

Article

Detection of Double-Compressed H.264/AVC Video Incorporating the Features of the String of Data Bits and Skip Macroblocks

Heng Yao ^{1,2} , Saihua Song ¹, Chuan Qin ¹, Zhenjun Tang ^{2,*} and Xiaokai Liu ¹

¹ Shanghai Key Lab of Modern Optical System, and Engineering Research Center of Optical Instrument and System, Ministry of Education, University of Shanghai for Science and Technology, Shanghai 200093, China; hyao@usst.edu.cn (H.Y.); sshusst@163.com (S.S.); qin@usst.edu.cn (C.Q.); lxkust@163.com (X.L.)

² Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, Guilin 541004, China

* Correspondence: zjtang@gxnu.edu.cn; Tel.: +86-152-9599-0968

Received: 21 October 2017; Accepted: 11 December 2017; Published: 11 December 2017

Abstract: Today's H.264/AVC coded videos have a high quality, high data-compression ratio. They also have a strong fault tolerance, better network adaptability, and have been widely applied on the Internet. With the popularity of powerful and easy-to-use video editing software, digital videos can be tampered with in various ways. Therefore, the double compression in the H.264/AVC video can be used as a first step in the study of video-tampering forensics. This paper proposes a simple, but effective, double-compression detection method that analyzes the periodic features of the string of data bits (SODBs) and the skip macroblocks (S-MBs) for all I-frames and P-frames in a double-compressed H.264/AVC video. For a given suspicious video, the SODBs and S-MBs are extracted for each frame. Both features are then incorporated to generate one enhanced feature to represent the periodic artifact of the double-compressed video. Finally, a time-domain analysis is conducted to detect the periodicity of the features. The primary Group of Pictures (GOP) size is estimated based on an exhaustive strategy. The experimental results demonstrate the efficacy of the proposed method.

Keywords: video forensics; double video compression; string of data bits; skip macroblock

1. Introduction

The rapid development of video-compression technology has made it possible to use digital video technology in many different fields (e.g., digital TV broadcasts, video conferences, digital video surveillance). With the automatic manufacturing of cheap digital devices, digital cameras are widely used in daily life. In addition, due to the implementation of high-speed Internet and the popularity of various video-sharing sites (e.g., YouTube), more and more users are uploading videos to the Internet. Digital videos may have undergone editing operations throughout the spreading process. Users may be able to enhance the quality of the video after shooting a digital video. After downloading a video from the site, the user can perform a variety of video transformations: copy a video scene, video zoom-in or zoom-out, adjustment of video color and contrast, and subtitle insertion.

Research to identify the authenticity of the multimedia (e.g., images, videos, audios) has attracted greater attention in the past few years. At first, tampering-detection research mainly focused on images, and the existing techniques for image forgery detection can be roughly grouped into three aspects: image forensic-hashing techniques [1–3], image fragile-watermarking techniques [4,5], and image passive-forensic techniques [6–8]. Although the methods using the former two techniques can provide more accurate detection accuracy, the main disadvantage of hashing and watermarking techniques

is that the hash information needs to be extracted in advance and watermark information needs to be embedded during the image-generating process. However, for passive-forensic techniques, the inspector can authenticate the image without any prior knowledge, in other words, passive-forensic methods are suitable for a wider range of practical applications. Inspired by image forgery-detection applications, the forgery-detection research has fast expanded to other media formats. To detect the tampering history of digital videos, great efforts have been invested in protecting the content of videos with digital-watermarking techniques [9]. Like image forgery detection, the main disadvantage of video-watermarking techniques is that the watermark information needs to be embedded during the video-recording process. Unlike watermarking techniques, passive video-forensic techniques identify the authentication of the videos without adding any extra information during the generation and spread processes. Therefore, video-forensic techniques have become the primary research direction of copyright protection in recent years [10,11].

In most video-tampering cases, the existing video-editing tools cannot directly operate on compressed video. As such, the process of editing a video is divided into three steps: decompress the input compressed-video sequence, edit the content of the decompressed sequence, and recompress the edited video again. In other words, the digital videos on the network are likely to have been compressed at least twice. Once a video sequence is decompressed into the pixel domain and a secondary compression is conducted, some of the primary encoding information is clearly lost. Hence, it is not possible to easily obtain the previous compression history information by analyzing the last encoded stream information. In the digital video-compression process, different coding standards (e.g., conventional MPEG-2 and MPEG-4 encoders, and the latest H.264/AVC and HEVC [12] standards) can be selected according to the application scenarios.

In all video-coding standards, a Group of Pictures (GOP) is used as the encoding unit. The GOP includes three video frame types: intra-coded frame (I-frame), predictive frame (P-frame) and bi-directionally predictive frame (B-frame). According to the difference between the primary and secondary compressed GOP structures, the research on double-compressed video detection can be grouped into two aspects: detecting double-compressed video with the same GOP structure or detecting double-compressed video with different GOP structures.

For the double compressed video with the same GOP structure, some detection methods have been proposed by using different features. Through utilizing a convex pattern in the distribution of quantized DCT coefficients, Su and Xu [13] proposed an approach to effectively detect a double MPEG-2 compression at various output bit rates. In [14], Wang and Farid utilized the time domain motion residual sequence of P-frames to detect a digital video-frame deletion or frame insertion operation. Based on the probability distribution of the first digits of the non-zero MPEG quantized alternating current (AC) coefficients, Chen and Shi [15] proposed a detection scheme to effectively detect doubly MPEG compressed videos for both the variable bit rate mode and constant bit rate mode. Combining the first digit distribution of quantized AC coefficients and the DCT coefficients, Sun et al. [16] proposed an approach with serial support vector machine (SVM) architecture to estimate the original bit rate scale in a double-compressed video. Jiang et al. [17] proposed an approach to detect double MPEG-4 compression artifacts based on Markov statistics.

For double-compressed videos with various GOP structures, most of the previously mentioned methods failed to detect this kind of forgery. The reason may be that some original I-frames were coded as P-frames in the secondary compression and that these two kinds of frames were applied to two different predictive-coding strategies. Since the fixed GOP has been used, Wang and Farid [18] detected a double compression with various GOP by using the periodicity of the average prediction residual sequence. In [19], Aghamaleki and Behrad used the quantization error of the P-frame prediction residuals to detect a double video compression and locate the deleted or inserted frames in the time domain. Stamm et al. [20] proposed an automatic double-compression detection method for spectral peak characteristics based on the average prediction residuals of the P-frames. Vazquez-Padin et al. [21] proposed a double-compression detection method based on the

variation characteristics of the macroblock prediction type in the double coded P-frames and estimated the primary GOP size of the to-be-detected videos. For a double-compressed video with intense foreground motion, He et al. [22] proposed a method to detect a double MPEG-4 compression based on a local motion vector field analysis, conducted after a static background segmentation. More recently, Bestagini et al. [23] proposed a coding footprint to identify the secondary codec and estimated the GOP size used in the first encoding process. The method in [23] is based on using the same codec and encoding parameters to recode the video sequence, which can generate similar characteristics with the former sequence.

The aforementioned double-compression detection methods are primarily based on the MPEG-4 compression standard. However, the videos coded by the H.264/AVC standard are more prevalent in practical terms. Thus, in this paper, by analyzing the time-domain periodic characteristics of H.264/AVC videos, we propose a simple, but effective, double-compressed H.264/AVC video-detection method incorporating the periodic features of the string of data bits and skip macroblocks. The rest of this paper is organized as follows. Sections 2 and 3 introduce the motivation and concrete steps of the proposed method, respectively. Section 4 presents the experimental results and Section 5 concludes the paper.

2. Periodic Artifacts for Double-Compressed H.264/AVC Videos

2.1. Periodic Artifact in the P-Frame String of Data Bits

The original video sequence contains a large amount of data that needs to be stored and transmitted. In order to compress the original video, a variety of compression algorithms are used to reduce the redundancy in the video content. As the most prevalent video-compression standard, the H.264/AVC video-coding technique reduces the redundancy in both the space and time domains through the integration of intra- and inter-prediction coding. Similar to the MPEG-2 and MPEG-4 standards, the video frames in the H.264/AVC standard are divided into several consecutive GOPs. To suppress the channel noise and the error propagation caused by the frame loss during the decoding process, each GOP is independently coded.

