IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 4, 2021, accepted February 23, 2021, date of publication February 26, 2021, date of current version March 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3062468

A Compressed-Domain Robust Video Watermarking Against Recompression Attack

HAO DING[1], RUIXIN TAO[1], JING SUN[2], JIN LIU[3], FAN ZHANG[3], XIAOPING JIANG[1], (Member, IEEE), AND JIANJIN LI[4]

[1]Hubei Key Laboratory of Intelligent Wireless Communications, College of Electronics and Information Engineering, South Central University for Nationalities, Hubei 430074, China
[2]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
[3]Beidou Intelligent Technology Company Ltd., Shenzhen 518055, China
[4]Institute of Computer Science, University of Clermont Auvergne, 63001 Clermont-Ferrand, France

Corresponding author: Jing Sun (sunjing528@163.com)

This work was supported in part by the National Key Research and Development Project of China under Grant 2019YFB2102500, in part by the National Natural Science Foundation of China under Grant 61702563, in part by the Natural Science Foundation of Guangdong Province China, in part by the Natural Science Foundation of Hubei Province China under Grant 2019CFC924, and in part by the funds of South-Central University for Nationalities under Grant CZT20001.

ABSTRACT With the development of communication networks and the widespread use of mobile terminals, videos are increasingly shared and distributed among mobile users. During the process, the video is recompressed to a certain file size on the sending side and sent to the receiving side via the server. This makes robust video watermark with low complexity to resist recompression attacks become an important issue to address. The proposed video watermarking method in this article is specially designed for resisting recompression attacks when quantization parameter (QP) is greatly increased. In the proposed method, by using the texture information and motion information of video, the invariance of video content under different quantization parameters could be found to help improve the anti-recompression attack ability of video watermark. Moreover, the proposed framework does not use a location map which has security risk to locate the watermark. It aims at finding the optimal location of watermark embedding adaptively according to the feature of video content itself. The experiment results indicate that the proposed video watermarking method can not only achieve greater robustness against recompression attack but also effectively limit the degradation in video perceptual quality.

INDEX TERMS Compressed domain, recompression attack, video watermarking.

I. INTRODUCTION

The rapid growth of fast communication networks and the widespread use of mobile terminals make videos frequently shared and distributed among mobile users. During the process, the video is transferred to the receiver through the server. Considering the storage pressure of the server, the bandwidth bottleneck in receiving network, and the performance in receiving terminal equipment, etc., the server will ask the sender to recompress the video locally until it meets the requirements of the publishing rules, e.g. format and the size of video files, and then upload it to the server. In practical applications, a typical user is more willing to get a smaller video size by using of increasing the quantization parameter (QP) greatly during the process of recompression.

The associate editor coordinating the review of this manuscript and approving it for publication was Nilanjan Dey.

For copyright protection and piracy tracking, a video is embedded with a unique watermark so that the users who distribute pirated videos can be identified when piracy occurs [1], [2]. During the process of recompression, first the compressed bit stream is uncompressed in raw video using the decoder. Then, the video is compressed again using the encoder for different QP values. Since the change of QP, the compressed domain parameters such as transform coefficient, block structures, and prediction modes may change, which may destroy the embedded watermark hidder in the video, while it is more difficult to locate these watermark. Therefore, video watermark algorithm to resist recompression attack when QP greatly increases has become an important issue to attention.

The watermarks can be embedded before encoding (i.e. uncompressed-domain watermarking) [3]–[11] or during encoding (i.e. in-the-loop watermarking) [12], [13], both

completely compressing each video with the high computing complexity and the poor real-time performance [2]. Besides, compressed-domain watermarking [15]–[36], which also called out-of-the-loop watermarking, directly embeds the watermarks into the compressed videos. Compressed domain watermarking schemes can save massive computing resources since the watermarks can be embedded into it only by partially decoding and encoding the video. Therefore, low-complexity and high real-time compressed domain watermarking schemes plays an indispensable role in the field of video watermarking. However, it has its own intrinsic problems, i.e. poor robustness to attacks, which is the current research hotspots of the compressed-domain video watermarking. Robustness-related attacks that video watermarking have several types of attacks, including normal image processing attacks, geometric attacks, temporal synchronization attacks, malicious tampering attacks, etc [1], [2]. Moreover, compressed-domain video watermarking must be combined with the complex video coding standard. Thus, it is more difficult to resist different types of attacks at the same time [2]. Most of the existing literature focuses on specific types of attacks according to the requirements of the application scenario [14]. In practical applications, the recompression attack is one of the major attacks for compressed domain video watermarking since it can occur unintendedly during the video content adaptation and transmission. Therefore, designing a video watermarking scheme in compressed domain which is robust to recompression attack is really desirable and applicable.

Considerable achievements have been made in many works [15], [18], [19], [22], [23], [28], [29], [32], [35], [36] on this problem. Mansouri *et al.* [15] used number of non-zero quantization coefficients (NNZ) to determine the embedding area of the watermark, and adjusts the effect of the watermark on video quality through a priority matrix. Liu *et al.* [18] used BCH syndrome code (BCH code) technique to preprocess the watermark data, so that the erroneous extracted watermark can be corrected. During the recompression process, the algorithm can resist the recompression attack with an unchanged QP, but since the error-correcting ability of the BCH code is limited, when the QP changes greatly, it can not resist recompression attack.

Song *et al.* [19] proposed a watermarking algorithm based on prediction mode, and the algorithm embedded the watermark in 8×8 blocks of the I frame to improve the robustness. Yang *et al.* [22] also embedded the watermark by modifying the intra prediction mode. However, these algorithms [19], [22] cannot maintain robustness when QP changes greatly since the intra prediction mode is mostly altered to next prediction mode which may alter after recompression attack [23].

Gaj *et al.* [28] embedded the watermark in the residual matrix of 4×4 luma transform block (LTB), and ensure the robustness by modifying the 3×3 matrix in the upper left corner. Dutta and Gupta [29] proposed a blind watermarking algorithm, which uses the spatio-temporal features of the video and a random key to select the watermark embedding block. Zhou and Wang [35] proposed a watermarking algorithm based on the spatial domain, embedding the watermark in 4×4 LTB by modifying multiple coefficients. Gaj *et al.* [36] embedded the watermark in the block with less motion information in the I frame, and modify the 4×4 discrete cosine transform (DCT) coefficient to change the NNZ difference in the consecutive intra predicted frames to embed watermark. Gaj *et al.* [23] embedded the watermark by changing the intra prediction mode of the video, by grouping of the intra prediction modes such a way that the mode change due to recompression can be closed within a group, to resist recompression attack.

During the process of recompression, the change of QP will change compressed domain parameters, therefore, it is difficult to find the locations where the watermark embeds, in further to extract the watermark correctly. In [22], [23], [28], [29], [32], [35], [36], in order to locate these watermark in the compressed video watermarking schemes, they embedded the watermark in the specified locations. In these framework, since the detector or the decoder may not find the same locations due to the changes after recompression or imposed attacks, they used a location map or an embedding position template by private key describing embedding location which is used to find the specified locations where the watermark embeds. However, these locations are vulnerable to be identified by attackers [37], [38]. Thus, the security issue is the major problem in these methods [39], [40].

In practical applications, the video watermarking scheme can resist recompression attack when QP increases greatly have more urgent needs and practical significance. The existing compressed domain watermarking schemes are that they are fragile against recompression attack when QP increases substantially [15], [18], [19] or they have security issue since they need location map when extract watermark [22], [23], [28], [29], [32], [35], [36]. In this article, we try to to realize blind extraction without location map to improve the security of the video watermark scheme as well as it has good performance on robustness to recompression attack when QP increases greatly.

In this work, we propose a robust video watermarking algorithm against recompression attack in the compressed domain. In the proposed method, we embedded the watermark in the last P frame of the group of pictures (GOP) structure to reduce drift distortion. Using the texture information and motion information of video to find the invariance of video content to improve the anti-recompression attack ability. That is to say, the best block to embed watermark can be selected by the feature of the video content itself instead of using a location map to locate these blocks. Specifically, suitable candidate blocks for watermark embedding are selected adaptively according to video content feature which is calculated by the residual factor and energy factor and motion vector entropy. And then the watermark is embedded by modifying the positive or negative of the first AC coefficient of the quantization coding after discrete cosine transform (QDCT)

coefficient of the 4 × 4 watermark embedding block. The proposed framework is robust against recompression attack without using original video or location map when extract watermark. Moreover, it effectively limits degradation in perceptual quality as well.

