Automatic Detection of Object-Based Forgery in Advanced Video

Shengda Chen, Shunquan Tan, *Member, IEEE*, Bin Li, *Member, IEEE*, and Jiwu Huang, *Senior Member, IEEE*

***Abstract*— Passive multimedia forensics has become an active topic in recent years. However, less attention has been paid to video forensics. Research on video forensics, and especially on automatic detection of object-based video forgery, is still in its infancy. In this paper, we develop an approach for automatic identification and forged segment localization of object-based forged video encoded with advanced frameworks. The proposed approach starts with a frame manipulation detector. An automatic algorithm is proposed to identify object-based video forgery based on the frame manipulation detector. Then, a two-stage automatic algorithm is provided to accurately locate the forged video segments in the suspicious video. To construct the proposed frame manipulation detector, motion residuals are generated from the target video frame sequence. We regard the object-based forgery in video frames as image tampering in the motion residuals and employ the feature extractors that are originally built for still image steganalysis to extract forensic features from the motion residuals. The experiments show that the proposed approach achieves excellent results in both forged video identification and automatic forged temporal segment localization.**

***Index Terms*— Motion residual, object-based video forgery, steganalysis, video forensics.**

I. Introduction

With the wide availability of powerful media editing tools, it becomes much easier to manipulate or even tamper with digital media without leaving any perceptible traces. This leads to an increasing concern about the trustworthiness of digital media contents [1], and there is a pressing need to develop effective forensic techniques to verify the authenticity, originality, and integrity of media contents. *Passive forensics*, which aims at this purpose and only exploits the intrinsic statistical characteristics of digital media itself, becomes an active topic in recent years [2]. A substantial number of image forensic techniques have been developed to perform diverse tasks such as detecting the evidence of image forgery [3], [4], tracing the processing history of an image [5], and identifying the source of an image [6]. While some efforts on video forensics have also been made in the meantime, most of the existing video forensic algorithms either expose the evidence of side effects of forgery or detect the so-called *frame-based forgery*, which refers to the manipulations that insert or delete frames. In [7] and [8], detection methods of double compression on video were proposed to expose the evidence of nonoriginality. Stamm *et al.* [9] developed a frame deletion fingerprint based on the group of pictures (GOP) structure used in an MPEG video encoder. An antiforensic technique was also proposed in [9] to remove that frame deletion fingerprint. Research results in [10] revealed that intrinsic noise introduced during the video acquisition can also act as the inconsistent trails for forensics. Less attention has been paid to the forensics of *object-based forgery*, which adds new objects to a video scene or removes existing objects from it. Please note that object-based forgery is a common video tampering method since the object added into or removed from a video is usually critical to the contents that video conveys. Therefore, the attention paid to the forensics of object-based video forgery does not match its importance. Hsu *et al.* [11] proposed a method to detect naïve video forgery, which utilize computerized automatic inpainting. Zhang *et al.* [12] proposed a detection scheme of the forgery of the sole moving object in scene based on geometrical inconsistencies. Subramanyam and Emmanuel [13] proposed a forgery detection technique to detect object-based splicing, which relies on the GOP-based histogram of oriented gradients feature matching. Conotter *et al.* [14] proposed a specific forensic method to detect the forgery of objects in ballistic motion based on physical inconsistencies. Chen *et al.* [15] proposed a forensic method to detect the forgery of single moving object with absolutely static background relying on the statistical features of object contour. All of the above works are devoted to the forensic analysis of naïve tampering, or the manipulation of simplified scenes or specific objects. Furthermore, as far as we know, there has been no report on the video forensic method that can locate the object-based forged temporal segments in forged video.

In this paper, we develop an approach for automatic identification, and furthermore forged segment localization

Manuscript received January 11, 2015; revised May 1, 2015 and July 20, 2015; accepted August 18, 2015. Date of publication August 26, 2015; date of current version October 27, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61332012, Grant 61402295, and Grant 61572329; in part by the Guangdong National Science Foundation under Grant 2014A030313557; and in part by the Shenzhen Research and Development Program under Grant GJHZ20140418191518323 and Grant JCYJ20140418182819173. This paper was recommended by Associate Editor V. Monga. *(Corresponding author: Shunquan Tan.)*

S. Chen is with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: chshda@foxmail.com).

S. Tan is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with Shenzhen Key Laboratory of Media Security, Shenzhen 518060, China (e-mail: tansq@szu.edu.cn).

B. Li and J. Huang are with the College of Information Engineering, Shenzhen University, Shenzhen 518060, China, and also with Shenzhen Key Laboratory of Media Security, Shenzhen 518060, China (e-mail: libin@szu.edu.cn; jwhuang@szu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2015.2473436

of object-based forged video generated by skilled forgers and encoded with advanced video encoding standards. The proposed approach starts with a frame manipulation detector that is used to detect the traces in the video frames left by the suspicious manipulations. An automatic algorithm is proposed to identify object-based forgery in advanced video on the basis of the output of the frame manipulation detector. Furthermore, we put forward a two-stage solution to automatically locate the boundaries of the forged temporal segments from the suspicious videos: in the first stage the coarse boundaries of the forged segments are located, and then in the second stage an algorithm is proposed to fine-tune the boundaries based on the votes of the base learners of the frame manipulation detector. To construct the proposed frame manipulation detector, motion residuals are generated from the target video frame sequence using collusion operators. We point out that the obtained motion residuals can be treated as still images, and the object-based forgery in video frames can be regarded as image tampering in the motion residuals extracted from the corresponding video frames. The feature extractors originally built for still image steganalysis are adopted in our work to extract forensic features from the motion residuals. Then, the resulting features are used as the inputs of the frame manipulation detector.

The rest of this paper is organized as follows. In Section II, we first analyze the characteristics of object-based forgery in advanced video and the traces that forgery operation left. Then, the extraction of motion residuals is described in detailed. The proposed identification and automatic segment localization algorithm for object-based video forgery is described in Section III. We present extensive experimental results in Section IV. Finally, the paper is concluded in Section V.

II. Analysis of Object-Based Forgery in Advanced Video

A. Object-Based Forgery in Advanced Video

Fig. 1(a) gives the diagram of object-based video forgery procedure. In general, videos are in compressed format. Therefore, when a pristine video undergoes some kinds of object-based forgery, the first step is to decompress it to a sequence of individual frames and each frame can be regarded as a still image. Then, the frames in the selected segments of the sequence are tampered with, while the rest of the frames remain untouched. After all the manipulations are finished, the resulting frame sequence is recompressed to generate a forged version. Please note that those untouched frames in a forged video do contain some artifacts introduced in recompression though they do not have any perceptible difference compared with their original counterparts. Based on this scenario, we classify all frames in a test video into three categories.

1) *Pristine Frames:* The frames in a pristine compressed video stream that do not undergo any manipulation.
2) *Forged Frames:* The frames in a forged video stream that have undergone tampering operations.
3) *Double-Compressed Frames:* The frames in a forged video that remain untouched, but still have undergone recompression.

Please note that a forged video itself is a double-compressed video stream since re-encoding/recompression is indispensable in forgery. Therefore, a forged frame is also double compressed.

To forensically analyze the effect of object-based forgery, we provide a brief overview of video compression. Although there are various video compression techniques, most of them are of the same infrastructure. The similarity between the neighboring frames is exploited by video encoders via predicting a certain frame from its neighbors and then only encoding the prediction errors. Specifically, an encoded video stream consists of a series of successive GOP. Each GOP in turns contains three types of frames: I-frames (intra-coded frames), P-frames (predictive-coded frames), and B-frames (bipredictive-coded frames). An I-frame indicates the beginning of a GOP. It contains the full picture and is independently encoded as a still image. P-frames contain motion-compensated difference information relative to the preceding frames. B-frames are encoded in a similar manner as P-frames. However, their motion-compensated difference information can be relative to not only the preceding frames but also the posterior frames. Basically, P-frames and B-frames all rely on the I-frame in a GOP. All types of frames in a GOP exhibit strong correlations. In early video compression standards such as MPEG I/II, the structure of each GOP is fixed. In advanced frameworks such as H.264/MPEG-4, GOP have much more flexible structures. For example, H.264 encoder can generate shorter GOP for rapidly changing scenes because the accuracy of motion-compensated prediction greatly decreases as the objects in a scene change their trajectories abruptly. After the manipulation of the objects, the absolute positions of the I-frames and the lengths of the GOP in the recompressed object-based forged video will be completely different compared with what in the corresponding original version, which makes traditional GOP-based forensic methods such as what proposed in [9] and [13] failed. An illustration of an object-based forgery procedure in an advanced video is shown in Fig. 1(b). In this case, the forger tries to erase one plane from a video. The video contains four GOP, each of which possesses one leading I-frame and subsequent three B-frames/P-frames. He first decompresses the video into a sequence of individual frames. Then, the segment with the plane appearing on the scene is selected to perform object-based forgery, in which the plane in each individual frame is completely erased. Finally, the resulting frame sequence is recompressed to generate a new video stream. Since the rapidly moving plane is erased from the video and what remained in the scene is a slowly moving cloud, longer GOP structures are used to encode the resulting video. As shown at the bottom of Fig. 1(b), the recompressed video only contains two GOP, while the second and the fourth I-frames in the original video are replaced by two P-frames (the two slices with gray oblique stripe pattern). The flexibility of the GOP structure in advanced video encoding frameworks presents great challenge to the forensics of object-based forgery.

