mathematics

MDPI

Article

Anti-Recompression Video Watermarking Algorithm Based on H.264/AVC

Di Fan [1,2,3], Huiyuan Zhao [1], Changying Zhang [1], Hongyun Liu [1] and Xiaoming Wang [1,*]

[1] College of Electronic Information Engineering, Shandong University of Science and Technology, Qingdao 266590, China; skd992372@sdust.edu.cn (D.F.); zhy1021196704@163.com (H.Z.); zcy06072023@163.com (C.Z.); luh0420@163.com (H.L.)
[2] College of Electrical and Information Engineering, Hunan University, Changsha 410082, China
[3] National Engineering Research Center of RVC, Changsha 410082, China
* Correspondence: skd992388@sdust.edu.cn

Abstract: The problem of information security and copyright protection of video is becoming increasingly prominent. The current video watermarking algorithm does not have strong anti-compression, which has a significant impact on the visual effect of video. To solve this problem, this paper proposes a video watermarking algorithm based on H.264/AVC. The algorithm combines the non-zero quantization coefficient and the energy factor to select the appropriate chroma subblock, and then an optimized modulation is designed to embed the watermark into its DCT quantization coefficients in order to minimize the number of modifications of the subblocks. The invisibility and robustness experiments of the algorithm are conducted in the paper, and the Structural Similarity Indexes are above 0.99, and the False Bit Rates are all below 0.03. The results show that the algorithm has good invisibility, anti-compression performance and obvious advantages compared with other similar methods.

Keywords: video watermarking; H.264/AVC; recompression attack; requantization attack

MSC: 68T10

check for updates

Citation: Fan, D.; Zhao, H.; Zhang, C.; Liu, H.; Wang, X. Anti-Recompression Video Watermarking Algorithm Based on H.264/AVC. *Mathematics* **2023**, *11*, 2913. https://doi.org/10.3390/math11132913

Academic Editors: Cairong Zhao, Wankou Yang and Can Gao

Received: 24 May 2023
Revised: 19 June 2023
Accepted: 26 June 2023
Published: 29 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Media technology and internet technology have provided great convenience for current video production and dissemination. However, the information security and copyright protection of video have also become prominent. Although digital watermarking technology can play a role in copyright protection, the current video watermarking algorithm has insufficient ability to resist re-compression.

At present, video watermarking is mainly divided into two categories: digital video watermarking algorithm based on the original domain or compressed domain [1]. The video watermarking algorithm based on the original domain embeds and extracts the watermark in the original video sequence without compression coding. It can be divided into video watermarking algorithms based on the space domain [2,3] or the transform domain [4–7]. Jerisha [2] et al. proposed a video watermarking algorithm in the space domain, which converts the original video frame from the original color space to HSV color space, and the watermark is embedded in a pair of coefficients in the HSV space. Fan [4] et al. proposed a video watermarking algorithm based on pseudo 3D-DCT (three-dimensional cosine transform). The algorithm performs the pseudo 3D-DCT transform on the low frequency component of NSCT (nonsubsampled contourlet transform) of the key frame group, and the encryption watermark is embedded into the NMF (non-negative matrix factorization) decomposition of the DC coefficient matrix of the 3D-DCT transform, which is strongly robust. Zhang [5] et al. proposed a blind video watermarking algorithm

based on tensor decomposition, which uses the parity quantization of the maximum core tensor to realize the embedding and extraction of video watermarking.

Video watermarking based on the compressed domain is embedded and extracted in the process of video encoding and decoding or in the compressed video stream. Common video compression standards include MPEG-X [8], H.264/AVC [9,10], H.265/HEVC [11,12] and 3D-HEVC [13,14]. The video watermarking algorithm in H.264 encoding mode can be divided into three types: the video watermarking algorithm based on motion vector [15,16]; DCT (discrete cosine transform) coefficient; and entropy encoded codeword. Yang [15] et al. proposed a fragile video watermarking algorithm based on motion vector, extracted the feature code from the energy block of I frame, used the minimum Lagrange function search, and embedded the watermark in the motion vector with the best rate distortion performance in the macroblock with small segmentation mode. Sun [16] et al. proposed a bit-watershed video watermarking algorithm based on moving target detection. The algorithm embeds the watermark in the level codeword after P-frame entropy coding.