This article is organized as follows. In Section 2, we perform a coding process analysis in video coding and the motivation of the article. In Section 3, a compressed-domain robust video watermarking to resist recompression attack is proposed. Then, experiments and results analysis illustrated in Section 4. And finally, conclusion are given and discussed in Section 5.

II. CODING PROCESS ANALYSIS AND MOTIVATION

In video coding, transforming the image in the spatial domain to the frequency domain will produce some transform coefficients with little correlation, which can be compressed and encoded, that is, transform coding. In the transform coding process, the residual information is transformed and quantized to remove the frequency domain correlation, where in the residual information is calculated as follows.

$$X(i,j) = cur(i,j) - pred(i,j) \tag{1}$$

Here, cur(i,j) represents the current value of each pixel, and pred(i,j) is the predicted pixel value which is obtained by performing motion compensation on the previously existing reference frame through predictive coding. After the residual information X is obtained, DCT transform is performed as follows.

$$Y(i,j) = [C_f X(i,j) C_f^T] \otimes E_f \tag{2}$$

where

$$C_f = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}$$

$$E_f = \begin{pmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{pmatrix}$$

Here, E_f is a constant matrix, $a = \frac{1}{2}, b = \sqrt{\frac{2}{5}}$. Generally speaking, the DCT coefficients obtained after DCT transformation have a large dynamic range. Therefore, quantization of the DCT coefficients and combination with coding techniques such as entropy coding can effectively reduce the value space of the signal to achieve better compression. The basic forward quantization operation is given in (3) as follow.

$$Z(i,j) = round[Y(i,j)/Q_{step}] \tag{3}$$

where, Y(i, j) is the conversion coefficient in matrix Y, and Z(i, j) is the output quantization coefficient, Q_{step} is the quantizer step size which is determined by QP.

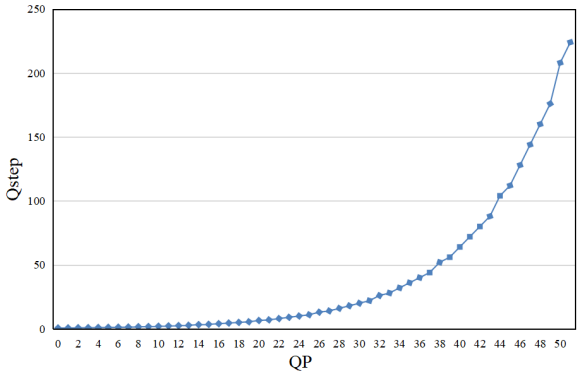

FIGURE 1. The relationship between QP and Q_{step}.

In H.264 encoding, the quantization process must also complete the multiplication of E_f in the DCT transform, so (3) can be expressed as:

$$Z(i,j) = round\left(W(i,j)\frac{PF}{Q_{step}}\right) \tag{4}$$

where, $W(i,j) = C_f X(i,j) C_f^T$, PF is a^2, ab/2 or $b^2/4$ depending on the position (i, j) of the matrix E_f as follows.

$$PF = \begin{cases} a^2 & (0,0)(2,0)(0,2) or (2,2) \\ b^2/4 & (1,1)(1,3)(3,1) or (3,3) \\ ab/2 & otherwise \end{cases} \tag{5}$$

In order to avoid any division operations, (4) is converted as follows.

$$Z(i,j) = round\left(W(i,j)\frac{MF}{2^{qbits}}\right) \tag{6}$$

where

$$MF = \frac{PF}{Q_{step}} 2^{qbits} \tag{7}$$

$$qbits = 15 + floor(QP/6) \tag{8}$$

In (6), Z(i, j) is the value after quantization which is QDCT. Fig.1 shows the relationship curves of QP and Q_{step} in H.264 / AVC [41]. It is expressed that the Q_{step} will change in a large range due to the increase or decrease of QP by 1, which will have a great influence on the value of the quantization transformation coefficient. As shown in Fig.2, the QDCT value of a block under different QP compression is given. When QP is large, the value of QDCT decrease greatly, and the number of QDCT equal to zero in a block increase greatly.

During the process of recompression, when QP increases from one value (e.g. QP = 16) to a large value (e.g. QP = 32), the video will lose some details and the bit rate will be reduced, which lead to lower video quality and enhanced video distortion, while the compressed domain parameters such as transform coefficient, block structures, prediction modes may change, these will affect the embedded watermark greatly. Mansouri *et al.* [15] and Liu *et al.* [18] are based on NNZ in a block in order to determine the watermark embedding block. When QP increases or decreases

69	-8	8	-2
-4	-3	-1	-1
0	-2	-2	0
-1	-1	0	0

QP=16

28	-4	2	-2
-1	-1	0	-6
-1	-5	-3	-1
-1	0	-1	0

QP=20

6	-2	2	-4
-1	-1	0	0
-1	-1	-2	0
0	0	-1	0

QP=28

-1	-1	1	-2
0	1	-2	-1
-1	-1	0	0
0	0	0	0

QP=32

FIGURE 2. Value of QDCT in a block when compression using different QP.

during the process of recompression, the number of nonzero transform coefficients (NNZ) in a block gradually reduce lead to increases difficulty to the location of the watermark embedding block. In Song *et al*. [19], the watermark embedding block is selected through the prediction mode. Since the selection of prediction mode is also related to QP in the coding rules and the change of QP will affect the prediction mode, it is more difficult to find the watermark embedding block. In order to solve this problem, some researchers [22], [23], [28], [29], [32], [35], [36] proposed to use location maps to record the locations of watermark embedding blocks. Although the location map improves the robustness of the algorithm, it also increases the security risk of the algorithm, because these location maps are vulnerable to be identified by attackers. Moreover, in the case of using location map, their algorithms have acceptable robustness when QP increases or decreases in a small range. And some of them does not present good performance when QP increases greatly.

In this article, we try to realize blind extraction without location map to improve the security of the video watermark scheme as well as it has good performance on robustness to recompression attack when QP increases greatly. The three main novelties and contributions in this article are as follows.

1) Blind extraction without location map. According to feature of video content, using the texture and motion information of the video to find the optimal location of watermark embedding adaptively, which improves robustness of the anti-recompression attack. And without using location map to extract watermark, it improves security of the video watermark scheme.

2) Good versatility. Since the method of selection the optimal watermark embedding blocks is based on the feature of video content itself, it is not limited by the coding standard and can be applied to other common video coding standards. Moreover, the method of finding optimal block of watermark embedding can be applied to the current mainstream video watermarking algorithms to furtherly improve the security and robustness.

3) Low complexity and high real-time performance. The proposed algorithm is based on compressed domain and embedding process is simple and fast, which has high real-time performance and is suitable for large-scale video distribution applications.

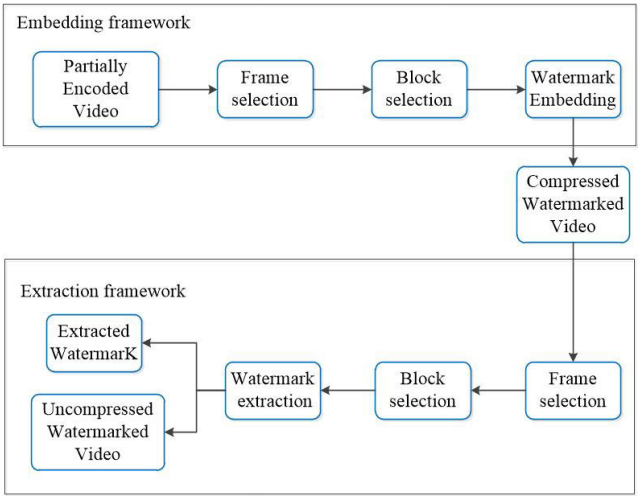

FIGURE 3. Watermarking algorithm framework.

III. PROPOSED METHOD

In this section, the algorithm proposes a robust video watermarking framework against recompression attack. As shown in Fig. 3. First, the algorithm uses texture information and motion information in the last P frame of each GOP to detect blocks to be embedded in the watermark according to the video content features. And then, the article describes the watermark embedding and extraction process, respectively. The watermark is embedded by changing the amplitude of the first AC coefficient in the QDCTs of 4×4 blocks.

A. SELECT WATERMARK EMBEDDING BLOCKS

Frames are divided into different sizes blocks in the interprediction process, and the reference frame blocks is used to predict each block. If these reference frame blocks are embedded the watermark, then the watermark noise will propagate to the frames which are predicted. Thus, drift distortion will occur and affect the video quality. The algorithm reduces the drift distortion by embedding the watermark in the last P frame of the GOP, because the last P frame of the GOP will not as a reference frame, this will reduce the influence of the reference frame due to the modification of the current frame.