However, to generate an object-based forged video without leaving any perceptible traces, object-based video forgery itself must be an elaborate and tedious operation. It usually requires

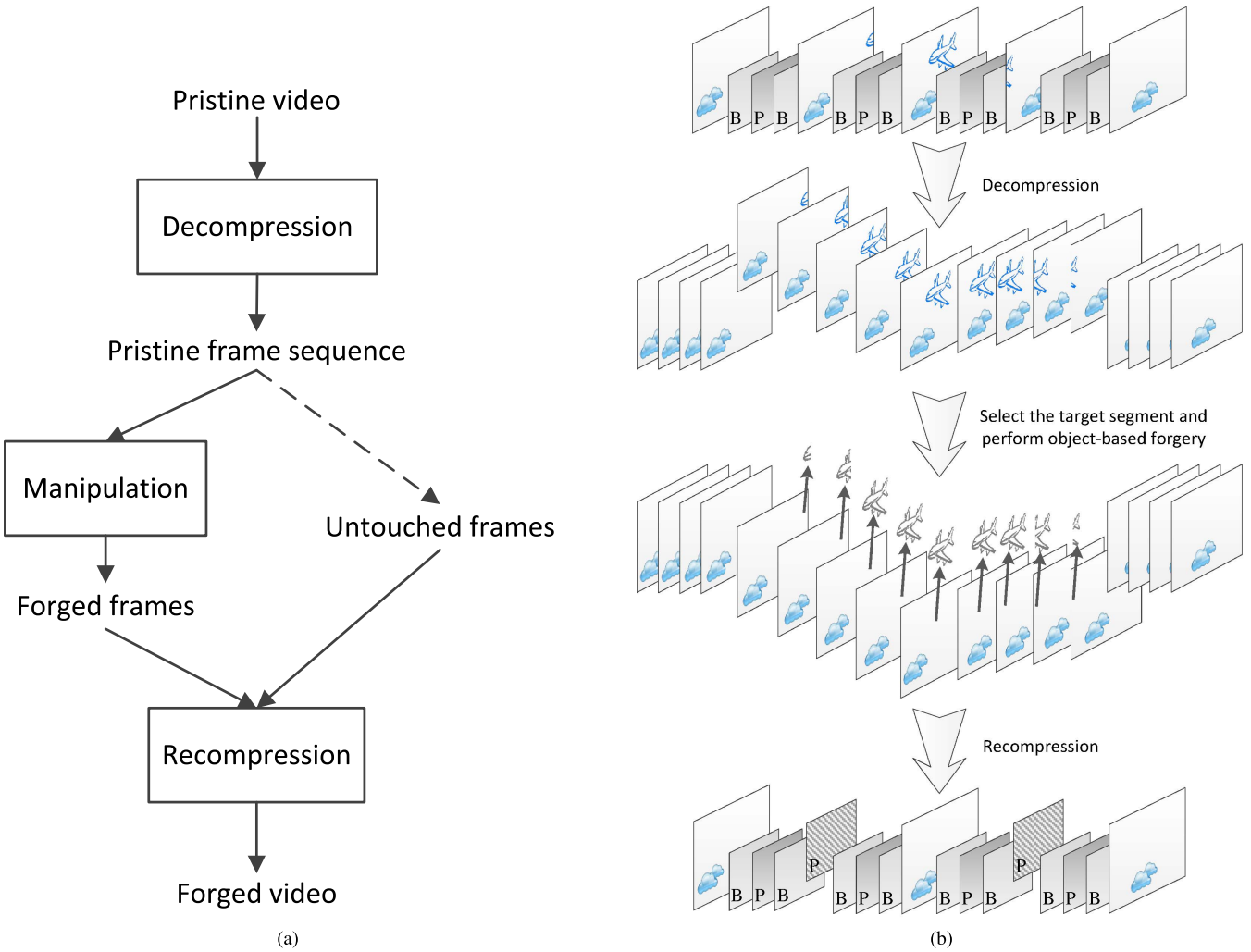


Fig. 1. (a) Diagram of an object-based video forgery procedure. (b) Illustration of an object-based forgery in an advanced video. The slices represent individual video frames, among which the slices with P and B marks denote the P-frames and B-frames, respectively, while the slices with figures denote the I-frames in an encoded video stream, or the decompressed version of all types of video frames in a decompressed video frame sequence.

some postprocessing operations such as contour blurring, contrast adjusting, video in-painting, and video layer fusion, which in turns inevitably alter some inherent statistical properties of the tampered pristine video. It is expected that the alteration of those inherent statistical properties can be detected effectively if they can be well modeled. For a target video, its inherent statistical properties can be extracted from the corresponding motion residual sequence, as detailed in Section II-B.

B. Extraction of Motion Residual Using Collusion Operators

The inherent statistical properties of a video can be divided into two categories: the *intra-frame inherent properties* that describe its spatial characteristics, and the *inter-frame inherent properties* that describe its temporal characteristics. The strong correlations among the neighboring frames in a video imply that each frame in a local temporal window comprises two parts: the motion part and the static part. The static part is identical to the basic anchor frame of the local temporal window, while the motion part is the motion residual relative to that anchor frame. For each frame in a video, its motion residual, as the principal part of the visual information presented by that frame, contains a substantial portion of the intra-frame properties of that frame. Furthermore, the motion residual also contains the inter-frame inherent properties of the corresponding frame since it represents the temporal changes from that frame to the basic anchor frame. Since the motion residuals contain both the intra-frame and inter-frame inherent properties of the corresponding frames, they become our primary analysis object. Actually, each GOP in an encoded video stream can be considered as a local temporal window. The I-frame in a GOP represents an anchor frame and the P-frames/B-frames in that GOP are actually the motion residuals of the corresponding I-frame. However, the flexible structure of GOP in advanced video framework makes GOP-based video forensic method unattainable. As an alternative, collusion operators are adopted in the extraction of motion residual in our proposed method [16], [17].

Denote a sequence of decompressed video frames of length N as

$$\mathbb{V} \triangleq \{F^{(1)}, F^{(2)}, \ldots, F^{(N)}\}, \quad N \in \mathbf{Z} \tag{1}$$

where $F^{(k)} = (F^{(k)}_{i,j}) \in \{0, \ldots, 255\}^{n_1 \times n_2}$ represents the kth decompressed video frame, which is actually an 8-bit

gray-scale still image of $n_1 \times n_2$. A collusion operation inside a temporal window of the target video frame sequence, which centered at frame $F^{(k)}$ with the window size of $L = 2 \times L_h + 1$ (L_h is the number of the left/right neighbors of $F^{(k)}$), is defined as:

$$\begin{aligned} C^{(k)} &= \left(C_{i,j}^{(k)}\right) \\ &= \mathfrak{C}\left[\left(F_{i,j}^{(k-L_h)}\right), \ldots, \left(F_{i,j}^{(k)}\right), \ldots, \left(F_{i,j}^{(k+L_h)}\right)\right] \end{aligned} \quad (2)$$

where $C^{(k)}$ is the colluded result for $F^{(k)}$ and $\mathfrak{C}$. The collusion operator $\mathfrak{C}$ is actually an aggregate function that groups the pixels in the corresponding coordinates of every frames in the temporal window to generate $C_{i,j}^{(k)}$. The motion residual of $F^{(k)}$, is defined as

$$\begin{aligned} R^{(k)} &= |F^{(k)} - C^{(k)}| \\ &= \left(R_{i,j}^{(k)}\right) = \left(\left|F_{i,j}^{(k)} - C_{i,j}^{(k)}\right|\right) \end{aligned} \quad (3)$$

where $|\cdot|$ denotes the absolute value.

Two collusion operators used in our experiments are defined in the following equations, in which $\mathfrak{C}_{\text{MIN}}$ and $\mathfrak{C}_{\text{MEDIAN}}$ represent minimum collusion and median collusion, respectively:

$$\mathfrak{C}_{\text{MIN}} \triangleq \min_{l \in [-L_h, L_h]} \left\{F_{i,j}^{(k+l)}\right\} \quad (4a)$$

$$\mathfrak{C}_{\text{MEDIAN}} \triangleq \tilde{F}_{i,j}^{((L+1)/2)} \quad \text{given } \left\{\tilde{F}_{i,j}^{(l)}\right\} \text{ is a sorted version of } \left\{F_{i,j}^{(k+l)}\right\}. \quad (4b)$$

Note that according to (4a) and (4b), $C^{(k)} \in \{0, \ldots, 255\}^{n_1 \times n_2}$. Therefore, $F^{(k)} - C^{(k)} \in \{-255, \ldots, 255\}^{n_1 \times n_2}$. As a consequence, the absolute value operator $|\cdot|$ in (3) is necessary to limit the range of $R^{(k)}$ in $\{0, \ldots, 255\}^{n_1 \times n_2}$. The resulting $R^{(k)}$ can be regarded as an 8-bit gray-scale still image. Consider one degradation situation in which $F_{i,j}^{(k-L_h)} = \cdots = F_{i,j}^{(k)} = \cdots = F_{i,j}^{(k+L_h)}$. No matter what collusion operator is adopted, the colluded result $C^{(k)}$ is equal to $F^{(k)}$, which determines that $R^{(k)}$ is an all-zero matrix, namely, a null motion residual. The degradation situation mentioned above represents a completely static scene in the temporal collusion window. Increasing motion in the scene leads to increasing difference between $F^{(k)}$ and $C^{(k)}$, which in turns results in an $R^{(k)}$ with growing elements. Therefore, $C^{(k)}$ describes the object migration inside a temporal collusion window and $R^{(k)}$ acts as a measure of the motion in the window. This is why we call $R^{(k)}$ the motion residual of a collusion window. $R^{(k)}$ does not rely on the underlying flexible GOP structure and therefore is suitable for our proposed method that is aimed at advanced video. From (3), we can also see that with the introduction of motion residual, object-based video forgery turns into the modification of the pixel values in the corresponding motion residuals. Therefore, object-based video forgery can be regarded as image tampering in motion residuals.

The pristine video clip illustrated in Fig. 2(a) recorded a surveillance scene. The two men (marked with white dotted circles) were completely erased from the scene in the corresponding forged video clip, as shown in Fig. 2(b). The motion residual of the frame No. 40 in the pristine video clip and the corresponding forged one are shown in Fig. 2(c). We can observe that the relative static background in the scene corresponds to the dark area in the motion residuals. The gray shapes of the objects in motion can be seen in those two motion residuals. The faster the object is moving, the more obvious and wider the ghost shape of the object becomes. As the top figure of Fig. 2(c) shows, the shapes of the two quickly moving men are wider than those of the two slowly strolling girls. Since the two men were erased from the scene in the forged video clip, the region that contains their ghost shapes in the top figure turns into a dark region in the bottom figure (the corresponding two regions are marked with white dashed rectangle). From Fig. 2(c), it is hard to find any perceptible traces of forgery. However, some inconsistent trails for forensics still can be accessed via careful analysis in this special case. The figures of the two walking men caused subtly luminance variety of the light reflected by the polished floor, which becomes visible after increasing the contrast of $R^{(40)}$. In Fig. 2(d), a layer of light gray shadow corresponding to the variety of luminance roughs out the floor tile pattern, as the arrows indicate. However, as shown in Fig. 2(e), in the same region of the contrast-increased $R^{(40)}$ of the forged video, the floor tile pattern rendered by the light gray shadow disappears abruptly around the region where the two walking men erased from. It is obvious that to erase the two men, the forger covered that region with a static moving-object-free floor image that is cut from other frames. Similar to this special case, inconsistent trails of object-based video forgery can be found by carefully manual analysis. However, it is infeasible to identify forged videos in a mass of video data merely by manual analysis. Fortunately, state-of-the-art steganalytic techniques can be utilized for identification and automatic segment localization of object-based forgery, as revealed in Section III.

