Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8848553, 11 pages
https://doi.org/10.1155/2020/8848553

WILEY | Hindawi

Research Article

A Semi-Fragile Video Watermarking Algorithm Based on H.264/AVC

Chen Li, Yi Yang, Kai Liu, and Lihua Tian

School of Software Engineering, Xi'an Jiaotong University, Xi'an, China

Correspondence should be addressed to Lihua Tian; lhtian@xjtu.edu.cn

Received 27 March 2020; Revised 8 May 2020; Accepted 26 May 2020; Published 20 June 2020

Academic Editor: Huimin Lu

With the increasing application of advanced video coding (H.264/AVC) in the multimedia field, a great significance to research in video watermarking based on this video compression standard has been established. We propose a semifragile video watermarking algorithm, which can simultaneously implement frame attack and video tamper detection, herein. In this paper, the frame number is selected as the watermark information, and the relationship of the discrete cosine transform (DCT) nonzero coefficients is used as the authentication code. The 4×4 subblocks, whose DCT nonzero coefficients are sufficiently complex, are selected to embed the watermark. The parities of these nonzero coefficients in the medium frequency are modulated to embed watermarks. The experimental results show that the visual quality of the embedded watermarked video is virtually unaffected, and the algorithm exhibits good robustness. Furthermore, the algorithm can correctly implement frame attack and video tamper detection.

1. Introduction

The rapid development of internet and multimedia technology has brought about the increasing popularity of digital videos, such as network TVs, online videos, and mobile videos [1–5]. These video applications greatly enrich people's lives and engender ample convenience. However, the proliferation of videos and video-sharing also causes a variety of serious problems. With the help of various video editing tools, people can easily tamper and edit public video content illegally, as well as copy and spread it freely. However, these behaviors endanger the legitimate rights of the copyright owners. Recently, several major video hosting portals, such as Youku and Tudou, have been involved in copyright disputes, causing enormous losses to media manufacturers and restricting the application of digital multimedia. To protect the copyright ownership of original videos or to provide the content authentication of original works has become an urgent problem to grapple with. As an effective method to resolve the problem of copyright protection and content authentication of multimedia, digital watermarking technology has become a research focus in the field of information security [6–8]. The basic idea of video watermarking is to add some extra information to the original video without affecting its visibility, which can provide the evidence of copyright or integrity authentication for the video content when necessary.

With the acquirement of video quality, video sizes are becoming progressively larger. Due to their storage capacity and bandwidth, nearly all digital videos are transmitted with compression coding on the Internet or other transmission channels. Thus, video watermarking algorithms combining video compression standards are more applicable. As a new generation of video coding standard, the application of the advanced video coding (H.264/AVC) is widespread because of its higher compression efficiency and better network affinity. Presently, research on video watermarking algorithms based on H.264/AVC is becoming increasingly more active.

Usually, video watermarking is categorized into three types, namely, the fragile, robust, and semifragile watermarks. The fragile watermarking scheme is used to validate integrity authentication, and thus, it should be sensitive for all video manipulations with good transparency and large watermark capacity. Robust watermarking needs to resist most common video processing activities, such as recompression and filtering, and perhaps presents a greater sacrifice

on transparency and watermark capacity. It is mainly applied for copyright protection. Semifragile watermarking is insensitive to common video processing operations, but it is sensitive to malicious attacks, and thus, it is mainly applied in tamper detection.

These different types of watermarking schemes based on (H.264/AVC) have developed in varying degrees in recent years, i.e., all types of watermarking schemes based on H.264/AVC have undergone considerable development. More researchers are beginning to concentrate on the semifragile video watermarking method because of its application for tampering detection [9–14]. Among them, the proposed algorithms in [9–11] can detect and locate tampering with different precision levels. Farfoura [10] provides a semifragile watermarking scheme for H.264/AVC. The scheme has a certain robustness for content-preserving manipulations and is sensitive to manipulations with content-changing. The algorithm utilizes dual watermarking. One watermark is embedded into I-frames to detect spatial tampering, and the other watermark is embedded into P-frames to identify temporal attack. Xu et al. [13] proposed a novel semifragile watermark, which generates feature codes through block energy and then embeds watermarks by using discrete cosine transform (DCT) coefficients on the diagonal, which has little effect on video quality. Cedillo [14] calculated the variance of the original pixel and the reference pixel in the 4×4 blocks and selected the subblock with the smallest variance as the block to be embedded, and then hid the watermark content according to the coding characteristics of the prediction mode value. The algorithm has good transparency, but it needs to calculate the variance value of 4×4 blocks, which has a large computational complexity. In [9], a semifragile video watermarking algorithm is proposed. In the context-adaptive variable-length coding (CAVLC) entropy encoding process of luminance 4×4 block DCT coefficients, the watermark is embedded by modifying the last nonzero coefficient after zig-zag scanning. However, the algorithm is more complicated in design. In [15], a chaotic semifragile watermarking algorithm for copyright authentication is proposed. The time information of the video frame is used as a parameter combined with the modulation of the chaotic algorithm to be embedded in the video, thereby generating embedded watermark information and embedding it into the DCT coefficient. During the parsing process, mismatch of time information can be used to reveal tampering in the time domain. The disadvantage is that it is an uncompressed domain algorithm and cannot be applied to H.264/AVC. Facciol and Farrugia [16] proposed a reversible watermarking algorithm that applies differential extension rules to 4×4 DCT block quantized coefficients. The algorithm firstly performs 16 quantized 4 s based on zig-zag scanning at the encoding end. The 4×4 DCT coefficients are divided into eight groups, and it is determined whether or not each set of coefficients can be embedded in the watermark, and then the watermark is embedded therein.