In the following, a scheme is proposed for against recompression attack for the interpredicted P-frames based on the concept of visual sensitive areas. Visual sensitive areas represent textured and motion-rich blocks which are detected by the video content. When QP increases significantly, the value of the QDCT of the blocks in visual sensitive areas changed dramatically, however, the texture and motion distribution of the video content itself will not change with the change of QP. In other words, the blocks in visual sensitive areas still have more texture and motion information compared with other areas. When we embed the watermark into these blocks in the visual sensitive areas, it is easier to locate these blocks

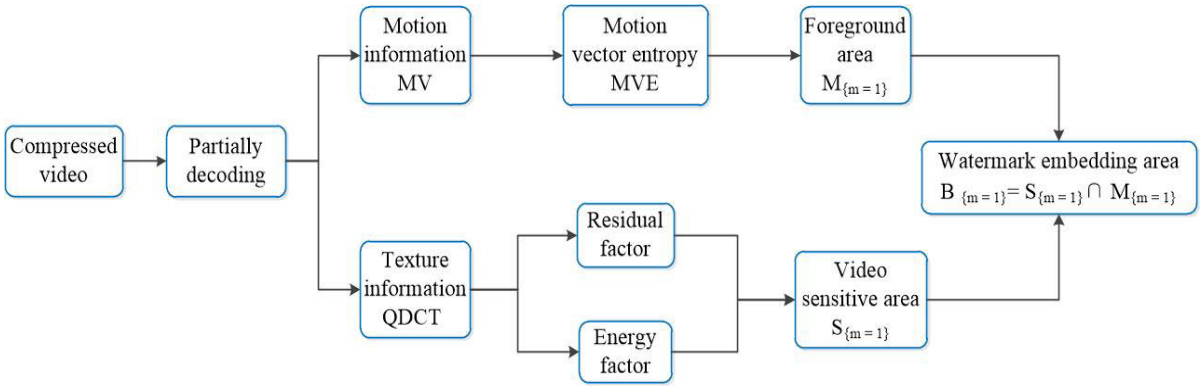

FIGURE 4. Detect embedded watermark blocks.

when extracting the watermark than other areas. Specifically, the algorithm uses the texture information of the video to calculate the residual factor and energy factor through the QDCTs of the blocks, and fuse the two factors to detect the visual sensitive area. Then we use the motion vector entropy (MVE) which is calculated by the motion information to further filter these visual sensitive areas, to obtain the watermark embedded blocks. The flow chart is shown in Fig.4.

The watermark embedding area is calculated by the QDCTs and motion vectors in the process of video coding, which contain the texture information and motion information of video content. Texture factor related to texture information contains two indexes: residual factor and energy factor. Motion factor related to motion information contains one index: motion vector entropy. In the following operation, we use 4×4 block as a basic processing unit.

1) TEXTURE FACTORS

a: RESIDUAL FACTOR(Rf)

In the video encoding process, the information will be concentrated into low frequency coefficients and most of the intermediate or high frequency coefficients are transformed and quantified to zero. The residual factor is an indicator used to count the number of the nonzero QDCTs of a 4×4 block. In the proposed method, the residual factor represents the complexity of the texture features of the block. Although changing QP greatly changes the specific value of the QDCTs in each block, the distribution of the number of nonzero coefficients is relatively stable. In other words, compared with the regions with less information, the number of nonzero QDCTs of the block in the information-rich region is still greater after being compressed with different QP. The residual factor defined as in (9).

$$Rf(i) = \|C\|_0 \tag{9}$$

where, i represents the i_{th} 4×4 block, and C represents a QDCT coefficient matrix obtained after quantization transform coding of a 4×4 block. $||C||_0$ is the number of non-zero elements in vector C. The texture features of visual sensitive areas are more complex, and their residual factor is generally greater than the other areas.

b: ENERGY FACTOR(Ef)

Energy factor (Ef) is a concept in the image domain, which represents the amount of information in a block. In the video coding process, Ef is the sum of the absolute values of the QDCTs of a 4×4 block. In the proposed method, the energy factor represents the complexity of the texture features of the block. Although changing QP greatly changes the specific value of the QDCTs in each block, the energy distribution of blocks is relatively stable. In other words, the area with large energy still have large energy after recompression with different QP. Energy factor defined as in (10).

$$Ef\ (i) = \sum_{j=1}^{4} \sum_{k=1}^{4} |c\ (j, k)| \tag{10}$$

where, i represents the i_{th} 4×4 block, and c(j, k)represents the coefficient of the j_{th} row and k_{th} column of QDCT coefficient matrix C of each 4×4 block.

c: FACTOR FUSION

The appropriate fusion method will based on whether the features are independent, and whether their interaction diminishes or enhances visual sensitive area. In our algorithm, we note that Rf and Ef are independent with each other, because that we could imagine a block in the video with low Rf and high Ef, and vice versa. Also, the area is not only likely to contain structural mutation area (large Ef), but also contains areas that are not easily predict from previous frames (large Rf) when both Rf and Ef are large, The combined action of Ef and Rf is likely to increase visual sensitive area. Therefore, our fusion involves both additive and multiplicative combination of Rf and Ef, shown as follows in (11).

$$S(i) = \mathrm{N}\left[Ef(i) + Rf(i) + Ef(i) \cdot Rf(i)\right] \tag{11}$$

where, i represents the i_{th} 4×4 block, and N represents the normalization operation. If the texture characteristics of the area are more obvious, the larger the Rf(i) and Ef(i), the larger

FIGURE 5. Visual sensitive blocks. (a) Shows the visual sensitive area, (b) Shows the watermark embedding area.

the S(i) will be. The S(i) in the area with larger values still have large S(i) values after recompression with different QPs. The blocks with large S(i) values are more easily noticed by the human and the content structure is more complicated and not easily lost during the video transmission process. This algorithm marks the S(i) = 1 block as $S_{\{m=1\}}$ and keeps it as the video sensitive area. The detailed process of video sensitive areas generation given in Algorithm I. Fig.5 (a) shows the visual sensitive areas detected by the Algorithm I.

Algorithm 1 : Generate Video Sensitive Areas

Data: Compressed Video
Result: Video sensitive areas
for each the last P frame of each GOP of the video **do** /* Select frames */
  for each 4 × 4 block in a frame **do** /*Detect blocks*/
    Calculate the number of the non-zero QDCT coefficients in each 4 × 4 block; /* Residual factor*/
    Calculate the sum of the absolute values of QDCT coefficients in each 4 × 4 blocks;
/*Energy factor*/
    S = Norm(Rf+Ef+Rf·Ef); /*Factor fusion*/
    if S = 1**then**
      The current block is marked as $S_{\{m=1\}}$
      /*Detect visual sensitive areas */
    else
      Skip to the next block;

2) MOTION FACTOR

The motion vectors (MVs) of a video often records the important information of the video scene in the spatio-temporal neighborhood. MV represents the offset between the current block and the best matching block in the reference frame. Due to the motion of the object, the different MVs will be generated in the corresponding spatio-temporal neighborhood. Among of them, the MV representing the background area will appear to be consistent over a large area, and the MV representing the moving object, especially when the object motion is more flexible, will be very different from each other.

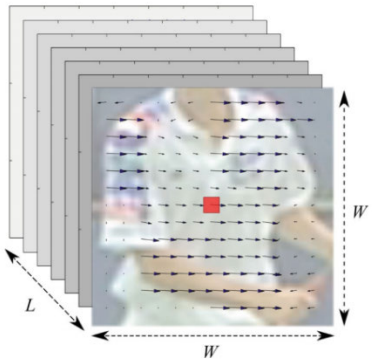

FIGURE 6. MV map of the block.

Article [42] used the concept of information entropy to mark the foreground area through statistical MV. As shown in Fig. 6, it is a cube centered on a block b (shown in red) with a size of 4 × 4. This cube is defined as the causal spatio-temporal neighborhood of block b, with a size of W×W×L. Where, W represents the space size, the size is twice the size of the fovea, L represents the time size, here set to 200 ms. It is worth noting that all the MVs in the frame are mapped to 4 ×4 blocks, which is convenient for statistics of motion information, since the macroblock size is different in video coding.

The motion information MV in the causal spatio-temporal neighborhood of 4 × 4 blocks is used to calculate the MVE of this block [42]. As follow in (12)

$$MVE\,(b) = -\frac{1}{\log N}\sum_{i\in H(\Theta(b))}\frac{n(i)}{N}\cdot\log\left(\frac{n(i)}{N}\right) \quad (12)$$

where,

$$N = \sum n\,(i)$$

In (12), H(•) is the histogram, i is the abscissa of the histogram, which represents the bin index. In bin i, the number of MVs is counted as n(i). Θ(b) represents the motion cube which is associated with the block b. 1/ log N is used for normalization, so that the maximum value of MVE is 1. According to statistics, the MVE of the foreground in the video is often greater than the MVE of the background. Therefore, this motion information entropy can be used to select the foreground area in the video, which be mark as $M_{\{m=1\}}$.