III. Identification and Automatic Segment Localization of Object-Based Forgery

As shown in Fig. 3, the proposed approach starts with a frame manipulation detector based on motion residuals to identify forged video. After that, an automatic algorithm is provided for a given suspicious video to locate the forged temporal segments, which can be divided into two stages. First, a recursive procedure is presented to locate the coarse boundaries of the forged segments. Then, an accurate positioning algorithm is used to fine tune the boundaries of the forged segments.

A. Frame Manipulation Detector and Forgery Identification

How to model the intra-frame and inter-frame inherent properties of a pristine video is the key issue of successful detection of object-based forgery in advanced video. From Section II-B, we have already known that a motion residual can be regarded as an 8-bit gray-scale still image and object-based video forgery can be regarded as image tampering in motion residuals. Since image forensics is the art of detecting image tampering/processing, we can use image forensic methods to detect tampering in motion residuals. Let us go a step further. Li *et al.* [3] and Qiu *et al.* [4] have revealed that tampering/processing in still images can be modeled as image data

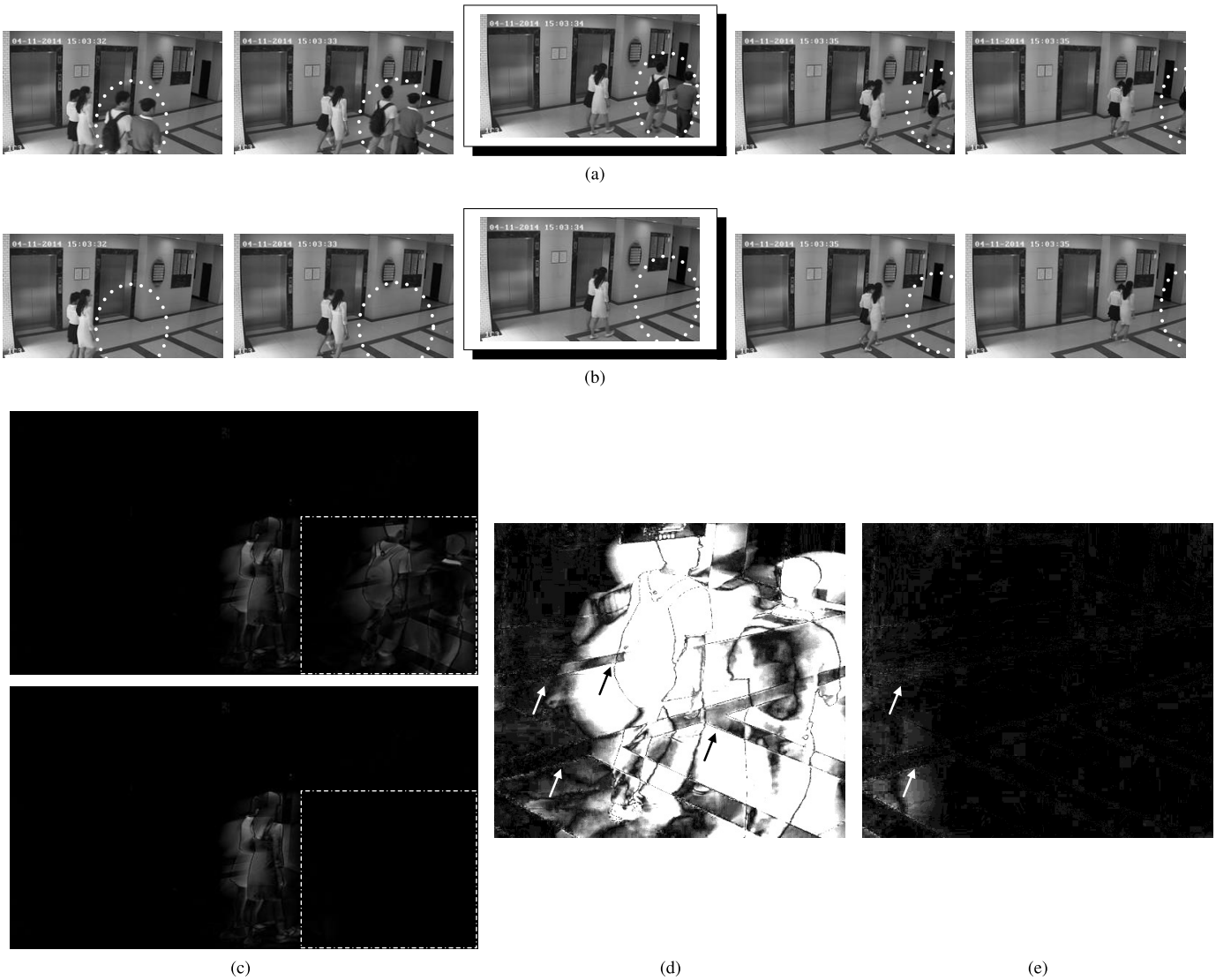


Fig. 2. (a) Representative frames of a pristine video clip. Left to right: frames 1 to 80 in steps of 20. (b) Representative frame of the corresponding forged version of the pristine video clip shown by (a), in which two walking men were erased from the scene. (c) Top and bottom: $R^{(40)}$, namely, the motion residual of frame 40 of the pristine video clip shown by (a) and its forged version shown by (b), respectively. The corresponding frames in (a) and (b) are emphasized by shadow, $L_h = 9$ and $\mathfrak{C}_{\text{MEDIAN}}$ are adopted in the collusion. (d) Enlargement of the rectangular region surrounded by dashed lines in the top figure of (c). (e) Enlargement of the rectangular region surrounded by dashed lines in the bottom figure of (c). The data values in (e) are remapped to fill the entire intensity range $[0, 255]$ so that its contrast can be increased. To facilitate the comparison between (d) and (e), the data values in (d) which are less than or equal to the maximum value in (e) are also remapped to $[0, 255]$, while the values higher than that maximum are mapped to the rightmost intensity.

hiding/steganography and state-of-the-art image steganalytic features can be used to detect them. Image steganalysis is the art of detecting data hidden in cover image by means of steganography. Using motion residuals as intermediates, we can borrow some powerful statistical features from image steganalysis to model the alteration of the inherent properties of a video clip introduced by object-based forgery.

Seven state-of-the-art image steganalytic features, CC-PEV [18], [19], SPAM [20], CDF [21], CF* [22], SRM [23], CC-JRM [24], and J + SRM [24], are extracted from the motion residuals to model the intra-frame and inter-frame inherent properties contained in them, which are of 548, 686, 1234, 7850, 34 671, 22 510, and 35 263 dimensions, respectively.[1] Among them, SPAM and SRM are oriented to spatial-domain images, while CC-PEV, CF*, and CC-JRM are for frequency-domain images. Different from CF*, both CC-PEV and CC-JRM are combined with Cartesian calibration. CDF is the union of CC-PEV and SPAM features, while J + SRM is the union of SRMQ1 (the reduced version of SRM with fixed quantization) and CC-JRM. Both of them are cross-domain feature sets. From the steganalytic point of view, the best detection is usually achieved by extracting features directly in the domain the embedding modification occurs in [25]. Since in a video stream either the I-frames themselves or the motion vectors in the P-frames and B-frames are compressed using frequency-domain lossy compression scheme, the intra-frame inherent properties of the frames can be modeled by frequency-domain oriented feature sets, such as CC-PEV, CF*, and CC-JRM. The inter-frame inherent properties in the motion residuals are synthesized via collusion. The temporal variety of the moving objects is

[1] Adopt the abbreviations in http://dde.binghamton.edu/download/feature_extractors/ for the feature extractors listed above.

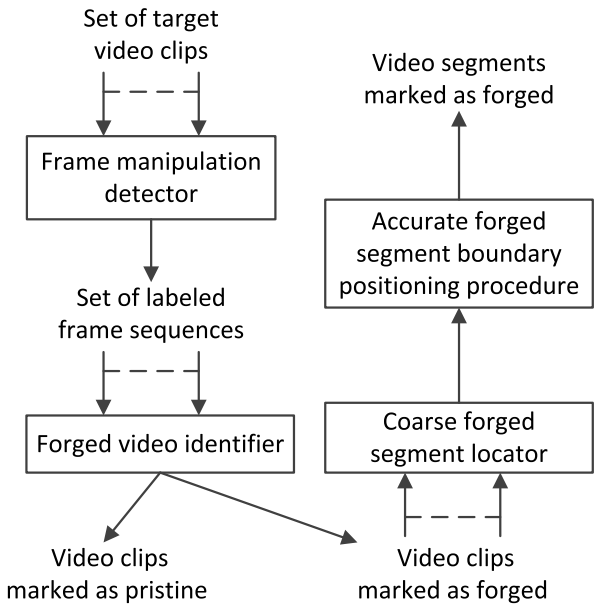


Fig. 3. Diagram of our proposed approach.

