post-processing. The result shows that the post-processing phase can merge and refine two adjacent duplicated segments into a complete fake clip.

4.4. *Comparison with Wang and Farid's method*[21]

Next, we compare the proposed scheme's performance with that of Wang and Faird's method[21] in terms of the precision, recall, accuracy, DC, and computation time.

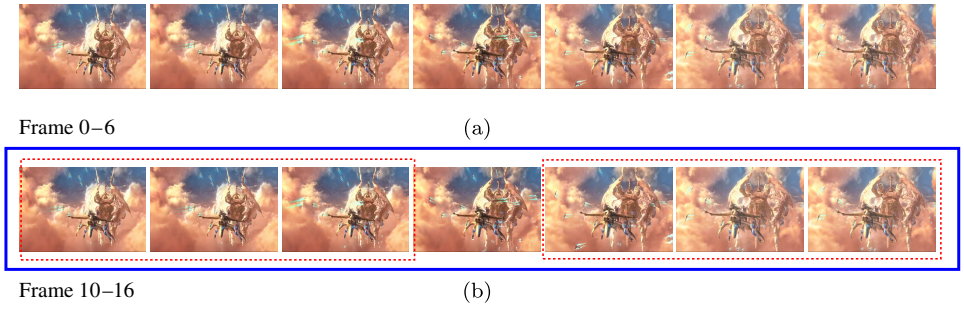

Fig. 6. An example of post-processing; (a) original segment, (b) duplicated segment (color online).

Table 3. The precision, recall, accuracy, DC, and computation times of the compared schemes.

Scheme	Precision	Recall	ζ	DC	Computation Time (s/frame)
Ref. 21	0.310	0.788	0.867	0.341	264.67
Proposed	0.849	1	1	0.951	109.6

As shown in Table 3, the precision and recall rates of the proposed scheme are 0.849 and 1, respectively. The recall rate indicates that the scheme can detect actual duplicates in different kinds of videos. The scheme is effective because it utilizes more information (i.e. the color histogram in the RGB color space) as a feature in the coarse search. The feature is used to find a small set of candidate segments, including actual replicas, for further analysis. The precision rate (0.849) means that there are relatively few mis-detected duplicates (i.e. false alarms) under the proposed scheme. However, most of the actual replicas among the candidates are detected by using the feature in the fine search. The results demonstrate that the proposed scheme performs well, and the impact of lossy re-compression on frame duplication detection is reduced. In addition, the precision and recall rates of the approach in Ref. 21 are lower than those of the proposed scheme, as shown in Table 3. This means that compared with the proposed scheme, few actual duplicates but a number of misclassified video segments (i.e. false alarms) are found by the method in Ref. 21.

Table 3 also shows that the accuracy rates of the proposed scheme and the approach in Ref. 21 are 1 and 0.867, respectively; while the DC rates are 0.951 and 0.341, respectively. These results demonstrate that, for different kinds of videos, our scheme can detect and localize actual duplicates more effectively than Wang and Faird's method.

The computation times of the compared schemes are also shown in Table 3. The total computation time of our scheme is 41.4% less than that required by Ref. 21. The key reason is that the proposed scheme is a coarse-to-fine approach to reduce the size of the search space during the coarse search. In addition, compared with Ref. 21, a few number of candidate segments are examined in the fine search.

To summarize, the experiment results demonstrate that, in terms of precision, recall, accuracy, DC, and computation time, the proposed scheme outperforms Ref. 21 on different kinds of videos.

5. Concluding Remarks and Future Work

We have presented a passive-blind scheme for detection of frame duplication forgery in videos. The scheme is a coarse-to-fine approach that analyzes the similarity between two video clips in the temporal and spatial domains. It is comprised of four phases: candidate segment selection, spatial similarity measurement, frame duplication classification, and post-processing. To find duplicated candidates in the temporal domain, we use the histogram difference of two adjacent frames in the RGB

color space as a feature. Then, to evaluate the similarity of the image content, a block-based algorithm is used to measure the spatial correlation of each frame in the candidate segment with the corresponding frame in the query template. Based on the spatial and temporal analysis, we construct a classifier to detect duplicated segments. To solve the partial detection problem, a post-processing technique is used to examine, merge, and refine two adjacent detected candidates into a complete duplicated video clip.

To evaluate our scheme's performance, we conducted experiments on 15 fake video clips compiled from different kinds of videos. For the 15 clips, the scheme's precision and recall rates were 0.84 and 1, respectively; while the accuracy and DC rates were 1 and 0.95, respectively. The results show that detection of frame duplication forgery can be achieved. In addition, the results demonstrate the efficacy of the proposed scheme in detecting and localizing duplicated clips. Finally, the results also demonstrate that the proposed scheme outperforms the method in Ref. 21 in terms of precision, recall, accuracy, and computation time.

A direction for future work is to explore and combine other features for reducing the number of candidates selected in the coarse search in order to raise the efficiency of the proposed scheme. In addition, near-duplicate video detection is an important issue in some applications such as data mining, TV broadcasting, online video usage monitoring, copyright enforcement, and so on. We will apply the proposed scheme to achieve near-duplicate video detection in the future.

Acknowledgment

This research was supported by the National Science Council, Taiwan, under the grant of NSC 101-2221-E-212-019. Moreover, the authors would like to thank the editor and reviewers for their valuable suggestions.