3) WATERMARK EMBEDDING AREA

In order to resist the recompression attack with different QPs, the algorithm use the invariance of video content, considering both the richer texture features and the more intense motion features of the video to select blocks to embed the watermark. This detailed process is given in Algorithm II. The intersection of the visual sensitive area $S_{\{m=1\}}$ and the foreground area $M_{\{m=1\}}$ is used to embed the watermark as follows in (13). The blocks in watermark embedding area are used to embed watermark. Fig. 5(b) shows the blocks in watermark

Algorithm 2 : Select Watermark Embedding Block

Data: Compressed Video
Result: Watermark embedding block
for each the last P frame of each GOP of the video **do** /*Select frames*/
  for each 4×4 block in a frame **do** /*Select blocks*/
    Calculate the motion vector entropy MVE for each block with MVs;
    Mark foreground area as $M_{\{m=1\}}$/*Detect foreground area $M^{*}_{\{m=1\}}$/
  for each 4×4 block in a frame **do**
    if Current block is $\{M_{\{m=1\}} \cap S_{\{m=1\}}\}$ **then**
      The current block is marked as $B_{\{m=1\}}$ /*Select watermark embedding block */
    else
      Skip to the next block;

embedding area detected by the Algorithm II.

$$B_{\{m=1\}} = M_{\{m=1\}} \cap S_{\{m=1\}} \tag{13}$$

B. PROPOSED VIDEO WATERMARKING ALGORITHM

1) WATERMARK EMBEDDING

In order to improve the security of the watermark sequence, the watermark information need be preprocessed before embedding the watermark. In gerenal, Arnold scrambling algorithm is used to encrypt watermark information. First, the compressed domain video is partial decoding to obtain MV and QDCT. Then, the 4×4 blocks $B_{\{m=1\}}$ for embedding are detected based on the method in section A. Embed the pre-processed watermark data into the QDCT coefficients of the block. The results of an element QDCT transform are DC coefficients and AC coefficients. The algorithm works on AC coefficients since such coefficients are less sensitive to embedding error compared to DC coefficients. Finally, all the QDCTs are entropy encoded to obtain the embedded video bitstream. The watermark embedding algorithm is described in Algorithm III, specifically:

Step 1: Use the Arnold scrambling algorithm to preprocess the watermark sequence.

$$W = \{w(i) |\ i = 1, 2, \cdots L, w(i) \in \{0, 1\}\} \tag{14}$$

where, L represents the length of the watermark sequence.

Step 2: Detect the blocks to embed watermark using the method in Section A.

Step 3: Embed the watermark information in the blocks $B_{\{m=1\}}$. For each block in $B_{\{m=1\}}$, watermark is embedded in the first quantized AC coefficient c_1 in the block and c_1 are modified as follows.

1) If the embedded bit is 1, c1 is modified as follows.

$$\begin{cases} c_1 = c_1 + 1 & if \ \ c_1 \geq 0 \\ c_1 = -c_1 + 1 & if \ \ c_1 < 0 \end{cases} \tag{15}$$

2) If the embedded bit is 0, c1 is modified as follows.

$$\begin{cases} c_1 = c_1 - 1 & if \ \ c_1 < 0 \\ c_1 = -c_1 - 1 & if \ \ c_1 \geq 0 \end{cases} \tag{16}$$

Algorithm 3 : Watermark Embedding

Data: Unwatermarked Video, pre-processed watermark
Result: Watermarked Video
for each block $B_{\{m=1\}}$ that satisfies in Alogrithm II **do**
  if W(i)= 1 **then** /*If the watermark is 1, then c_1 = positive number. */
    if $c_1 >= 0$ **then**
      $c_1 = c_1 + 1$;
    else
      $c_1 = -c_1 + 1$;
  if W(i)= 0 **then** /*If the watermark is 0, then c_1 = negative number.*/
    if $c_1 < 0$ **then**
      $c_1 = c_1 - 1$;
    else
      $c_1 = -c_1 - 1$;

Algorithm 4 : Watermark Extraction

Data: Watermarked Video
Result: Unwatermarked Video
for each block $B_{\{m=1\}}$ that satisfies Alogrithm II **do**
if $c_1 >= 0$ **then** /*Extract watermark*/
  Watermark bit is 1; /*Extracted watermark bit is 1*/
else
  Watermark bit is 0; /*Extracted watermark bit is 0*/

2) WATERMARK EXTRACTION

The extraction operation is shown Algorithm IV. After entropy decoding, the extraction of watermark is performed at the decoder. The watermark embedding blocks $B_{\{m=1\}}$ are detected according to Section A, and watermark is extracted in the first quantized AC coefficient c_1 in these blocks as follows.

$$\begin{cases} w(i) = 1 & if \ \ c_1 \geq 0 \\ w(i) = 0 & if \ \ c_1 < 0 \end{cases} \tag{17}$$

IV. EXPERIMENTS AND RESULTS

The proposed algorithm was implemented in the H.264 / AVC reference encoding software version JM16.0, and performance of this algorithm was tested in the environment of Matlab2018b. In order to check the effectiveness of the algorithm, the experiment selects test sequences that satisfy various scenarios. For example, Mobile and Tennis have higher texture areas and motion, while Carphone and Claire sequences have lower texture changes and very little motion. The specific parameters of the experimental environment

TABLE 1. Experimental Set-Up

Parameters for Watermarking	Values Taken
H.264/AVC	JM 16.0
GOP structure	I-P-P-P
Quantization Parameter (QP)	16
Video Sequence Used	Mobile, Claire, Tempete, Carphone, Flower, Coastguard , Ice , Paris, Huskey, Football, Garden, Stefan, Foreman, Tennis, Bus, Hall, Soccer
Video resolutions	176×144,352×288,416×240,576×432
Frame rate (fps)	30

and detailed sequence rates are shown in Table 1. The algorithm mainly evaluates the performance of the watermarking scheme from the aspects of robustness, visual quality, security and capacity.

A. ROBUSTNESS ANALYSIS AGAINST RECOMPRESSION ATTACKS

The main purpose of this algorithm is to use the invariance of the video content to resist recompression attacks of different QPs, and to ensure that the embedded watermark will not be distorted. Therefore, the robustness of the algorithm after different QP recompression attacks has become an important evaluation performance index.

To prove the robustness of the watermarking algorithm, the methods calculate the bit error rate (BER) which is the probability of error into one bit and similarity (Sim) of the extracted watermark. These two indicators are defined as follows:

$$Sim = \frac{\sum_{i=1}^{a}\sum_{j=1}^{b}\left[M(i,j) \times \hat{M}(i,j)\right]}{\sqrt{\sum_{i=1}^{a}\sum_{j=1}^{b} M(i,j)^2} \times \sqrt{\sum_{i=1}^{a}\sum_{j=1}^{b} \hat{M}(i,j)^2}} \tag{18}$$

$$BER = \frac{\sum_{i=1}^{a}\sum_{j=1}^{b}\left[M(i,j) \oplus \hat{M}(i,j)\right]}{a \times b} \tag{19}$$

where M (i, j) and $\hat{M}$(i, j) are the original and obtained watermark information, respectively, a × b is the size of the embedded data, M (i, j) ×$\hat{M}$(i, j) represents the multiplication of the corresponding positions of the two matrices. ⊕ represents for exclusive OR. Similarity (Sim) and bit error rate (BER) have been used to evaluate the robustness performance of data hiding [43], [44]. In this section, Sim and BER are used for the evaluation of robustness of the proposed work against recompression attack and compared with the state-of-the-art literature. The maximum Sim and minimum BER can be 1 and 0 with the proposed method which has best robustness to against recompression attack.

Table 2 and Fig. 7 presents comparison of BER of test video sequences with and without selection of watermark embedding block, A represents that the watermark is embedded in the candidate blocks which are detected based on the proposed method, B represents that the watermark embedding block is randomly selected. In the experiments, the QP = 20 is chosen as the initial QP and the QP of the recompression attack during the recompression procedure is from 16 to 23. In Fig. 7, the horizontal axis represents QP and the vertical axis represents BER. From Table 2 and Fig. 7, it is found that the BER using the randomly selected watermark embedding block is much larger than embedding the watermark in the candidate block. Moreover, in the method of embedding the watermark in the candidate block, after recompression, the BER increases slowly with the increase of QP, and the overall value is relatively small. Therefore, embedding the watermark in the candidate block described in Section III greatly improves the robustness of the algorithm against recompression with different QPs.