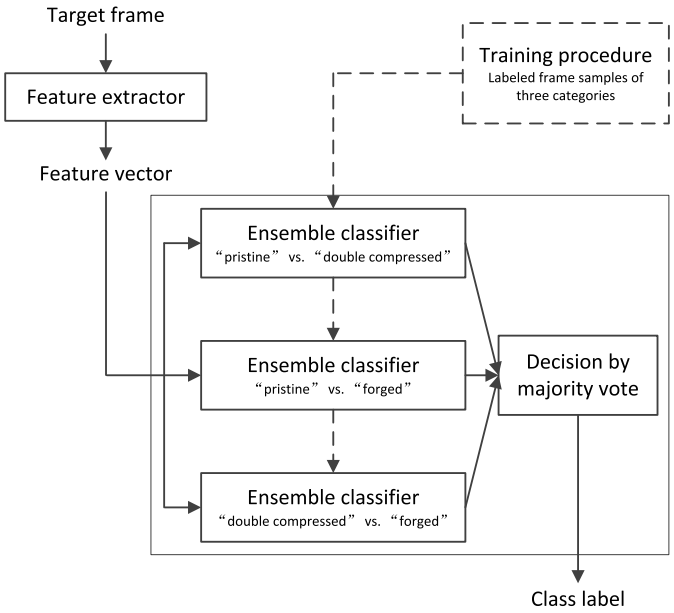


Fig. 4. Diagram of the ternary classifier used in our proposed approach.

converted into different gray levels in the resulting motion residual during the collusion procedure. Therefore, the inter-frame inherent properties can be regarded as a spatial-domain one. As a result the spatial-domain based feature sets might be useful in modeling this type of properties. Composite feature sets, like CDF and J + SRM, are supposed to be the best option owing to the fact that they can model intra-frame inherent properties via frequency-domain features and model inter-frame inherent properties via spacial-domain features simultaneously.

As mentioned in Section II-A, the observed frames can be classified into three categories: the pristine frames, the forged frames, and the double-compressed frames. Since a forged frame is also double compressed, if we are able to distinguish between pristine frames and double-compressed frames, we can reveal the nonoriginality of a target video clip and mark it as suspicious. Furthermore, if we can make a distinction between the forged frames and the double-compressed frames, we can pick out an actual object-based forged video from the suspicious ones and locate the forged segments in the forged video. In a word, the proposed frame manipulation detector should be a ternary classifier. Ensemble classifier described in [22] is adopted in our work to construct the frame manipulation detector. It is actual a random forest that consists of multiple base learners each trained on a standalone subvector of the feature vector selected uniformly at random. Fisher linear discriminants are adopted as base learners due to their simplicity. The ensemble classifier makes its decision by fusing all the decisions of individual base learners using majority voting strategy. However, since the ensemble classifier [22] was specifically designed to solve binary classification problem, the ternary classification problem in object-based forgery detection has to be converted to multiple binary classification problems. The one-versus-one strategy is adopted in our proposed approach [26]. As represented by Fig. 4, three independent ensemble classifiers are separately trained in the one-versus-one strategy. Each of them receives the samples of a pair of classes from the frame samples of three categories, namely, the pristine versus double compressed pair, the pristine versus forged pair, and the double compressed versus forged pair, and learns to distinguish between the two classes assigned to it. In the testing phase, a higher level majority voting strategy is applied: all three ensemble classifiers are applied to the feature vector of a target frame and vote on the candidate class the target frame belongs to. The class that has the most votes becomes the final decision of the combined classifier. Certainly, there is the possibility that a majority vote cannot be achieved. For instance, suppose in Fig. 4 the top, the middle, and the bottom ensemble classifiers vote on pristine, forged, and double compressed, respectively, then there is no majority vote. In such a situation our proposed frame manipulation detector will randomly guess the class the target frame belongs to. However, as revealed by the experiments in Section IV, this is a rare event.

Given a target video clip, each frame in the video clip is used to extract its corresponding motion residual. Then, a feature vector is extracted from each motion residual using an image steganalytic feature set. The frame manipulation detector takes a sequence of feature vectors as input and outputs a label sequence in which the corresponding video frames are classified as pristine, double compressed, or forged. On the basis of the label sequence, we can identify the category of the target video clip. Before dig into the detailed procedure, it is worthy of re-emphasizing that since re-encoding/recompression is indispensable in forgery, a forged video clip is also double compressed. As a result, our proposed video clip identification algorithm is a two-stage procedure in which the pristine video clips and naïve double-compressed video clips are screened out successively until only the forged video clips remain.

In the first stage, we determine a target video clip is pristine, or suspected to have undergone tampering operations. Please note that due to the unavoidable false classification, some frames in the target video clip may be assigned wrong labels by the frame manipulation detector. Therefore, a simple majority strategy is adopted to screen out the pristine video clips. A target video clip will be classified as pristine if the majority of the frames it contains (more than 50%) are classified as pristine, or else it is suspected to have undergone tampering operations. A suspicious video clip still might be actually innocent in the case that it is just a double-compressed version of the original pristine video clip and no tampering operations occurs between the decompression and recompression. Therefore, the second stage is introduced and a further classification is performed to filter away naïve double-compressed video clips from the suspects. As mentioned in Section II-A, an object-based forged video stream is usually part tampered, namely, only the frames in some segments of the corresponding video frame sequence are tampered. Each forged segment contains a certain number of successive forged frames. Isolated forged frame is unlikely to appear in real object-based forgery scenario due to the tampering in isolated frame makes no sense in perception. Hence, in the secondary classification, the existence of a certain number (set as five in our experiments) of successive frames classified as forged marks the existence of the forged segments, which in turn marks a forged video.

B. Localization of the Forged Segments

After the two-stage classification, the forged videos have been identified. The next step is to locate the forged segments. With regards to a video marked as forged, it only contains two categories of frames: those forged and those nonforged (double compressed or rarely pristine). Our extensive experiments further revealed that the frames with false predicted labels, namely, the forged frames misclassified as nonforged or the nonforged frames misclassified as forged, intensively appear around the boundaries of the actual forged segments, which implies accurately locating the boundaries of the forged segments a challenging task. Therefore, we propose a two-stage algorithm as a solution to locate the forged segments. In the first stage, the coarse boundaries of the forged segments are located, and then in the second stage, an accurate positioning algorithm is adopted to fine tune the left and right boundaries of each forged segment.

The algorithm that coarsely locates the forged segments is a recursive procedure. Let 0 indicates those nonforged frames and 1 indicates those forged frames. A video marked as forged can be expressed by the corresponding $0/1$ sequence denoted by $\mathbb{V}' \triangleq \{f^{(1)}, f^{(2)}, \ldots, f^{(N)}\}$, $f^{(i)} \in \{0, 1\}$, $N \in \mathbf{Z}$. With the consideration of false classification rate, a threshold $T_C = 90\%$ is set for the localization of coarsely forged segments. Taking the indexes of the left boundary frame and the right boundary frame of $\mathbb{V}'$ as input, the algorithm will return $\mathbb{V}'$ itself if there are more than T_C frames in $\mathbb{V}'$ belonging to forged category. Otherwise, two $0/1$ subsequences are extracted from $\mathbb{V}'$. The first subsequence contains the first three quarter frames of $\mathbb{V}'$, while the second contains the last three quarter frames. We continue to check those two subsequences to see whether they contain more than T_C forged frames or not. Stop the recursive checking procedure for a subsequence and return the subsequence itself when the threshold condition is reached, or else the first $3/4$ and the last $3/4$ frames of the subsequence will be extracted as the new subsequences. The same recursive checking procedure will be performed on the two new subsequences. In every iterations, the proportion of the forged frames in the corresponding video frame subsequence can be obtained by evaluating the quotient of the summation of $f^{(i)}$ in the subsequence divided by the length of the subsequence. After the recursion is finished, its output is a union set of the segments in each of which more than T_C frames are forged. Overlapping segments are merged by the union operation.

The accurate positioning algorithm employs the double compressed versus forged classifier that is picked out from the frame manipulation detector trained for forgery identification. The classifier makes its decision by fusing all decisions of individual base learners using majority voting strategy. However, the votes of the base learners not only just merely produce classification result but also can act as an approximate measure of the divergence between a target frame and the category it might be classified to. The more votes a certain category gets from the base learners, the more likely the present target frame belongs to that category. For K base learners in the double compressed versus forged ensemble classifier, let -1 indicates the vote received by the double-compressed category and $+1$ indicates the vote received by the forged category. Denote the vote of the ith base learner by v_i, the divergence measure based on the votes of the base learners (VD for short) are defined as:

$$\mathrm{VD} = \sum_{i=1}^{K} v_i, \quad v_i \in \{-1, +1\}. \tag{5}$$

The corresponding frame is classified as forged by the double compressed versus forged ensemble classifier if $\mathrm{VD} > 0$, otherwise classified as double compressed. But the capability of VD goes beyond that. Denote the VD of a certain frame $F^{(\cdot)}$ by $\mathrm{VD}_{F^{(\cdot)}}$. Assume that there are two frames $F^{(i)}$ and $F^{(j)}$ classified as double compressed, which means $\mathrm{VD}_{F^{(i)}} < 0$ and $\mathrm{VD}_{F^{(j)}} < 0$. However, if $|\mathrm{VD}_{F^{(i)}}| \gg |\mathrm{VD}_{F^{(j)}}|$, which means the double-compressed votes $F^{(i)}$ receives are much more than those $F^{(j)}$ receives, one immediate deduction is that the base learners decide $F^{(i)}$ is much more likely to be double-compressed frame compared with $F^{(j)}$. In other words, the probability of misclassification of $F^{(j)}$ is much higher than that of $F^{(i)}$.