Combining the advantages and disadvantages of the abovementioned existing algorithms, we herein propose an algorithm for simultaneously realizing video time domain and spatial domain integrity authentication. In our algorithm, the frame number is utilized as the watermark information, and the numerical relationship of the DCT coefficients is adopted as the authentication code. The watermark is embedded by changing the DCT coefficients of the luminance subblocks with more nonzero coefficients. Finally, the integrity of the video is detected by the watermark information, and the tamper detection is implemented by the authentication code.

This paper is organized as follows: in Section 2, the generation of watermarking and authentication code is given first, and then two parts of the scheme are briefly described in Section 3. Experimental results are given in Section 4, and the conclusions are drawn in the last section.

2. Generation of Watermark and Authentication Code

2.1. Generation of Watermark. To identify and verify a frame attack, one idea is to embed the frame number as watermark information into the current frame, such that the watermark information extracted from the video represents the frame number information of the video frame. If the extracted watermark can match the frame number, it indicates that the video is not subject to frame attack. Otherwise, the frame attack can be determined according to the relationship between the specific watermark and the frame number.

Here, we propose a new semifragile video watermarking algorithm, which is intended to encode the number of video frames into a binary sequence, and embed this binary sequence as watermark information into the current frame. After the watermark is extracted, the watermark sequence is decoded to a decimal number again. Finally, this value is compared with the number of the video frame to verify and identify the frame attack.

When the video is recompressed, the watermark extraction would be misplaced due to the transition of the prediction mode. A large number of experimental results prove that the higher the texture complexity of the video, the better the robustness of the watermark. To improve the robustness of the watermark, we choose to embed the watermark into the video region with a complex texture. To improve the correctness of frame number restoration, the watermark sequence is divided into three segments and embedded in the video in a loop. The specific watermark information scheme is as follows:

Step 1: combine the number of frames of a common video; we choose to encode the number value of the video frame into an 18-bit binary sequence. For example, if the frame number is 10, the binary sequence converted is 000000000000001010. The sequence does not have to be 18-bit, this can be decided according to the actual situation.

Step 2: divide the binary sequence into three groups. The first six values are the first group, the middle six values are the second group, and the last six values are the third group. We will embed these three sets of binary sequences as the watermark values to the frame. For example, if the frame number is 10, the binary sequence is 000000000000001010, the first set of sequences is 000000, the second set of sequences is 000000, and the third set of sequences is 001010.

Step 3: the watermark information sequence is handled by Arnold scrambles to obtain a secure watermark sequence. Finally, the watermark sequence should be embedded in the video repeatedly.

2.2. Generation of Authentication Code. The attack identification and verification of the frame can be implemented by the watermark information, and the tamper detection of the video requires the participation of the authentication code. The type of attack that the video is subjected to is usually determined based on the degree of change of the authentication code. Therefore, the authentication code needs to be insensitive to conventional attacks but sensitive to malicious attacks.

The main goal of major video tampering is to alter the interesting targets, which normally have complicated textures rather than the background, which is the flat region [17]. Therefore, we generate the authentication code from the region with complex texture region according to the DCT coefficients. In our algorithm, we select the numerical relationship of the video DCT coefficients as the authentication code, and the experimental results show that the authentication code is not sensitive to conventional attacks but to malicious attacks. The algorithm chooses to embed a watermark on the last nonzero coefficient. Therefore, the numerical relationship between the second last nonzero coefficient and the third last nonzero coefficient is specifically selected as the authentication code of the algorithm. The specific authentication code scheme is as follows:

Step 1: traverse each macroblock of each frame of video to determine the prediction mode of the macroblock. If the prediction mode of the macroblock is 16×16, skip it. If the prediction mode of the macroblock is 4×4, the execution continues.

Step 2: determine whether the macroblock includes six or more subblocks, which have at least three nonzero coefficients, if not, skip it. Otherwise, the execution continues.

Step 3: for the macroblock that satisfies the abovestated conditions, six subblocks with the most nonzero coefficients are selected, and the authentication code R is extracted in the order of the number of nonzero coefficients. Each subblock extracts a 1-bit authentication code, and each macroblock extracts a 6-bit authentication code. The authentication information is generated by comparing the relationship between the second last nonzero coefficient, $m1$, and the third last nonzero coefficient, $m2$, in each subblock, and the authentication codes are saved. The specific scheme is shown in Equation (1).

$$R = \begin{cases} 1 & \text{if } m1 \geq m2 \\ 0 & \text{else} \end{cases} \tag{1}$$

Step 4: after the aforementioned steps, we would get the authentication code array A[i]($i = 1, \cdots, M$, M is the sum of the authentication codes) and the authentication code flag array flag [j], corresponding to each 4×4 macroblock ($j = 1, \cdots, P$, P is the sum of all 4×4 macroblocks). Meanwhile, the watermarking image, including logo information are scrambled and converted into a binary sequence, W, through dimension reduction. Then, the length of the sequence, W, is calculated. Perform exclusive or operations on each element of array A and the corresponding element of array W in sequence, and the result is saved in array B.

3. Watermark Embedding and Extraction Algorithm

3.1. Watermark Embedding Algorithm. Different from robust video watermarking, a semifragile video watermark is often derived from the characteristics of the video itself. To avoid misplacing the extracted watermark, the macroblock that does not satisfy the certain condition can also be embedded with a watermark with a value of -1 to achieve the purpose of "placeholder." We choose to modify the parity of nonzero coefficients in a complex 4×4 macroblock to embed the watermark. The specific watermark embedding scheme is as shown in Figure 1, and the detailed steps are as follows:

Step 1: the watermark binary sequence $M = \{wm1, wm2, \cdots, wm18\}$ is equally divided into three sets of binary sequences.

Step 2: traverse all the macroblocks in the current frame, and select a macroblock that satisfies the following condition to perform the embedding of the binary watermark. The complex 4×4 macroblocks include six or more subblocks, which have at least three nonzero coefficients. To avoid nonsynchronization of the extracted watermark, all other macroblocks in the frame that do not satisfy the abovementioned conditions are embedded into six watermark information units with a value of -1, and the video content is not modified.