To verify the effectiveness of the proposed algorithm, the robustness against recompression of the proposed algorithm is evaluated and compared with [15] and [29]. Fig. 8 shows the comparison between the proposed algorithm and the article [15]. To ensure the comparability of the algorithms in the experiment, the QP = 24 is chosen as the initial QP, and the QP of the recompression attack is 26, which are the same as [15]. In Fig. 8, the horizontal axis represents video sequences and the vertical axis represents robustness.

TABLE 2. Comparison of BER of Test Video Sequences With and Without Selection of Watermark Embedding Block. A Represents That the Watermark is Embedded in the Blocks Which are Detected Based on the Proposed Method; B Represents That the Watermark Embedding Block is Randomly Selected

Video sequence		QP=16	QP=17	QP=18	QP=19	QP=20	QP=21	QP=22	QP=23
Stefan	A	0.1313	0.1117	0.1352	0.1286	0.1147	0.1234	0.143	0.1608
	B	0.3554	0.3362	0.3064	0.2909	0.2118	0.2781	0.3255	0.3414
Hall	A	0.1202	0.1234	0.1169	0.1363	0.1352	0.1202	0.1034	0.1477
	B	0.3454	0.3282	0.2818	0.2499	0.2058	0.2327	0.2849	0.3041
News	A	0.1059	0.0977	0.0932	0.0947	0.1034	0.1117	0.1313	0.2086
	B	0.3584	0.3251	0.2964	0.2762	0.2357	0.2551	0.2779	0.2922
Soccer	A	0.1547	0.1391	0.1508	0.1469	0.0846	0.1253	0.1951	0.2358
	B	0.3584	0.3011	0.2752	0.2481	0.2177	0.2469	0.2941	0.3355
Foreman	A	0.1223	0.1329	0.1152	0.0864	0.0752	0.1484	0.1925	0.2253
	B	0.3487	0.3099	0.2774	0.2441	0.1917	0.2348	0.2766	0.326
Bus	A	0.1487	0.1168	0.1012	0.0925	0.0881	0.1125	0.1643	0.2011
	B	0.3558	0.3441	0.2965	0.2528	0.2081	0.2497	0.2873	0.3241

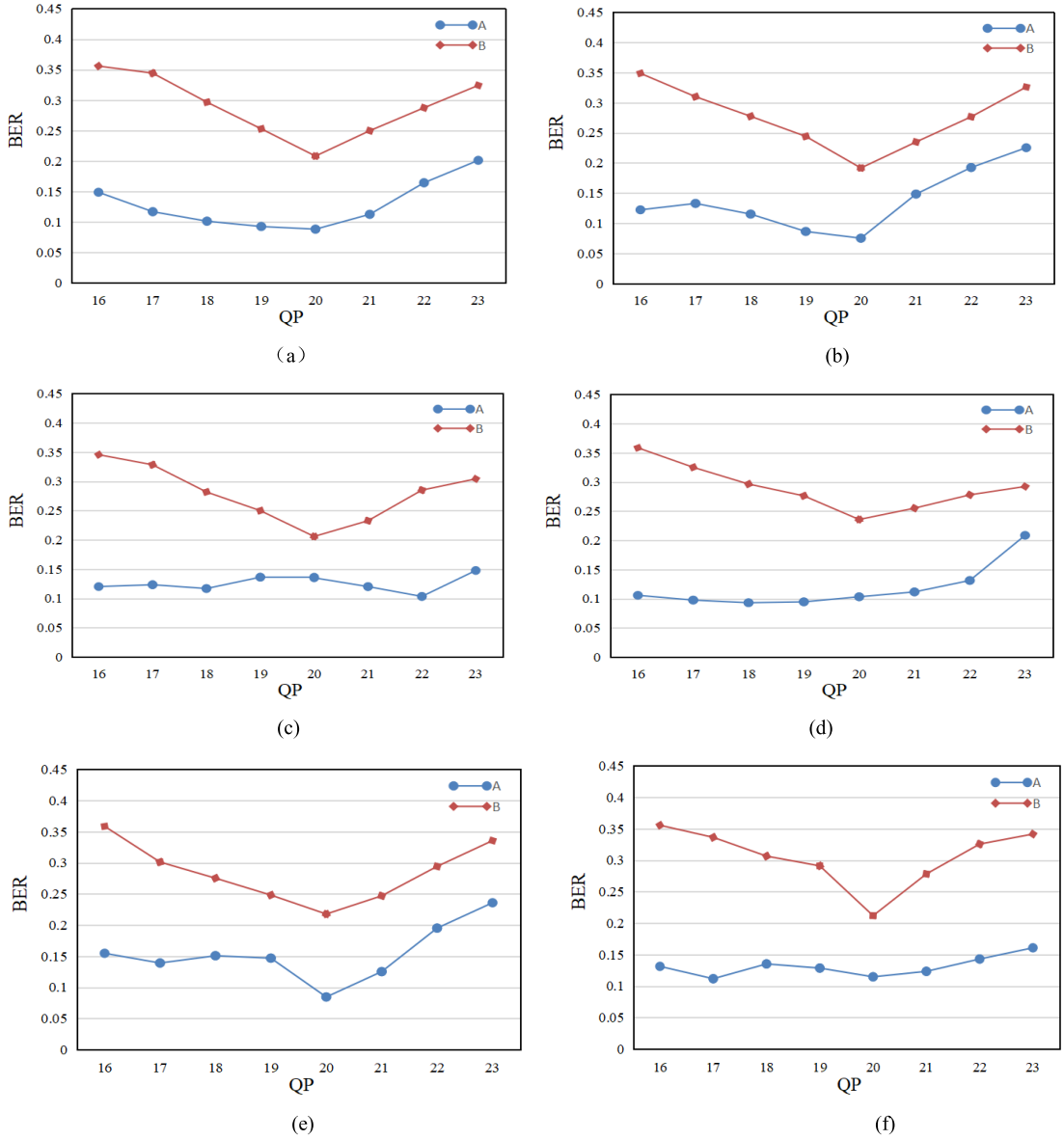

FIGURE 7. Comparison of BER on different test video sequences:(a)Bus, (b)Foreman, (c)Hall, (d)News, (e)Soccer, (f)Stefan. A represents that the watermark is embedded in the blocks which are detected based on the proposed method; B represents that the watermark embedding block is randomly selected.

The formula of robustness is as follows.

$$Robutness = 1 - BER \tag{20}$$

As can be seen from the Fig. 8 that the robustness of the proposed algorithm is better than the algorithm [15]. Like article [15], article [18] did not detect the watermark embedding area through location map, which improves the security of the algorithm. In article [18], the 4×4 blocks of QDCTs that are not all zero could be selected as watermark embedding blocks and embedding the watermark by changing the parity of the QDCTs. Nevertheless, as the change of QP is larger, the number and value of QDCTs in a 4×4 block that are not zero changed a lot, the embedding position of the watermark are easily lost. And the algorithm uses BCH code to correct the extracted watermark, but the number of error codes will exceed the error correction ability of BCH code when the QP changes greatly, therefore, it is unable to resist recompression attacks when QP increases greatly.

Fig. 9 shows the robustness comparison between the proposed algorithm and the algorithm [29]. To ensure the comparability of the algorithms in the experiment, the QP = 28 is chosen as the initial QP, and the QP of the recompression attack is 30, which are the same as [29]. In Fig. 9, the horizontal axis represents video sequences and the vertical axis represents robustness. We observed that when QP change from 28 to 30 during the process of recompression, the robustness

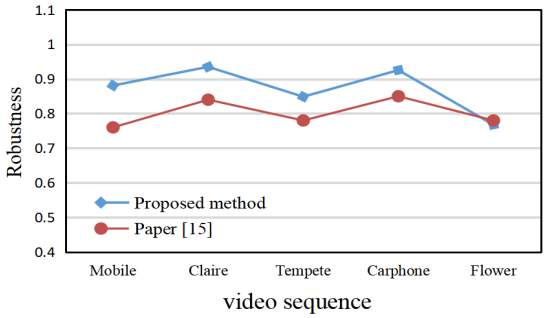

FIGURE 8. Comparing the robustness against recompression attack with in [15].

