IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 17, 2019, accepted August 26, 2019, date of publication September 11, 2019, date of current version September 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2940366

An Intra-Drift-Free Robust Watermarking Algorithm in High Efficiency Video Coding Compressed Domain

YANGMING ZHOU, CHENGYOU WANG, (Member, IEEE), AND XIAO ZHOU
School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai 264209, China

Corresponding author: Chengyou Wang (wangchengyou@sdu.edu.cn)

This work was supported in part by the Shandong Provincial Natural Science Foundation, China, under Grant ZR2017MF020, in part by the National Natural Science Foundation of China under Grant 61702303, and in part by the Science and Technology Development Plan Project of Weihai Municipality under Grant 2018DXGJ07.

ABSTRACT The existing watermarking algorithms in H.265/high efficiency video coding (HEVC) compressed domain cannot achieve intra-frame distortion drift-free and resist the recompression and signal processing attacks simultaneously. Therefore, the application of HEVC watermarking in copyright protection is limited in practice. To solve the problem, this paper proposes an intra-drift-free robust watermarking algorithm, which uses a multi-coefficients modification method to embed the watermark into intra prediction residual pixels of 4×4 luminance transform blocks in the spatial domain. Based on the spatial-domain embedding method, watermark information can be extracted from the blocks that have been attacked and merged into 8×8 size, which improves the robustness. To further improve the performance of the watermarking algorithm, the proposed algorithm uses a combination of three thresholds according to the application requirements. Experimental results on a publicly available video database (JCT-VC) have justified the effectiveness of the proposed algorithm which has better imperceptibility and robustness than existing algorithms. In addition, the proposed algorithm has high real-time performance and is suitable for application scenarios where the video is distributed on a large scale.

INDEX TERMS Video signal processing, watermarking, robustness, H.265/high efficiency video coding (HEVC), compressed domain, intra-drift-free.

I. INTRODUCTION

With the popularity of full-high-definition (FHD) and ultra-high-definition (UHD) videos, the compression performance of H.264/advanced video coding (AVC) can no longer satisfy the coding demand. As the latest video coding standard, H.265/high efficiency video coding (HEVC), with approximately twice the compression performance compared to AVC, has gradually replaced AVC as the mainstream coding standard [1], [2]. At the same time, the severe piracy of HEVC video has had a bad impact on the video industry and needs solving urgently.

As one of the information hiding technologies, robust video watermarking identifies the ownership of the data by embedding secret information, i.e. watermark, into the video to achieve copyright protection [3]. There are two main metrics to evaluate the performance of a robust video watermarking: imperceptibility and robustness [4]. Imperceptibility means that the watermark embedded in the video cannot be perceived by people's eyes and is difficult to be detected by detection tools. That is to say, the embedding of the watermark cannot cause a significant impact on video quality. Robustness is the ability of watermarking to resist attacks, which means the embedded watermark can still be extracted from the video subjected to common signal processing attacks or malicious attacks. Real-time performance, depending on complexity, is a special requirement of video watermarking, because the embedding and extraction speed of the watermark should not be lower than the frame rate of the video. Bit increase ratio (BIR) is used to measure the increase in the bit rate. Capacity is also a criterion for evaluating watermarking. However, since the number of frames in the video is extremely large, capacity is not so important in video watermarking [5].

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

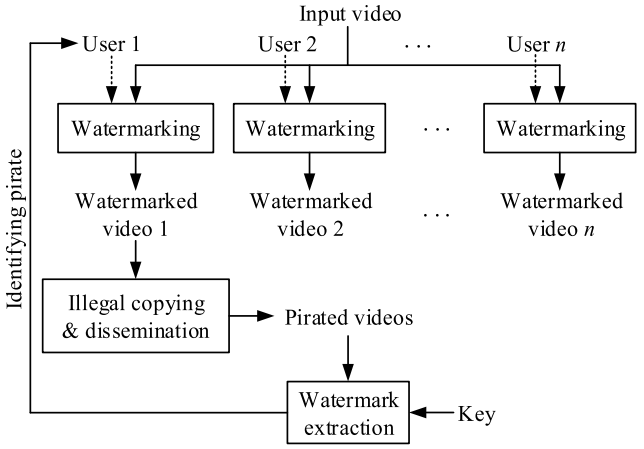


FIGURE 1. Framework of video watermarking for copyright protection and piracy tracking.

For copyright protection and piracy tracking, each video is embedded with a unique watermark so that the users who distribute pirated videos can be identified when piracy occurs, as shown in Fig. 1. Although the compression performance of HEVC is much better than that of AVC, the computational complexity of HEVC is increased. If HEVC is used to compress video for different users, computing resources will increase linearly with the number of users, which will affect the application of video watermarking in practice [6]. Compressed-domain watermarking [7]–[26], also called out-of-the-loop watermarking, directly embeds the watermarks into the compressed videos. If the input video is uncompressed, the watermark embedding process only needs to compress the video once; if the input video is compressed, the watermark can be embedded in it only by partially decoding and encoding the video, saving massive computing resources.

According to the different embedding positions of the watermark, compressed-domain watermarking can be divided into watermarking based on residual signal [7]–[22] and watermarking based on coding information [23]–[26]. Watermarking based on residual signal is embedding the watermark into the predicted discrete cosine transform (DCT) or discrete sine transform (DST) coefficients. Watermarking based on coding information is embedding watermark into prediction information, block partition information, or motion vector, etc. Compared with another kind of watermarking, watermarking based on residual signal is relatively robust and often used as robust watermarking.

Besides, the watermark can also be embedded before encoding (i.e. uncompressed-domain watermarking) [27]–[29] or during encoding (i.e. in-the-loop watermarking) [30]–[32], both completely compressing every video with high computing complexity and poor real-time performance [6]. Therefore, low-complexity and high real-time compressed-domain watermarking plays an indispensable role in the field of video watermarking, especially for HEVC videos. However, it has its own intrinsic problems, i.e., distortion drift and poor robustness to attacks, which are the current research hotspots of compressed-domain watermarking. Although considerable achievements have been made in many works [7]–[22] on these two problems, the existing watermarking algorithms still cannot solve these two problems at the same time.

In this paper, a novel robust watermarking algorithm for HEVC is proposed to eliminate distortion drift and resist various attacks simultaneously. The three main novelties and contributions in this paper are as follows:

1) The multi-coefficients modification embeds watermark information into residual pixels and enables the watermarking to achieve intra-drift-free and high robustness at the same time.
2) The residual pixels-based embedding method enables watermark bits to be extracted from the watermarked blocks that have been merged into larger blocks.
3) The three thresholds applicated can improve and balance the imperceptibility and robustness of the watermarking algorithm.

The rest of this paper is organized as follows. In Section II, a presentation of related work is given. Section III starts with a brief review of related HEVC techniques, and then analyzes the existing problem of HEVC watermarking. Section IV presents the proposed watermarking algorithm. Experimental results of the proposed algorithm are presented in Section V. Conclusions and remarks on possible further work are given finally in Section VI.

II. RELATED WORK

The distortion caused by watermark embedding propagates and accumulates during the prediction process, which will cause serious visual distortion of a video and extremely decrease the imperceptibility of watermarking. As the intra-frame distortion drift has a greater impact on video quality than the inter-frame distortion drift, this paper mainly discusses the intra-frame distortion drift (hereafter referred to as distortion drift). To reduce the effect of watermark embedding on video quality, many scholars and researchers are engaged in the research of eliminating distortion drift. In the field of AVC watermarking, Huo *et al.* [7] analyzed the relationship between distortion propagation mode and different integer transform coefficients, and used partial integer transform coefficients to compensate drift distortion. Ma *et al.* [8] defined three conditions with different prediction modes in 4×4 luminance transform blocks (LTBs), and selected the corresponding DCT paired-coefficients to embed the watermark. This method keeps the residual pixels used for prediction unchanged, thus avoiding the distortion drift. Based on [8], Lin *et al.* [9] proposed a method for embedding watermark in all 4×4 LTBs, improving the capacity of watermarking. According to the prediction mode of adjacent blocks, Liu *et al.* [10] selected the LTBs which are not the intra prediction reference blocks, and embedded the watermark into their DCT coefficients. However, watermarking for AVC cannot be directly applied to HEVC watermarking

due to the difference of coding techniques between HEVC and AVC.

In the field of HEVC watermarking, inspired by [8], the existing methods to achieve intra-drift-free mainly use the multi-coefficients modification to embed the watermark. Like [8], Liu *et al.* [11] defined three conditions according to the prediction modes of adjacent blocks and embedded the watermark into DST coefficients in 4×4 LTB using the corresponding multi-coefficients modification. Chang *et al.* [12] extended the three conditions in [11] to five, and then proposed a multi-coefficients modification method in 8×8 blocks. Gaj *et al.* [13] used the same method to embed multiple level watermark in the difference of transform coefficients between the motion coherent blocks of two consecutive I-frames, achieving better visual quality. However, the embedded watermarks are easily destroyed because robustness is not being considered in these watermarking algorithms.

Compared with uncompressed-domain watermarking, the robustness of compressed-domain watermarking is unsatisfactory due to the limitation of the coding architecture. Therefore, how to improve the robustness of watermarking in the compressed domain has become an interesting and important research field. In the field of AVC watermarking, Mansouri *et al.* [14] proposed a low-complexity robust watermarking algorithm based on the number of nonzero (NNZ) transform coefficients, defining a priority matrix to adjust the impact of watermark on bit rate and video quality. Gaj *et al.* [15] used a detection method for motion coherent regions in the compressed domain to embed the watermark into the motion objects of each shot of the video, resisting geometric attacks such as rotation and scaling. Chen *et al.* [16] proposed a robust watermarking algorithm based on just noticeable difference (JND). This algorithm adjusts the watermark strength through the JND mask, embeds the watermark into transform coefficients that do not cause intra-frame distortion drift, and uses cross-correlation detection to ensure robustness. Gong and Lu [17] divided the blocks into texture blocks, edge blocks, and smooth blocks, and embedded watermarks into them with different intensities respectively. In addition, this algorithm compensates the drift by subtracting the mean value of distortion from the watermark, which has good robustness and video quality.

In the field of HEVC watermarking, Dutta *et al.* [18] embedded the watermark invisibly into the low-frequency non-zero quantized transform coefficients (QTCs) of 4×4 LTB in I-frame, according to the temporal and spatial characteristics of HEVC video. To exploit the watermarking capacity in P-frames, Dutta and Gupta [19] proposed a bit-rate-controllable watermarking method for P-frame, whose BIR can be controlled and robustness is high. Gaj *et al.* [20] proposed a recompression-against robust watermarking algorithm, and embedded the watermark by altering the NNZ of 4×4 transform block (TB). The advantage of this algorithm is that it is still possible to extract watermark bits from those TBs correctly, although some watermarked 4×4 TBs are merged into 8×8 TBs.