The GOP structure contains three types of frames: I-frame, P-frame, and B-frame. The I-frame, also known as the intra-frame coding frame, is the starting frame for each GOP. It does not rely on the other frame during the coding process. It should be noted that the H.264/AVC coding standard applied a new intra-prediction coding technology that can effectively reduce the string of the data bits (SODBs) of the I-frames. The P-frame, also known as the forward predictive coding frame, executes compression coding by using a one-way motion compensation technique. In this way, only the residuals predicted by the I-frame or P-frame, before the current coding frame and the motion compensation prediction values, are encoded. The B-frame, also known as the bi-directional predictive coding frame, has a similar encoding process to the P-frame, except for the use of a two-way motion compensation technique to generate the current encoding frame residual.

The values of the orthogonal transformed coefficients and prediction errors are quantified to represent the output values with fewer bits, during the process of video-compression coding. Once the secondary compression process is conducted using another GOP structure, the encoding style of the original I-frame may be converted to that of the P-frame. Due to the use of the different quantization steps during the various encoding types, the P-frames of the double-compressed video illustrates periodic statistical characteristics. Denote $F = \{F_t, t = 1, 2, \dots, T\}$ as the original video frame set, where T is the total frame number. Note here, we only consider the I-frame and P-frames in each GOP. Figure 1 shows the schematic of the double compression for the cases of conversion from an I-frame to a P-frame. As shown in Figure 1, during the process of the primary compression, F_{t-1} is the last frame of the $(i-1)^{\text{th}}$ GOP and it is encoded using the inter-frame coding mode. For F_t , located at the starting frame of the i^{th} GOP, the quantization process of the primary compression on the frames of F_t and F_{t-1} can be expressed as:

$$Q_t^1 = Q_{\Delta 1}[\Gamma(F_t - \hat{F}_t)] = \left\lfloor \frac{\Gamma(F_t - \hat{F}_t)}{\Delta_1} \right\rfloor, \quad (1)$$

$$Q_{t-1}^1 = Q_{\Delta 2}[\Gamma(F_{t-1} - \tilde{F}_{t-1})] = \left\lfloor \frac{\Gamma(F_{t-1} - \tilde{F}_{t-1})}{\Delta_2} \right\rfloor, \quad (2)$$

where Q_t^n stands for the n^{th} -time quantized value of the t^{th} frame, $Q_{\Delta 1}[\cdot]$ and $Q_{\Delta 2}[\cdot]$ denote the intra-frame and inter-frame quantization operations, respectively, Γ represents the transformation coding operation, superscript $\hat{\cdot}$ represents the prediction operation during the intra-frame coding process, Δ_1 and Δ_2 denote the corresponding quantization steps, respectively, where the former is greater than the latter, and superscript $\tilde{\cdot}$ stands for the inter-frame motion compensation prediction process, which is obtained from the current frame F_{t-1} and its preceding decompressed frame, denoted by \bar{F}_{t-2} .

After the single compression is conducted, F_t and F_{t-1} can be approximately restored as:

$$\bar{F}_t = \Gamma^{-1}[Q_{\Delta 1}^{-1}(Q_t^1)] + \hat{F}_t, \quad (3)$$

$$\bar{F}_{t-1} = \Gamma^{-1}[Q_{\Delta 2}^{-1}(Q_{t-1}^1)] + \tilde{F}_{t-1}. \quad (4)$$

Once the video sequence undergoes a secondary compression, \bar{F}_t and \bar{F}_{t-1} will again be treated as the input frames for the compression. As shown in Figure 1, both frames are likely to be appointed as the P-frames, to apply the inter-frame coding mode at the same time. Note that both frames are grouped into the same GOP in the secondary compression. Similar to (1) and (2), the secondary compression process for the $(t-1)^{\text{th}}$ and t^{th} frames can be expressed as:

$$\begin{aligned} Q_{t-1}^2 &= Q_{\Delta 2}[\Gamma(\bar{F}_{t-1} - \tilde{\bar{F}}_{t-1})] = \left\lfloor \frac{\Gamma(\bar{F}_{t-1} - \tilde{\bar{F}}_{t-1})}{\Delta_2} \right\rfloor \\ &= \left\lfloor \frac{\Gamma\{\Gamma^{-1}[Q_{\Delta 2}^{-1}(Q_{t-1}^1)] + \tilde{F}_{t-1} - \tilde{\bar{F}}_{t-1}\}}{\Delta_2} \right\rfloor, \\ &\cong \left\lfloor \frac{\Gamma\{\Gamma^{-1}[Q_{\Delta 2}^{-1}(Q_{t-1}^1)]\}}{\Delta_2} \right\rfloor = Q_{t-1}^1 \end{aligned} \quad (5)$$

$$\begin{aligned} Q_t^2 &= Q_{\Delta 2}[\Gamma(\bar{F}_t - \tilde{\bar{F}}_t)] = \left\lfloor \frac{\Gamma(\bar{F}_t - \tilde{\bar{F}}_t)}{\Delta_2} \right\rfloor \\ &= \left\lfloor \frac{\Gamma\{\Gamma^{-1}[Q_{\Delta 1}^{-1}(Q_t^1)] + \hat{F}_t - \tilde{\bar{F}}_t\}}{\Delta_2} \right\rfloor \\ &= \left\lfloor \frac{\Delta_1}{\Delta_2} Q_t^1 + \frac{\Gamma(\hat{F}_t - \tilde{\bar{F}}_t)}{\Delta_2} \right\rfloor \end{aligned} \quad (6)$$

Note that, for (5), during the primary and secondary compression processes, $\tilde{\bar{F}}_{t-1}$ and $\tilde{\bar{F}}_{t-1}$ both use the same inter-frame prediction mode and the same reference frame. Thus, the predicted values after their two motion compensations are approximately equal (i.e., $\tilde{\bar{F}}_{t-1} \cong \tilde{\bar{F}}_{t-1}$). According to (5) and (6), the quantized value Q_t^2 is greater than Q_{t-1}^2 . In other words, it is necessary to use a longer string of data bits to represent the t^{th} frame than the $(t-1)^{\text{th}}$ frame. Furthermore, there are two different types of double-compressed P-frames: the frame specified as the P-frame during both compression processes (referred to herein as the P-P-frame) (e.g., $(t-1)^{\text{th}}$ frame) and the frame specified as the I-frame during the first compression process, while being converted to a P-frame in the secondary compression process (referred to herein as the I-P-frame) (e.g., t^{th} frame).

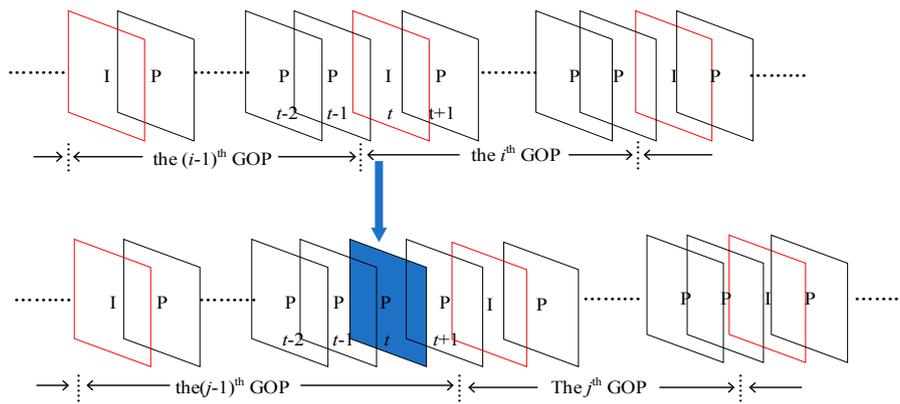


Figure 1. The schematic of the double compression for the cases of conversion from an I-frame to a P-frame.