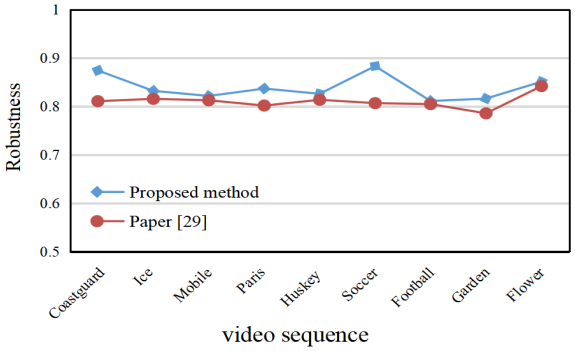

FIGURE 9. Comparing the robustness against recompression attack with in [29].

of the proposed algorithm to resist recompression attack is better than the algorithm [29]. Moreover, the algorithm [29] is to detect the watermark embedded block through the location map. The presence of the location map reduces the security of the algorithm during video transmission. The algorithms [35] [23] also uses the location map to detect the watermark embedding block. Article [35] detects the watermark embedding block through the pseudo-random sequence as the key, and then uses the invariance of the residual coefficient to embed the watermark. Article [23] embeds the watermark by modifying the prediction mode of the embedding block. The robustness of these algorithms is based on the premise that the location map is not lost and can be completely read by the recipient without tampering.

In order to verify the robustness of the proposed algorithm when the QP of recompression increases greatly, the experiment tested the robustness of the six sequences when the initial QP is 16 and the QP ranges from 8 to 32 during the process of recompression. As shown in Fig.10, the horizontal axis represents the QP ranges from 8 to 32 during the process of recompression, and the vertical axis represents Sim and Robustness, specifically, the blue histogram represents similarity, and the red histogram represents robustness. We observed that the algorithm has higher robustness and similarity when the recompressed QP is unchanged. When QP decreases, the robustness and similarity can basically reach more than 90%. When the QP increases gradually, the robustness and similarity gradually decreases. In the article [28], [23], in the case of using location map, the robustness of the algorithm reduced to around 0.6 when the QP change from 16 to 32 in recompression. And in article [35], the robustness of the algorithm reduced to around 0.7 when the QP change from 16 to 28 in recompression. Based on above analysis, it is indicated that the proposed algorithm has good robustness to resist recompression attack even without using the location map.

B. PERFORMANCE OF VISUAL QUALITY

The actual visual quality of the video sequences Stenfan and Foreman is shown in Fig.11, in it, (a) and (c) represent unwatermarked video, (b) and (d) represent watermarked video. The changes in two image will not be significant and the human eye is almost imperceptible to the watermark embedded in the video. The reason of this is that the proposed method only changes the block in the last frame in a GOP and modifies one AC coefficient in the block. On the other hand, the proposed method embeds the watermark in the vigorous moving and rich texture block of the video and these blocks will be allocated more coding resources during the compression process, so that the minor changes can be ignored.

In order to further prove the performance of visual quality, the algorithm uses the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) to calculate the degree of video quality change after embedding the watermark. PSNR is an evaluation standard that can be derived on the basis of the mean square error (MSE) to determine the similarity between two images. The unit of PSNR is db. Generally speaking, the larger the value of PSNR, the smaller the difference between the two videos. As follow in (21).

$$PSNR = 10\log_2\left(\frac{(2^n-1)^2}{MSE}\right) \tag{21}$$

Among them, MSE is the mean square error, the formula is as follows.

$$MSE = \frac{\int_{i=1}^{H}\int_{j=1}^{W}(X(i,j)-Y(i,j))^2}{H\times W} \tag{22}$$

where, H and W represent the width and height of the image, and X(i,j) represents the pixel value at the coordinates (i,j) in the image without the watermark embedded. Y(i,j) represents the pixel value at the coordinates (i,j) in the image after embedding the watermark.

The structural similarity SSIM formula is as follows:

$$SSIM(X,Y) = l(x,y) * c(x,y) * s(x,y) \tag{23}$$

where

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2+\mu_y^2+C_1} \tag{24}$$

$$c(x,y) = \frac{2\rho_x\rho_y + C_2}{\rho_x^2+\rho_y^2+C_2} \tag{25}$$

$$s(x,y) = \frac{\rho_{xy} + C_3}{\rho_x\rho_y + C_3} \tag{26}$$

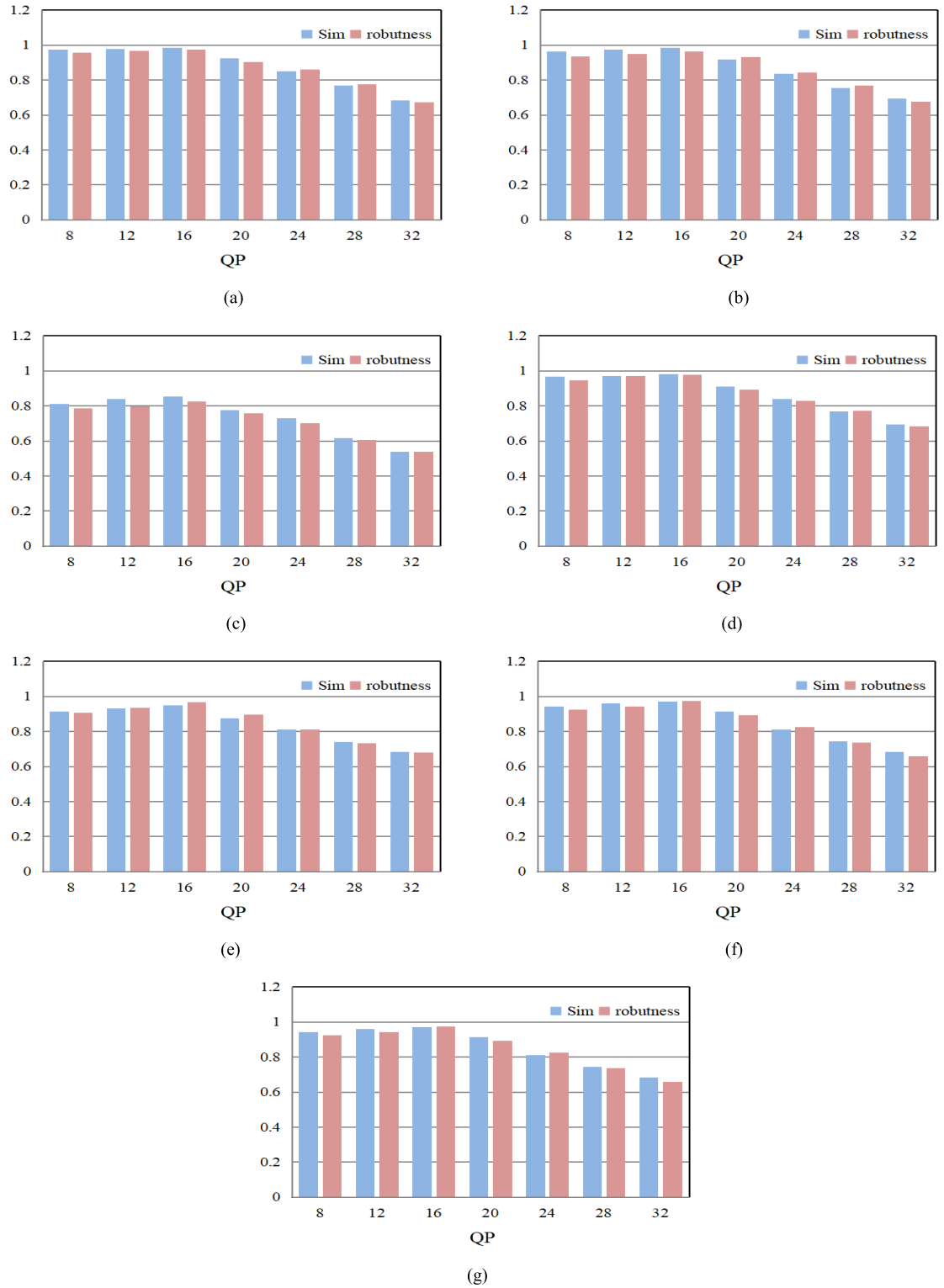

FIGURE 10. Sim and robustness against recompression with different QPs on different test video sequences:(a)Carphone, (b)Claire, (c)Flower, (d)Foreman, (e)Hall, (f)Mobile, (g)Tempete.

In it, l(x, y) calculates ligthness similarity, c(x, y) calculates contrast similarity, and s(x, y) calculates structural similarity. μ_x represents the average brightness of the image without watermark embedded, μ represents the average brightness of the image embedded with watermark, ρ_x^2 and ρ_y^2 represents the variance between the video before the watermark

FIGURE 11. Comparison of 25th frame of original video and watermarked video. (a) And (c) Represent unwatermarked video, (b) And (d) represent watermarked video.