From above, there have been considerable research works on watermarking in the compressed domain. However, the intra-drift-free HEVC watermarking algorithm with high robustness is relatively rare. Based on [11], Liu *et al.* [21] used the BCH code to encode the watermark data for pre-processing. When the watermark is extracted incorrectly, the BCH can be used for error correction. The algorithm not only prevents distortion drift but also has certain robustness, so it can resist the recompression attack with an unchanged quantization parameter (QP). However, the error-correcting ability of BCH is limited. If the number of error codes exceeds the error-correcting ability of BCH, BCH will be invalid. Therefore, this algorithm cannot resist recompression attacks when QP changes greatly. Besides, signal processing attacks are not considered in [21]. Another state-of-the-art algorithm was proposed by Gaj *et al.* [22]. The algorithm first applies the inverse transform to the dequantized DST coefficients of 4×4 LTB, and obtains a residual matrix. Then the watermark is embedded in the 3×3 residual matrix that locates in the top left corner which does not work as a reference, and the 4×4 residual matrix is re-transformed and re-quantized. Although the algorithm has good robustness, the distortion drift is not completely avoided. The reason is that the re-quantization will lead to information loss, resulting in minor distortion of the reference pixels, and then the distortion will propagate to the neighboring blocks by prediction process.

III. RELATED TECHNIQUES AND PROBLEM ANALYSIS

A. HEVC ARCHITECTURE

Compared with AVC, the coding architecture of HEVC, which still adopts a similar hybrid coding architecture, has not changed fundamentally. The portion of the red dotted frame in Fig. 2 shows the block diagram of HEVC encoder, mainly including prediction, transform, quantization, entropy encoding, and other modules. A compressed video consists of the residual signal, prediction information, and other coding information. Although the architecture of HEVC is similar to that of AVC, HEVC introduces new coding techniques in almost every module, greatly improving compression performance.

The core coding unit in the AVC standard is a macroblock whose maximum size is 16×16 [1]. With the increase of video resolution, AVC's limitations of coding methods based on conventional macroblock are becoming more and more obvious. Hence, HEVC applies a coding tree unit (CTU), similar to a macroblock in concept, but its size can be set by an encoder, which can exceed 16×16. To represent video content more flexibly and effectively, HEVC defines a series of new video frame partitioning modes, including coding unit (CU), prediction unit (PU), and transform unit (TU). Each CTU is divided into several CUs via the form of quadtree, and each CU contains PUs and TUs associated

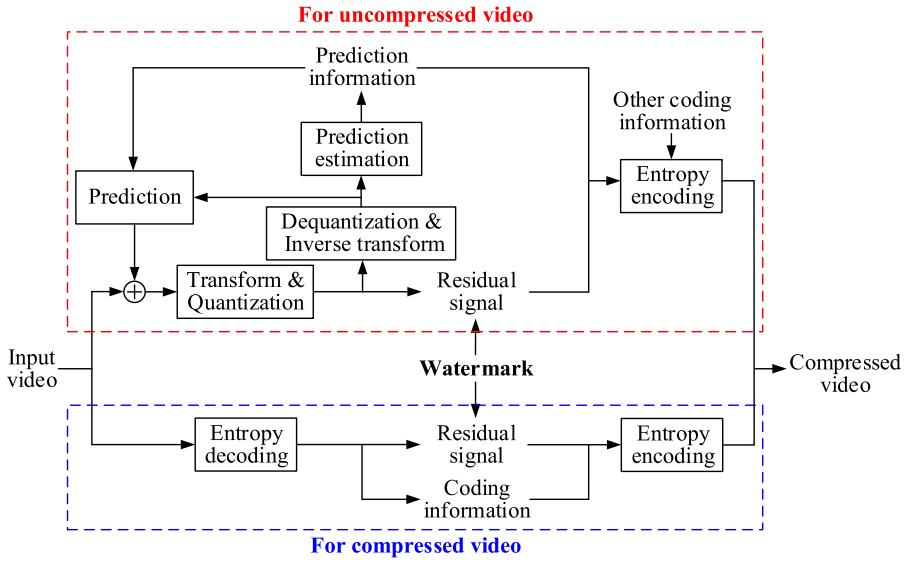


FIGURE 2. Block diagram of watermarking based on residual signal in the compressed domain.

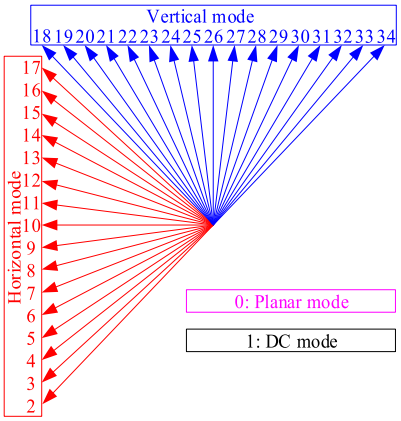


FIGURE 3. Prediction modes of 4 × 4 luminance block in HEVC.

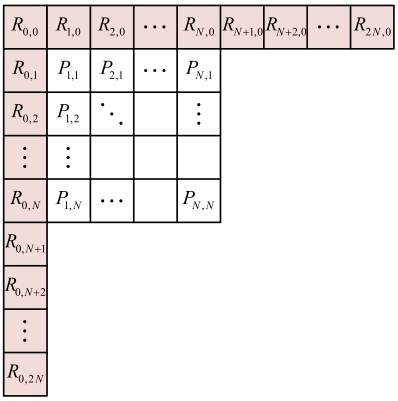


FIGURE 4. HEVC intra prediction template.

with it. It is worth noting that a unit of each type consists of a luminance block, two chroma blocks, and some associated syntax elements. For instance, a PU consists of a luminance prediction block (PB) and two chroma PBs. Different from AVC which only uses 4 × 4 and 8 × 8 DCT transform, TB in HEVC adopts DCT of 4 × 4 to 32 × 32 size, and supports 4 × 4 DST for 4 × 4 LTB. Large-size DCT is one of the important techniques to improve coding efficiency in HEVC, which enables the encoder to enhance the compression ratio when processing flat regions that often appear in high-resolution videos [1].

In AVC, there are only 9 prediction modes for intra prediction, whereas HEVC provides 35 intra prediction modes for 4 × 4 luminance block, including 33 angle predictions, DC prediction mode, and Planar prediction mode [33], as shown in Fig. 3. The increased prediction modes can match the complex textures in the video better, get enhanced prediction effects, and remove spatial redundancy more effectively. The prediction modes are defined based on the PB, but the practical intra prediction process is based on the TB. The standard stipulates that PB can be divided into TBs in the form of quadtree, and TBs in the same PB share the same prediction mode. In the actual prediction process, each TB is predicted by the pixels around itself as reference. The intra prediction template is shown in Fig. 4. $R_{x,y}$ represents a reconstruction pixel of the neighboring blocks for $x = 0, 1, \cdots, 2N, y = 0$ and $x = 0, y = 1, 2, \cdots, 2N$, where x and y are horizontal and vertical coordinates of the pixel, respectively, and N is the size of the current block. $P_{x,y}$ represents a prediction pixel of the current block for $x, y = 1, 2, \cdots, N$ based on reference $R_{x,y}$ [33].

After 4 × 4 luminance block is predicted by intra prediction, HEVC uses integer DST to transform the residual, with the integer DST transform matrix $\boldsymbol{H}$ is [1]:

$$\boldsymbol{H} = \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix}. \quad (1)$$

The correction matrix $\boldsymbol{E}$ used in the scaling process is a matrix of the same size as $\boldsymbol{H}$ whose all elements are 1/128. In addition, the scaling and the quantization are performed together in HEVC encoder. The process of DST transform and quantization can be written as follows:

$$\begin{aligned}\boldsymbol{Y} &= (\boldsymbol{H}\boldsymbol{X}\boldsymbol{H}^{\mathrm{T}}) \cdot \boldsymbol{E} \cdot \boldsymbol{E}^{\mathrm{T}}/Q_{\mathrm{step}} \\ &= \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix} \boldsymbol{X} \begin{bmatrix} 29 & 74 & 84 & 55 \\ 55 & 74 & -29 & -84 \\ 74 & 0 & -74 & 74 \\ 84 & -74 & 55 & -29 \end{bmatrix} \\ &\quad \times \frac{1}{128} \times \frac{1}{128}/Q_{\mathrm{step}}, \end{aligned} \tag{2}$$

where "$\cdot$" represents element-wise multiply, $\boldsymbol{X}$ represents the original residual pixel matrix, $\boldsymbol{Y}$ represents the quantized DST coefficient matrix, and Q_{step} represents the quantization step corresponding to a QP. The relationship between Q_{step} and QP can be given by:

$$Q_{\mathrm{step}} \approx 2^{(\mathrm{QP}-4)/6}. \tag{3}$$

At the decoder, $\boldsymbol{Y}$ is inversely quantized and inversely transformed to obtain the residual pixel matrix $\boldsymbol{X}'$ as follows:

$$\begin{aligned}\boldsymbol{X}' &= (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{H}) \cdot \boldsymbol{E} \cdot \boldsymbol{E}^{\mathrm{T}} \times Q_{\mathrm{step}} \\ &= \begin{bmatrix} 29 & 74 & 84 & 55 \\ 55 & 74 & -29 & -84 \\ 74 & 0 & -74 & 74 \\ 84 & -74 & 55 & -29 \end{bmatrix} \boldsymbol{Y} \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix} \\ &\quad \times \frac{1}{128} \times \frac{1}{128} \times Q_{\mathrm{step}}, \end{aligned} \tag{4}$$

where $\boldsymbol{X}' \neq \boldsymbol{X}$ because of the quantization loss. It should be noted that the operations in HEVC are all integer operations, and the results are downward rounded. In the practical coding process of HEVC, division is often replaced by a bit shift operation.

This section gives a brief introduction of the HEVC techniques related to the watermark in this paper and other techniques are not described in detail.

B. EXISTING PROBLEM

The application of various efficient new techniques in HEVC, although enhancing compression performance, has brought greater difficulties to watermarking. Compared with AVC, HEVC watermarking is more prone to distortion drift and less likely to have high robustness. As presented in Section II, the recent works aiming to solve the two problems simultaneously for HEVC are only [21] and [22].

Fig. 2 shows the block diagram of watermarking based on residual signal in the compressed domain, where the portion of the red dotted frame is for the uncompressed video and the blue is for the compressed video. The embedding positions are 4 × 4 luminance blocks of residual signal for uncompressed or compressed video. After embedding the watermark in the residual signal, the residual signal and the original encoding information are entropy encoded without complex re-prediction estimation, and the compressed video with the watermark is obtained. It means that the decoder will directly use original prediction information and watermarked residual to reconstruct frames. When current block pixels are distorted due to watermark embedding, and neighboring block uses the distorted pixels as the reference pixels, the distortion drift occurs.

To avoid distortion drift, Liu *et al.* [21] analyzed the modification conditions of coefficients without distortion drift, and pointed out that the multi-coefficients conditions which an additive watermark matrix $\boldsymbol{W}_{\mathrm{mc}}$ needs to satisfy are as follows:

$$\boldsymbol{W}_{\mathrm{mc}} = \begin{bmatrix} \delta_{11} & 0 & -\delta_{11} & \delta_{11} \\ \delta_{21} & 0 & -\delta_{21} & \delta_{21} \\ \delta_{31} & 0 & -\delta_{31} & \delta_{31} \\ \delta_{41} & 0 & -\delta_{41} & \delta_{41} \end{bmatrix} \text{ or } \begin{bmatrix} \delta_{11} & \delta_{12} & \delta_{13} & \delta_{14} \\ 0 & 0 & 0 & 0 \\ -\delta_{11} & -\delta_{12} & -\delta_{13} & -\delta_{14} \\ \delta_{11} & \delta_{12} & \delta_{13} & \delta_{14} \end{bmatrix}, \tag{5}$$

where δ_{i1} is the watermark strength in the i-th row of the first, third, and fourth columns of the matrix, and δ_{1j} is the watermark strength in the j-th column of the first, third, and fourth rows of the matrix. The coefficients in the third and fourth columns or the third and fourth rows are compensation coefficients, whose absolute value is equal to δ_{i1} or δ_{1j}. Then, the error matrices $\boldsymbol{\Delta}$ of decoded residual can be written as follows:

$$\boldsymbol{\Delta} = \begin{bmatrix} 0 & 0 & e_{13} & 0 \\ 0 & 0 & e_{23} & 0 \\ 0 & 0 & e_{33} & 0 \\ 0 & 0 & e_{43} & 0 \end{bmatrix} \text{ or } \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ e_{31} & e_{32} & e_{33} & e_{34} \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{6}$$

where e_{ij} is the error of pixel in the i-th row of the j-th column. If the blocks adjacent to the watermarked block do not use the last row or the last column for prediction, the distortion will not propagate.