VD is based on the steganalytic features fed to it, which in turns are based on the motion residual between the decompressed video frame and its colluded result. For a forged video, our investigative experiments revealed that the position of a frame in the collusion window causes less effect to its VD as long as the actual category of the video frames in the collusion window is homogeneous, namely, all the video frames in the window are forged or its opposite. However, the proportion of the two categories, double compressed and forged in the collusion window causes more effect to the VD

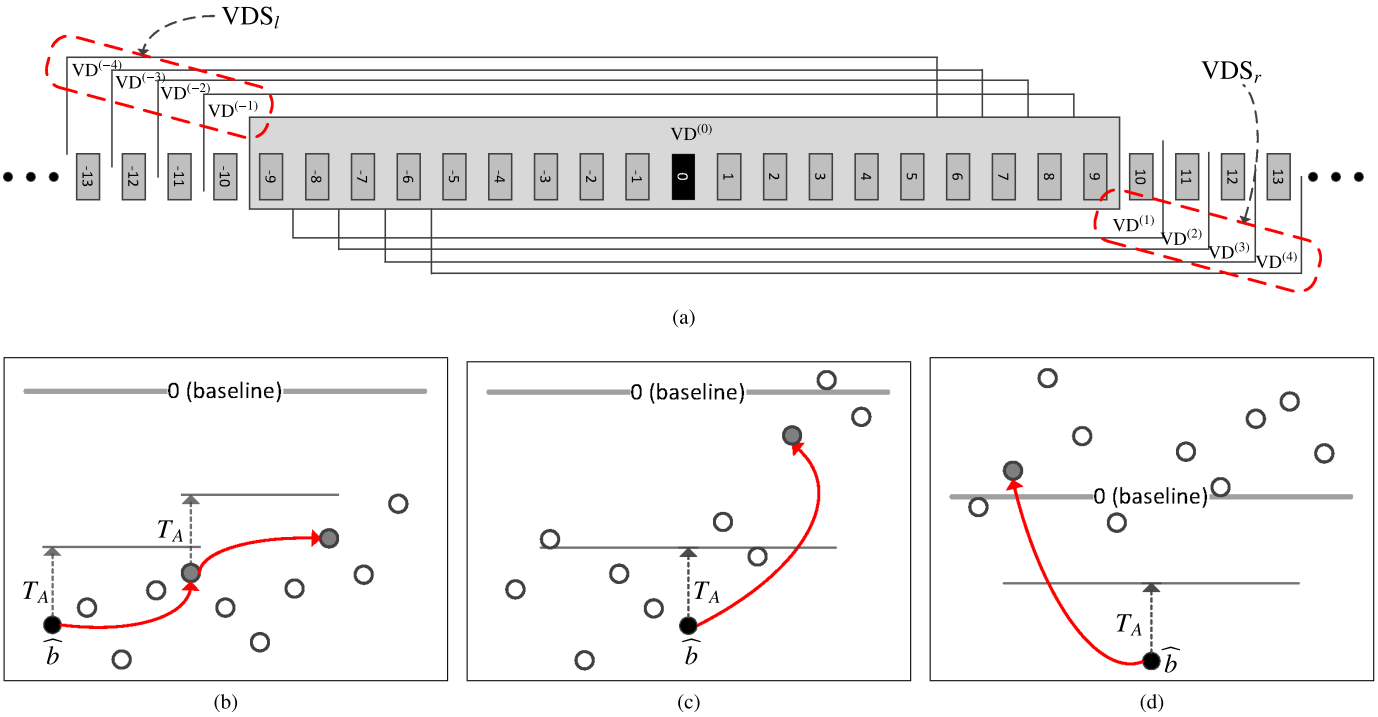


Fig. 5. (a) Diagram of a shifting collusion window in which $L_h = 9$ and $L_s = 4$. Every frame is marked with its relative temporal positions with respect to $\widehat{b}$ (the frame with 0 mark). Those contained in the central gray rectangle are the frames included in the collusion window for $\mathrm{VD}^{(0)}$. Other shifted collusion windows are marked by cascaded transparent rectangles. (b) Diagram of step 2 of the proposed accurate positioning algorithm. The gray horizontal line denotes zero, the baseline of VD. The black circle denotes the current index of $\widehat{b}$ and the gray circles denote the target index $\widehat{b}$ changed in an individual iteration. (c) Diagram of step 3 of the proposed accurate positioning algorithm. (d) Diagram of step 4 of the proposed accurate positioning algorithm.

of a video frame. When the length of the collusion window is fixed, the one-frame window shifting operation that brings in a forged frame and throw away a double-compressed frame at the same time will make the value of the VD change significantly in the direction of the positive axis. Otherwise, double compressed in and forged out will make the value of that VD change significantly in the opposite direction. Based on the above analysis, the accurate positioning algorithm is detailed as follows.

1) Denote the M coarsely located forged segments, namely, the output of the recursive procedure of the first stage, by the indexes of the boundary frames, $\{(lb_i, rb_i)\}$, $i = 1, \ldots, M$. Denote the shift length of collusion window by $L_s = \lfloor L_h/2 \rfloor$, where $\lfloor \cdot \rfloor$ rounds down the numeral to nearest integer. Traverse $\{(lb_i, rb_i)\}$, $i = 1, \ldots, M$. Let $\widehat{b}$ denotes either boundary index of a given (lb_i, rb_i), the collusion window is shifted from $[\widehat{b} - L_h - L_s, \widehat{b} + L_h - L_s]$ to $[\widehat{b} - L_h + L_s, \widehat{b} + L_h + L_s]$ and the VDs generated from the corresponding colluded results are recorded as $\{\mathrm{VD}^{(k)}\}, k = -L_s, \ldots, L_s$.
2) If the shifting of the collusion window causes little effect to its VD, then the new frame brought in should not belong to opposite category. Let $\mathrm{VD}^{(0)}$ corresponds to the base collusion window in which $\widehat{b}$ is at the center. $\mathrm{VDS}_l = \{\mathrm{VD}^{(k)}\}, k = -1, \ldots, -L_s$ comes from a continually left-shifting collusion window, while $\mathrm{VDS}_r = \{\mathrm{VD}^{(k)}\}, k = 1, \ldots, L_s$ is from a right-shifting collusion window. An illustration of a shifting collusion window is shown in Fig. 5(a), from which we can see the collusion windows for every colluded results in $\{\mathrm{VD}^{(k)}\}$. The colluded results contained in VDS_l or VDS_r are marked as well. For each coarsely located forged segment (lb_i, rb_i), if $\widehat{b}$ is equal to its left boundary lb_i, then the centroid of that forged segment is located at its right-hand side. Otherwise, the centroid is at the left-hand side of $\widehat{b}$ if $\widehat{b}$ is equal to the right boundary rb_i. For simplicity, we only consider the situation that $\widehat{b} = lb_i$ in the following algorithm. What for $\widehat{b} = rb_i$ behaves similarly except that VDS_l and VDS_r exchange their positions in the expressions, and $\widehat{b}$ moves to the opposite direction in the index tuning operations. Suppose the category of the frame with index $\widehat{b}$ is double compressed, repeatedly executes $\widehat{b} := \widehat{b} + L_s - 1$ whenever the following condition is satisfied:

$$\frac{\sum_{\mathrm{VD}^{(k)} \in \mathrm{VDS}_r} \mathrm{VD}^{(k)}}{L_s} - \mathrm{VD}^{(0)} \leq T_A \tag{6}$$

as demonstrated in Fig. 5(b). Otherwise, suppose that frame is forged, repeatedly executes $\widehat{b} := \widehat{b} - L_s + 1$ whenever the following condition is satisfied:

$$\mathrm{VD}^{(0)} - \frac{\sum_{\mathrm{VD}^{(k)} \in \mathrm{VDS}_l} \mathrm{VD}^{(k)}}{L_s} \leq T_A \tag{7}$$

where T_A is a predefined threshold.
3) Conversely, if the shifting of the collusion window causes huge effect to its VD, then the new frame

brought in by the shifting operation should belong to opposite category. Suppose the category of the frame with index $\widehat{b}$ is double compressed, and the following two conditions are satisfied:

$$\begin{cases} \dfrac{\sum_{\mathrm{VD}^{(k)}\in\mathrm{VDS}_r}\mathrm{VD}^{(k)}}{L_s}-\mathrm{VD}^{(0)} > T_A \\ \dfrac{\sum_{\mathrm{VD}^{(k)}\in\mathrm{VDS}_l}\mathrm{VD}^{(k)}}{L_s}-\mathrm{VD}^{(0)} \le T_A. \end{cases} \tag{8}$$

The final position of the left boundary of the corresponding forged segment is equal to $\widehat{b}+\widehat{k}$, where $\widehat{k}$ is determined by

$$\widehat{k}=\underset{1\le k\le L_s}{\operatorname{argmax}}\{\mathrm{VD}^{(k)}-\mathrm{VD}^{(k-1)}\} \tag{9}$$

where $\mathrm{VD}^{(k)} \in \mathrm{VDS}_r \cup \{\mathrm{VD}^{(0)}\}$, as demonstrated in Fig. 5(c). Otherwise, suppose the category of the frame with index $\widehat{b}$ is forged, and the following two conditions are satisfied:

$$\begin{cases} \mathrm{VD}^{(0)}-\dfrac{\sum_{\mathrm{VD}^{(k)}\in\mathrm{VDS}_l}\mathrm{VD}^{(k)}}{L_s} > T_A \\ \mathrm{VD}^{(0)}-\dfrac{\sum_{\mathrm{VD}^{(k)}\in\mathrm{VDS}_r}\mathrm{VD}^{(k)}}{L_s} \le T_A. \end{cases} \tag{10}$$

The final position of the left boundary is equal to $\widehat{b}-\widehat{k}$, where $\widehat{k}$ is determined by

$$\widehat{k}=\underset{-L_s\le k\le -1}{\operatorname{argmin}}\{\mathrm{VD}^{(k)}-\mathrm{VD}^{(k+1)}\} \tag{11}$$

where $\mathrm{VD}^{(k)} \in \mathrm{VDS}_l \cup \{\mathrm{VD}^{(0)}\}$.

4) In some rare cases, boundaries with isolated VDs have to be handled. Suppose the category of the frame with index $\widehat{b}$ is double compressed, and the following two conditions are satisfied:

$$\begin{cases} \dfrac{\sum_{\mathrm{VD}^{(k)}\in\mathrm{VDS}_r}\mathrm{VD}^{(k)}}{L_s}-\mathrm{VD}^{(0)} > T_A \\ \dfrac{\sum_{\mathrm{VD}^{(k)}\in\mathrm{VDS}_l}\mathrm{VD}^{(k)}}{L_s}-\mathrm{VD}^{(0)} > T_A. \end{cases} \tag{12}$$

$\widehat{b} := \widehat{b}-L_s+1$ until the two conditions for double-compressed boundaries in step 3) are satisfied, as demonstrated in Fig. 5(d). Otherwise, suppose the frame with index $\widehat{b}$ is forged, and the following two conditions are satisfied:

$$\begin{cases} \mathrm{VD}^{(0)}-\dfrac{\sum_{\mathrm{VD}^{(k)}\in\mathrm{VDS}_l}\mathrm{VD}^{(k)}}{L_s} > T_A \\ \mathrm{VD}^{(0)}-\dfrac{\sum_{\mathrm{VD}^{(k)}\in\mathrm{VDS}_r}\mathrm{VD}^{(k)}}{L_s} > T_A. \end{cases} \tag{13}$$

$\widehat{b} := \widehat{b}+L_s-1$ until the two conditions for forged boundaries in step 3) are satisfied.