TABLE 3. The PSNRs and SSIM of Different Video Sequences

Video sequence	PSRN	SSIM
Mobile	44.0677	0.9951
Claire	47.2399	0.9878
Tempete	45.8183	0.9903
Carphone	46.7473	0.9894
Flower	47.4142	0.9924
Basketball	44.7928	0.9805
Stefan	43.4667	0.9845
Soccer	45.8566	0.9827
Foreman	43.9074	0.9941
Tennis	42.1343	0.9878
Bus	45.7443	0.9901
Hall	42.8455	0.9763

embedding and after the watermark embedding, ρ_{xy} is the covariance between the video before the watermark embedding and after the watermark embedding.

$$\mu_x = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} X(i,j)}{H \times W} \tag{27}$$

$$\rho_x^2 = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} (X(i,j) - \mu_x)^2}{H \times W - 1} \tag{28}$$

$$\rho_{xy} = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} \left((X(i,j) - \mu_x)\left(Y(i,j) - \mu_y\right)\right)}{H \times W - 1} \tag{29}$$

C1, C2, C3 are constants. Generally, the value of SSIM is larger, the similarity between the two pictures is stronger and the distortion is lower.

Table 3 shows the PSNR and SSIM of different watermarked sequences when QP is 16. We observed that from the data in the table that the average SSIM of the video after watermark embedding is above 98%, and the PSNR are above 40db. This shows that video information changes little after embedded the watermark, which further proves the algorithm achieve the better imperceptibility.

To sum up, we can conclude from the PSNR and SSIM that this method can effectively limits the decrease in perceptual quality as well.

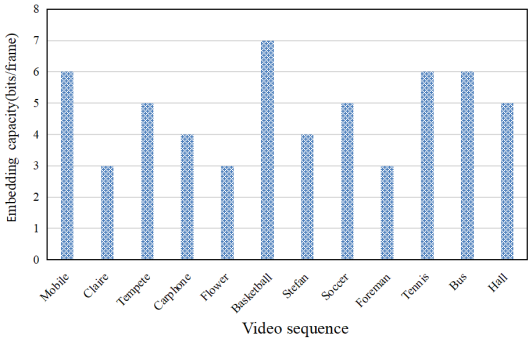

FIGURE 12. Video sequence embedding capacity.

C. EMBEDDING CAPACITY

The watermark capacity illustrates the number of watermark bits embedded in unit time or in a single video. The video embedding capacity analysis of the proposed method is shown in Fig. 12, which shows the embeded watermark bits of average one frame of different sequences. In it, the horizontal coordinate represents different video sequences, and the vertical coordinate represents the embedded bits of average each frame. In this method, the embedding capacity varies according to the video content. The watermark capacity will increase if the video has vigorous moving and rich texture. It is common that the watermark capacity, imperceptibility, and robustness are mutually restricted. According to different practical requirements, a tradeoff occurs between the capacity and the visual quality if you want to get a higher embedding capacity. The proposed method is suitable for practical application scenarios with low watermark embedding capacity and high security requirements.

D. OTHER PERFORMANCE ANALYSIS

In this section, the other performance analysis will be presented in the following. First, blind watermark: In the proposed method, the original video is not needed during the process of watermark extraction, which makes the application of this algorithm more extensive and practical; Second, anti-drift distortion: Since the algorithm embeds the watermark in the last frame of each GOP, the changes and effects on the current frame will not spread to the next frame, the drift distortion is reduced and the robustness of this algorithm is greatly improved; Third, bit rate variation: Because the embedding capacity of each frame of video is not much, and the algorithm is to embed watermark by modifying a coefficient of some blocks in a frame. Therefore, the actual video sequence has fewer bits to be changed, and the bit rate increase is lower, which can meet the limitation of channel bandwidth. Finally, security analysis: In the proposed

algorithm, the location of the watermark embedding block can be automatically generated based on the video content information, and without additional information such as location map or position template, to avoid the risk of location map being identified by attackers and faces tampered and loss during transmission. Therefore, the proposed method not only ensures the robustness of the algorithm, but also improves the security of the algorithm.

V. CONCLUSION

In the existing literature, the previous watermark schemes are fragile against recompression attack when QP increases greatly or they embeds the watermark in the specified locations while it has security issue since they need location map where extract watermark. In this work, a robust compressed domain video watermarking algorithm against recompression attack with different QPs is proposed. The major contribution of this article is that we have used the texture and motion information of the video content itself to find the optimal location of watermark embedding adaptively, the candidate blocks in the optimal location have better robustness to resist recompression attack than other areas. From the experiments and results, we can conclude that the proposed scheme improves the robustness against recompression attack than the existing schemes without location map when QP increases greatly, while it has better or similar robustness against recompression attack compared to the existing schemes which have location map. And we realize blind extraction without location map to improve the security of the proposed scheme. Besides, the proposed algorithm is based on compressed domain and embedding process is simple and fast, which has high real-time performance and is suitable for large-scale video distribution applications. Finally, since this algorithm is based on the features of the video content, it is not limited by the coding standard and can be applied to other video coding standards. Moreover, the method of finding optimal block of watermark embedding can be applied to the current mainstream video watermarking algorithms to furtherly improve the security and robustness. In the future, how to combine with the complex video coding standard in the compression domain to resist geometric attacks is the main content of our next research.