However, the algorithm [21] embedding the watermark by changing the parity of the QTCs is unable to resist the signal processing attacks and recompression attacks with changed QPs. According to (2), the change of Q_{step} will change the QTCs, resulting in the damage of the watermark based on parity. The signal processing attacks can affect the quality of the uncompressed video, and also change the parity of QTCs.

In addition, as discussed in Section III.A, the types and sizes of blocks in HEVC are more abundant and flexible than those in AVC, so it tends to cause the loss of watermark information. Since HEVC uses a rate-distortion (RD) optimization [1] to partition blocks and select prediction modes in each recompression, residual signals may be completely changed. When the video is attacked, part of the 4 × 4 TBs will be repartitioned and merged into 8 × 8 or even larger TBs, resulting in the watermark bits not being able to extract from them. Furthermore, if the watermark detection algorithm fails to recognize the watermarked blocks which have been merged into larger blocks, the subsequently extracted watermark bits will not synchronize with the embedded watermark bits,

which is called synchronization error in this paper. Besides, the change of prediction modes also affects the accuracy of the watermark extraction. In [22], the watermark is embedded by changing the relationship of size between two specific transform coefficients in high-NNZ LTBs, which is relatively robust. However, the drift-compensate method in [22] cannot make the watermarking drift-free.

From above, the problem is that the existing algorithms cannot meet the requirements of imperceptibility and robustness at the same time, which is the motivation of this paper.

IV. PROPOSED VIDEO WATERMARKING ALGORITHM

To solve the problem in Section III.B, based on the research and analysis of existing algorithms and HEVC standard, the proposed method embedding watermark into the predicted residual pixels is presented in Section IV.A. Section IV.B and Section IV.C describe the watermark embedding and extraction processes, respectively.

A. PROPOSED WATERMARK EMBEDDING METHOD

In some works, like [14], [18], and [22], the LTBs with higher NNZ are selected to be embedded by the watermark, because these blocks have lower probabilities of changes in block partition sizes and prediction modes after recompression. Moreover, the higher NNZ usually means the more complicated the texture. The human visual system (HVS) is less sensitive to the high-frequency regions with complex texture, where the distortion is not easy to be perceived. The algorithm [21] needs to select blocks that satisfy the condition of prediction mode. If combined with the NNZ-based block selection, the capacity of watermarking will be limited. Therefore, the proposed algorithm uses a matrix $\boldsymbol{W}$ which satisfies both the conditions in (5) as follows [11]:

$$\boldsymbol{W} = \begin{bmatrix} \delta & 0 & -\delta & \delta \\ 0 & 0 & 0 & 0 \\ -\delta & 0 & \delta & -\delta \\ \delta & 0 & -\delta & \delta \end{bmatrix}, \tag{7}$$

where δ is the watermark strength. In the additive embedding scheme of this paper, the embedding process of watermarking can be expressed as:

$$\boldsymbol{Y}_{\mathrm{w}} = \boldsymbol{Y} + \boldsymbol{W}, \tag{8}$$

where $\boldsymbol{Y}_{\mathrm{w}}$ represents the watermarked quantized DST coefficient matrix. Then, the errors $\boldsymbol{\Delta}$ caused by the embedded watermark in the spatial domain can be quantitatively calculated according to (4) and (8) as follows:

$$\begin{aligned} \boldsymbol{\Delta} &= (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{Y}_{\mathrm{w}}\boldsymbol{H}) \cdot \boldsymbol{E} \cdot \boldsymbol{E}^{\mathrm{T}} \times Q_{\mathrm{step}} - (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{H}) \cdot \boldsymbol{E} \cdot \boldsymbol{E}^{\mathrm{T}} \times Q_{\mathrm{step}} \\ &= (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}) \cdot \boldsymbol{E} \cdot \boldsymbol{E}^{\mathrm{T}} \times Q_{\mathrm{step}} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 3\delta Q_{\mathrm{step}} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{9}$$

It can be seen that the errors are controllable and only appear on the pixel in the third row of the third column of the 4×4 LTB. The residual pixel matrix $\boldsymbol{X}'$ can be written as:

$$\boldsymbol{X}' = \begin{bmatrix} x'_{11} & x'_{12} & x'_{13} & x'_{14} \\ x'_{21} & x'_{22} & x'_{23} & x'_{24} \\ x'_{31} & x'_{32} & x'_{33} & x'_{34} \\ x'_{41} & x'_{42} & x'_{43} & x'_{44} \end{bmatrix}, \tag{10}$$

In (10), the only distorted element is x'_{33}, which will not be used as a reference for intra prediction in any condition. The disadvantage of this 9-coefficients modification is that if δ is large, then the distortion $\boldsymbol{\Delta}$ will be tremendous, even exceeding the boundary of the pixel value. The existing robust watermarking algorithms like [14], [18], and [22], embed watermarks by changing the size relationship among QTCs, usually requiring a large degree of modification in coefficients. For example, when $\delta = 5$ and $\mathrm{QP} = 16$, the distortion in the spatial domain is:

$$\boldsymbol{\Delta} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 60 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{11}$$

Due to excessive distortion, the visual quality of video will be significantly degraded, and the imperceptibility cannot be guaranteed. As a result, the watermarking based on the size relationship among QTCs is not suitable to use the 9-coefficients modification method.

From above, we have known that small change of the values of QTCs will lead to a large change of residual in the spatial domain. Moreover, Section III has illustrated that the flexible block partitioning and prediction mode in HEVC make transform coefficients difficult to be robust. Therefore, this paper proposes embedding the watermark information into the residuals in the spatial domain rather than into transform coefficients. It is noteworthy that embedding watermark into spatial residual is realized via modifying QTCs.

Because only the pixel x'_{33} will be altered, we consider finding a feature and altering the size of x'_{33} to change the relationship between x'_{33} and the feature of the current block to embed the watermark information. However, after recompression, the prediction mode may change, leading to the change of the prediction values of the pixels in the current block and destroying the relationship among the pixels. Therefore, the chosen relationship between x'_{33} and the feature needs to be robust to recompression and the change of prediction values. A useful way is to find a feature whose prediction value is close to that of x'_{33}. To determine this feature, the related theory of HEVC intra prediction is further analyzed.

Taking angle mode 27 in a 4×4 block as an example, the intra prediction pixel values determination in HEVC using the template in Fig. 4 is shown in Fig. 5. To improve the prediction accuracy, HEVC interpolates the reference pixel of the corresponding prediction direction and the next reference pixel adjacent to it with a precision of 1/32 to get the prediction value $P_{x,y}$. The interpolation is performed linearly

as follows [33]:

$$P_{x,y} = [(32 - w_y) \times R_{i,0} + w_y \times R_{i+1,0} + 16]/32, \quad (12)$$

where w_y is the weighting between the two reference pixels $R_{i,0}$ and $R_{i+1,0}$. Reference index i and weighting parameter w_y are calculated based on the displacement d corresponding to the selected prediction direction as:

$$c_y = (y \times d)/32, \quad (13)$$
$$w_y = (y \times d)\&31, \quad (14)$$
$$i = x + c_y, \quad (15)$$

where & denotes a bitwise AND operation. According to (12) to (15), the prediction value $P_{3,3}$ is obtained as an interpolation of $R_{3,0}$ and $R_{4,0}$. $P_{3,2}$ and $P_{3,4}$ on the same line as $P_{3,3}$ are obtained by interpolation with the same reference pixels and will be very close to $P_{3,3}$. However, the change of prediction direction will make $P_{3,2}$ and $P_{3,4}$ deviate from the prediction line of $P_{3,3}$ and not close to $P_{3,3}$. For other prediction values around $P_{3,3}$ (in the yellow region of Fig. 5), their reference pixels are adjacent to the reference pixels of $P_{3,3}$. As adjacent pixels in the spatial domain have a certain correlation, these prediction values are also close to $P_{3,3}$, though not as close as $P_{3,2}$ and $P_{3,4}$ to $P_{3,3}$. The mean value of the eight prediction values around $P_{3,3}$ is marked as $\bar{P}$. Fig. 6 shows the probability density of the difference of $P_{3,3}$ and $\bar{P}$ of all 4×4 TBs in the first 20 frames of the compressed video BasketballPass with QP $= 16$. It shows that most of the differences of $P_{3,3}$ and $\bar{P}$ are in range $[-10, 10]$, and $\mathrm{P}(|P_{3,3} - \bar{P}| \leq 10) = 0.900$. Moreover, $\mathrm{P}(P_{3,3} = \bar{P}) = 0.251$ is the highest probability in Fig. 6. Therefore, the relationship between $P_{3,3}$ and $\bar{P}$ is generally stable.

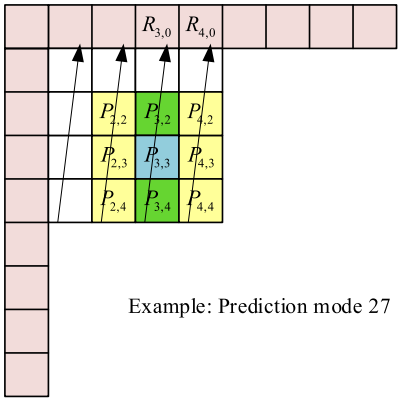


FIGURE 5. Intra prediction pixel values determination of prediction mode 27 in HEVC.

Based on the analysis above, we found that the difference of the residual pixel x'_{33} and the mean value $\bar{x}'$ of the eight pixels around x'_{33} is robust and appropriate for watermark embedding.

For example, as shown in Fig. 7, after recompression, the prediction mode is changed, resulting in changes in residual pixels and QTCs. However, the mean value $\bar{x}'$ in the brown

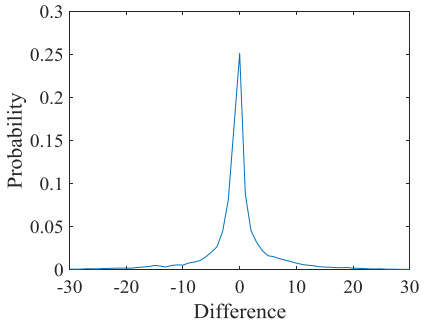


FIGURE 6. Probability density of the difference of $P_{3,3}$ and $\bar{P}$ of all 4×4 TBs in the first 20 frames of the compressed video BasketballPass with QP $= 16$.