IV. Experiments

We test the proposed algorithm on SYSU-OBJFORG data set (will be publicly available in the near future), where all of the video clips are extracted from primitive video footages of several static commercial surveillance cameras, which are 3 Mbits/s, 1280 × 720 (720p) H.264/MPEG-4 encoded video streams with frame rate of 25 frames/s. 100 video clips are directly cut from the video footages, all of which can be regarded as pristine since they have not undergone any type of manipulation. The lengths of all the video clips are about eleven seconds. One object-based forged video clip is generated from each of the pristine video clips. Every forged video clip contains one or two forged segments that last from 1 to 5 s. The object-based forgery in those segments includes adding/erasing moving figures and changing the positions of the figures in the scene. It is guaranteed that no perceptive traces can be easily found. Then, all forged video clips are recompressed using the same parameters as those used in the corresponding pristine video clips. To verify the performance of the proposed algorithm for lower resolution videos, another video database is generated via converting every video clips in the basic object-based forged video database to their 3 Mbits/s, 640 × 360 low-resolution version. Further on, to verify the performance of the proposed algorithm for low bitrate videos, we reduce the bitrate of the video clips in the basic database by half (namely from 3 to 1.5 Mbits/s) and generate the third video database. For convenience, we refer to the three video databases as the basic database, the low-resolution database, and the low bitrate database, respectively, in the rest of this paper. Fig. 2(a) and (b) shows one of the pristine video clips and its corresponding forged version. There are totally about 11 000 pristine frames selected to undergo object-based forgery manipulations. According to the best knowledge of the authors, SYSU-OBJFORG is the largest object-based forged video database ever reported in the literature.

A. Experiment Setups

The basic, the low-resolution, and the low bitrate video databases are, respectively, used in the following experiments. 50% of the video clips are randomly selected from the pristine group. They constitute the training set along with their corresponding forged versions. The rest 50% video clips are for testing. All the experiments are repeated ten times and the average results are reported. The following criteria are used in the experiments, in which Σ stands for the number of the set elements.

Pristine frame accuracy (PFACC):

$$\frac{\Sigma\text{ Correctly classified pristine frames}}{\Sigma\text{ Pristine frames}}.$$

Forged frame accuracy (FFACC):

$$\frac{\Sigma\text{ Correctly classified forged frames}}{\Sigma\text{ Forged frames}}.$$

Double-compressed frame accuracy (DFACC):

$$\frac{\Sigma\text{ Correctly classified double-compressed frames}}{\Sigma\text{ double-compressed frames}}.$$

Frame accuracy (FACC):

$$\frac{\Sigma\text{ Correctly classified frames}}{\Sigma\text{ All the frames}}.$$

Video accuracy (VACC):

$$\frac{\Sigma\text{ Correctly classified video clips}}{\Sigma\text{ All the video clips}}.$$

Precision of forged segment localization (Precision):

$$\frac{\Sigma \text{ Forged frames with correct labels}}{\Sigma \text{ All the frames labeled as forged}}.$$

Recall of forged segment localization (Recall):

$$\frac{\Sigma \text{ Forged frames with correct labels}}{\Sigma \text{ All the forged frames}}.$$

Balanced F-score of forged segment localization (F1 score):

$$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

VACC is the only one criterion measured on video clips. Except VACC, the rest of the criteria are applied on frames. PFACC, FFACC, DFACC, and FACC are used to evaluate the performance of the proposed frame manipulation detector, while Precision, Recall, and F1 score are used to evaluate the performance of the proposed localization algorithm for forged segments. Please note that FFACC and Recall (or Precision) are two completely different criteria although they are similar in formulas. Quite a few frames in the suspicious forged video clips may be reassigned a new label during the segment localization procedure. As a result, the number of the frames labeled as forged when calculating Recall (or Precision) is always different from the number of the frames classified as forged by the frame manipulation detector in the calculation of FFACC.

B. Experimental Results

In Fig. 6, the influences of different lengths $L = 2 \times L_h + 1$ of collusion window on the two collusion operators, $\mathfrak{C}_{\text{MIN}}$ and $\mathfrak{C}_{\text{MEDIAN}}$, are analyzed. Two relatively simple feature sets, CC-PEV and SPAM are adopted in this experiment due to their fast feature extraction speed. FACC, Precision, Recall, and F1 score are used as measure in the experiment. As shown in Fig. 6(a)–(d), for CC-PEV feature set, the performance of $\mathfrak{C}_{\text{MEDIAN}}$ gets promoted steadily while that of $\mathfrak{C}_{\text{MIN}}$ get degraded along with the increase of the length of the collusion window. The satisfactory experimental results of $\mathfrak{C}_{\text{MEDIAN}}$ are achieved with $L_h = 9$, which is better than that of $\mathfrak{C}_{\text{MIN}}$ with $L_h = 5$. On the other hand, as shown in Fig. 6(e)–(h), for SPAM feature set, the increase of the length of the collusion window does not affect largely on the performance of $\mathfrak{C}_{\text{MIN}}$. Under the same condition, the performance of $\mathfrak{C}_{\text{MEDIAN}}$ is always inferior to that of $\mathfrak{C}_{\text{MIN}}$. From Fig. 6, we can also see that as a frequency-domain oriented feature set, CC-PEV is always better than SPAM. Despite the high computing complexity of $\mathfrak{C}_{\text{MEDIAN}}$ due to its sort operator, it achieves the best performance when $L_h = 9$ for CC-PEV feature set. Therefore, $\mathfrak{C}_{\text{MEDIAN}}$ with $L_h = 9$ is the default collusion operator in the following experiments.

The performance of the proposed frame manipulation detector and its three constituent independent ensemble classifiers are shown in Table I. The feature set used here is CC-PEV. The features of all the frames in the 200 training and testing video clips in the basic video databases are used as the input of the trained frame manipulation detector. The quantity of the frames belonging to a certain category is compared with the quantity of the frames with a certain label assigned by the top-level frame manipulation detector and the three bottom-level ensemble classifiers. Please note that as mentioned in Section III-A, since a forged frame is also double compressed, the forged frames are referred to as in double-compressed category when analyzing the performance of the bottom-level pristine versus double-compressed classifier. From Table I(a), we can see that the bottom-level ensemble classifiers can distinguish rather well between pristine and the rest two categories. It is relatively harder to distinguish forged frames from the innocent double-compressed frames. However, the double compressed versus forged classifier still correctly recognize 95.85% double-compressed frames and 85.32% forged frames. Since the decision of the top-level classifier is made by majority votes of the three bottom-level classifiers, it is no doubt that the proposed frame manipulation detector can achieve excellent performance, as shown in Table I(b). Further on, additional data are provided to verify that for our proposed frame manipulation detector the situation a majority vote cannot be achieved is a rare event. For all the frames (totally 58 442 frames) in the 200 video clips, only 84 frames are classified by the bottom pristine versus double-compressed classifier as pristine and at the same time classified by the bottom pristine versus forged classifier as forged. Furthermore, in those 84 frames, only 10 frames are classified by the bottom double compressed versus forged classifier as double compressed. The majority votes cannot be achieved for those ten frames. Nevertheless, the probability that such a situation arises is merely about 0.017% ($10/58\,442 \approx 0.017\%$). There are other voting combinations of the three bottom ensemble classifiers that result in no majority vote, however, our experimental results also show that none of the probability of each of those voting combinations exceeds 0.02%.

In Fig. 7, two investigative experiments are conducted on the basic 1280×720 object-based forged video database. The feature set used in the experiments is CC-PEV. For each frame in the forged videos, the collusion window is shifted in the segment $[-L_s, L_s]$ and the corresponding VDs are generated. Fig. 7(a) shows the percentage distribution of the difference between $\text{VD}^{(0)}$ and $\{\text{VD}^{(k)}\}, k = -L_s, \ldots, L_s$ for each frame under the constraint that the shift of the window does not bring in new frames belonging to opposite category, while Fig. 7(b) shows the percentage distribution of the difference between $\text{VD}^{(0)}$ and $\{\text{VD}^{(k)}\}, k = -L_s, \ldots, L_s$ for each frame in the case that the one-frame window shifting operation brings in a forged frame and throw away a double-compressed frame at the same time, and vice versa. From Fig. 7(a), we can see that the distribution achieves its peak at difference value 0 and the majority of the distribution is contained in a small segment of difference values $[-8, 8]$, which verifies that the position of a decompressed video frame in the temporal window of its corresponding collusion causes less effect to its VD. However, Fig. 7(b) reveals a completely different result. That is, the distribution has quite a wide peak with gently sloping sides. The width of the broad component of the distribution exceeds $[-8, 8]$, indicating that in this case the shift of the collusion window causes more effect to the VD of the corresponding frame. The two investigative experiments