Step 3: embed the watermark in the macroblock that meets the requirements of Step 2. Next, embed six binary watermarks in each luma macroblock, and select the watermark sequence value to be embedded, according to the remainder of the macroblock number divided by three.

Step 4: for the macroblock that satisfies the abovestated conditions, select six subblocks with the most nonzero coefficients, and embed the watermark in the order of the number of nonzero coefficients.

Step 5: for each filtered subblock, square the sum of the second last nonzero coefficient, $m1$, and the third last nonzero coefficient, $m2$, in each subblock. The watermark is embedded by modifying the last nonzero coefficient to satisfy the identical parity of the sum and the watermark value. The specific modification approach is as follows:

if $wm = 0$ and $LNZ^2 > sum$:

$$LNZ = \begin{cases} \lfloor \sqrt{sum} \rfloor & LNZ \geq 0 \\ -\lceil \sqrt{sum} \rceil & LNZ < 0 \end{cases} \tag{2}$$

if $wm = 1$ and $LNZ^2 \leq sum$:

$$LNZ = \begin{cases} \lfloor \sqrt{sum} \rfloor & LNZ \geq 0 \\ -\lceil \sqrt{sum} \rceil & LNZ < 0 \end{cases} \tag{3}$$

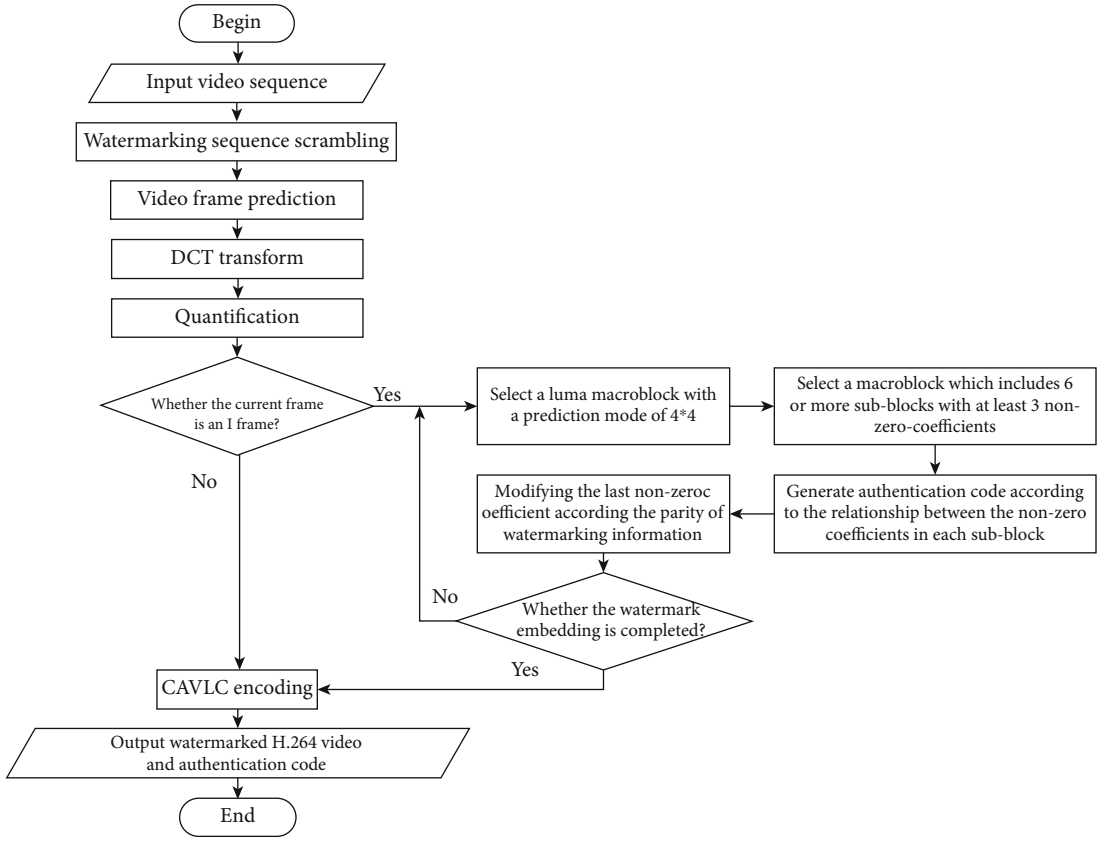

Figure 1: Flow chat of watermark embedding.

where *LNZ* is the last nonzero coefficient in *NNZ*, *sum* is the sum of all nonzero coefficients, *wm* is the embedded watermark, and $\lfloor\rfloor, \lceil\rceil$ represent rounding up and rounding down.

3.2. Watermark Extraction Algorithm and Authentication Code Judgment. The process of watermark extraction is equivalent to the inverse process of watermark embedding. The extracted watermark information contains the information of the frame number, so the frame number needs to be restored. The specific watermark extraction scheme is shown as Figure 2, and the detailed steps are as follows:

Step 1: traverse all the macroblocks in the current frame, and select a macroblock that satisfies the following condition to perform the embedding of the binary watermark. The prediction mode is 4×4 and includes six or more subblocks, which have at least three nonzero coefficients. For all macroblocks in the frame that do not satisfy the abovementioned conditions, extract six watermarks with a value of -1.

Step 2: if the macroblock satisfies the aforementioned conditions, the six subblocks with the most nonzero coefficients are selected, and the watermark is extracted according to the order of the number of nonzero coefficients. If the square of the last nonzero coefficient is larger than the square sum of the second last and third last nonzero coefficients, then the corresponding watermark is one. If the square of the last nonzero coefficient is less than or equal to the square sum of the second last and third last nonzero coefficients, then the corresponding watermark value is zero. At the same time, the authentication code is restored.