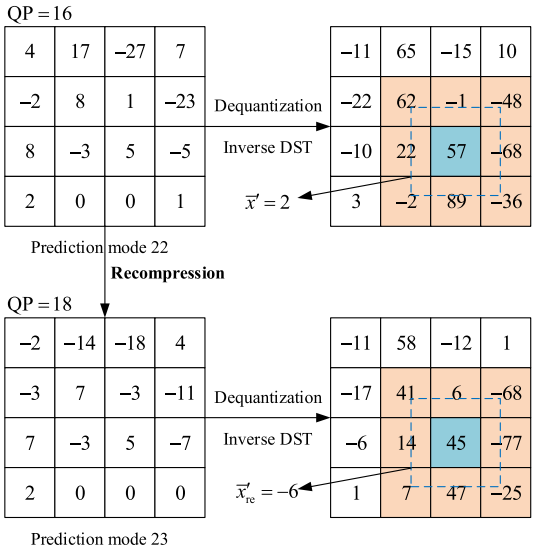


FIGURE 7. A 4×4 block from the first frame of BasketballPass before and after recompression.

region of the original residual is 2 and the mean value $\bar{x}'_{\mathrm{re}}$ after recompression is -6, so the difference of the original residual is 55 and after recompression, it is 51. The differences in the original residual and the recompressed residual are close, even when the residual pixels themselves are very different. Based on this research, a watermark embedding method exploiting the relationship between the residual pixel and the mean value of pixels is proposed in this paper.

Furthermore, based on the proposed embedding method, watermark bits can be extracted from the watermarked blocks whose size becomes 8×8 after recompression. Fig. 8 shows the watermark extraction from the merged 8×8 blocks embedded by an embedding method based on transform coefficient and the proposed embedding method based on residual pixel, where T and Q represent the DST transform and quantization respectively, RP represents repartition, T^{-1} and Q^{-1} represent the inverse DCT transform and dequantization respectively, and w and $\tilde{w}$ represent the embedded and extracted watermark bit respectively.

Suppose that $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$, $\boldsymbol{Y}_3$, and $\boldsymbol{Y}_4$ are four coefficient matrices in the same CU. Then the watermarked coefficient

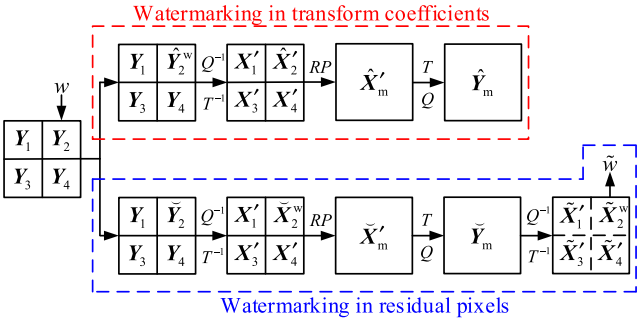


FIGURE 8. Watermark extraction from the merged 4 × 4 blocks embedded by an embedding method based on transform coefficient and the proposed embedding method based on residual pixel.

matrices $\hat{\boldsymbol{Y}}_2^{\mathrm{w}}$ and $\breve{\boldsymbol{Y}}_2$ are obtained by embedding the watermark bit into $\boldsymbol{Y}_2$ using the method based on transform coefficient and the proposed method based on residual pixel respectively, where the superscript w indicates that the watermark bit can be extracted directly.

As shown in the portion of the red dotted frame in Fig. 8, in the method based on transform coefficient, the four coefficient matrices are inverse transformed and dequantized to residual matrices $\boldsymbol{X}'_1$, $\hat{\boldsymbol{X}}'_2$, $\boldsymbol{X}'_3$, and $\boldsymbol{X}'_4$. After repartition, the four 4 × 4 residual matrices in the CU are replaced by an merged 8 × 8 residual matrix $\hat{\boldsymbol{X}}'_{\mathrm{m}}$. Then, $\hat{\boldsymbol{X}}'_{\mathrm{m}}$ is transformed and quantized to an 8 × 8 quantized DCT coefficient matrix $\hat{\boldsymbol{Y}}_{\mathrm{m}}$ from which the watermark bit cannot be extracted.

For the embedding method based on residual pixel in the blue dotted frame, although the watermark bit is also embedded by modifying the coefficients, the watermark information exists in the residual matrix $\breve{\boldsymbol{X}}_2^{\mathrm{w}}$. After recompression, the 8 × 8 quantized DCT coefficient matrix $\breve{\boldsymbol{Y}}_{\mathrm{m}}$ in the merged block is obtained. To extract the watermark bit, the 8 × 8 coefficient matrix $\hat{\boldsymbol{Y}}_{\mathrm{m}}^{\mathrm{w}}$ is inversely transformed and dequantized to get the 8 × 8 residual pixel matrix, and then the obtained 8 × 8 matrix is directly divided into four 4 × 4 submatrices. Since the relationship between x'_{33} and $\bar{x}'$ is robust to the change of prediction values, the watermark information may exist even if the repartition results in the change of predicted pixels. Thus, the watermark bit can be extracted from the 4 × 4 submatrix of the corresponding embedding position.

For larger merged blocks, such as blocks with sizes of 16 × 16 and 32 × 32, the predicted values may change greatly because the pixels on the lower or right side of the blocks are far from the reference pixels. Because the possibility of correctly extracting watermark bits is too low and the extraction process increases the computational complexity of the algorithm, the proposed algorithm does not extract watermark bits from these blocks.

B. WATERMARK EMBEDDING

As discussed in Section IV.A, the 4 × 4 LTBs whose NNZ is higher than a threshold N_{th} in I-frame are selected. To improve the security of the proposed algorithm, a pseudo-random number generator is used to generate a pseudo-random sequence as a key and select the candidate blocks. For each candidate block, dequantization and inverse DST functions in HEVC are applied to the QTC matrix $\boldsymbol{Y}$ of the candidate block to get residual matrix $\boldsymbol{X}'$. Then, the mean value $\bar{x}'$ of the eight pixels around the residual pixel x'_{33} in $\boldsymbol{X}'$ is calculated. When the watermark bit to be embedded is 1, if the difference of x'_{33} and $\bar{x}'$ is less than or equal to a threshold R_{th}, then the 9-coefficients modification is used to make the difference greater than R_{th}, otherwise no modification is needed. When the watermark bit to be embedded is 0, if the difference of $\bar{x}'$ and x'_{33} is less than R_{th}, then make the difference greater than or equal to R_{th}, otherwise no modification is needed. The application of the threshold R_{th} can improve the robustness at the cost of the imperceptibility.

Since the distortion only occurs on x'_{33}, if the embedding strength of the watermark is too high, it will produce distortion similar to Salt & Pepper noise, which will decrease the imperceptibility subjectively. To solve this problem, a threshold α_{th} is set. If the distortion caused by embedding the watermark bit in the current block exceeds $3\alpha_{\mathrm{th}}Q_{\mathrm{step}}$, the watermark will not be embedded in this block.

To avoid synchronization error caused by block size change, a location map $\boldsymbol{L}_{\mathrm{w}}$ is used to record the locations of the watermarked LTBs after embedding the watermark.

The watermark embedding algorithm is described in Algorithm 1. Note that the operations in the watermark embedding algorithm are integer operations.

C. WATERMARK EXTRACTION

The process of watermark extraction is very simple. First, according to the recorded location in $\boldsymbol{L}_{\mathrm{w}}$, the 4 × 4 watermarked LTBs are found. For each watermarked LTB, the dequantization and inverse DST functions in HEVC are applied to $\boldsymbol{Y}_{\mathrm{w}}$ and the watermarked residual $\boldsymbol{X}_{\mathrm{w}}$ is obtained. When the watermarked block determined by $\boldsymbol{L}_{\mathrm{w}}$ is merged into an 8 × 8 block, the dequantization and inverse DCT are applied to the 8 × 8 block, and an 8 × 8 residual matrix is obtained. Then the matrix is divided into four 4 × 4 matrices from which the watermark bits are extracted. The mean value $\bar{x}_{\mathrm{w}}$ in $\boldsymbol{X}_{\mathrm{w}}$ is calculated and compared with x_{33}^{w} or $\hat{x}_{33}^{\mathrm{w}}$. If x_{33}^{w} or $\hat{x}_{33}^{\mathrm{w}}$ is greater than $\bar{x}_{\mathrm{w}}$, the extracted watermark bit is 1; otherwise, the extracted watermark bit is 0.

Although the watermark extraction is not blind, it only needs the location information of the watermarked blocks with a small amount of data, but does not need the huge original video data or the original watermark data.

The watermark extraction algorithm is described in Algorithm 2.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, imperceptibility, robustness, BIR, and real-time performance are evaluated by several experiments. To prove the effectiveness of the proposed algorithm, standard test video sequences with different resolutions and textures [34] are used in experiments and a binary random

Algorithm 1 Watermark Embedding

Input: original video $\boldsymbol{V}$, watermark sequence $\boldsymbol{w}$
Output: watermarked video $\boldsymbol{V}_{\mathrm{w}}$, location map $\boldsymbol{L}_{\mathrm{w}}$
Watermark Embedding Procedure:
for each I-frame $\boldsymbol{F}_i$ of $\boldsymbol{V}$ **do**
for each 4×4 LTB $\boldsymbol{B}_j$ in $\boldsymbol{F}_i$ **do**
Calculate the NNZ value N_j of $\boldsymbol{B}_j$;
Generate a pseudorandom bit p_r using a key;
if $N_j \geq N_{\mathrm{th}}$ and $p_k = 1$ **then**
Apply dequantization and inverse DST functions in HEVC to $\boldsymbol{Y}$ and obtain the residual $\boldsymbol{X}'$;
Calculate the mean value $\bar{x}'$ of the eight pixels around the residual pixel x'_{33} in the third row of the third column of $\boldsymbol{X}'$;
Set a flag bit b_{f} indicating whether the watermark bit has been embedded;
if $w_k = 1$ **then**
if $R_{\mathrm{th}} - 3\alpha_{\mathrm{th}} Q_{\mathrm{step}} < x'_{33} - \bar{x}' \leq R_{\mathrm{th}}$ **then**
$\delta = (\bar{x}' - x'_{33} + R_{\mathrm{th}})/3Q_{\mathrm{step}} + 1$;
$Y_{\mathrm{w}} = Y + W$ $(x^{\mathrm{w}}_{33} = x'_{33} + 3\delta Q_{\mathrm{step}})$;
$b_{\mathrm{f}} = 1$;
elseif $x'_{33} - \bar{x}' > R_{\mathrm{th}}$ **then**
$b_{\mathrm{f}} = 1$;
else
$b_{\mathrm{f}} = 0$;
end if
end if
if $w_k = 0$ **then**
if $R_{\mathrm{th}} - 3\alpha_{\mathrm{th}} Q_{\mathrm{step}} \leq \bar{x}' - x'_{33} < R_{\mathrm{th}}$ **then**
$\delta = (x'_{33} - \bar{x}' + R_{\mathrm{th}} - 1)/3Q_{\mathrm{step}} + 1$;
$Y_{\mathrm{W}} = Y - W$ $(x^{\mathrm{w}}_{33} = x'_{33} - 3\delta Q_{\mathrm{step}})$;
$b_{\mathrm{f}} = 1$;
elseif $\bar{x}' - x'_{33} \geq R_{\mathrm{th}}$ **then**
$b_{\mathrm{f}} = 1$;
else
$b_{\mathrm{f}} = 0$;
end if
end if
if $b_{\mathrm{f}} = 1$ **then**
Record location in $\boldsymbol{L}_{\mathrm{w}}$;
end if
end if
end for
end for
End Procedure