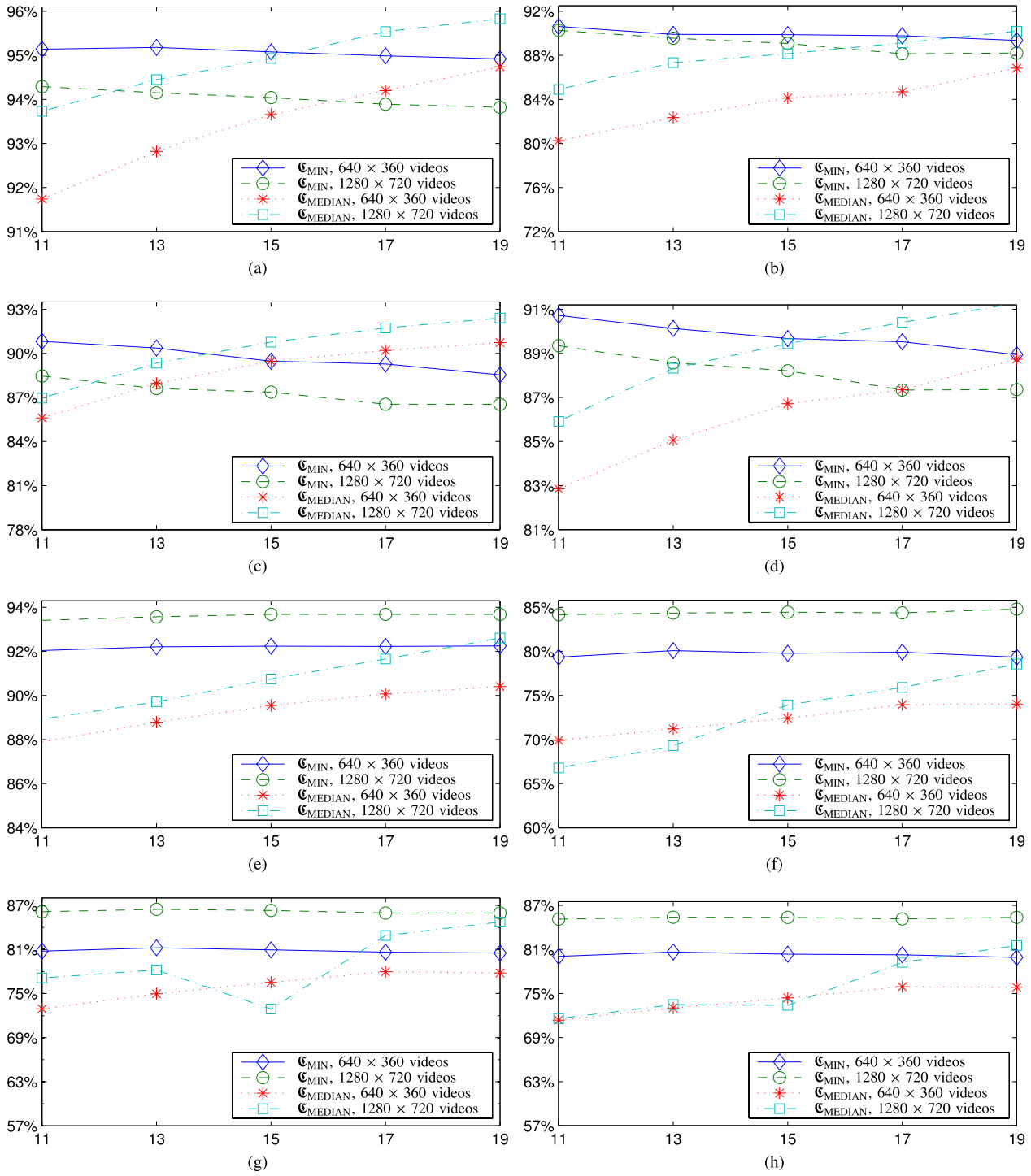


Fig. 6. Comparison of the influence of different lengths $L = 2 \times L_h + 1$ of collusion window and two collusion operators ($\mathfrak{C}_{\mathrm{MIN}}$ and $\mathfrak{C}_{\mathrm{MEDIAN}}$). In each figure, the abscissa represents L, and the ordinate represents the value of the given criterion. (a) FACC, CC-PEV feature set. (b) Precision, CC-PEV feature set. (c) Recall, CC-PEV feature set. (d) F1 score, CC-PEV feature set. (e) FACC, SPAM feature set. (f) Precision, SPAM feature set. (g) Recall, SPAM feature set. (h) F1 score, SPAM feature set.

lay an experimental base for the accurate forged segment boundary positioning algorithm. As a result, T_A is set as 8 in the following experiments.

The results of the conclusive experiments are shown in Table II, in which all of the seven steganalytic feature sets mentioned in Section III-A are adopted. In these experiments $L_h = 9$, $T_A = 8$, and $\mathfrak{C}_{\mathrm{MEDIAN}}$ is used as the collusion operator. We compare our approach with the forensic method proposed in [15] (CHEN-6D for short), the newest detection scheme of object-based forgery up to now in the literature, as far as we know. However, CHEN-6D can only be used to identify forged video clips (that is to say it simply output VACC).

TABLE I

(a) PERFORMANCE OF THE THREE CONSTITUENT INDEPENDENT ENSEMBLE CLASSIFIERS IN FIG. 4. THE FIGURES IN THE TABLE DENOTE THE PERCENTAGE OF A CERTAIN CLASS OF FRAME SAMPLES THAT GET ASSIGNED A CERTAIN LABEL BY THE CORRESPONDING CLASSIFIER. THE BEST RESULTS IN EACH COLUMN ARE BOLD. (b) PERFORMANCE OF THE FRAME MANIPULATION DETECTOR IN WHICH THE DECISION IS MADE BY MAJORITY VOTE, AS SHOWN IN FIG. 4

"pristine" vs. "double compressed"		
Assigned \ Actual	Pristine	Double compressed
Pristine	**99.90%**	0.21%
Double compressed	0.10%	**99.79%**

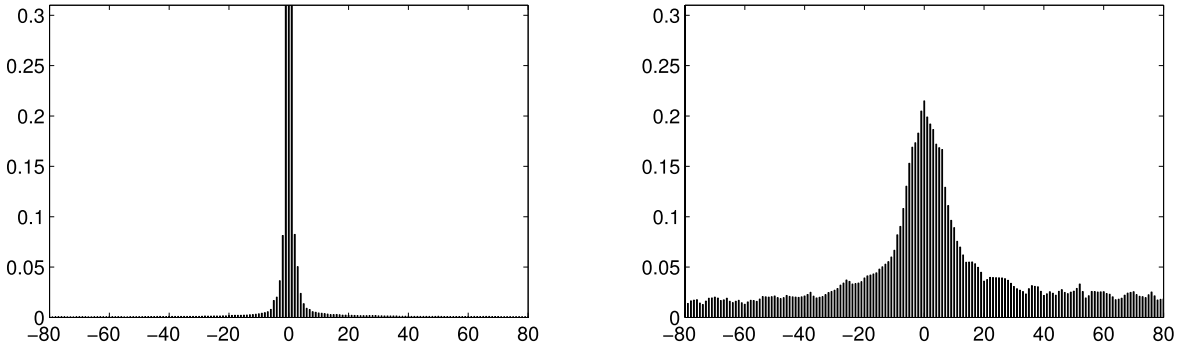
"pristine" vs. "forged"		
Assigned \ Actual	Pristine	Forged
Pristine	**99.98%**	0.13%
Forged	0.02%	**99.87%**

"double compressed" vs. "forged"		
Assigned \ Actual	Double compressed	Forged
Double compressed	**95.85%**	14.68%
Forged	4.15%	**85.32%**

(a)

Decision by majority vote			
Assigned \ Actual	Pristine	Double compressed	Forged
Pristine	**99.90%**	0.19%	0.05%
Double compressed	0.08%	**95.71%**	14.65%
Forged	0.02%	4.10%	**85.30%**

(b)



(a) (b)

Fig. 7. (a) Percentage distribution of the difference between $\mathrm{VD}^{(0)}$ and $\{\mathrm{VD}^{(k)}\}$, $k = -L_s, \ldots, L_s$ under the constraint that the shift of the window does not bring in new frames belonging to opposite category. (b) Percentage distribution of the difference between $\mathrm{VD}^{(0)}$ and $\{\mathrm{VD}^{(k)}\}$, $k = -L_s, \ldots, L_s$ in the case that the one-frame window shifting operation brings in a forged frame and throw away a double-compressed frame at the same time, and vice versa.

As a result we can only compare our proposed approach with CHEN-6D in VACC criterion.[2]

From Table II(a) and (b), we can see that the reduced resolution does not have a large effect on the performance of the proposed approach. In general, all of the seven criteria steadily rise along with the increment of the dimensionality of the feature sets adopted in the experiments. With similar dimensionality, the feature sets oriented to frequency domain are always perform better than those oriented to spatial domain. CF* is a special case. It even performs worse than CC-PEV. One reasonable explanation is that the Cartesian calibration adopted by CC-PEV and CC-JRM is a critical factor to their superior performance. Compounded feature sets, like CDF and J + SRM, perform the best with similar dimensionality in most cases, which verifies our analysis in Section III-A. However, from Table II(c), we can see that the reduction of bitrate makes greater impact on the performance of the proposed approach. The results indicate that for low bitrate videos, their inherent statistical properties sensitive to forgery are weakened during the compression procedure. However, CC-PEV, the feature set with the smallest dimensionality unexpectedly performs the best in most cases, which reveals CC-PEV is more robust to video bitrate reduction in forgery detection. The improvement in F1 score gained by the accurate positioning algorithm is non-negligible, which is 0.3%–0.5% for 3 Mbits/s 1280×720 videos, 0.4%–0.7% for 3 Mbits/s 640×360 videos, and 1.5 Mbits/s 1280×720 videos. The more advanced the steganalytic feature set is, the more minor the contribution the accurate positioning algorithm makes. This is because that those more advanced steganalytic feature sets perform better in the classification of the frames around the forged segment boundary, which in turns reduce the promotion space left for the accurate positioning algorithm. However, better gains in low resolution and low bitrate videos indicate that the accurate positioning algorithm becomes more useful for the forged segment localization in lower resolution/bitrate videos. From Table II, it can also be seen that the superior performance of our method compared with CHEN-6D is quite

[2]In CHEN-6D optimal representative frames have to be manually chosen before feature extraction. As a result for the sake of a fair comparison between our automatic framework and CHEN-6D, the procedure of manually choosing representative frames is omitted, and instead all the frames of a given video clip are used for feature extraction in the experiments with regard to CHEN-6D.