Note that the P-P-frame applies to the same inter-frame coding mode during both compression processes. Moreover, the quantization parameters of the two compression processes are the same. For the I-P-frame, the intra-frame and inter-frame coding modes are applied during the primary and secondary compression processes, respectively. Therefore, taking F_t as an example, the time domain correlation between the reference and current frame is weak, relative to the P-P-frame. In other words, although both the $(t - 1)^{th}$ and t^{th} frames are both encoded as P-frames in the secondary compression process, more data bits are needed to represent the t^{th} frame, than the $(t - 1)^{th}$ frame, due to the weaker correlation. To verify this view, Figure 2 illustrates an example for this circumstance. Here, the YUV sequence *hall* (All standard test sequences are downloaded from <http://media.xiph.org/video/derf/>.) was doubly compressed with the first and second GOP length settings at 12 and 15, respectively. Figure 2 illustrates the corresponding data bits for each I-frame and P-frame, where the blue solid line and red dashed line represent the data bits after the primary and secondary compressions, respectively. Figure 2 also shows that for a double compression, besides the obvious peaks at the second I-frame indices, there are still some lower peaks located at the first I-frame indices, which have been encoded as a P-frame during the secondary compression.

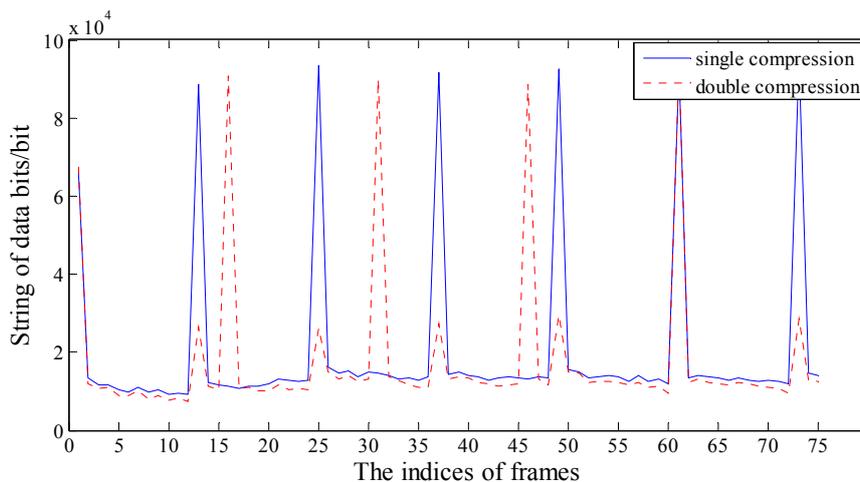


Figure 2. The comparison of a string of data bits between single and double compressions on the sequence *hall*. The sizes of the Groups of Pictures (GOPs) for the single and double compressions are 12 and 15, respectively.

2.2. Periodic Feature of Skip Macroblocks for Double-Compressed H.264/AVC Videos

In Section 2.1, we explained the periodic artifacts for the string of data bits of P-frames after a double compression with different GOP structures. This characteristic can be used as a significant clue to detect the double compression, in most circumstances. However, some videos experience dramatic changes in the video content (e.g., a test sequence *coastguard*, due to the camera moving up and down during the shooting process). Thus, the background is changed dramatically and the periodic characteristic is no longer observed. This is observed in Figure 3, where the red solid stem denotes the 73rd frame, an I-P-frame that is now submerged in its surrounding values. In this example, the 75th frame, which is a P-P-frame, has more data bits than the 73rd frame. Thus, the enhancement of the periodic artifact for the double-compressed videos with moving content is significant for double-compression forgery detection.

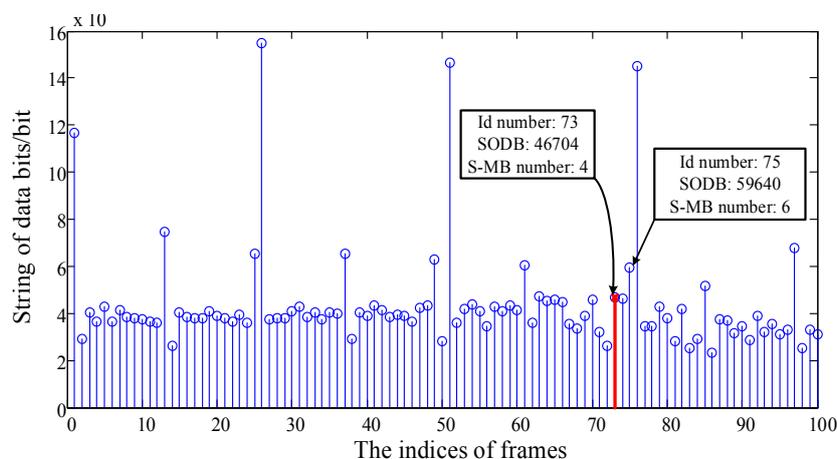


Figure 3. The string of data bits for the double-compressed test sequence of *coastguard*. The 73rd and 75th frames are an I-P-frame and P-P-frame, respectively.

To suppress the interference of the scene transformation (e.g., the 75th frame in Figure 3), we propose an I-P-frame feature enhancement method for H.264/AVC videos. During the encoding process of the H.264/AVC standard, each P-frame is permitted to have three types of macroblocks: intra-frame encoding macroblocks (I-MB), prediction-encoding macroblocks (P-MB) and skip macroblocks (S-MB). It is worth noting that S-MB is one type of macroblock that ignores macroblock compression coding; it only needs to mark the macroblock type in the frame stream. Since the I-P-frames and their surrounding P-P-frames originally belonged to different GOPs in the primary compression process, a large number of macroblock types are converted during the I-P-frame coding. More specifically, S-MBs are converted into P-MBs for prediction coding. In addition, the number of I-MBs corresponding to the more motion dramatic regions in the video frame are increased accordingly.

For ease of comprehension, Figure 4 illustrates the comparison of numbers for all three macroblock types between the P-P-frame and the I-P-frame after the double-compression process. In Figure 4, we also take the sequence *coastguard* as an example. The primary and secondary GOP sizes were set at 12 and 25, respectively. As shown in Figure 4, the 72nd, 74th and 75th frames all belong to the P-P-frame and the 73rd frame belongs to the I-P-frame. The lower panel in Figure 4 illustrates the macroblock types of the 72nd, 73rd, 74th and 75th frames in the double-compressed video, respectively. Here, the S-MB, I-MB (8×8 size), I-MB (4×4 or 16×16 size) and P-MB are denoted by yellow, orange, red and blue dots, respectively. It can be observed that the number of S-MBs for each P-P-frame is greater than that for each I-P-frame, even for some scene-switching frames (e.g., the 75th frame). For the I-P-frames (e.g., the 73rd frame in Figure 4), a large number of P-MBs are converted into I-MBs. The S-MBs are converted into P-MBs at the same time as the preceding frame. This artifact is caused by the conversion of the prediction-coding mode, from the I-frame to the P-frame.

In general, the quantization matrix and quality factor of the I-frame are different from that of the P-frame. The I-frame is successively or indirectly the reference frame, as designated by the subsequent un-coded frame (P-frame or B-frame). In addition, for the static background, if there is no residual information or motion vector, the encoder will use the S-MB to improve the coding efficiency. However, in the case of severe background changes or frequent scene switching, once the decompressed I-frame is encoded in the P-frame prediction coding mode during the secondary compression process, a slight change with respect to its reference frame is inevitably generated. Thus, the relative displacement is generated to obtain the prediction vector. The S-MB cannot be used directly, so it is converted to the P-MB for inter-frame prediction coding. In a word, the number of S-MBs in the I-P-frame is smaller than that of the other P-P-frames.