Algorithm 2 Watermark Extraction

Input: watermarked video $\boldsymbol{V}_{\mathrm{w}}$, location map $\boldsymbol{L}_{\mathrm{w}}$
Output: extracted watermark sequence $\boldsymbol{w}'$
Watermark Extraction Procedure:
Read $\boldsymbol{L}_{\mathrm{w}}$ for watermark location;
for each I-frame $\hat{\boldsymbol{F}}_i$ of $\boldsymbol{V}_{\mathrm{w}}$ **do**
for each watermarked LTB $\hat{\boldsymbol{B}}_j$ in $\hat{\boldsymbol{F}}_i$ **do**
if the size of $\hat{\boldsymbol{B}}_j$ is 4×4 **then**
Apply dequantization and inverse DST functions in HEVC to $\boldsymbol{Y}_{\mathrm{w}}$ and obtain the residual $\boldsymbol{X}_{\mathrm{w}}$;
Calculate the mean value $\bar{x}_{\mathrm{w}}$ of the eight pixels around the residual pixel x^{w}_{33} in the third row of the third column of $\boldsymbol{X}_{\mathrm{w}}$;
if $x^{\mathrm{w}}_{33} - \bar{x}_{\mathrm{w}} > 0$ **then**
$\hat{w}_k = 1$;
else
$\hat{w}_k = 0$;
end if
end if
if $\hat{\boldsymbol{B}}_j$ is merged into 8×8 **then**
Apply dequantization and inverse DCT functions in HEVC to $\boldsymbol{Y}_{\mathrm{w}}$ and obtain the residual $\boldsymbol{X}_{\mathrm{w}}$;
Calculate the corresponding mean value $\bar{x}_{\mathrm{w}}$ of the eight pixels around the residual pixel x^{w}_{33}, x^{w}_{37}, x^{w}_{73}, or x^{w}_{77}, represented as $\hat{x}^{\mathrm{w}}_{33}$;
if $\hat{x}^{\mathrm{w}}_{33} - \bar{x}_{\mathrm{w}} > 0$ **then**
$\hat{w}_k = 1$;
else
$\hat{w}_k = 0$;
end if
end if
end for
end for
End Procedure

sequence is applied as a watermark. The detailed resolution information of the test video sequences is listed in Table 1. The experiments embed the watermark in the first 20 frames of each video. The performance of the proposed algorithm is compared with [22] and [21] which aim at avoiding distortion drift and mainly resisting recompression attacks. All these three algorithms are implemented using the HEVC reference software HM 16.20 with "encode_intra_main.cfg" as config document. To ensure the comparability of the algorithms in the experiment, the QP $= 16$ ($Q_{\mathrm{step}} = 4$) is chosen as the initial QP, which is the same as [22]. The embedding capacity of all three watermarking algorithms is 100 bits/frame.

TABLE 1. Detailed resolution of the test videos.

Video sequence	Resolution
BasketballPass	416×240
BlowingBubbles	416×240
BasketballDrill	832×480
BQMall	832×480
FourPeople	1280×720
Kristen&Sara	1280×720
BasketballDrive	1920×1080
BQTerrace	1920×1080
Traffic	2560×1600
PeopleOnStreet	2560×1600

A. EVALUATION CRITERIA

In this paper, the average peak signal-to-noise ratio (PSNR) and the double-stimulus continuous quality scale (DSCQS) [35] are used to evaluate the imperceptibility from both objective and subjective aspects, respectively.

The PSNR is the most common and widely used objective evaluation index for evaluating video quality, which is given by:

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{3/2 \times 255^2}{\mathrm{MSE_Y} + \mathrm{MSE_U} + \mathrm{MSE_V}}\right)(\mathrm{dB}), \quad (16)$$

where $\mathrm{MSE_Y}$, $\mathrm{MSE_U}$, and $\mathrm{MSE_V}$ are the mean square errors (MSEs) of Y, U, and V components, respectively. The MSE is defined as:

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (f_{ij} - \hat{f}_{ij})^2, \quad (17)$$

where M and N represent the height and width of the video frame, respectively, f_{ij} represent the pixel in the i-th row of the j-th column of the original frame $\boldsymbol{F}$, and $\hat{f}_{ij}$ represent the pixel in the i-th row of the j-th column of the frame to be evaluated $\hat{\boldsymbol{F}}$. $\mathrm{MSE_U}$ and $\mathrm{MSE_V}$ will not change, because watermark is only embedded in the Y component. The experiments use 20 frames of each video sequence as test sequences. Hence, the PSNR in the experiments is the mean value of the PSNRs of the 20 frames. The higher the PNSR, the better the objective imperceptibility.

However, due to the non-linear behavior of the HVS, the PSNR is not always a real reflection of the perceptual quality of a video, especially the temporal flicker. Therefore, DSCQS, a subjective evaluation method, is adopted to reflect the perceptual quality. The DSCQS method simultaneously displays the original video and the watermarked video to the participants at random positions, and the participants are not told which video is the original video. The participants compare the two videos and then give subjective evaluation scores with regard to the visual quality of the two videos, respectively. The evaluation score is divided into five levels, as shown in Table 2. Then, the scores of the participants are averaged to obtain mean opinion score (MOS). The higher the MOS, the better the subjective imperceptibility.

TABLE 2. Subjective scoring criteria for video quality evaluation.

Score	Scale
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

The robustness is evaluated in terms of bit error ratio (BER) and normalized cross-correlation (NCC), which can be represented as follows:

$$\mathrm{BER} = \frac{1}{L} \sum_{i=1}^{L} |w_i - w'_i|, \quad (18)$$

$$\mathrm{NCC} = \frac{\sum_{i=1}^{L} w_i \times w'_i}{\sqrt{\sum_{i=1}^{L} w_i^2} \times \sqrt{\sum_{i=1}^{L} w'^2_i}}, \quad (19)$$

where w_i and w'_i denote the i-th original and extracted watermark bit, respectively, and L is the length of the watermark sequences $\boldsymbol{w}$ and $\boldsymbol{w}'$. BER and NCC are adopted to estimate the error ratio and similarity between $\boldsymbol{w}$ and $\boldsymbol{w}'$, respectively. A better robust watermarking should have lower BER and higher NCC.

The BIR is defined as the percentage of bit rate increase per embedded bit, which is given by [18]:

$$\mathrm{BIR} = \frac{n_\mathrm{w} - n_\mathrm{o}}{n_\mathrm{c} \times n_\mathrm{o}}, \quad (20)$$

where n_w and n_o are the number of bits used for coding the watermarked and original video sequences, respectively, and n_c is the actual capacity, i.e., the number of watermark bits actually embedded in the video sequence. Since the bit rate of watermarked video stream must meet the channel bandwidth limitation, the BIR should be as low as possible.

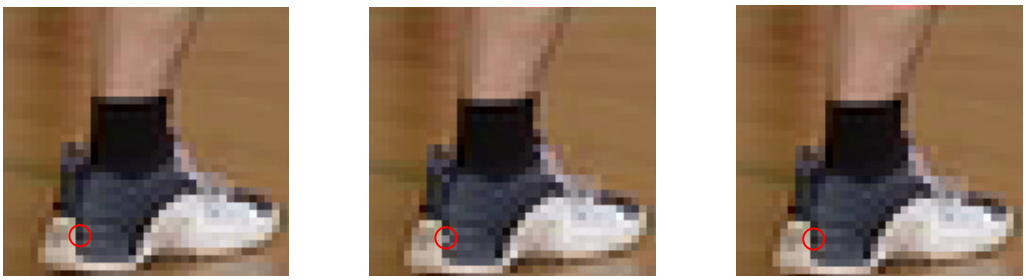
B. THRESHOLDS DETERMINATION

The three important thresholds, N_th, α_th, and R_th, aiming to improve the imperceptibility and robustness in the proposed algorithm, need to be determined.

As mentioned in Section IV, the high-NNZ regions are robust to recompression, having high probability of maintaining a block size of 4×4. Embedding watermark in texture areas with high NNZ can improve the imperceptibility. However, the extreme large threshold N_th will result in a small watermarking capacity, thus not meeting the practical application. Gaj *et al*. [22] determined that the threshold $N_\mathrm{th} = 12$ is most appropriate through experiments, so the proposed algorithm also uses this threshold in the experiments.

The threshold α_th is set to reduce the subjective effect of watermarking on visual quality and is a threshold concerning embedding strength δ. When $Q_\mathrm{step} = 4$, the distortion on x'_{33} is 12δ by (9). In the experiments, we set $\alpha_\mathrm{th} = 2$, then $\delta \leq 2$, i.e., the distortion is not greater than 24. Fig. 9 shows the subjective visual quality of texture regions in the first watermarked frames of BasketballPass video sequence that applies and does not apply the threshold $\alpha_\mathrm{th} = 2$. It can be seen that there is a distinct bright spot similar to Salt noise in the red circle of Fig. 9(b). Although in the texture or edge regions to which the HVS is insensitive, it is easy to be perceived because of the high brightness of the bright spot. With the threshold α_th, the bright spot is discarded and the subjective visual quality is improved, as shown in Fig. 9(c).

The application of the threshold R_th is an important part of the proposed algorithm to improve the robustness. An experiment is implemented to illustrate the effect of R_th on watermarking. In this experiment, a specific evaluation index,



(a) (b) (c)

FIGURE 9. Subjective performance evaluation of the application of the threshold α_{th} on the first frame of BasketballPass: (a) original, (b) watermarked without α_{th}, and (c) watermarked with α_{th}. The embedded watermark bits are all 1.

called modification ratio (MR), is designed as follows:

$$\mathrm{MR} = \frac{N_{\mathrm{m}}}{N_{\mathrm{m}} + N_{\mathrm{um}}}, \tag{21}$$

where the range of MR is [0, 1], and N_{m} and N_{um} are the number of blocks embedded with modified and unmodified coefficients, respectively. The higher the MR value, the more blocks with coefficient modification, the worse the imperceptibility and security. At the same time, if the MR is too small, the watermark can be extracted from unwatermarked videos and cannot play the role of watermarking. Therefore, the closer the MR to 0.5, the better the imperceptibility. Table 3 gives the imperceptibility and the robustness against recompression of QP $=$ 24 with a different R_{th}. As the increase of R_{th}, the MR and the NCC increase, the PSNR and the BER decrease. That is to say, the greater the R_{th}, the better robustness and the worse imperceptibility. To trade off the imperceptibility and robustness, $R_{\mathrm{th}} = 3$ is adopted in the following experiments.

TABLE 3. Imperceptibility and robustness against recompression of QP = 24 with a different R_{th}.

Video sequence	R_{th}	Imperceptibility		Robustness	
		PSNR	MR	BER	NCC
BasketballPass	0	47.311	0.448	0.206	0.790
	1	47.279	0.487	0.180	0.818
	3	47.229	0.553	0.151	0.847
	5	47.177	0.626	0.113	0.888
	7	47.137	0.664	0.085	0.915
	9	47.052	0.730	0.064	0.935
BasketballDrill	0	46.934	0.462	0.187	0.810
	1	46.927	0.500	0.172	0.827
	3	46.918	0.565	0.122	0.877
	5	46.905	0.620	0.094	0.905
	7	46.891	0.668	0.071	0.929
	9	46.877	0.704	0.071	0.929

The selection of the three thresholds is not fixed. The thresholds determined in this section are selected based on the experimental conditions in this paper. In practice, appropriate thresholds can be selected according to actual needs and conditions.