Step 3: after extracting all the watermarks, the watermark sequences are grouped and classified. The watermark sequence should be classified into three groups. If the macroblock number is divided by three and the remainder is one, the watermark sequence corresponding to the macroblock belongs to the first group, and the other types are the same. The most frequently occurring watermark sequence is the correct result. The three sets of sequence values are then integrated into an 18-bit binary sequence that is converted to a decimal number, which is the number of the current frame.

Here, we should first determine whether the whole video has been maliciously attacked by the extracted watermark information. If the watermark information is completely correct, we assume that the video has not been tampered maliciously. Otherwise, the correct frame number cannot be extracted from the watermark sequence, and we presume that the video may be attacked unconventionally. In this case, the authentication code is adopted to determine the occurrence of tampering and locate the specific modification area in the tampered frames.

Given that we make use of the relationship between the second last nonzero coefficient, $m1$, and the third last nonzero coefficient, $m2$, as the authentication code, recompression can be resisted. If malicious content tamper in a frame occurs, it can be recognized because there would be a large

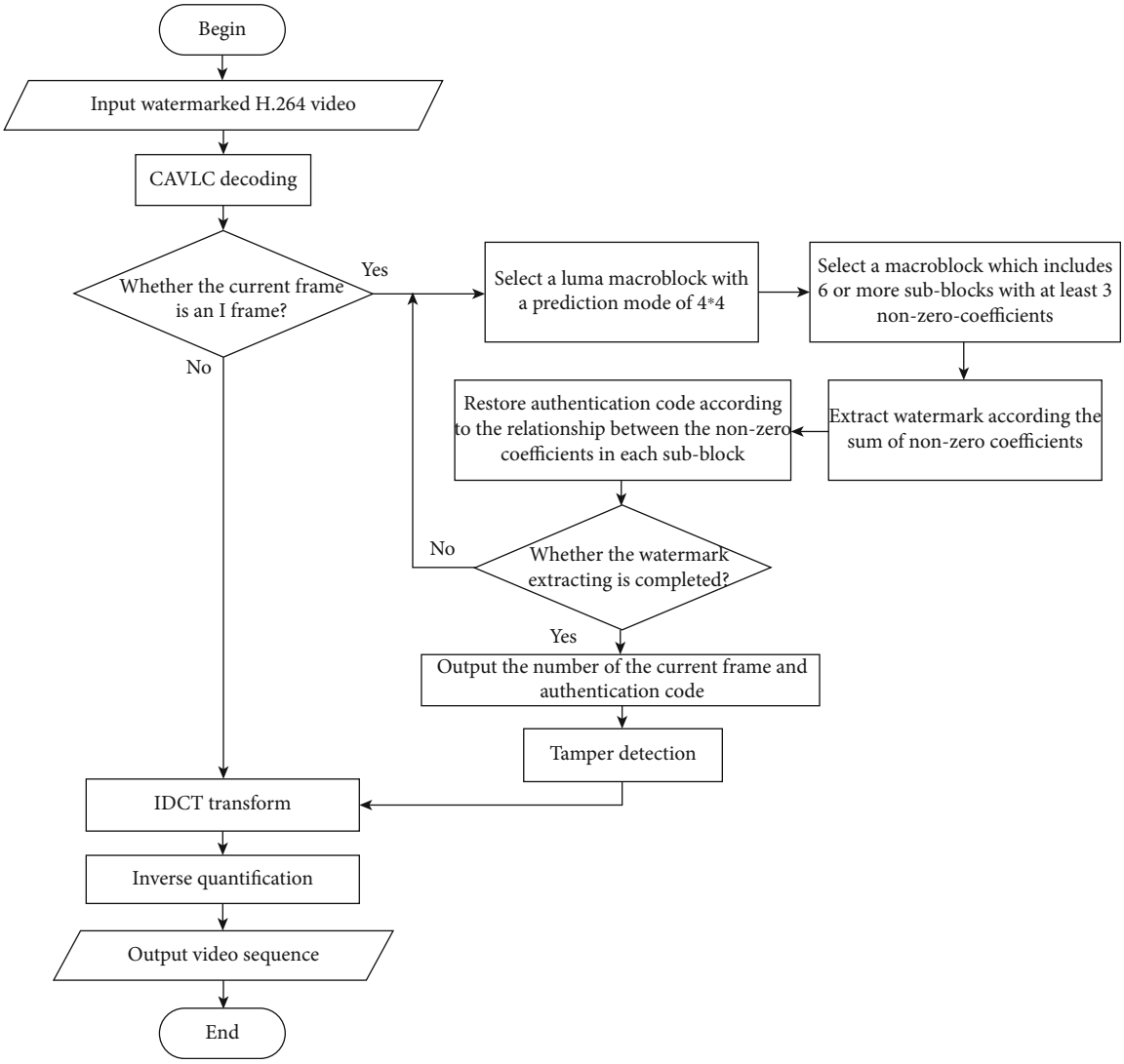

Figure 2: Flow chat of watermark extraction.

change in the authentication code. The specific criteria for judgment are as follows:

Step 1: firstly, check whether the extracted watermark has problems in the macroblock. If the watermark information does not synchronize, continue to Step 2. Otherwise, we consider that there is no tampering.

Step 2: if the frame number based on the watermark extraction is incorrect, it is indicative that the video has been subjected to conventional processing or malicious tampering. Firstly, we should judge whether the flag of this macroblock is variant. If the flag is changed, the extracted information of the current macroblock is distant from the information in the encoder, and we can directly affirm that the current macroblock has been tampered. Otherwise, continue to Step 3.

Step 3: under the condition that the flag is changeless and the watermark is extracted successfully, we generate the authentication code through the same method in encoding. The number of 6-bit authentication code changed is necessary to make further decisions.