Based on the observation of the macroblock artifacts for the I-P frames, we enhance the periodic characteristics for the double compressed H.264/AVC videos by incorporating the skip macroblock features. In this paper, for the t^{th} frame, we use the ratio of the total number of macroblocks, denoted by N_t , and the number of S-MBs, denoted by S_t , to quantify this feature:

$$\varepsilon_t = \frac{N_t}{S_t} \tag{7}$$

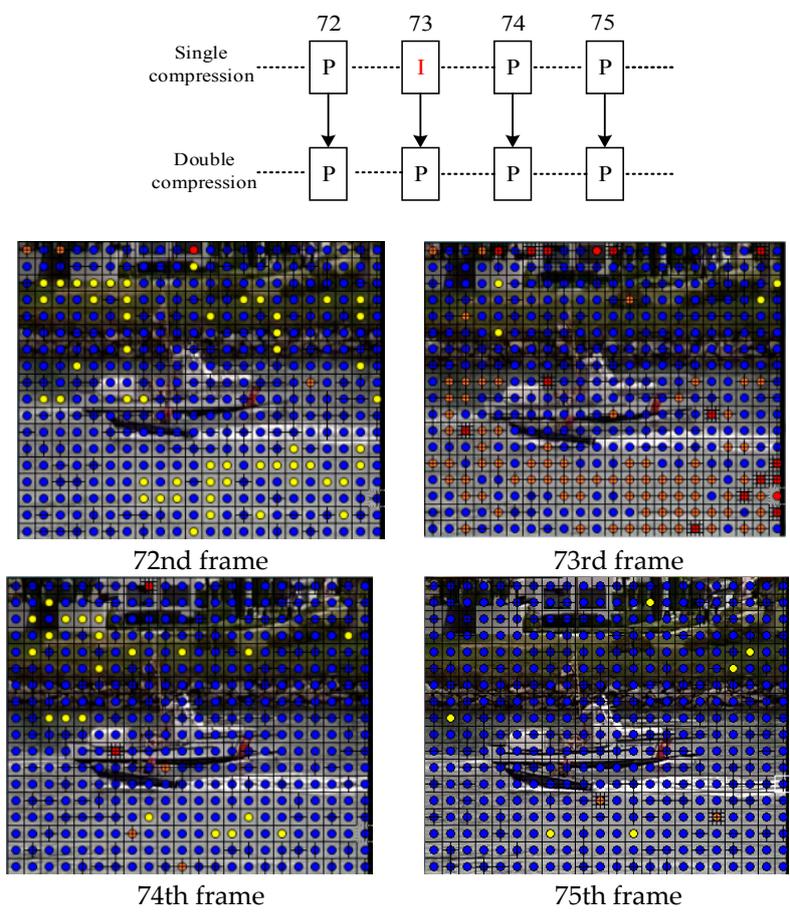


Figure 4. The macroblock-type comparison for the I-P and P-P frames. The yellow dots denote the skip macroblocks.

3. Proposed Method

The periodic artifacts for the double-compressed H.264/AVC videos have been explained in Section 2. The proposed method to detect the double-compression manipulation will be introduced

in this section. In addition, the GOP size of the primary compression can be estimated using the proposed method. Figure 5 illustrates the three-step schematic diagram of the proposed method: coding parameter extraction, feature sequence calculations, and periodic characteristic analysis. More specifically, the string of the data bits and the skip macroblock number for each I-frame and P-frame, and the GOP size of the to-be-detected video, can be extracted directly from the video stream. The conversion of the S-MBs in the video frame is calculated and the final feature sequence is generated by combining the features of the string of the data bits and the S-MBs. Finally, the feature sequence is analyzed using the time domain periodic analysis method in order to obtain the detection result and estimate the primary GOP size. Details for Steps 2 and 3 will be presented in Sections 3.1 and 3.2, respectively.

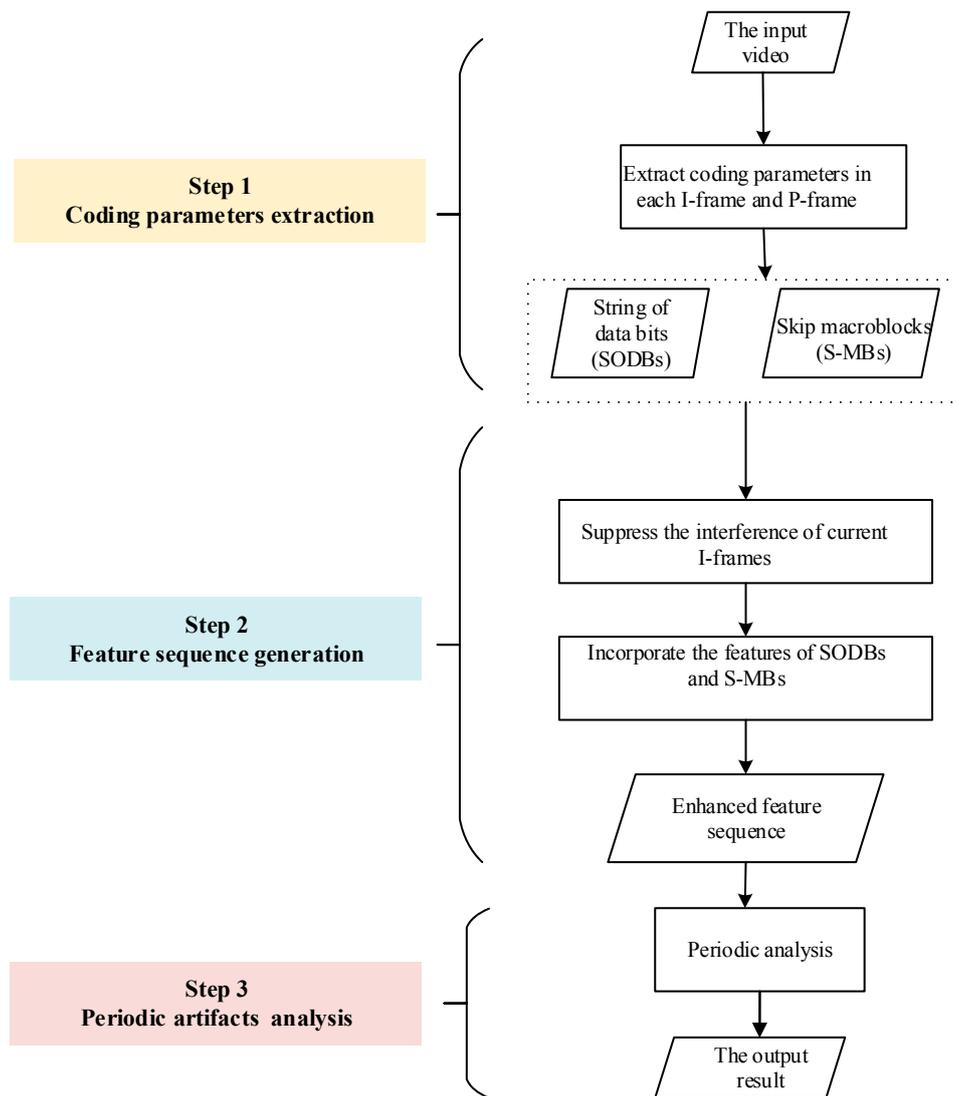


Figure 5. Schematic diagram of the proposed method.

3.1. Generation of the Feature Sequence

After extracting the SODB and the macroblock types for each frame, as demonstrated in Section 2.1, the original I-frame is likely converted to a P-frame for the inter-frame prediction coding in the secondary compression process. Therefore, for a double-compression video, the SODBs of the P-frames illustrate the periodic peaks. This attribute is used as the characteristic of the double-compression detection in the proposed method. To further expose the underlying periodic artifact, a pre-process is

conducted by replacing the value of the SODBs at the current I-frames by the average SODBs of their corresponding, preceding and next frames:

$$D_{kG_2+1} = \frac{D_{kG_2} + D_{kG_2+2}}{2} \quad k = 2, 3, 4, \dots, \frac{T}{G_2}, \quad (8)$$

where G_2 is the GOP size of the latest compression process, T is the total frame number, and D_t is the SODB of the t^{th} frame. By doing so, the interference of the extremely high SODBs of the current I-frames are suppressed. Taking the sequence *hall* as an example, as in Figure 2, a comparison of the single and double compression, after the preprocess, is shown in Figure 6. Figure 6 illustrates that the peaks in the current I-frame have been successfully suppressed.