C. PERFORMANCE OF IMPERCEPTIBILITY

To verify the effectiveness of the proposed algorithm, we test and compare the imperceptibility of [22], [21], and the proposed algorithm. The algorithm [21] uses the same (63,7,15) BCH code as in the original literature. The algorithm [22] does not use inter-frame drift compensation method, because the experiments in this paper only use I-frames as the experimental objects, and no inter-frame drift is produced.

TABLE 4. The PSNRs of original video and watermarked videos with [22], [21], and the proposed algorithm.

Video sequence	Original	[22]	[21]	Proposed
BasketballPass	47.597	45.761	47.075	47.229
BlowingBubbles	47.002	45.172	46.505	46.676
BasketballDrill	47.003	43.955	46.882	46.918
BQMall	46.865	46.299	46.749	46.782
FourPeople	48.082	45.811	47.975	48.032
Kristen&Sara	48.287	47.899	48.221	48.240
BasketballDrive	48.196	47.569	48.166	48.176
BQTerrace	47.789	47.431	47.761	47.769
Traffic	47.460	47.204	47.447	47.452
PeopleOnStreet	47.854	47.045	47.839	47.844

Table 4 gives the comparison of the PSNRs among compressed videos that do not have watermarks and those embedded the three watermarks respectively. It can be seen that the PSNR of the proposed algorithm is higher than that of the other two algorithms, and closer to the original videos. Note that as the video resolution increases, the difference of PSNR between the three algorithms decreases, because the fixed embedding capacity of 100 bits/frame has less impact on high-resolution videos. The PSNR of the proposed algorithm is only a little higher than that of [21] in high-resolution videos, but if the embedding capacity is large enough, the difference will become obvious. The PSNR of [22] is much lower than that of the other two algorithms, because its method cannot eliminate the distortion drift and the distortion accumulates to a certain extent. Whereas the algorithm [21] and the proposed algorithm completely avoid the distortion drift. The PSNR of the proposed algorithm is the highest among the three algorithms, which can prove that the objective imperceptibility of the proposed algorithm is better than that of [22] and [21].

Then, the subjective perception experiment was carried out using the DSCQS method. Since the duration of the test videos in the subjective experiment should not be too short, the first 100 frames of each test video were selected as experimental objects and 100 watermark bits were embedded in each frame. The original and the watermarked videos were simultaneously presented to 15 participants at the rate of 25 frames per second. The videos were repeatedly played four times in each test, and paused for two seconds after playing two times. After each pair of videos was played, participants assessed the quality of the two test videos, respectively. For the videos with resolution equal to or less than 832×480, the experiment played them on a 22 inch HP monitor. For the other videos with higher resolution, each pair of videos was played on the two same 22 inch HP monitors with the same setting parameters simultaneously and respectively. The experimental data are normalized to make the MOSs of the reference (original) videos being 4.5. The comparison of the MOSs on the ten test videos with the 95% confidence interval is shown in Fig. 10.

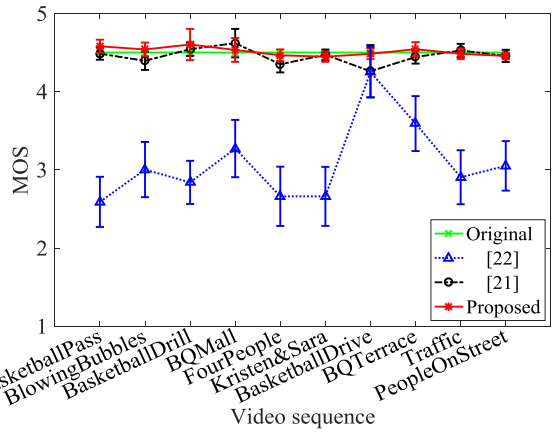


FIGURE 10. Comparison of the DSCQS MOSs of the proposed algorithm with [22] and [21] on the ten test videos.

Due to the uncontrollability of the propagation direction of drift distortion and the randomness of the watermark embedding position, the MOS of [22] fluctuates greatly. In addition to the better visual effects on the video sequence BasketballDrive, the other nine videos have obvious temporal flicker, which degrades the perceived quality of these videos.

For instance, the PSNR of the 100 frames of PeopleOnStreet is 47.4204, which is a considerable result. However, in the subjective test, the overall brightness of an area of the 17-th frame of the video is degraded due to the distortion drift, resulting in a serious temporal flicker. For the algorithm [21] and the proposed algorithm, the test videos have excellent visual effects because there is no distortion drift. As to the proposed algorithm, the participants could not perceive the difference between the original videos and the watermarked videos, and the MOS curve of the proposed algorithm almost coincided with that of the original videos. The reason is that the proposed algorithm not only achieves intra-drift-free, but also embeds the watermark in the high-frequency texture regions to which the HVS is insensitive.

It can be concluded that in HEVC compressed domain, intra-drift-free is an important condition for watermarking algorithms to be imperceptible. The proposed algorithm achieves intra-drift-free, and the imperceptibility of the proposed algorithm is better than [22] and [21] in both objective and subjective aspects.

D. PERFORMANCE OF ROBUSTNESS

The robustness against recompression and common signal processing attacks of the proposed algorithm is evaluated and compared with [22] and [21]. Fig. 11 and Fig. 12 present the comparison of the BERs and the NCCs against recompression attacks, respectively. In this experiment, the implementation of the recompression attacks is to decompress the watermarked compressed videos in Section V.B into the original videos using HEVC decoder, and then encode the original videos with different QPs using HEVC encoder respectively, where the used QPs are from 8 to 28 with a step of 2.

As can be seen from Fig. 11 and Fig. 12, all the three watermarking algorithms have the lowest BERs and the highest NCCs when QP remains unchanged. However, with the change of QP, the BER of [21] quickly reaches about 0.5, and the NCC rapidly drops below 0.5. The main reason is that Q_{step} divided by the transform coefficients changes during the quantization process, which leads to the parity change of the QTCs. It is worth noting that the NCC < 0.5 means that more extracted watermark bits are 0. The reason is that more watermarked blocks are merged into larger blocks where watermark bits cannot be extracted, and this paper replaces those unextractable bits with 0. The algorithm [21] does not use the threshold N_{th}, so a large number of watermarked blocks are merged into larger blocks when QP becomes large. The BERs and the NCCs of [22] and the proposed algorithm increase and decrease with the increase of QP respectively, and are acceptable when QP $\leq$ 24. When the QP used in recompression is less than the original QP except QP $=$ 10, the BERs and the NCCs of [22] and the proposed algorithm keep acceptable. At this time, the failure of watermark extraction is mainly caused by the change of prediction mode rather than quantization distortion. There is a special point QP $=$ 10 on the horizontal axis, where the BERs and the NCCs of [22] and the proposed algorithms are very close to those of QP $=$ 16, and the NCC of [21] decreases or even approaches zero. When QP $=$ 10, $Q_{\text{step}} = 2$. It is half the value of QP $=$ 16 ($Q_{\text{step}} = 4$). Under this circumstance, the HEVC recompression hardly produce quantization errors and the RD function tends to maintain the original block partition and prediction mode, so the algorithm [22] and the proposed algorithm have achieved good results. However, because most QTCs carrying watermark information in [21] are twice as large as the original, almost all even numbers, the extracted watermark bits are almost all 0, resulting in extremely low NCC. Compared with [22] and [21], the

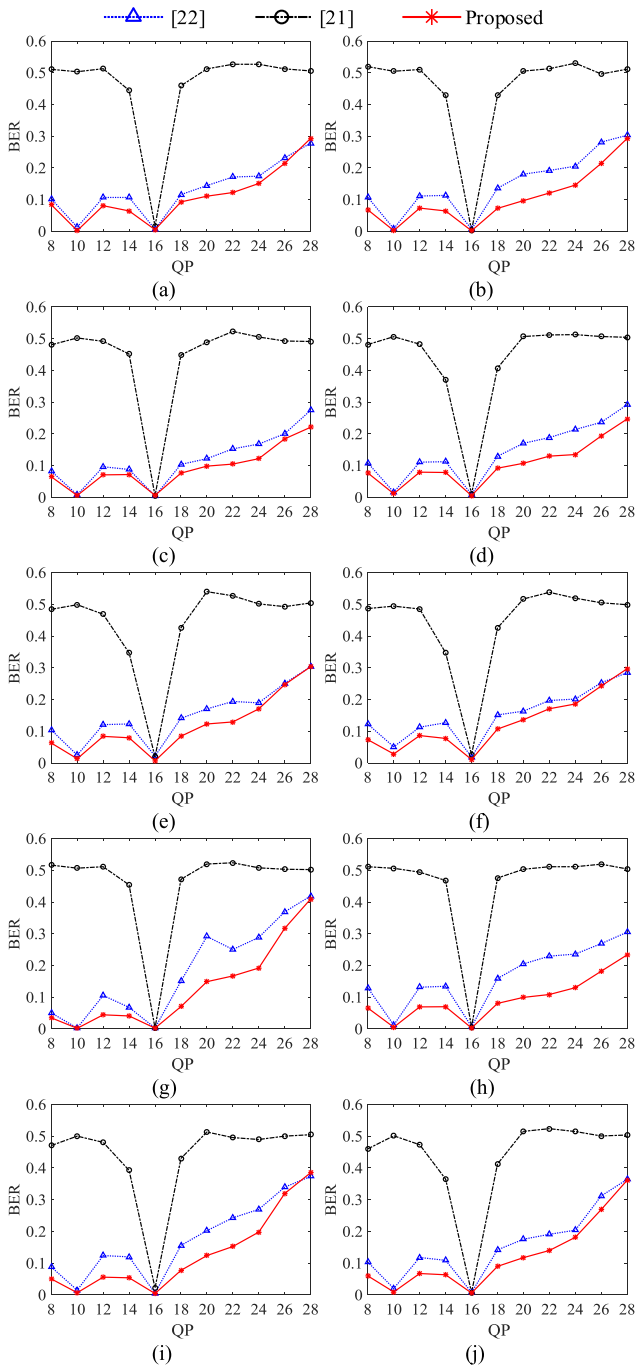


FIGURE 11. Comparison of the BERs of the proposed algorithm with [22] and [21] against the recompression attacks on different test videos: (a) BasketballPass, (b) BlowingBubbles, (c) BasketballDrill, (d) BQMall, (e) FourPeople, (f) Kristen&Sara, (g) BasketballDrive, (h) BQTerrace, (i) Traffic, and (j) PeopleOnStreet.