References

1. Y. L. Chen and C. T. Hsu, Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection, *IEEE Trans. Inform. Forensics Security* **6** (2011) 396–406.
2. Y. L. Chen and C. T. Hsu, Image tampering detection by blocking periodicity analysis in JPEG compressed images, in *Proc. IEEE 10th Workshop on Multimedia Signal Processing* (2008), pp. 803–808.
3. C. Cotsaces, N. Nikolaidis and I. Pitas, Video shot detection and condensed representation, *IEEE Signal Process. Mag.* **23** (2006) 28–37.
4. A. P. Dhawan, *Medical Image Analysis* (John Wiley & Sons, Inc, New Jersey, 2003).
5. A. Hanjalic, Shot-boundary detection: Unraveled and resolved, *IEEE Trans. Circuits Syst. Video Technol.* **12** (2002) 90–105.
6. C. C. Hsu, T. Y. Hung, C. W. Lin and C. T. Hsu, Video forgery detection using correlation of noise residue, in *Proc. IEEE Int. Workshop on Multimedia Signal Processing* (2007), pp. 170–174.

7. C. L. Huang and B. Y. Liao, A robust scene-change detection method for video segmentation, *IEEE Trans. Circuits Syst. Video Technol.* **11** (2001) 1281−1288.
8. P. Kakar, N. Sudha and W. Ser, Exposing digital image forgeries by detecting discrepancies in motion blur, *IEEE Trans. Inform. Forensics Security* **13** (2011) 443−452.
9. P. Kalar, S. Natarajan and W. Ser, Detecting digital image forensics through inconsistent motion blur, in *Proc. IEEE Int. Conf. Multimedia Expo* (2010), pp. 486−491.
10. M. Kobayashi, T. Okabe and Y. Sato, Detecting video forgeries based on noise characteristics, *LNCS* **5414** (2009) 306−317.
11. W. N. Lie, G. S. Lin and S. L. Cheng, Dual protection of JPEG images based on informed embedding and two-stage watermark extraction techniques, *IEEE Trans. Inform. Forensics Security* **1** (2006) 330−341.
12. G. S. Lin, M. K. Chang and S. T. Chiu, A feature-based scheme for detecting and classifying video-shot transitions based on spatio-temporal analysis and fuzzy classification, *Int. J. Pattern Recogn. Artificial Intell.* **23** (2009) 1179−1200.
13. G. S. Lin, M. K. Chang and Y. L. Chen, A passive-blind scheme for image forgery detection based on content-adaptive quantization table estimation, *IEEE Trans. Circuits Syst. Video Technol.* **21** (2011) 421−434.
14. G. S. Lin, Y. T. Chang and W. N. Lie, A framework of enhancing image steganography with picture quality optimization and anti-steganalysis based on simulated annealing algorithm, *IEEE Trans. Multimedia* **12** (2010) 345−357.
15. C. S. Lu and H. Y. Mark Liao, Multipurpose watermarking for image authentication and protection, *IEEE Trans. Image Process.* **10** (2001) 1579−1592.
16. Y. X. Peng and C. W. Ngo, Clip-based similarity measure for query-dependent clip retrieval and video summarization, *IEEE Trans. Circuits Syst. Video Technol.* **16** (2006) 612−627.
17. A. C. Popescu and H. Farid, Exposing digital forgeries in color filter array interpolated images, *IEEE Trans. Signal Process.* **53** (2005) 3948−3959.
18. C. W. Su, H. Y. Liao, H. R. Tyan, K. C. Fan and L. H. Chen, A motion-tolerant dissolve detection algorithm, *IEEE Trans. Multimedia* **7** (2005) 130−140.
19. Y. Su, J. Xu, B. Dong, J. Zhang and Q. Liu, A novel source MPEG-2 video identification algorithm, *Int. J. Pattern Recogn. Artificial Intell.* **24** (2010) 1311−1328.
20. Y. Su, J. Zhang, Y. Han, J. Chen and Q. Liu, Exposing digital video logo-removel forgery by inconsistency of blur, *Int. J. Pattern Recogn. Artificial Intell.* **24** (2010) 1027−1046.
21. W. Wang and H. Faird, Exposing digital forgeries in video by detecting duplication, in *ACM Multimedia and Security Workshop* (2007).
22. W. Wang and H. Farid, Exposing digital forgeries in interlaced and de-interlaced video, *IEEE Trans. Inform. Forensics Security* **2** (2007) 438−449.
23. H. R. Wu and K. R. Rao, *Digital Video Image Quality and Perceptual Coding* (CRC Press, Taylor & Francis Group, New York, 2006).

Guo-Shiang Lin was born in Tainan, Taiwan, R.O.C., in 1971. He received his Ph.D. from the Department of Electrical Engineering, National Chung Cheng University, Taiwan, R.O.C., in May 2005. In August 2005, he joined the Department of Computer Science and Information Engineering, Da-Yeh University, Taiwan, R.O.C., where currently he is an Associate Professor. He was a Visiting Professor at the Department of Mathematics and Computer Science, University of Münster, Germany, from June to September 2009. His current research interests include image/video forensics, multimedia signal processing and analysis, 2D-to-3D image/video conversion, computer vision, and pattern recognition.

Jie-Fan Chang received his B.S. degree in 2009 from the Da-Yeh University, Changhua Taiwan, R.O.C. Currently he is pursuing his M.S. degree at National Chung Hsing University, Taichung, Taiwan, R.O.C. His research interests include digital image and video forensics, and data mining.