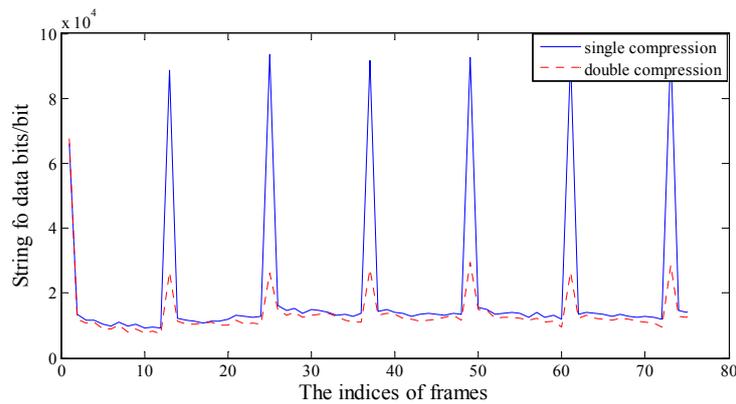


Figure 6. A comparison of the string of data bits between the single and double compression on the sequence *hall* after a key frame suppression. The sizes of the GOPs for the single and double compression are 12 and 15, respectively.

To further suppress the interference of scene switching, a P-frame skip macroblock feature is extracted, according to (7). The periodic artifact in the SODBs is then enhanced by incorporating the skip macroblock feature as:

$$E_t = \begin{cases} \varepsilon^{-1}D_t & \text{if } t = kG_2 + 1 \\ \varepsilon D_t & \text{otherwise} \end{cases}, \quad (9)$$

where E_t is the enhanced data bits feature of the t^{th} frame, incorporating the S-MB feature. At this point, we have obtained the feature sequence $\{E_t, t = 1, 2, \dots, T\}$, which will be used to expose the periodic artifact in Step 3.

3.2. Detection of Double Compression and Estimation of the Primary GOP Size

The feature sequence $\{E_t, t = 1, 2, \dots, T\}$ is used to conduct a time-domain analysis to detect the double compression and estimate the primary GOP size. A time-domain analysis method, similar to [22], is used to obtain the final result. Since the detector does not know the primary GOP size, the set containing all primary GOP size candidates is initially identified. It is reasonable to assume that the maximum value of the primary GOP size is limited and is much smaller than the length of the video. Based on this assumption, the maximum candidate value of the primary GOP size, denoted by $\hat{G}_{1 \max}$, is set at:

$$\hat{G}_{1 \max} = \min(150, \lfloor T/10 \rfloor). \quad (10)$$

Thus, the set $\hat{G}_1 = \{2, 3, \dots, \hat{G}_{1 \max}\}$ contains all probable primary GOP sizes. The value of 1 is not included in the set \hat{G}_1 , because the I-P frame does not appear, in this case. Hence, it is not realistic for a practical H.264/AVC compression with GOP size being set at 1.

The function $\Lambda(\hat{G}_1)$ is then defined to evaluate the periodic characteristics presented in the feature sequence $\{E_t, t = 1, 2, \dots, T\}$ for each candidate \hat{G}_1 in \hat{G}_1 :

$$\Lambda(\hat{G}_1) = \frac{1}{\lfloor T/\hat{G}_1 \rfloor} \sum_{j=0}^{\lfloor T/\hat{G}_1 \rfloor - 1} \sum_{i=1}^{\hat{G}_1} E_{j\hat{G}_1+i} \quad (11)$$

Considering the issue of some frame deletion, probably conducted before the secondary compression, each frame in its primary GOP has a probability that the intra-frame predictive coding is performed as a key frame in the secondary compression process. Thus, in this method, and for each \hat{G}_1 , it is assumed that each frame in \hat{G}_1 is likely to be the original I-frame and each frame is traversed to obtain the final $\Lambda(\hat{G}_1)$.

After computing the $\Lambda(\hat{G}_1)$ with all candidate \hat{G}_1 , we seek the maximum and the second value among all $\Lambda(\hat{G}_1)$. We then compare their differences with a pre-defined threshold T_Λ . It is worth pointing out that the strategy of setting T_Λ in the proposed method refers to [22], and T_Λ is sought according to the receiver operating characteristic (ROC) curve analysis on a large number of samples with the fixed parameters. If the difference is smaller than T_Λ , the video is identified as a single-compressed video; otherwise, it is identified as a double-compressed video, and its primary GOP size G_1^* can be estimated as:

$$G_1^* = \arg \max_{\hat{G}_1 \in \hat{G}_1} \Lambda(\hat{G}_1). \quad (12)$$

In order to better explain the basic flow of the proposed method, a pseudo-code implementation of double-compressed video detection and primary GOP size estimation is described in Algorithm 1, where $\max1()$ and $\max2()$ are the functions to seek the maximum and the second value among the input data set.

Algorithm 1. Pseudo code of double-compressed video detection and primary GOP size estimation.

Input: video sequence $\{F_t, t = 1, 2, \dots, T\}$, the last GOP size G_2 , and threshold T_Λ

1: Count string of data bits (SODB) of each frame F_t , denote D_t

2: Count total number of macroblocks and number of skip macroblocks (S-MB), denote N_t and S_t , respectively

3: **for** $t = 1$ to T **do**

4: $\varepsilon_t = N_t / S_t$

5: **if** $\text{mod}(t, G_2) = 1$

6: $D_t = (D_{t-1} + D_{t+1})/2$

7: $E_t = D_t / \varepsilon_t$

8: **else** $E_t = \varepsilon_t D_t$

9: **end if**

10: **end for**

11: **for** $m = 2$ to $\min(150, \lfloor T/10 \rfloor)$ **do**

12: $\Lambda_m = \frac{1}{\lfloor T/m \rfloor} \sum_{j=0}^{\lfloor T/m \rfloor - 1} \sum_{i=1}^m E_{jm+i}$

13: **end for**

14: **if** $[\max1(\Lambda_m) - \max2(\Lambda_m)] > T_\Lambda$

15: $R = 1$

16: $G_1^* = \underset{m}{\operatorname{argmax}} \Lambda_m$

17: **else** $R = 0$

18: **end if**

Output: the double-compression indicator R (1 for double compressed and 0 for single compressed), and estimated primary GOP size G_1^* in case of $R = 1$

4. Experimental Results

In this section, the efficacy of the proposed method will be demonstrated on some standard test sequences, which will be provided in Section 4.1.

4.1. Test Dataset

To verify the proposed method, we used a variety of video sequences encoded with different parameters. Table 1 lists all used encoding parameters in our dataset, including the primary compression codec c_1 , the primary video-compression bit rate R_1 , its corresponding GOP size G_1 , the secondary compression codec c_2 , the secondary compressed bit rate R_2 and its corresponding GOP size G_2 .

Table 1. List of parameter settings used to build the dataset.

Parameters	Setting Values
c_1	MPEG-2(libavcodec), MPEG-4(libavcodec), H.264/AVC
R_1	{300, 500, 700, 900, 1100}(kbps)
G_1	10, 15, 30, 40
c_2	H.264/AVC
R_2	{300, 500, 700, 900, 1100}(kbps)
G_2	9, 16, 33, 50

In the experiments, 11 CIF (resolution 352×288) YUV sequences were used as the original videos. They include: *akiyo*, *bowing*, *container*, *foreman*, *deadline*, *hall*, *coastguard*, *news*, *paris*, *sign-irene*, and *silent*. Note that all YUV video sequences in our dataset not only include fixed-camera captured video, they also include a moving camera shoot. All selected video sequences are shown in Figure 7. To reduce the computational complexity, the experimental analysis only compressed the first 300 frames of the YUV sequence. The compression process used a different bit rate and codec to generate the target. MPEG-2, MPEG-4 and H.264/AVC standard codecs were used during the primary compression process. The H.264/AVC standard codec was fixed during the second compression. The first compressed bit rate value R_1 and the second compressed bit rate value R_2 were set at {300, 500, 700, 900, 1100} kbps. The primary and secondary compression GOP size G_1 and G_2 were set at {10, 15, 30, 40} and {9, 16, 33, 50}, respectively. Therefore, for each YUV sequence, 60 singly compressed videos and 1200 doubly compressed videos can be generated, according to the different combinations of parameters.