TABLE II

COMPARISON OF THE EXPERIMENTAL RESULTS OF SEVEN STEGANALYTIC FEATURE SETS. THE BEST RESULTS IN EACH COLUMN ARE BOLD. (a) EXPERIMENTAL RESULTS FOR 1280 × 720 VIDEOS. (b) EXPERIMENTAL RESULTS FOR 640 × 360 VIDEOS. (c) EXPERIMENTAL RESULTS FOR 1280 × 720 VIDEOS, BITRATE REDUCED BY HALF

	Identification					Forged segment localization			
	PFACC	FFACC	DFACC	FACC	VACC	Precision	Recall	F1 score	F1 score gained by the accurate positioning algorithm
CC-PEV	99.90%	83.94%	95.22%	95.71%	99.80%	90.48%	**91.80%**	91.13%	0.45%
SPAM	99.71%	76.86%	89.03%	92.47%	99.00%	78.90%	83.04%	80.92%	**0.48%**
CF*	99.50%	77.55%	93.64%	94.15%	99.50%	87.06%	85.87%	86.46%	0.37%
CDF	99.96%	84.07%	95.67%	95.88%	99.70%	90.20%	91.01%	90.60%	0.39%
SRM	99.92%	76.40%	93.21%	93.70%	98.40%	83.10%	82.68%	82.89%	0.28%
CC-JRM	99.96%	84.39%	**97.82%**	**96.59%**	99.90%	**93.15%**	91.51%	**92.32%**	0.34%
J+SRM	**99.99%**	**84.90%**	97.56%	**96.59%**	**100%**	92.80%	91.58%	92.18%	0.32%
CHEN-6D	—	—	—	—	76.70%	—	—	—	—

(a)

	Identification					Forged segment localization			
	PFACC	FFACC	DFACC	FACC	VACC	Precision	Recall	F1 score	F1 score gained by the accurate positioning algorithm
CC-PEV	99.63%	82.03%	93.41%	94.60%	99.70%	86.69%	90.24%	88.43%	0.67%
SPAM	99.30%	69.27%	87.12%	90.30%	96.60%	73.02%	76.01%	74.49%	**0.73%**
CF*	99.73%	73.00%	93.20%	93.04%	97.60%	83.68%	80.36%	81.99%	0.62%
CDF	99.89%	82.25%	94.29%	95.05%	99.70%	87.55%	**90.62%**	89.05%	0.55%
SRM	99.86%	70.92%	91.80%	92.24%	97.10%	79.30%	78.49%	78.89%	0.43%
CC-JRM	99.90%	**83.56%**	96.91%	96.16%	99.70%	90.01%	90.34%	90.18%	0.48%
J+SRM	**99.91%**	83.38%	**97.27%**	**96.23%**	**99.90%**	**90.28%**	90.28%	**90.28%**	0.46%
CHEN-6D	—	—	—	—	71.45%	—	—	—	—

(b)

	Identification					Forged segment localization			
	PFACC	FFACC	DFACC	FACC	VACC	Precision	Recall	F1 score	F1 score gained by the accurate positioning algorithm
CC-PEV	99.87%	**70.27%**	89.68%	91.57%	**98.40%**	78.01%	**78.68%**	**78.34%**	**0.72%**
SPAM	99.07%	57.82%	81.54%	86.43%	91.90%	61.85%	65.93%	63.82%	0.69%
CF*	99.43%	61.27%	86.62%	89.34%	92.5%	71.42%	73.26%	72.33%	0.65%
CDF	**99.92%**	69.78%	89.87%	91.56%	97.20%	75.94%	77.61%	76.76%	0.61%
SRM	99.53%	51.86%	89.01%	88.23%	91.67%	64.53%	60.43%	62.41%	0.52%
CC-JRM	99.60%	62.91%	**95.84%**	**92.50%**	93.75%	**79.34%**	70.89%	74.88%	0.48%
J+SRM	99.61%	63.15%	94.99%	92.28%	95.83%	78.50%	71.01%	74.57%	0.51%
CHEN-6D	—	—	—	—	65.32%	—	—	—	—

(c)

apparent in identification of forged video clips, especially for low-resolution and low bitrate videos.

V. CONCLUSION

In this paper, we developed an approach for automatic identification and forged segment localization of object-based forged video that is encoded with advanced frameworks. The major contributions of this paper are as follows.

1) By analyzing the similarity between the object-based forgery and steganography, we have converted the detection of object-based forgery in a video clip into the detection of hidden data in the motion residuals of the corresponding video frames.
2) We have adopted state-of-the-art image steganalytic techniques to detect the alteration of the inherent properties inside the motion residuals of the video frames.
3) We have proposed a forged video detector and a two-stage automatic algorithm that can be used to accurately locate the forged video segments in the forged video. The experiments on the largest object-based forged video database in the literature show that our approach achieves excellent results.

Our future work will focus on more precise localization algorithms that can detect the actual location of forged objects in the video scene.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for providing valuable comments and suggestions. The authors would also like to thank the members of DDE Laboratory, State University of New York, Binghamton, for sharing their research codes for steganalysis on the webpage http://dde.binghamton.

edu/download/. They would also like to thank Prof. G. Yang with Hunan University, China, for permission to use his codes in their experiments.

REFERENCES

[1] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, "Vision of the unseen: Current trends and challenges in digital image and video forensics," *ACM Comput. Surv.*, vol. 43, no. 4, pp. 26–40, 2011.

[2] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, May, 2013.

[3] H. Li, W. Luo, and J. Huang, "Countering anti-JPEG compression forensics," in *Proc. 19th IEEE Int. Conf. Image Process. (ICIP)* Sep./Oct. 2012, pp. 241–244.

[4] X. Qiu, H. Li, W. Luo, and J. Huang, "A universal image forensic strategy based on steganalytic model," in *Proc. 2nd ACM Inf. Hiding Multimedia Secur. Workshop (IH&MMSec)*, 2014, pp. 165–170.

[5] F. Huang, J. Huang, and Y. Q. Shi, "Detecting double JPEG compression with the same quantization matrix," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 848–856, Dec. 2010.

[6] W. Luo, Y. Wang, and J. Huang, "Detection of quantization artifacts and its applications to transform encoder identification," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 810–815, Dec. 2010.

[7] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double quantization," in *Proc. 11th ACM Workshop Multimedia Secur. (MMSec)*, 2009, pp. 39–48.

[8] D. Liao, R. Yang, H. Liu, J. Li, and J. Huang, "Double H.264/AVC compression detection using quantized nonzero AC coefficients," *Proc. SPIE*, vol. 7880, p. 78800Q, Feb. 2011.

[9] M. C. Stamm, W. S. Lin, and K. J. R. Liu, "Temporal forensics and anti-forensics for motion compensated video," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 4, pp. 1315–1329, Aug. 2012.

[10] M. Kobayashi, T. Okabe, and Y. Sato, "Detecting forgery from static-scene video based on inconsistency in noise level functions," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 883–892, Dec. 2010.

[11] 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 Multimedia Signal Process. (MMSP)*, Oct. 2008, pp. 170–174.

[12] J. Zhang, Y. Su, and M. Zhang, "Exposing digital video forgery by ghost shadow artifact," in *Proc. 1st ACM Workshop Multimedia Forensics (MiFor)*, 2009, pp. 49–54.

[13] A. V. Subramanyam and S. Emmanuel, "Video forgery detection using HOG features and compression properties," in *Proc. IEEE 14th Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2012, pp. 89–94.

[14] V. Conotter, J. O'Brien, and H. Farid, "Exposing digital forgeries in ballistic motion," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 283–296, Feb. 2012.

[15] R. Chen, G. Yang, and N. Zhu, "Detection of object-based manipulation by the statistical features of object contour," *Forensic Sci. Int.*, vol. 236, pp. 164–169, Mar. 2014.

[16] M. Wu, W. Trappe, Z. J. Wang, and K. J. R. Liu, "Collusion-resistant fingerprinting for multimedia," *IEEE Signal Process. Mag.*, vol. 21, no. 2, pp. 15–27, Mar. 2004.

[17] K. Su, D. Kundur, and D. Hatzinakos, "Statistical invisibility for collusion-resistant digital video watermarking," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 43–51, Feb. 2005.

[18] J. Kodovský and J. Fridrich, "Calibration revisited," in *Proc. 11th ACM Workshop Multimedia Secur. (MMSec)*, 2009, pp. 63–74.

[19] T. Pevny and J. Fridrich, "Merging Markov and DCT features for multi-class JPEG steganalysis," *Proc. SPIE*, vol. 6505, pp. 301–304, Feb. 2007.

[20] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 215–224, Jun. 2010.

[21] J. Kodovský, T. Pevny, and J. Fridrich, "Modern steganalysis can detect YASS," *Proc. SPIE*, vol. 7541, p. 754102, Jan. 2010.

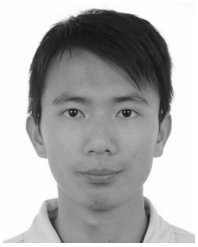
[22] J. Kodovský, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 432–444, Apr. 2012.

[23] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.

[24] J. Kodovský and J. Fridrich, "Steganalysis of JPEG images using rich models," *Proc. SPIE*, vol. 8303, p. 83030A, Feb. 2012.

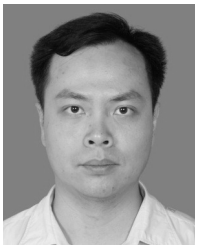
[25] J. Fridrich, J. Kodovský, V. Holub, and M. Goljan, "Breaking HUGO—The process discovery," in *Proc. 13th Inf. Hiding Workshop (IH)*, 2011, pp. 85–101.

[26] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *Ann. Statist.*, vol. 26, no. 2, pp. 451–471, 1998.



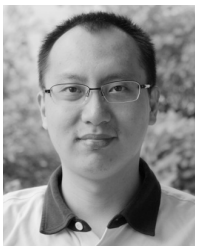
Shengda Chen received the B.S. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2013, where he is currently working toward the master's degree in computer science and technology.

His research interests include multimedia forensics and information security.



Shunquan Tan (M'10) received the B.S. degree in computational mathematics and applied software and the Ph.D. degree in computer software and theory from Sun Yat-sen University, Guangzhou, China, in 2002 and 2007, respectively.

He is currently a Lecturer with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He is also a member of the Shenzhen Key Laboratory of Media Security. His research interests include steganography, steganalysis, multimedia forensics, and deep machine learning.



Bin Li (S'07–M'09) received the B.E. degree in communication engineering and the Ph.D. degree in communication and information system from Sun Yat-sen University, Guangzhou, China, in 2004 and 2009, respectively.

He was a Visiting Scholar with New Jersey Institute of Technology, Newark, NJ, USA, from 2007 to 2008. He is currently an Associate Professor with Shenzhen University, Shenzhen, China, which he joined in 2009. He is also a member of Shenzhen Key Laboratory of Media Security. His research interests include image processing, multimedia forensics, and pattern recognition.

Jiwu Huang (M'98–SM'00) received the B.S. degree from Xidian University, Xi'an, China, in 1982; the M.S. degree from Tsinghua University, Beijing, China, in 1987; and the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences, Beijing, in 1998.

He is currently a Professor with the College of Information Engineering, Shenzhen University, Shenzhen, China, and also with Shenzhen Key Laboratory of Media Security. His research interests include multimedia forensics and security.

Dr. Huang served as an Associate Editor of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. He serves as a member of the IEEE SPS Information Forensics and Security Technical Committee.