Figure 3: Original logo image.

If there are K or more authentication codes that change greatly in the current macroblock, it means that the macroblock has been tampered with maliciously. Otherwise, it means that the macroblock has been subjected to conventional video processing. Here, K is a threshold, and it should be adjusted by the complexity of the video. Denote the authentication code sequence in the decoding as array C, and perform exclusive or operations on each element of array A and the corresponding element of array C in sequence, and the result is saved as W'. Rearranging W' into a two-dimensional array, the logo image can be reconstructed.

(a) Original image (b) Watermarked image (c) Pixel difference

Figure 4: Video frame before embedding watermark and after embedding watermark.

4. Results and Discussion

In this section, we would test the performance of our algorithm under different conditions. We realize our algorithm based on the reference joint test model (JM) software JM by H.264 using the version 8.6. Standard video sequences, including akiyo_qcif, container_qcif, mobile_qcif, news_qcif, tempete_qcif, carphone_qcif, coastguard_qcif, silent_cif, highway_cif, and flower_cif, are provided to demonstrate the effectiveness of the proposed watermarking algorithm. The size of all videos are 176×144 and 352×288 with quarter common intermediate format (QCIF) and common intermediate format (CIF), which adopts the baseline encoding mode. The frame rate is set to 30 frames per second, and the default quantization parameter is set to 28. The logo image is 32×32 two-valued images, as shown as Figure 3.

4.1. Imperceptibility Test. To test the imperceptibility, we provide the comparison of the original sequence and the same reconstructed frame with embedded watermark for the different video. By the space limitation, we provide three video results, which are shown in Figure 4, whose left column is some frame of original sequence, and middle column is the corresponding frame of the same sequence with watermarking, and the right column is the pixel difference of both as seen in Figure 4; the video quality is almost unchanged by subjective judgment.

To better evaluate the invisibility of our watermarking method in an objective way, we introduce two evaluation standards of video quality evaluation, i.e., PSNR (peak signal to noise ratio) and SSIM (structural similarity index) [18] to compare the original video and the video with the embedded watermark. To eliminate the influence of randomness, we give the average of the PSNR and SSIM for all the frames in the video and compare our algorithm to [10]. The comparison results are as shown in Figures 5 and 6, respectively. Normally, if the PSNR is higher than 30, the quality of the processed image is better. From Figure 5, we can see that the PSNRs of our algorithm for various formats and different videos are all higher than 34.0, which proves that our method has inconsiderable effect on video quality. We also compare to [19], the result is shown in Figure 6. Though our method is slightly inferior than [19], the difference is very small, and other indicators have greater ground. From Figure 7, we can know that all the SSIMs of our algorithm are above 0.960, which is very close to 1, and this also proves that the image quality is fine. All SSIM results of different videos are better than [10], as shown in Figure 8. Because we only embed the information into the areas with more complex textures, the impact it has on the video quality is minimal. The abovementioned experiment results evidently prove this fact.

4.2. Bit Change Rate Test. For a watermarking method, the bit-change rate should be also considered. After all, nobody likes the size of the compressed video becoming bigger after

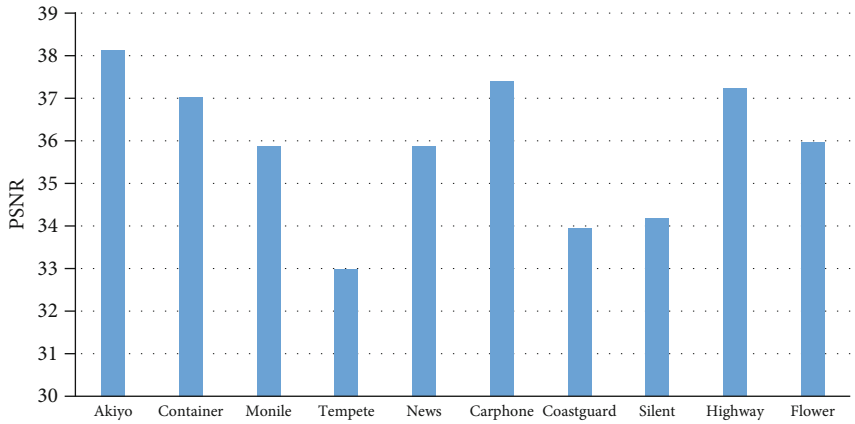

Figure 5: PSNR of proposed method for different sequences.

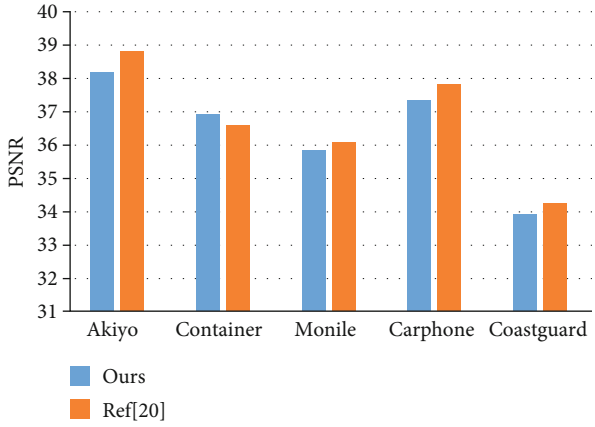

Figure 6: PSNR comparison for different video sequences.