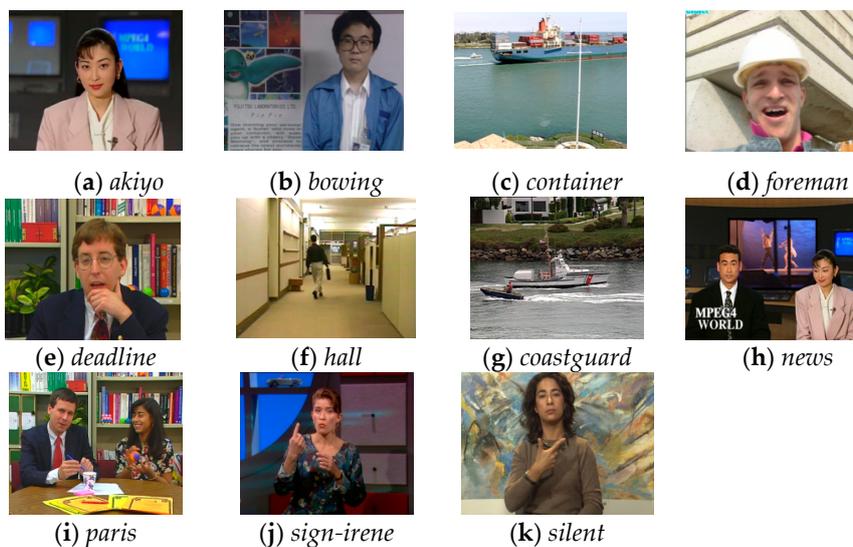


Figure 7. Screenshots of the standard test sequences used in our experiments.

4.2. Detection Results for the Double Compression

In this section, the proposed double-compression method is evaluated with different parameter settings, including bit rates and GOP size. Figure 6 illustrates an example of our detection results for the test sequence *coastguard*, where the primary GOP and secondary GOP were set at 10 and 16, respectively. Figure 8 illustrates that, by using the enhanced feature of the string of the data bits, the interference of frame 75 has been partially suppressed and the periodicity for all peak values is detected. We also prepare the proposed method with other state-of-the-art methods, including He et al. [22], Stamm et al. [20] and Vazquez-Padin et al. [21].

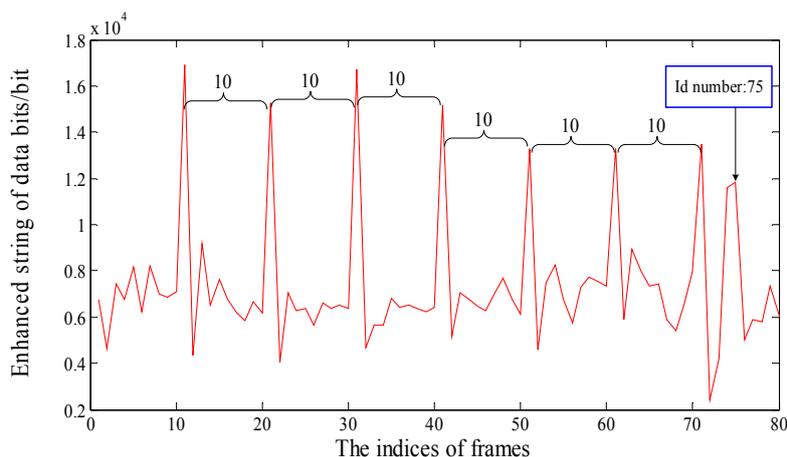


Figure 8. Enhanced feature of the string of data bits for the test sequence *coastguard*. The periodic artifact and primary GOP size are observable in the figure.

The coding bit rate is a significant parameter for video compression. Accordingly, we first evaluated our method with different bit rate settings. For a fixed setting of (R_1, R_2) , through modifying the other parameters of all test videos, we can generate $11(\text{sequence number}) \times 3(c_1 \text{ number}) \times 4(G_1 \text{ number}) = 132$ single compressed videos. These test videos were double compressed using the different setting of G_2 to generate $132 \times 4 = 528$ double-compressed sample videos. Following [22], we obtained the receiver operating characteristic (ROC) curve to seek the optimal threshold T_Λ with a fixed combination of (R_1, R_2) . The optimal detection accuracy was also obtained at the same time. Table 2 lists the detection accuracy with different bit rate settings where the best results for each combination of (R_1, R_2) are bold and underlined. Note that all of the accuracy data contains two aspects: whether the single- and double-compressed videos were correctly detected. Table 2, in most cases, shows that the method proposed in this paper has a better performance for double-compression detection than other methods in different target bit rate videos. The greater the value of the bit rate means that the quality of the compressed video is better. When $R_1 < R_2$, all methods have satisfactory accuracies. When R_1 is small, the detection accuracies of the proposed method are slightly lower than that of [22]. However, with the increase in R_1 , the proposed method exhibits slightly better performance than all of the rest of the methods. In the case of $R_1 > R_2$, the performance of all of the methods has a different degree of decline. The cause for this kind of decline for the primary compression is due to the higher bit rate, meaning a smaller quantization step. Thus, a variety of periodic artifact features used in all methods are less obvious to detect. However, our method still has higher accuracies than the other methods; this is due to the incorporation of the skip macroblock feature.

Table 2. Detection accuracy with different bit rate settings.

$R_1 \backslash R_2$	Methods	300	500	700	900	1100
300	He et al.	<u>0.997</u>	<u>0.999</u>	<u>0.999</u>	<u>0.997</u>	0.992
	Stamm et al.	0.965	0.980	0.945	0.888	0.809
	Vazquez-Padin et al.	0.875	0.991	0.997	0.996	0.999
	Proposed	0.985	<u>0.999</u>	0.998	0.992	<u>0.998</u>
500	He et al.	0.860	<u>0.987</u>	<u>0.999</u>	0.994	0.994
	Stamm et al.	0.832	0.927	0.962	0.925	0.900
	Vazquez-Padin et al.	0.489	0.922	0.940	0.976	0.968
	Proposed	<u>0.909</u>	0.981	<u>0.999</u>	<u>0.997</u>	<u>0.995</u>
700	He et al.	0.763	0.836	<u>0.983</u>	0.983	0.990
	Stamm et al.	0.727	0.793	0.873	0.900	0.892
	Vazquez-Padin et al.	0.549	0.750	0.870	0.926	0.915
	Proposed	<u>0.794</u>	<u>0.886</u>	0.979	<u>0.985</u>	<u>0.992</u>
900	He et al.	0.671	0.711	<u>0.899</u>	0.955	0.984
	Stamm et al.	0.633	0.706	0.784	0.836	0.871
	Vazquez-Padin et al.	0.540	0.605	0.732	0.885	0.852
	Proposed	<u>0.723</u>	<u>0.774</u>	0.815	<u>0.967</u>	<u>0.986</u>
1100	He et al.	<u>0.616</u>	0.620	0.776	0.833	0.916
	Stamm et al.	0.568	0.622	0.741	0.719	0.780
	Vazquez-Padin et al.	0.451	0.564	0.579	0.818	0.837
	Proposed	0.597	<u>0.677</u>	<u>0.797</u>	<u>0.864</u>	<u>0.944</u>

The detection accuracy was then evaluated with different GOP size settings. Similar to the previous experiment, for a fixed setting of (G_1, G_2) , we generate $11 \times 33 \times 5 = 165$ single-compressed videos. Based on these videos, we also generated $165 \times 5 = 825$ double-compressed videos. After an ROC curve fitting, the optimal threshold T_Λ was selected to obtain the best detection accuracy. Table 3 lists the detection accuracies with different GOP size settings. Table 3 shows that the proposed method has better performance for most cases, where best results for each combination of (G_1, G_2) are bold and underlined. Even when the primary GOP encoding size is relatively large, all detection rates are more than 0.80. The experimental results illustrate that the feature sequence $\{E_t, t = 1, 2, \dots, T\}$ generated in this method exhibits a more obvious periodicity than that of the other methods, especially in the case of videos with scene switching.