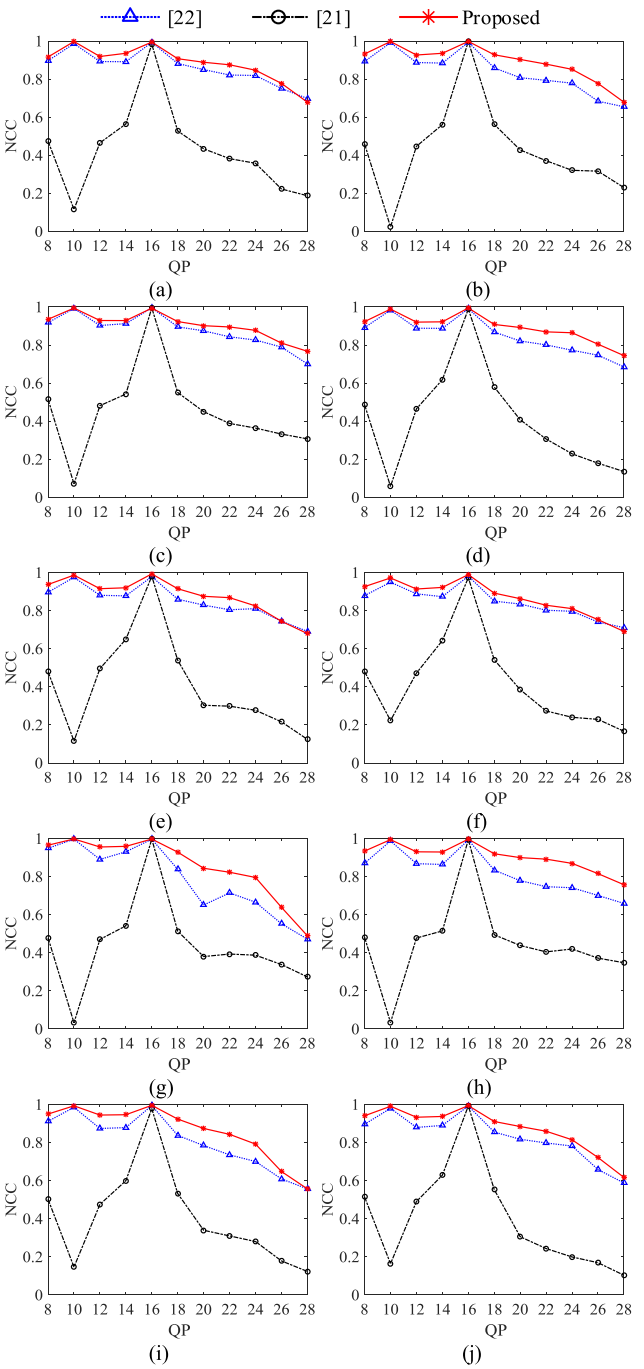


FIGURE 12. Comparison of the NCCs of the proposed algorithm with [22] and [21] against recompression attacks on different test videos: (a) BasketballPass, (b) BlowingBubbles, (c) BasketballDrill, (d) BQMall, (e) FourPeople, (f) Kristen&Sara, (g) BasketballDrive, (h) BQTerrace, (i) Traffic, and (j) PeopleOnStreet.

proposed watermarking algorithm has lower BER and higher NCC. It can be concluded that the proposed algorithm has the highest robustness against recompression attacks among the three algorithms.

Salt & Pepper noise, Gaussian noise, and Gaussian filter are three common signal processing attacks and used as attack tools to test the robustness of the proposed algorithm. In this experiment, the watermarked compressed videos are decompressed, processed by the three kinds of attacks respectively, and then recompressed with the original QP. Table 5 shows the BERs and the NCCs between the embedded watermarks and the extracted watermarks from the attacked videos using the proposed algorithm with [22] and [21]. It can be seen that the proposed algorithm is robust to these common signal

TABLE 5. Comparison of the robustness to Salt & Pepper noise, Gaussian noise, and Gaussian filter attacks of the proposed algorithm with [22] and [21].

Video sequence	Algorithm	Salt & Pepper noise				Gaussian noise						Gaussian filter (3×3)					
		Noise strength				σ^2						σ					
		0.01		0.02		0.0001		0.0002		0.001		0.4		0.5		0.6	
		BER	NCC	BER	NCC	BER	NCC	BER	NCC	BER	NCC	BER	NCC	BER	NCC	BER	NCC
BasketballPass	[22]	0.064	0.936	0.105	0.895	0.168	0.825	0.185	0.806	0.251	0.741	0.131	0.867	0.223	0.768	0.354	0.607
	[21]	0.208	0.788	0.323	0.657	0.484	0.452	0.517	0.416	0.503	0.459	0.351	0.645	0.519	0.421	0.506	0.383
	Proposed	0.072	0.928	0.096	0.904	0.106	0.896	0.133	0.866	0.224	0.773	0.090	0.909	0.151	0.845	0.252	0.736
BlowingBubbles	[22]	0.067	0.934	0.089	0.911	0.170	0.824	0.184	0.808	0.256	0.740	0.136	0.860	0.247	0.732	0.323	0.629
	[21]	0.149	0.850	0.263	0.732	0.501	0.452	0.506	0.459	0.502	0.486	0.377	0.614	0.493	0.435	0.502	0.382
	Proposed	0.059	0.941	0.105	0.895	0.094	0.907	0.115	0.884	0.208	0.790	0.086	0.915	0.157	0.843	0.251	0.745
BasketballDrill	[22]	0.055	0.946	0.089	0.911	0.145	0.850	0.181	0.813	0.237	0.759	0.129	0.867	0.222	0.766	0.339	0.621
	[21]	0.148	0.850	0.289	0.698	0.470	0.495	0.510	0.452	0.497	0.480	0.351	0.646	0.487	0.422	0.492	0.385
	Proposed	0.056	0.944	0.092	0.908	0.103	0.897	0.112	0.890	0.194	0.805	0.103	0.895	0.150	0.846	0.250	0.733
BQMall	[22]	0.069	0.932	0.105	0.895	0.171	0.824	0.188	0.805	0.281	0.704	0.156	0.839	0.274	0.703	0.356	0.586
	[21]	0.226	0.764	0.357	0.603	0.504	0.401	0.530	0.382	0.509	0.449	0.327	0.654	0.492	0.428	0.497	0.385
	Proposed	0.066	0.935	0.101	0.899	0.108	0.892	0.134	0.864	0.212	0.787	0.114	0.884	0.156	0.843	0.267	0.722
FourPeople	[22]	0.073	0.927	0.109	0.891	0.169	0.828	0.200	0.795	0.263	0.728	0.163	0.840	0.314	0.681	0.411	0.564
	[21]	0.194	0.796	0.306	0.658	0.481	0.456	0.510	0.433	0.502	0.425	0.295	0.683	0.462	0.455	0.492	0.394
	Proposed	0.055	0.945	0.105	0.894	0.116	0.884	0.132	0.868	0.214	0.786	0.095	0.905	0.203	0.791	0.309	0.674
Kristen&Sara	[22]	0.079	0.920	0.117	0.884	0.171	0.829	0.208	0.789	0.275	0.717	0.158	0.843	0.312	0.691	0.392	0.591
	[21]	0.261	0.710	0.352	0.595	0.495	0.437	0.503	0.424	0.482	0.445	0.287	0.697	0.485	0.432	0.511	0.387
	Proposed	0.071	0.929	0.107	0.894	0.134	0.865	0.140	0.861	0.215	0.785	0.122	0.878	0.224	0.774	0.338	0.649
BasketballDrive	[22]	0.045	0.955	0.068	0.931	0.192	0.791	0.222	0.756	0.318	0.640	0.135	0.857	0.275	0.683	0.338	0.604
	[21]	0.210	0.790	0.336	0.652	0.503	0.425	0.497	0.417	0.522	0.355	0.373	0.614	0.520	0.374	0.512	0.401
	Proposed	0.055	0.946	0.086	0.914	0.010	0.900	0.122	0.878	0.230	0.767	0.060	0.944	0.165	0.831	0.259	0.724
BQTerrace	[22]	0.062	0.939	0.097	0.904	0.183	0.811	0.204	0.789	0.287	0.696	0.171	0.823	0.268	0.710	0.361	0.592
	[21]	0.132	0.867	0.284	0.704	0.470	0.472	0.494	0.433	0.507	0.414	0.418	0.570	0.485	0.442	0.500	0.408
	Proposed	0.061	0.940	0.098	0.902	0.094	0.906	0.122	0.878	0.210	0.789	0.087	0.913	0.161	0.839	0.265	0.729
Traffic	[22]	0.061	0.939	0.088	0.913	0.219	0.763	0.239	0.740	0.310	0.654	0.160	0.836	0.336	0.624	0.442	0.455
	[21]	0.203	0.787	0.339	0.632	0.467	0.451	0.478	0.452	0.493	0.468	0.291	0.696	0.443	0.466	0.472	0.397
	Proposed	0.054	0.946	0.097	0.902	0.108	0.894	0.129	0.873	0.217	0.785	0.083	0.918	0.200	0.798	0.343	0.636
PeopleOnStreet	[22]	0.065	0.935	0.089	0.911	0.204	0.783	0.210	0.777	0.303	0.669	0.149	0.848	0.314	0.652	0.433	0.480
	[21]	0.210	0.780	0.318	0.652	0.493	0.402	0.485	0.420	0.501	0.432	0.260	0.725	0.472	0.412	0.505	0.348
	Proposed	0.061	0.940	0.097	0.906	0.129	0.872	0.139	0.863	0.228	0.775	0.096	0.905	0.198	0.801	0.323	0.672

processing attacks, and its performance is superior to that of [22] and [21] except that it is similar to [21] in resisting Salt & Pepper noise.

E. PERFORMANCE OF BIR

Table 6 shows the comparison of the BIRs (BIR $\times$ 10^{-4}) of the proposed algorithm with [22] and [21] on the ten video sequences, where the test videos are also the videos obtained in the experiment of Section V.B. Each test video is embedded with 100 watermark bits per frame, and each watermarked test video of 20 frames contains 2000 watermark bits. The results show that [22] and the proposed algorithm both have lower BIR than [21], especially for high-resolution video. The reason is that the watermarks of [22] and the proposed algorithm are embedded in the high-NNZ LTBs and do not increase the NNZ too much. The BIR of the proposed algorithm is slightly higher than that of [22], because the number of coefficients needed to be modified in each block for embedding watermark in the proposed algorithm is 9, which is relatively high. In addition, the MR of the proposed algorithm is greater than 0.5, which results in more blocks to be modified. The algorithm [21] has higher BIR because it needs more BCH check information to be embedded. Note that all these three algorithms need text files for recording location of watermark, where the files of the algorithm [22] and the proposed algorithm are around 750 bytes/frame and the file of the algorithm [21] is around 1500 bytes/frame.

TABLE 6. Comparison of the BIRs (BIR$\times 10^{-4}$) of the proposed algorithm with [22] and [21] on ten video sequences.

Video sequence	[22]	[21]	Proposed
BasketballPass	6.303	161.567	10.214
BlowingBubbles	3.040	44.647	6.363
BasketballDrill	1.018	25.688	2.320
BQMall	1.219	30.856	2.267
FourPeople	3.722	36.833	4.665
Kristen&Sara	4.063	43.907	4.798
BasketballDrive	0.237	4.512	1.009
BQTerrace	0.198	2.974	0.356
Traffic	0.267	5.224	0.468
PeopleOnStreet	0.390	5.772	0.538

F. REAL-TIME PERFORMANCE

In the experiments, all algorithms are implemented on the same platform of 3.30 GHz CPU, 4.00 GB RAM, Win 7 and Visual Studio 2013. To reduce the influence of errors and guarantee the reliability of experimental data, the data are averaged by the same experiment for five times.

TABLE 7. Comparison of execution time on BasketballPass and BasketballDrill.