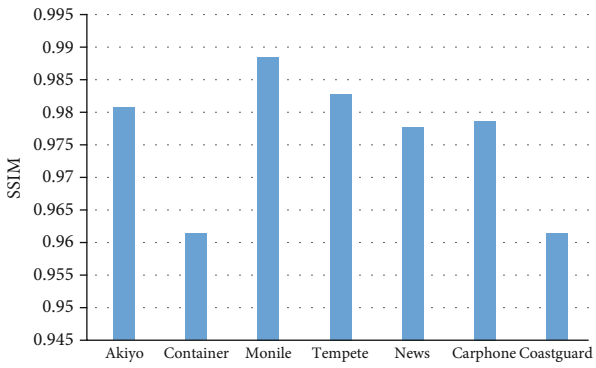

Figure 7: SSIM of proposed method for different sequences.

embedding the watermark. If the bit-change rate is smaller, we can get better video affinity [20]. We can calculate the bit-change rate as follows:

$$BIR = \frac{\left(BIR_{water}\text{-}BIR_{org}\right)}{\left(BIR_{org}\right)} \times 100 \quad (4)$$

The bit-change rate of our algorithm is very low, and the bit-increase rates of the video sequence of container, foreman, mobile, and news are only 0.08%, 0.06%, 0.04%, and 0.08%, respectively. Notwithstanding, the bit-rate changes in [17] are apparent. Comparing with [10, 17, 19], the results are shown in Figure 9. From the comparison results, we can see that our algorithm outperforms the compared ones. The smaller the bit-rate change, the better the real-time performance of the algorithm. Therefore, the proposed algorithm has a better real-time performance and is more suitable for real-time broadcast applications.

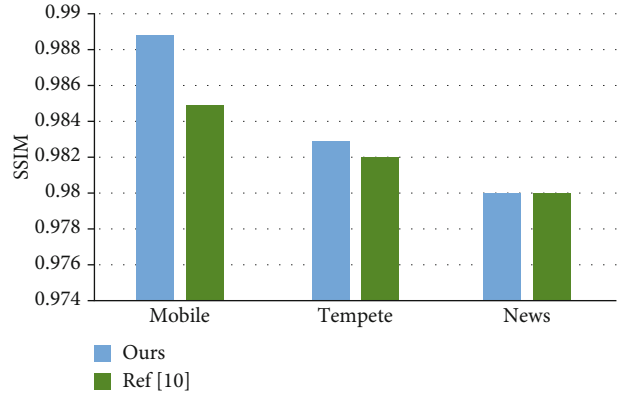

Figure 8: SSIM comparison for different video sequences.

4.3. Robustness Test. Several experiments are conducted to verify the robustness of our watermarking algorithm against recompression. At first, the extracted watermark is provided in Table 1, under the condition that the video with the embedded watermark information is recompressed with equal quantization values ($QP = 28$ and $QP = 34$). From Table 1, we can see that the logo image is clear, which reflects that our algorithm has better performance against recompression.

For objectively evaluating the performance of our algorithm against recompression, the NC (bit accuracy rate) and BER (bit error rate) are adopted to further measure the robustness of the algorithm. The closer the NC is to one and the nearer the BER is to zero, the video distortion from the watermark is less, and the robustness of the algorithm is better. The experimental results of the NC and BER are given

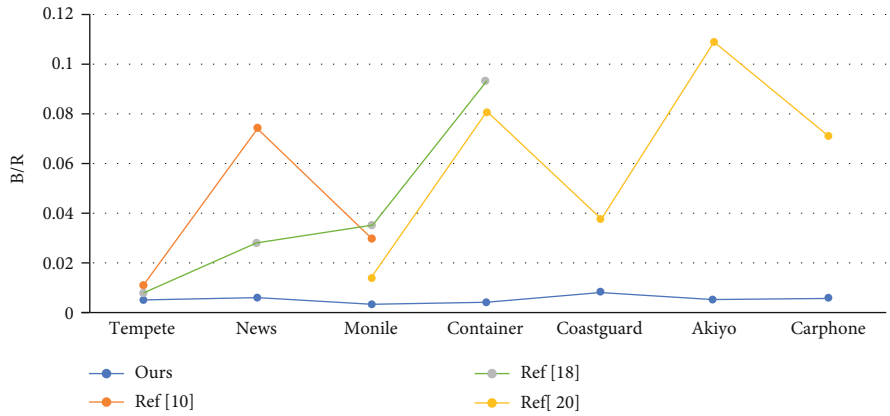

Figure 9: Bit rate contrast diagram.

Table 1: The extraction result under different QP.

Sequence	$QP = 28$	$QP = 34$
Akiyo		
Carphone		
Container		
Coastguard		
Mobile		
News		
Tempete		

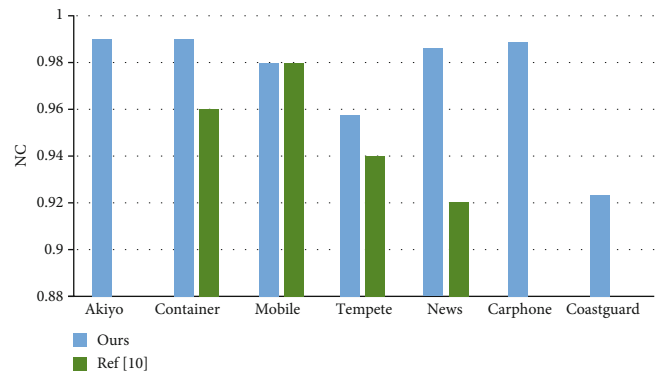

Figure 10: NC comparison under equal quantized recompression.