It is common to embed watermarking in DCT coefficients after video transform and coding. Liu [17] et al. proposed a video watermarking algorithm based on chromaticity residual DCT, selected the subblock with the largest residual, and embedded the watermark by modulating the three coefficients. Chen [18] et al. proposed a semi-fragile video watermarking algorithm that can achieve frame attack and video tamper detection at the same time. The frame number is used as the watermark information, and the relationship between non-zero DCT coefficients is used as the authentication code, and the watermark is embedded in subblocks with multiple non-zero coefficients. Xue [19] et al. designed a cost function combining block distortion, bit rate change and block complexity to select texture regions and embed the watermark with minimum distortion. Sun [20] et al. embedded the watermark into the DCT subblock by modulating the positive and negative coefficients of the watermark, and the algorithm can be applied to video sequences with different resolutions. Nguyen [21] et al. used an entropy encoder to encode the original video stream into intra-frame patterns and QDCT coefficients, and then constructed a two-dimensional histogram of the QDCT coefficients to embed the watermark by using the coefficient pair mapping (CPM) method.

In general, among the existing video watermarking algorithms, the embedded watermark has a great impact on the video quality, and the anti-recompression ability is insufficient. In order to solve this problem, this paper proposes a video watermarking algorithm based on H.264/AVC for anti-recompression. This algorithm embeds watermarking in the QDCT (quantification discrete cosine transform) coefficient of the video I frame chroma subblock. The modulation method combines the characteristics of video and watermark, which has little impact on the video quality. The experimental results show that the algorithm takes into account the invisibility and robustness, and significantly improves the anti-re-compression ability of the video.

2. H.264 Compression Standard and Watermark Embedding Location Selection

2.1. H.264/AVC Video Compression Standard

H.264 is a video coding standard issued by the joint video group composed of the International Organization for Standardization and ITU. It is one of the most widely used video coding methods at present. Figure 1 shows the principle block diagram of H.264 video coding [22]. H.264 includes I frame, P frame and B frame, wherein I frame is a full-frame compression coding frame, which is generated without reference to other pictures, and P frame is a prediction frame generated by reference to the previous I frame and P frame prediction coding. The main process of H.264 coding is to first perform DCT transform and quantization of the residual between the current frame and the predicted frame, and then entropy encode the obtained quantization coefficient sequence and write it into the code stream. According to the characteristics of the video frame, the encoder selects intra prediction or inter prediction to process the video frame, I frame uses intra prediction coding mode to encode, and P frame uses inter prediction to encode the image.

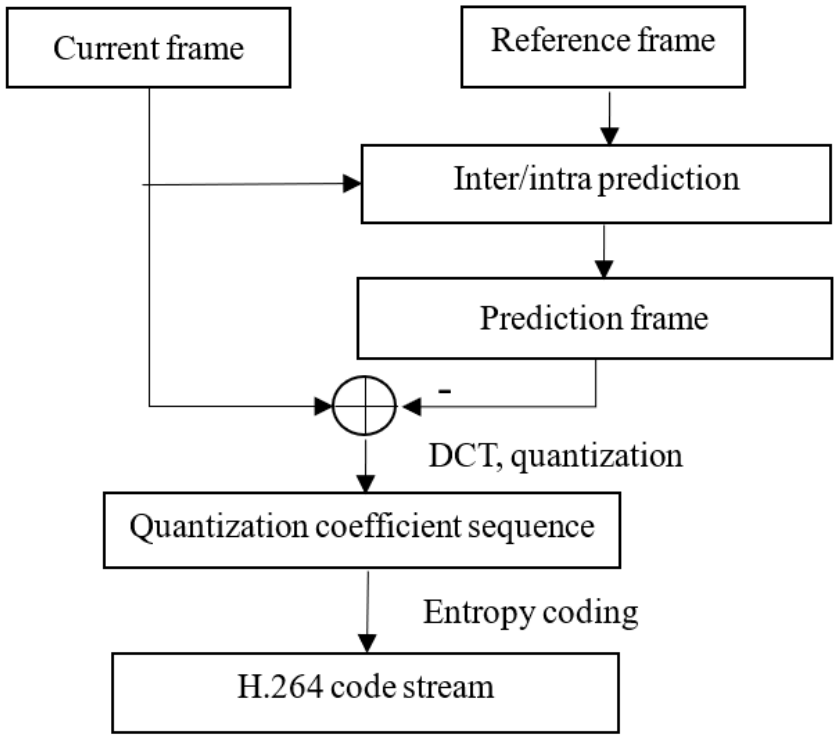


Figure 1. Structure diagram of H.264/AVC encoder.

The H.264 video decoding process is shown in Figure 2. The decoder obtains data information from the code stream, obtains transform coefficients through entropy decoding and reordering, and obtains residual blocks through inverse quantization and inverse DCT transformation. At the same time, the decoder uses the header information obtained from the code stream to make the intra prediction and generate prediction blocks. The decoder adds the prediction block and the residual block to obtain the frame image, which is filtered and reconstructed to obtain the decoded image.