REFERENCES

[1] M. Asikuzzaman and M. R. Pickering, "An overview of digital video watermarking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2131–2153, Sep. 2018.
[2] X. Yu, C. Wang, and X. Zhou, "A survey on robust video watermarking algorithms for copyright protection," *Appl. Sci.*, vol. 8, no. 10, p. 1891, Oct. 2018.
[3] S. Bose, S. R. Chowdhury, C. Sen, S. Chakraborty, T. Redha, and N. Dey, "Multi-thread video watermarking: A biomedical application," in *Proc. Int. Conf. Circuits, Commun., Control Comput.*, Nov. 2014, pp. 242–246.
[4] R. Thanki, V. Dwivedi, K. Borisagar, and S. Borra, "A watermarking algorithm for multiple watermarks protection using RDWT-SVD and compressive sensing," *Informatica*, vol. 41, pp. 479–493, Jun. 2017.
[5] M. Asikuzzaman, M. J. Alam, A. J. Lambert, and M. R. Pickering, "Imperceptible and robust blind video watermarking using chrominance embedding: A set of approaches in the DT CWT domain," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 9, pp. 1502–1517, Sep. 2014.
[6] L. E. Coria, M. R. Pickering, P. Nasiopoulos, and R. K. Ward, "A video watermarking scheme based on the dual-tree complex wavelet transform," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 466–474, Sep. 2008.
[7] A. A. Mohammed and N. A. Ali, "Robust video watermarking scheme using high efficiency video coding attack," *Multimedia Tools Appl.*, vol. 77, no. 2, pp. 2791–2806, Jan. 2018.
[8] Y. Tan, J. Qin, X. Xiang, W. Ma, W. Pan, and N. N. Xiong, "A robust watermarking scheme in YCbCr color space based on channel coding," *IEEE Access*, vol. 7, pp. 25026–25036, 2019.
[9] R. Z. Liu and T. N. Tan, "An SVD-based watermarking scheme for protecting rightful ownership," *IEEE Trans. Multimedia*, vol. 4, no. 1, pp. 121–128, Aug. 2002.
[10] M. Jafari Barani, P. Ayubi, M. Yousefi Valandar, and B. Yosefnezhad Irani, "A blind video watermarking algorithm robust to lossy video compression attacks based on generalized Newton complex map and contourlet transform," *Multimedia Tools Appl.*, vol. 79, nos. 3–4, pp. 2127–2159, Jan. 2020.
[11] X.-L. Liu, C.-C. Lin, and S.-M. Yuan, "Blind dual watermarking for color Images' authentication and copyright protection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 5, pp. 1047–1055, May 2018.
[12] M. Noorkami and R. M. Mersereau, "A framework for robust watermarking of H.264-encoded video with controllable detection performance," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 1, pp. 14–23, Mar. 2007.
[13] J. Zhang, A. T. S. Ho, G. Qiu, and P. Marziliano, "Robust video watermarking of H.264/AVC," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 2, pp. 205–209, Feb. 2007.
[14] S. Borra, N. Dey, A. S. Ashour, and F. Shi, "Digital image watermarking tools: State-of-the-art," in *Proc. 2nd Int. Conf. Inf. Technol. Intell. Transp. Syst.*, 2017, pp. 450–459.
[15] A. Mansouri, A. M. Aznaveh, F. Torkamani-Azar, and F. Kurugollu, "A low complexity video watermarking in H.264 compressed domain," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 649–657, Dec. 2010.
[16] Y. Liu, S. Liu, H. Zhao, and S. Liu, "A new data hiding method for H.265/HEVC video streams without intra-frame distortion drift," *Multimedia Tools Appl.*, vol. 78, no. 6, pp. 6459–6486, Mar. 2019.
[17] J. Yang and S. Li, "An efficient information hiding method based on motion vector space encoding for HEVC," *Multimedia Tools Appl.*, vol. 77, no. 10, pp. 11979–12001, May 2018.
[18] Y. Liu, H. Zhao, S. Liu, C. Feng, and S. Liu, "A robust and improved visual quality data hiding method for HEVC," *IEEE Access*, vol. 6, pp. 53984–53997, 2018.
[19] X. Song, S. Lian, W. Hu, and Y. Hu, "Digital video watermarking based on intra prediction modes for audio video coding standard," *Multimedia Syst.*, vol. 20, no. 2, pp. 195–202, Mar. 2014.
[20] M. Ghasempour and M. Ghanbari, "A low complexity system for multiple data embedding into H.264 coded video bit-stream," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 4009–4019, Nov. 2020.
[21] Y. Yang, Z. Li, W. Xie, and Z. Zhang, "High capacity and multilevel information hiding algorithm based on pu partition modes for HEVC videos," *Multimedia Tools Appl.*, vol. 78, no. 7, pp. 8423–8446, Apr. 2019.
[22] G. Yang, J. Li, Y. He, and Z. Kang, "An information hiding algorithm based on intra-prediction modes and matrix coding for H.264/AVC video stream," *AEU-Int. J. Electron. Commun.*, vol. 65, no. 4, pp. 331–337, Apr. 2011.
[23] S. Gaj, A. Sur, and P. K. Bora, "Prediction mode based H. 265/HEVC video watermarking resisting re-compression attack," *Multimedia Tools Appl.*, vol. 79, no. 25, pp. 18089–18119, 2020.
[24] H. Mareen, J. De Praeter, G. Van Wallendael, and P. Lambert, "A novel video watermarking approach based on implicit distortions," *IEEE Trans. Consum. Electron.*, vol. 64, no. 3, pp. 250–258, Aug. 2018.
[25] C. Wang, R. Shan, and X. Zhou, "Anti-HEVC recompression video watermarking algorithm based on the all phase biorthogonal transform and SVD," *IETE Tech. Rev.*, vol. 35, no. sup1, pp. 42–58, Dec. 2018.
[26] X. Ma, Z. Li, H. Tu, and B. Zhang, "A data hiding algorithm for H.264/AVC video streams without intra-frame distortion drift," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 10, pp. 1320–1330, Oct. 2010.
[27] D. Xu, R. Wang, and Y. Q. Shi, "Data hiding in encrypted H.264/AVC video streams by codeword substitution," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 4, pp. 596–606, Apr. 2014.
[28] S. Gaj, A. Kanetkar, A. Sur, and P. K. Bora, "Drift-compensated robust watermarking algorithm for H.265/HEVC video stream," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 13, no. 1, pp. 11:1–11:24, 2017.

[29] T. Dutta and H. P. Gupta, "An efficient framework for compressed domain watermarking in p frames of high-efficiency video coding (HEVC)-encoded video," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 13, no. 1, pp. 12:1–12:24, 2017.
[30] Q. Sheng, R. Wang, A. Pei, and B. Wang, "An information hiding algorithm for HEVC based on differences of intra prediction modes," in *Proc. ICCCS*. Nanjing, China: Springer, 2016, pp. 63–74.
[31] A. Joshi, V. Jain, S. Ladda, and R. Kumar, "Real-time implementation of blind and robust watermarking for HEVC video coding," in *Proc. IEEE Int. Symp. Smart Electron. Syst. (iSES) (Formerly iNiS)*, Hyderabad, India, Dec. 2018, pp. 58–63.
[32] S. Gaj, A. K. Rathore, A. Sur, and P. K. Bora, "A robust watermarking scheme against frame blending and projection attacks," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 20755–20779, Oct. 2017.
[33] J. Yang and S. Li, "An efficient informaiton hiding method based on motion vector space encoding for HEVC," *Multimedia Tools Appl.*, vol. 77, no. 10, pp. 11079–12001, 2018.
[34] T. Shanableh, "Altering split decisions of coding units for message embedding in HEVC," *Multimedia Tools Appl.*, vol. 77, no. 7, pp. 8939–8953, Apr. 2018.
[35] Y. Zhou, C. Wang, and X. Zhou, "An intra-drift-free robust watermarking algorithm in high efficiency video coding compressed domain," *IEEE Access*, vol. 7, pp. 132991–133007, 2019.
[36] S. Gaj, A. Sur, and P. K. Bora, "A robust watermarking scheme against re-compression attack for H.265/HEVC," in *Proc. 5th Nat. Conf. Comput. Vis., Pattern Recognit., Image Process. Graph. (NCVPRIPG)*, Patna, India, Dec. 2015, pp. 1–4.
[37] H. R. Lakshmi and S. Borra, "Digital video watermarking tools: An overview," *Int. J. Inf. Comput. Secur.*, vol. 11, no. 1, p. 1, 2019.
[38] B. Surekha, P. R. Babu, and G. N. Swamy, "Security analysis of 'A novel copyright protection scheme using visual cryptography,'" in *Proc. Int. Conf. Comput. Commun. Technol.*, vol. 11, no. 1, Dec. 2014, pp. 1–5.
[39] T. Dougan and K. Curran, "Man in the browser attacks," *Int. J. Ambient Comput. Intell.*, vol. 4, no. 1, pp. 29–39, Jan. 2012.
[40] S. Borra, R. M. Thanki, and N. Dey, *Digital Image Watermarking: Theoretical and Computational Advances*. Boca Raton, FL, USA: CRC Press, 2018, pp. 11–22.
[41] *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification*, document ITU-T Rec. H. 264 and ISO/IEC 14496-10 AVC, Joint Video Team, 2003.
[42] S. H. Khatoonabadi, I. V. Bajić, and Y. Shan, "Compressed-domain correlates of human fixations in dynamic scenes," *Multimedia Tools Appl.*, vol. 74, no. 22, pp. 10057–10075, Nov. 2015.
[43] R. J. Mstafa, K. M. Elleithy, and E. Abdelfattah, "A robust and secure video steganography method in DWT-DCT domains based on multiple object tracking and ECC," *IEEE Access*, vol. 5, pp. 5354–5365, 2017.
[44] H. A. Aly, "Data hiding in motion vectors of compressed video based on their associated prediction error," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 14–18, Mar. 2011.

HAO DING received the Ph.D. degree from Blaise Pascal University, France, in 2012. He is currently a Lecturer with the College of Electronics and Information Engineering, South Central University for Nationalities, Wuhan, China. His research interests include multimedia processing, neural networks, and compressed sensing.

RUIXIN TAO is currently pursuing the M.S. degree with the College of Electronics and Information Engineering, South Central University for Nationalities, Wuhan, China. Her research interests include signal process and multimedia security.

JING SUN received the Ph.D. degree from Wuhan University, China, in 2013. She is currently a Postdoctoral Fellow with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. Her research interests include multimedia processing, multimedia security, and pattern recognition.

JIN LIU received the Ph.D. degree from Wuhan University, China, in 2011. He is currently a Senior Engineer with Beidou Intelligent Technology Company Ltd., Shenzhen, China. His research interests include multimedia processing, cloud computing, and big data processing.

FAN ZHANG received the Ph.D. degree from the Huazhong University of Science and Technology, China, in 2007. He is currently a Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. His research interests include cloud security, big data processing, and data mining.

XIAOPING JIANG (Member, IEEE) received the Ph.D. degree from the Huazhong University of Science and Technology, China, in 2007. He is currently an Associate Professor with the College of Electronics and Information Engineering, South Central University for Nationalities, Wuhan, China. His research interests include signal process, video analysis, and wireless communication.

JIANJIN LI received the bachelor's degree in mathematics from Wuhan University, China, in 1987, the master's degree in computer science from INSA de Lyon, in 1989, and the Ph.D. degree in computer science from University Lyon 1, France, in 1992 (Ph.D. carried out with the LIP Laboratory, Ecole Normale Supérieure de Lyon, under the supervision of Pr. Yves Robert). She is currently an Associate Professor with the Institute of Computer Science, University of Clermont Auvergne.