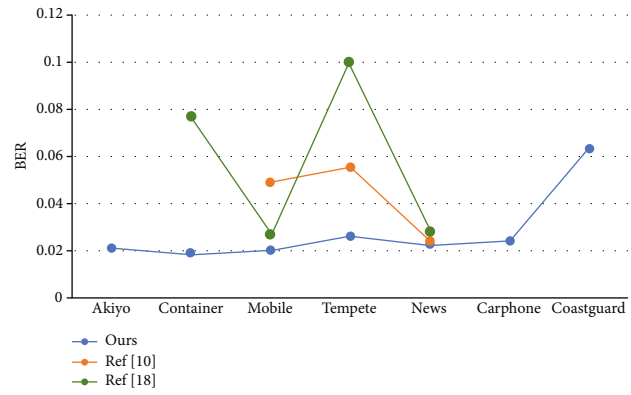

Figure 11: BER comparison under equal quantized recompression.

in Figures 10 and 11, respectively. From them, it is revealed that under the condition of equal quantization and recompression, the NC is higher than 0.91, and it is stable at the mean value of 0.97. When the NC is higher than 0.8, it means that this algorithm has decent robustness. In the meantime, the BER of all sample videos is lower than 0.01 and the average BER is 0.028, which indicates that the proposed method has good robustness for recompression. Compared with [17], our algorithm has less BER than [10, 17] for the same test video, which proves the algorithm has stronger robustness under equal compression quantization, as shown in Figure 11.

4.4. Tamper Detection Test. Based on our watermarking method, the authentication code is used to locate intraframe tampering, and the watermark is used to detect malicious operations, such as addition and deletion between frames. To test the performance of the algorithm to resist the interframe attacks, many experiments are designed, including frame dropping, frame addition, and frame replacement. In the experiment of the frame dropping attack, the frames 40–80 of the video sequence are dropped. The detected result of our method is shown in Figure 12. We can see clearly that our method can successfully extract the reserved frame

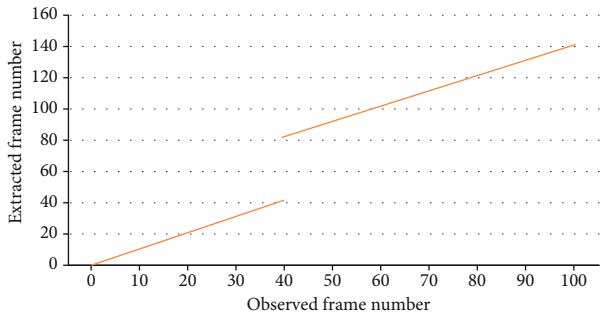

Figure 12: Temporal tampering frame dropping.

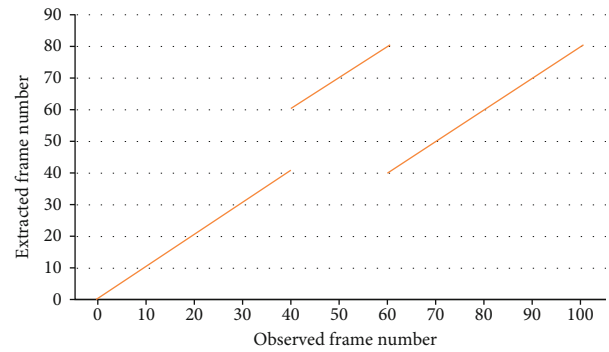

Figure 13: Temporal tampering frame addition.

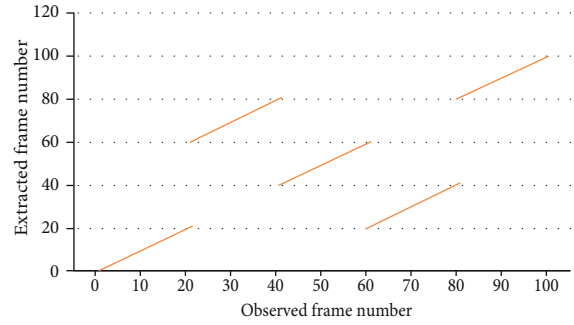

Figure 14: Temporal tampering frame replacing.

number (1–40) and identify the removal of frames 40–80. In the experiment of the frame adding attack, we added frames 61–80 to the 41st frame. The Figure 13 shows the results, wherein we can find our method is also effective for this attack. Finally, we experiment the method with frame replacing attack by substituting frames 61–80 with frames 21–40, and Figure 14 shows that our algorithm can detect such changes successfully.

Through experiments, we also provide the proportion of verification code changes of different video sequences after recompression. The statistical results are shown in Table 2 below.

Then, the proportion of verification code changes of different video sequences after tampering is also provided. The statistic results are shown in Table 3 below.

From the above tables, it is found that the rate of change of videos after recompression and tampering varies greatly.

Next, we tested the effectiveness of the proposed method to the tamper in a frame. Based on the method, if the current macroblock has K or more authentication codes changed, it indicates that the macroblock has been maliciously tampered. Otherwise, the macroblock is processed normally. The value of K is determined according to the actual situation. In this experiment, we implement the specific tamper to copy and paste the top right corner of the first frame of the mobile video sequence. By comparing the authentication codes of each macroblock whether changed to prove the effectiveness of the proposed method. The results of the experiment are shown in Figure 15. The right column of Figure 15 gives the location of the tamper, and we can see an evident region in the top right side.

The detection accuracy rate (DAR) is computed by $DAR = (TPR + TNR)/2$, where TPR and TNR denote true positive rate and true negative rate, respectively. Table 4 shows the experimental results that represent the DAR of our method on different QPs. Also, the results illustrate that our method is efficient to handle content based tamper detection.

Table 2: Rate of change of authentication code after recompression.

Sequence	QP	Rate of change
Akiyo	28	16.1%
Container	28	15.5%
Mobile	28	9.4%
News	28	19.3%
Tempete	28	17.6%
Carphone	28	19.6%
Coastguard	28	18.2%

Table 3: Rate of change of authentication code after tampering.

Sequence	QP	Rate of change
Akiyo	28	79.7%
Container	28	76.9%
Mobile	28	77.3%
News	28	82.5%
Tempete	28	80.7%
Carphone	28	75.6%
Coastguard	28	81.2%

5. Conclusions

Herein, we propose a semifragile video watermarking algorithm based on content authentication, which has good robustness to recompression under different QPs and can simultaneously implement frame attack and video tamper detection. The frame number is binary and is converted into an 18-bit long binary sequence as a watermark. The sequence is divided into three sets of NNZ residual coefficients embedded in the DCT of the video to detect frame attack in the video. To realize the tampering detection of the video, the

Figure 15: Tampering detection diagram.