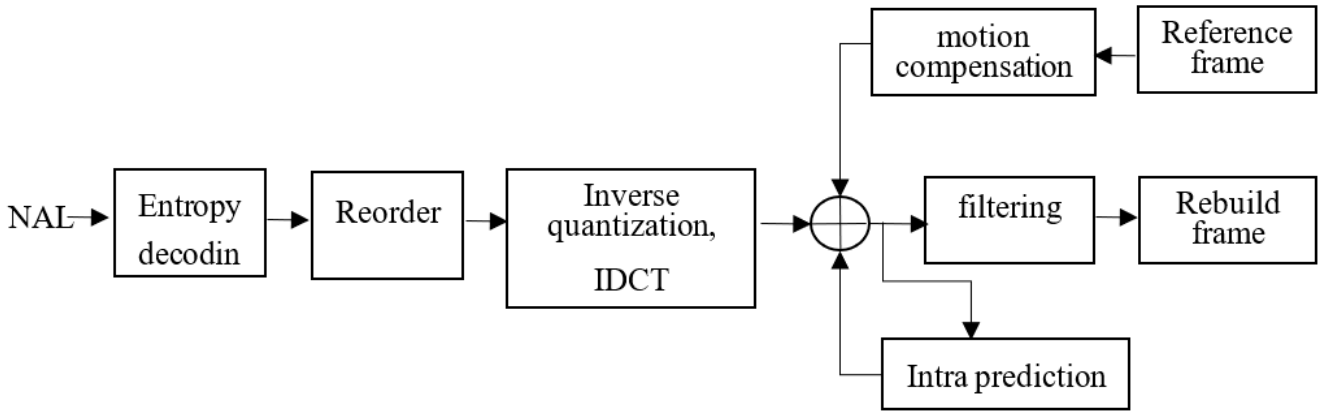


Figure 2. Structure diagram of H.264/AVC decoder.

2.2. Watermark Embedding Position Selection

The watermark is embedded in the encoding process of H.264. The embedding position can be selected from I frame or P frame, chroma block or brightness block. I frame will participate in the encoding process of P frame as a reference frame, and the experiment found that embedding watermarking on the chroma subblock can be more resistant to recompression than the brightness subblock, so embedding watermark on the QDCT coefficient of I frame chroma subblock has less impact on the video.

For the selection of I frame chroma subblock embedded with watermark, this paper combines the number of NNZ (non-zero coefficients) and Ef (energy factor) as the condition of subblock selection, that is, the definition function:

$$C(i) = NNZ(i) + Ef(i) \tag{1}$$

where i represents the i-th 4×4 chroma block; $NNZ(i)$ represents the number of non-zero coefficients of the i-th subblock; $Ef(i)$ represents the energy factor of the i-th subblock and is the sum of the absolute values of the quantization transformation coefficients of the i-th subblock.

The specific selection method is to first select the third non-zero quantization coefficient of all I frame chroma subblocks that is not 2 or -2, then calculate the C value of the selected subblocks and select several subblocks with the largest C value as the watermark embedding objects.

3. Watermark Embedding and Extraction Algorithm

3.1. Video Watermark Embedding Algorithm

The video watermark embedding algorithm framework in this paper is shown in Figure 3. The algorithm can be applied to color video. First all I-frames and their chromaticity subblocks are extracted for color video sequences in the compression process, and the subblocks to be embedded are selected according to the C-value size of each subblock; Then, according to the characteristics of the video and watermark, the modulation mode is determined adaptively; after modulation by the third non-zero quantized DCT coefficient of the subblock, the watermark is embedded; The watermarks are replayed back to the I frame and the H.264 encoding is continued. Finally, the H.264 encoded video containing watermark is outputted.

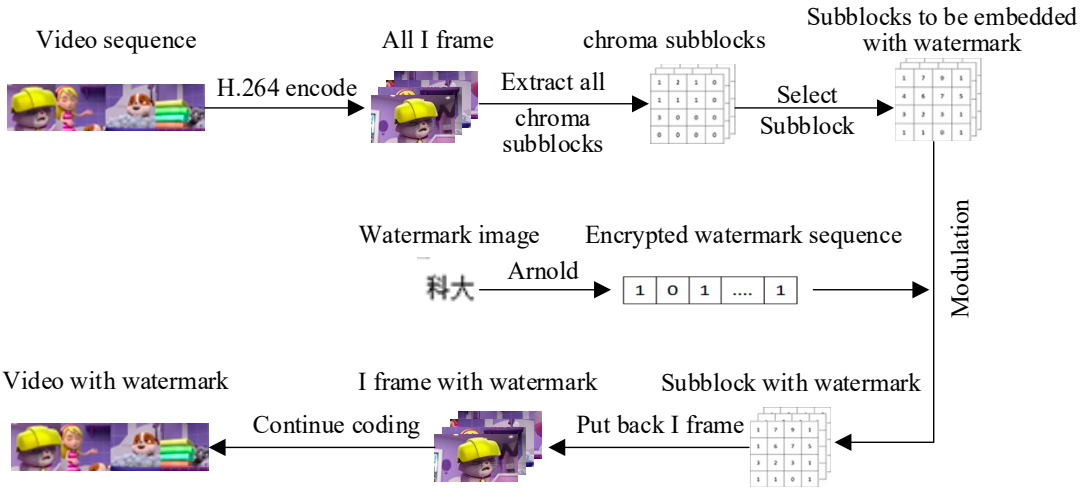


Figure 3. Watermark embedding algorithm framework.