Table 3. Detection accuracy with different GOP size settings.

$G_1 \backslash G_2$	Methods	9	16	33	50
10	He et al.	<u>0.934</u>	0.922	0.928	<u>0.908</u>
	Stamm et al.	0.892	0.857	0.854	0.811
	Vazquez-Padin et al.	0.794	0.855	0.853	0.841
	Proposed	0.927	<u>0.954</u>	<u>0.932</u>	0.906
15	He et al.	<u>0.921</u>	0.916	<u>0.924</u>	0.918
	Stamm et al.	0.892	0.857	0.854	0.811
	Vazquez-Padin et al.	0.749	0.855	0.853	0.841
	Proposed	<u>0.921</u>	<u>0.918</u>	0.910	<u>0.922</u>
30	He et al.	<u>0.892</u>	0.882	0.895	0.868
	Stamm et al.	0.827	0.812	0.811	0.768
	Vazquez-Padin et al.	0.721	0.817	0.818	0.769
	Proposed	0.854	<u>0.894</u>	<u>0.914</u>	<u>0.883</u>
40	He et al.	0.885	0.804	<u>0.889</u>	0.838
	Stamm et al.	0.850	0.707	0.776	0.727
	Vazquez-Padin et al.	0.726	0.742	0.803	0.770
	Proposed	<u>0.894</u>	<u>0.824</u>	0.876	<u>0.842</u>

To further demonstrate the superiority of the proposed method, we compare the detection performance of the proposed method with He et al. [22], Stamm et al. [20] and Vazquez-padin et al. [21] by applying non-parametric statistical tests. We used the STAC platform [24] to conduct the tests of statistical significance on the previously presented experimental results, with the objective of determining if statistical differences existed among the proposed method and comparative approaches [20–22]. Specifically, a Friedman ranking test [25] with a significance level of 0.05 was first performed, and the test results are presented in Table 4, where a lower rank indicates a better performance. As shown in Table 4, the proposed method has the lowest rank of 1.439, followed by [22] with a rank of 1.744. In addition, we next verified the proposed method outperforms the other methods by means of a Holm post hoc test method [26] with an alpha level of 0.05. Table 5 shows the Holm test results, where the proposed method is set as the control method, the null hypothesis (H0) is set so that there is no difference between the control method and the comparative method, and the Holm p -value refers to the adjusted p -value of the first combination of the ranking that performs significantly worse than one of the best groups. It can be seen in Table 5 that for Stamm et al. [20] and Vazquez-Padin et al. [21], the hypothesis H0 are both rejected, in other words, the proposed method is apparently better than [20,21]. However, for He et al. [22], the hypothesis H0 is accepted, that is, the proposed method is slightly better than [22] in terms of performance, but it is not improved significantly from a statistical point of view.

Table 4. Friedman ranking test results.

Methods	Rank
Proposed	1.439
He et al.	1.744
Stamm et al.	3.390
Vazquez-Padin et al.	3.427

Table 5. Holm post hoc test results.

Control Method	Comparative Method	Holm p -Value	Result
Proposed	He et al.	0.285	H0 is accepted
Proposed	Stamm et al.	0.000	H0 is rejected
Proposed	Vazquez-Padin et al.	0.000	H0 is rejected

4.3. Performance for Primary GOP Size Estimation

In this section, we evaluate our accuracy performance for the primary GOP size estimation with different secondary bit rate settings. Here, we selected six test videos as our test sequences: *akiyo*, *bowing*, *container*, *hall*, *paris* and *coastguard*. For a fixed parameter R_2 , we generated $6 \times 3 \times 4 \times 4 = 288$ double-compressed videos. We used the threshold T_Λ sought from the first experiment and identified the authentication of all test videos. For all correctly detected videos, we further estimated the corresponding primary GOP size G_1 for each video according to (12). We defined the estimation results as:

$$d = \begin{cases} 1, & \text{if } G_1^* = G_1 \\ 0, & \text{otherwise} \end{cases} . \quad (14)$$

Table 6 lists the estimation accuracy comparison of the proposed method for the different sequences with different secondary bit rates. To further intuitively show the estimation performance of the proposed method, Figure 9 illustrate a boxplot to graphically depict the estimation accuracy of the proposed method on six standard test sequences with different settings of R_2 . The bottom and top of the box are the first and third quartiles; the band inside the box is the median accuracy; and the ends of the whiskers represent the minimum and maximum of all accuracies. Table 6 illustrates that with an increase in R_2 , the estimation accuracy is also increased. In addition, observed from Table 6

and Figure 9, all sequences have similar estimation accuracies for a fixed R_2 , except for the *hall* and *coastguard* sequences. More specifically, the *hall* or *coastguard* performances are higher or lower than that of the other sequences. The probable reason for this circumstance is that *hall* is a surveillance video with a mass of static background. The features of the string of the data bits and the skip macroblock are both more significant than that of the other sequences. For *coastguard*, due to the fast-moving camera, the features for the I-P-frame are easily disturbed by the other P-P-frame with violent content changes.

Table 6. The estimation accuracy for the double-compressed H.264 videos for different sequences and secondary bit rate settings.

Sequence R_2 (kbps)	<i>akiyo</i>	<i>bowing</i>	<i>container</i>	<i>hall</i>	<i>paris</i>	<i>coastguard</i>
300	0.732	0.715	0.701	0.823	0.708	0.684
500	0.802	0.743	0.757	0.896	0.767	0.722
700	0.885	0.816	0.861	0.920	0.858	0.771
900	0.941	0.924	0.903	0.969	0.917	0.854
1100	0.989	0.965	0.924	0.983	0.951	0.872

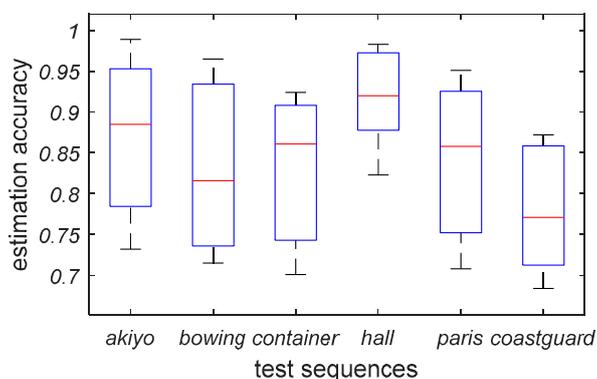


Figure 9. Boxplot of estimation accuracy of the proposed method for six standard test sequences with different settings of R_2 .

4.4. Time-Efficiency Evaluation

The implementation of the algorithm is based on the Visual Studio 2010 software development platform on the Windows operating system. It was run on a personal computer with an Inter Core i3-2310M CPU 2.10GHz processor and 4GB RAM. In our experiment, a standard test video library is generated by the FFmpeg video codec tool to create different parameters for the test videos. To evaluate the time efficiency of the proposed method, we conducted our algorithm on two standard test sequences, *akiyo* and *coastguard*, with a fixed frame number of 300 and with different primary and secondary GOP sizes, respectively. Table 7 lists their corresponding execution time. It can be seen in Table 7 that the average running time of the entire process for *akiyo* and *coastguard* are 129 ms and 197 ms, respectively; therefore, the results demonstrate the computation cost of the proposed method is satisfactory and the proposed method can be applied in practical video forgery detection. It is noteworthy that the execution time for *coastguard* is generally higher than *akiyo*. The reason for this difference is perhaps due to the fact that *akiyo* belongs to the videos with relatively static scenes and *coastguard* contains a variety of dramatically changing scenes. Thus, the execution complexity during the process of macroblock classification for scenes with dramatic changes is likely to be higher than that for static scenes.