Video sequence	Algorithm	Embedding time (ms)	Extraction time (ms)	Total time (ms)
BasketballPass	[22]	36.855	5.048	41.903
	[21]	34.058	5.597	39.655
	Proposed	32.478	5.622	38.100
BasketballDrill	[22]	136.745	6.736	143.481
	[21]	135.197	6.706	141.903
	Proposed	132.754	7.632	140.386

Table 7 shows the execution time of the watermark embedding and extraction on the first 20 frames of BasketballPass and BasketballDrill. The algorithm [21], which embeds nine times more than the watermark bits in videos, increases execution time by BCH (63,7,15) code. Moreover, the execution time of [21] in Table 7 does not cover BCH encoding and decoding time, otherwise the execution time would have been much longer than that of the other two algorithms. The algorithm [22] has the longest embedding time because it requires matrix transformations repeatedly and many floating-point operations. Since all the embedding procedures of the proposed algorithm are integer operations and simple, the embedding execution of the proposed algorithm has the lowest complexity and best real-time performance compared with [22] and [21]. Although the extraction time of the proposed algorithm is longer than that of [22], it is only 0.574 ms and 0.896 ms longer on BasketballPass and BasketballDrill, respectively, so it is acceptable. The extraction time of the proposed algorithm is relatively long because the proposed algorithm processes 4×4 and 8×8 blocks to improve the robustness, whereas the algorithms [22] and [21] only deal with 4×4 blocks. The total time of the proposed algorithm is the shortest among the three algorithms, so the proposed algorithm has high real-time performance.

To prove that the proposed algorithm as a compressed-domain watermarking algorithm has shorter computation time and uses less computational resources than uncompressed-domain and in-the-loop watermarking algorithms, the experiment of full encoding time is implemented and the analysis of the proposed algorithm with other two kinds of algorithms is presented.

The full encoding time, including coding time before embedding t_b^{20}, embedding time t_e^{20}, and coding time after embedding t_a^{20}, of the proposed algorithm on the first 20 frames of BasketballPass and BasketballDrill with watermark embedding process is tested, respectively, and the results are shown in Table 8. t_e^{20} of the two test videos are 72 and 79 times the sums of t_b^{20} and t_e^{20} of the two videos, respectively, indicating that the encoding time before embedding accounts for the majority of the full encoding time. The reason is that the coding time before embedding is mainly the time spent by HEVC for complex estimation and optimization, and the encoding time after embedding is the time spent by relatively simple entropy encoding.

TABLE 8. Comparison among coding time before embedding, embedding time, and coding time after embedding of the proposed algorithm on BasketballPass and BasketballDrill.

Video sequence	t_b^{20} (ms)	t_e^{20} (ms)	t_a^{20} (ms)
BasketballPass	18920	32	230
BasketballDrill	79262	133	880

Assume that there are m videos to be watermarked and compressed, and each video has n frames, of which n_I frames are I-frames. The embedding time of uncompressed-domain, in-the-loop, and the proposed compressed-domain watermarking algorithms for each frame are t_{uc} , t_{il}, and t_c, respectively. The coding time before embedding and the coding time after embedding for each frame are t_b and t_a, respectively, independent of the time used for embedding watermarks. Since the execution time increases linearly with the number of frames, the uncompressed-domain and in-the-loop algorithms take the total time t_{uc}^{total} and t_{il}^{total}, respectively:

$$t_{uc}^{total} = (t_b + t_a) \times m \times n + t_{uc} \times m \times n_I, \quad (22)$$

$$t_{il}^{total} = (t_b + t_a) \times m \times n + t_{il} \times m \times n_I. \quad (23)$$

The compressed-domain watermarking only executes one prediction process and takes the total time t_c^{total}:

$$t_c^{total} = (t_b + t_a \times m) \times n + t_c \times m \times n_I, \quad (24)$$

where the values of t_{uc} and t_{il} depend on specific watermarking algorithms. The differences between the computation time of the compressed-domain algorithm and that of the other two kinds of algorithms can be calculated, respectively:

$$t_{uc}^{total} - t_c^{total} = t_b \times (m-1) \times n + (t_{uc} - t_c) \times m \times n_I, \quad (25)$$

$$t_{il}^{total} - t_c^{total} = t_b \times (m-1) \times n + (t_{il} - t_c) \times m \times n_I. \quad (26)$$

According to the experimental results shown in Table 8, we know that $t_b \gg t_c$. Equations (25) and (26) can be approximately written as:

$$t_{uc}^{total} - t_c^{total} \approx t_{il}^{total} - t_c^{total} \approx t_b \times (m-1) \times n. \quad (27)$$

From (27), it can be concluded that the difference between t_{uc}^{total} and t_c^{total} and the difference between t_{il}^{total} and t_c^{total} increase linearly with the number of videos and frames. In practical applications, the number of frames of a video is enormous and the video will be distributed to a large number of users. Compared with uncompressed-domain and in-the-loop watermarking algorithms, the proposed algorithm can save a lot of computing time and computing resources. Therefore, the proposed algorithm is suitable for application scenarios where the video is distributed on a large scale.

VI. CONCLUSION

In this paper, based on the analysis of HEVC standard, an intra-drift-free robust watermarking algorithm is proposed for HEVC video using the robust relationship between a specific residual pixel and the mean value of the pixels around it. The proposed algorithm eliminates the intra-frame distortion drift and has high robustness to several attacks at

the same time. In addition, the use of the three thresholds improves the performance of the algorithm. The watermark bits can be extracted from the 8×8 blocks merged by the watermarked blocks, which further improves the robustness. Experimental results show that the proposed algorithm outperforms the state-of-the-art algorithms [21], [22] in terms of imperceptibility and robustness to recompression and common signal processing attacks. The BIR of the proposed algorithm is low and can satisfy the limitation of transmission bandwidth. Besides, the proposed algorithm has high real-time performance and is suitable for large-scale video distribution applications because of its fast embedding process.

In the future, to improve the security of the proposed algorithm, we will explore how to realize blind extraction without location map or any side information. Moreover, the proposed algorithm is currently unable to resist geometric attacks. Compared with uncompressed-domain watermarking, compressed-domain watermarking must be combined with the complex video coding standard. Therefore, it is more difficult to resist geometric attacks in the compression domain. How to resist geometric attacks will be the main content of our next research.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[2] W. Hamidouche, M. Raulet, and O. Déforges, "4 K real-time and parallel software video decoder for multilayer hevc extensions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 169–180, Jan. 2016.
[3] 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.
[4] X. Yu, C. Wang, and X. Zhou, "A survey on robust video watermarking algorithms for copyright protection," *Appl. Sci.*, vol. 8, no. 10, Oct. 2018, Art. no. 1891.
[5] Y. Liu, S. Liu, Y. Wang, H. Zhao, and S. Liu, "Video steganography: A review," *Neurocomputing*, vol. 335, pp. 238–250, Mar. 2019.
[6] H. Mareen, J. De Praeter, G. Van Wallendael, and P. Lambert, "A scalable architecture for uncompressed-domain watermarked videos," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 6, pp. 1432–1444, Jun. 2019.
[7] W. Huo, Y. Zhu, and H. Chen, "A controllable error-drift elimination scheme for watermarking algorithm in H.264/AVC stream," *IEEE Signal Process. Lett.*, vol. 18, no. 9, pp. 535–538, Sep. 2011.
[8] 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.
[9] T.-J. Lin, K.-L. Chung, P.-C. Chang, Y.-H. Huang, H.-Y. M. Liao, and C.-Y. Fang, "An improved DCT-based perturbation scheme for high capacity data hiding in H.264/AVC intra frames," *J. Syst. Softw.*, vol. 86, no. 3, pp. 604–614, Mar. 2013.
[10] Y. Liu, Z. Li, X. Ma, and J. Liu, "A novel data hiding scheme for H.264/AVC video streams without intra-frame distortion drift," in *Proc. IEEE 14th Int. Conf. Commun. Technol.*, Chengdu, China, Nov. 2012, pp. 824–828.
[11] 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.
[12] P.-C. Chang, K.-L. Chung, J.-J. Chen, C.-H. Lin, and T.-J. Lin, "A DCT/DST-based error propagation-free data hiding algorithm for HEVC intra-coded frames," *J. Vis. Commun. Image Represent.*, vol. 25, no. 2, pp. 239–253, Feb. 2014.
[13] S. Gaj, S. Rana, A. Sur, and P. K. Bora, "A drift compensated reversible watermarking scheme for H.265/HEVC," in *Proc. IEEE 18th Int. Workshop Multimedia Signal Process. (MMSP)*, Montreal, QC, Canada, Sep. 2016, pp. 1–6.
[14] 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.
[15] S. Gaj, A. S. Patel, and A. Sur, "Object based watermarking for H.264/AVC video resistant to rst attacks," *Multimedia Tools Appl.*, vol. 75, no. 6, pp. 3053–3080, Mar. 2016.
[16] W. Chen, Z. Shahid, T. Stütz, F. Autrusseau, and P. Callet, "Robust drift-free bit-rate preserving H.264 watermarking," *Multimedia Syst.*, vol. 20, no. 2, pp. 179–193, Mar. 2014.
[17] X. Gong and H.-M. Lu, "Towards fast and robust watermarking scheme for H.264 Video," in *Proc. 10th IEEE Int. Symp. Multimedia*, Berkeley, CA, USA, Dec.. 2008, pp. 649–653.
[18] T. Dutta and H. P. Gupta, "A robust watermarking framework for high efficiency video coding (HEVC)–encoded video with blind extraction process," *J. Vis. Commun. Image Represent.*, vol. 38, pp. 29–44, Jul. 2016.
[19] 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, Jan. 2017, Art. no. 12.
[20] 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.
[21] 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.
[22] 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, Jan. 2017, Art. no. 11.
[23] T. Kim, K. Park, and Y. Hong, "Video watermarking technique for H.264/AVC," *Opt. Eng.*, vol. 51, no. 4, Apr. 2012, Art. no. 047402.
[24] 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.
[25] 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.
[26] 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.
[27] K.-S. Kim, H.-Y. Lee, D.-H. Im, and H.-K. Lee, "Practical, real-time, and robust watermarking on the spatial domain for high-definition video contents," *IEICE Trans. Inf. Syst.*, vol. E91.D, no. 5, pp. 1359–1368, May 2008.
[28] 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.
[29] 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.
[30] 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.
[31] 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.
[32] S. Swati, K. Hayat, and Z. Shahid, "A watermarking scheme for high efficiency video coding (HEVC)," *PLoS ONE*, vol. 9, no. 8, Aug. 2014, Art. no. e105613.
[33] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792–1801, Dec. 2012.
[34] F. Bossen, *Common Test Conditions and Software Reference Configurations*, document JCTVC-L1100, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Jan. 2013.
[35] *Methodology for the Subjective Assessment of the Quality of Television Pictures*, document ITU-R. BT.500-11, Radiocommunication Sector of International Telecommunication Union, 2002.

YANGMING ZHOU received the B.S. degree in electronic information science and technology from Shandong University, Weihai, China, in 2018, where he is currently pursuing the M.E. degree in information and communication engineering. His current research interests include digital image/video processing, digital watermarking, and computer vision.

CHENGYOU WANG (M'16) received the B.E. degree in electronic information science and technology from Yantai University, China, in 2004, and the M.E. and Ph.D. degrees in signal and information processing from Tianjin University, China, in 2007 and 2010, respectively. He is currently an Associate Professor and a Supervisor of master's students with Shandong University, Weihai, China. His current research interests include digital image/video processing and analysis, computer vision, machine learning, and wireless communication technology.

XIAO ZHOU received the B.E. degree in automation from the Nanjing University of Posts and Telecommunications, China, in 2003, the M.E. degree in information and communication engineering from Inha University, South Korea, in 2005, and the Ph.D. degree in information and communication engineering from Tsinghua University, China, in 2013. She is currently an Associate Professor and a Supervisor of master's students with Shandong University, Weihai, China. Her current research interests include wireless communication technology, digital image processing, and computer vision.