Table 4: Accuracy evaluation of tamper detection.

Sequence	TPR	TNR	DAR
News	84.08%	84.24%	84.16%
Mobile	88.36%	87.68%	88.02%
Tempete	86.42%	85.32%	85.87%

size relationship of the DCT nonzero coefficient is selected as the authentication code to distinguish whether the video is subjected to normal operation or malicious tampering. The experimental results show that the algorithm has good watermark invisibility, and the algorithm uses a multivalued watermark to embed, which avoids the nonsynchronization of the extracted watermark and greatly reduces the BER value of the watermark after recompression. Compared with the current algorithms, the algorithm has better invisibility and robustness, and it also has a good ability of frame attack and video tamper detection.

Data Availability

The video we use to test can be downloaded from http://trace.eas.asu.edu/yuv/index.html.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under grant no. 61901356 and the HPC Platform of Xi'an Jiaotong University.

We would like to thank Editage (http://www.editage.com/) for English language editing.

References

[1] S. Serikawa and H. Lu, "Underwater image dehazing using joint trilateral filter," *Computers and Electrical Engineering*, vol. 40, no. 1, pp. 41–50, 2014.

[2] H. Lu, Y. Li, M. Chen, H. Kim, and S. Serikawa, "Brain Intelligence: go beyond artificial intelligence," *Mobile Networks and Applications*, vol. 23, no. 2, pp. 368–375, 2018.

[3] H. Lu, Y. Li, T. Uemura, H. Kim, and S. Serikawa, "Low illumination underwater light field images reconstruction using deep convolutional neural networks," *Future Generation Computer Systems*, vol. 82, pp. 142–148, 2018.

[4] Y. Sakai, H. Lu, J. K. Tan, and H. Kim, "Recognition of surrounding environment from electric wheelchair videos based on modified YOLOv2," *Future Generation Computer Systems*, vol. 92, pp. 157–161, 2019.

[5] X. Xu, H. Lu, J. Song, Y. Yang, H. T. Shen, and X. Li, "Ternary adversarial networks with self-supervision for zero-shot cross-modal retrieval," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2400–2413, 2020.

[6] S. Gaur and V. K. Srivastava, "A hybrid RDWT-DCT and SVD based digital image watermarking scheme using Arnold transform," *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, 2017, pp. 399–404, Noida, India, 2017.

[7] S. Dong, J. Li, and S. Liu, "Frequency domain digital watermark algorithm implemented in spatial domain based on correlation coefficient and quadratic DCT transform," in *2016 International Conference on Progress in Informatics and Computing (PIC)*, pp. 596–600, Shanghai, China, 2016.

[8] S. J. Horng, D. Rosiyadi, P. Fan, X. Wang, and M. K. Khan, "An adaptive watermarking scheme for e-government document images," *Multimedia Tools and Applications*, vol. 72, no. 3, pp. 3085–3103, 2014.

[9] X. L. Chen and H. M. Zhao, "A novel video content authentication algorithm combined semi-fragile watermarking with compressive sensing," in *2012 Second International Conference on Intelligent System Design and Engineering Application*, pp. 134–137, Sanya, Hainan, 2012.

[10] M. E. Farfoura, S.-J. Horng, J.-M. Guo, and A. Al-Haj, "Low complexity semi-fragile watermarking scheme for H.264/AVC authentication," *Multimedia Tools and Applications*, vol. 75, no. 13, pp. 7465–7493, 2016.

[11] A. K. Mairgiotis and D. Ventzas, "Video watermark detection in DCT domain for H.264/AVC and extensions through a hierarchical prior," *2nd IET International Conference on Intelligent Signal Processing 2015 (ISP)*, 2015, pp. 1–6, London, 2015.

[12] M. Fallahpour, S. Shirmohammadi, M. Semsarzadeh, and J. Zhao, "Tampering detection in compressed digital video using watermarking," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 5, pp. 1057–1072, 2014.

[13] D. Xu, R. Wang, and J. Wang, "A novel watermarking scheme for H.264/AVC video authentication," *Signal Processing: Image Communication*, vol. 26, no. 6, pp. 267–279, 2011.

[14] A. Cedillo-Hernandez, M. Cedillo-Hernandez, M. Garcia-Vazquez, M. Nakano-Miyatake, H. Perez-Meana, and A. Ramirez-Acosta, "Transcoding resilient video watermarking scheme based on spatio-temporal HVS and DCT," *Signal Processing*, vol. 97, pp. 40–54, 2014.

[15] S. Chen and H. Leung, "Chaotic watermarking for video authentication in surveillance applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 704–709, 2008.

[16] R. Facciol and R. Farrugia, "Robust Video Transmission using Reversible Watermarking Techniques," in *2010 IEEE International Symposium on Multimedia*, pp. 161–166, Taichung, Taiwan, 2010.

[17] L. Tian, H. Dai, and C. Li, "A semifragile video watermarking algorithm based on chromatic residual DCT," *Multimedia Tools and Applications*, vol. 79, no. 3-4, pp. 1759–1779, 2020.

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[19] N. Mehmood and M. Mushtaq, "Blind watermarking scheme for H.264/AVC based on intra 4x4 prediction modes," in *Future Information Technology, Application, and Service*, pp. 1–7, Springer, 2012.

[20] L. Tian, N. Zheng, J. Xue, and T. Xu, "A CAVLC-based blind watermarking method for H. 264/AVC compressed video," in *2008 IEEE Asia-Pacific Services Computing Conference*, pp. 1295–1299, Yilan, Taiwan, 2008.