Table 7. The execution time for the proposed method (time unit: ms).

Sequence	$G_1(G_2)$									Average
	10(9)	10(16)	10(33)	15(9)	15(16)	15(33)	30(9)	30(16)	30(33)	
<i>akiyo</i>	142	127	125	133	123	125	135	126	122	129
<i>coastguard</i>	207	196	180	207	210	172	222	199	177	197

5. Conclusions

This paper presents a simple, but effective, double-compressed H.264/AVC video-detection method. The feature of the string of the data bits for each P-frame was extracted and then incorporated with the skip macroblocks feature. Both features can directly be obtained from the H.264/AVC coding stream. A time-domain periodicity analysis was then conducted to detect the artifact of the enhanced feature of the SODBs of all I-P-frames and P-P-frames. Finally, the primary GOP size was further estimated by a traversal search. In the experiment, the validity of the proposed method was proven by constructing the video dataset, highlighting the robustness of the different coding parameters (including the bit rate, GOP size and codec) used in the primary and secondary compression processes, and the accuracy of the estimation for the primary GOP size. The contributions of this paper are mainly divided into two aspects: on the one hand, the feature of SODBs is directly extracted from H.264/AVC coding streams, and therefore, the computation cost of the proposed method is relative low and very suitable for real-time applications; and on the other hand, our method concerns the effect of primary compression on macroblock types, and thus the double-compression artifacts are further enhanced before the periodicity analysis.

Some issues remain to be resolved in future work. First, once the primary GOP size is set at an integer multiple of the secondary GOP size (e.g., $G_1 = 16$, $G_2 = 8$), the proposed method will be invalid, because there is no I-P-frame on this occasion. In addition, for inter-frame tampering video detection, the proposed method will also be invalid when the number of frames deleted or inserted is an integer multiple of the primary GOP length. These issues are the future directions of our research.

Acknowledgments: This work was supported in part by the National Natural Science Foundation of China (61702332, 61672354, 61562007), Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (MIMS16-03), and Guangxi Natural Science Foundation (2017GXNSFAA198222).

Author Contributions: Heng Yao designed the algorithm and wrote the paper; Saihua Song designed and conducted the experiments; Chuan Qin and Zhenjun Tang supervised the research work and provided modification suggestions; Xiaokai Liu analyzed the experimental results.

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

References

1. Wang, X.; Pang, K.; Zhou, X.; Zhou, Y.; Li, L.; Xue, J. A visual model-based perceptual image hash for content authentication. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1336–1349. [[CrossRef](#)]
2. Qin, C.; Chen, X.; Ye, D.; Wang, J.; Sun, X. A novel image hashing scheme with perceptual robustness using block truncation coding. *Inf. Sci.* **2016**, *361–362*, 84–99. [[CrossRef](#)]
3. Qin, C.; Chen, X.; Luo, X.; Zhang, X.; Sun, X. Perceptual image hashing via dual-cross pattern encoding and salient structure detection. *Inf. Sci.* **2018**, *423*, 284–302. [[CrossRef](#)]
4. Zhang, X.; Wang, S.; Qian, Z.; Feng, G. Reference sharing mechanism for watermark self-embedding. *IEEE Trans. Image Process.* **2011**, *20*, 485–495. [[CrossRef](#)] [[PubMed](#)]
5. Qin, C.; Ji, P.; Zhang, X.; Dong, J.; Wang, J. Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. *Signal Process.* **2017**, *138*, 280–293. [[CrossRef](#)]
6. Yao, H.; Wang, S.; Zhang, X.; Qin, C.; Wang, J. Detecting image splicing based on noise level inconsistency. *Multimed. Tools Appl.* **2017**, *76*, 12457–12479. [[CrossRef](#)]
7. Thai, T.H.; Cogramme, R.; Reira, F.; Doan, T.N.C. JPEG quantization step estimation and its applications to digital image forensics. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 123–133. [[CrossRef](#)]

8. Li, H.; Luo, W.; Qiu, X.; Huang, J. Image forgery localization via integrating tampering possibility maps. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1240–1252. [[CrossRef](#)]
9. Tew, Y.; Wong, K.S. An overview of information hiding in H.264/AVC compressed video. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 305–319. [[CrossRef](#)]
10. Bestagini, P.; Fontan, M.; Milani, S.; Barni, M.; Piva, A.; Tagliasacchi, M.; Tubaro, K.S. An overview on video forensics. In Proceedings of the IEEE 2012 20th European Signal Processing Conference, Bucharest, Romania, 27–31 August 2012; pp. 1229–1233.
11. Piva, A. An overview on image forensics. *ISRN Signal Process.* **2013**, *2013*, 1–22. [[CrossRef](#)]
12. Sullivan, G.J.; Ohm, J.; Han, W.J.; Wiegand, T. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [[CrossRef](#)]
13. Su, Y.; Xu, J. Detection of double-compression in MPEG-2 videos. In Proceedings of the International Workshop on Intelligent Systems and Applications (ISA), Wuhan, China, 22–23 May 2010; pp. 1–4.
14. Wang, W.; Farid, H. Exposing digital forgeries in video by detecting double quantization. In Proceedings of the 11th ACM Workshop on Multimedia and Security, Princeton, NJ, USA, 7–8 September 2009; pp. 39–48.
15. Chen, W.; Shi, Y.Q. Detection of double MPEG compression based on first digit statistics. In Proceedings of the International Workshop on Digital Watermarking, Newark, DE, USA, 8–10 October 2009; pp. 16–30.
16. Sun, T.; Wang, W.; Jiang, X. Exposing video forgeries by detecting MPEG double compression. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 1389–1392.
17. Jiang, X.; Wang, W.; Sun, T.; Shi, Y.Q.; Wang, S. Detection of double compression in MPEG-4 videos based on markov statistics. *IEEE Signal Process. Lett.* **2013**, *20*, 447–450. [[CrossRef](#)]
18. Wang, W.; Farid, H. Exposing digital forgeries in video by detecting double MPEG compression. In Proceedings of the 8th Workshop on Multimedia and Security, Geneva, Switzerland, 26–27 September 2006; pp. 37–47.
19. Aghamaleki, J.A.; Behrad, A. Inter-frame video forgery detection and localization using intrinsic effects of double compression on quantization errors of video coding. *Signal Process. Image Commun.* **2016**, *47*, 289–302. [[CrossRef](#)]
20. Stamm, M.C.; Lin, W.S.; Liu, K.J.R. Temporal forensics and anti-forensics for motion compensated video. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 1315–1329. [[CrossRef](#)]
21. Vazquez-Padin, D.; Fontani, M.; Bianchi, T.; Comesana, P.; Piva, A.; Barni, M. Detection of video double encoding with GOP size estimation. In Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), Tenerife, Spain, 2–5 December 2012; pp. 151–156.
22. He, P.; Jiang, X.; Sun, T.; Wang, S. Double compression detection based on local motion vector field analysis in static-background videos. *J. Vis. Commun. Image Represent.* **2016**, *35*, 55–66. [[CrossRef](#)]
23. Bestagini, P.; Milani, S.; Tagliasacchi, M.; Tubaro, S. Codec and GOP Identification in Double Compressed Videos. *IEEE Trans. Image Process.* **2016**, *25*, 2298–2310. [[CrossRef](#)] [[PubMed](#)]
24. STAC: Web Platform for Algorithms Comparison through Statistical Tests. Available online: <http://tec.citius.usc.es/> (accessed on 7 December 2017).
25. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [[CrossRef](#)]
26. Holm, S. A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* **1979**, *6*, 65–70.