3.1.1. Modulation Mode of Watermark

When embedding the watermark, the third quantization coefficient in the selected subblock needs to be modulated, which may affect the video quality. In order to reduce the impact, the subblock with large C value is selected in this paper. The modulation mode is determined according to whether the third quantization coefficient of the subblock is ± 1, and the strong interference with the third quantization coefficient of the subblock is ± 2 is also excluded; In addition, four modulation modes are given in this paper, and which modulation mode is selected is determined by parameters to minimize the number of modified subblocks. After analyzing and synthesizing the characteristics of existing methods and video subblocks, and after experimental verification, the optimized modulation method and the associated expressions are given. The four modulation modes are as follows:

Modulation mode 1:

$$Q3'_k = \begin{cases} Q3_k + 1 & if\ W(k) = 1\ and\ Q3_k = 1 \\ 1 & if\ W(k) = 0\ and\ Q3_k \neq 1 \\ Q3_k & other \end{cases} \quad (2)$$

Modulation mode 2:

$$Q3'_k = \begin{cases} Q3_k + 1 & if\ W(k) = 0\ and\ Q3_k = 1 \\ 1 & if\ W(k) = 1\ and\ Q3_k \neq 1 \\ Q3_k & other \end{cases} \quad (3)$$

Modulation mode 3:

$$Q3'_k = \begin{cases} Q3_k - 1 & if\ W(k) = 1\ and\ Q3_k = -1 \\ -1 & if\ W(k) = 0\ and\ Q3_k \neq -1 \\ Q3_k & other \end{cases} \quad (4)$$

Modulation mode 4:

$$Q3'_k = \begin{cases} Q3_k - 1 & if\ W(k) = 0\ and\ Q3_k = -1 \\ -1 & if\ W(k) = 1\ and\ Q3_k \neq -1 \\ Q3_k & other \end{cases} \quad (5)$$

where, $Q3_k$ is the third non-zero quantization coefficient of the k-th subblock; $Q3'_k$ is the third non-zero quantization coefficient of the kth subblock after modulation; $W(k)$ is the watermark sequence $(k = 1, 2, \cdots, 1024)$.

The parameters S_i of each modulation method are specified as follows:

$$S_1 = \sum_{k=1}^{1024} a(k), S_2 = \sum_{k=1}^{1024} \widetilde{a}(k) \quad (6)$$

$$S_3 = \sum_{k=1}^{1024} b(k), S_4 = \sum_{k=1}^{1024} \widetilde{b}(k) \quad (7)$$

where,$a(k) = \begin{cases} W(k)\ if\ Q3_k = 1 \\ \widetilde{W}(k)\ if\ Q3_k \neq 1 \end{cases}$, $b(k) = \begin{cases} W(k)\ if\ Q3_k = -1 \\ \widetilde{W}(k)\ if\ Q3_k \neq -1 \end{cases}$, If $a(k)$, $b(k)$ is 1, it means the k-th subblock needs to be modified; if it is 0, it means the k-th subblock does not need to be modified; the symbol "~" indicates the inverse. Therefore, S_i indicates the number of subblocks that need to be modified by the i-th modulation method. In this paper, the modulation method corresponding to the smallest value of S_i is selected for embedding the watermark, which can further reduce the impact of watermark embedding on the video to improve the anti-compression performance of the watermark.

3.1.2. Watermark Embedding Process

The detailed steps for watermark embedding are shown in Algorithm 1, as follows:

(1) The watermarked image of size 32×32 is encrypted by Arnold scrambling, and then it is converted into a watermark sequence $W(k)$ $(k = 1, 2, \cdots, 1024)$ of size 1×1024.
(2) The H.264/AVC encoder is used to encode the video to the end of quantization, exclude the third I frame chroma subblock with non-zero quantization factor of 2 or -2, calculate the C value of the remaining chroma subblocks, select the 1024 subblocks with the largest C value as the watermark subblocks to be embedded, and record the position of the subblocks in the D array.

(3) The S_i of each subblock to be embedded is judged, to determine the modulation mode according to their minimum value and to modify the value of $Q3_k$ to achieve watermark embedding, and embed 1-bit watermark information in each subblock.
(4) After embedding all the watermarks, the subblocks are put back to their original positions and video coding continued to obtain the H.264 video stream with watermarks.

Algorithm 1:	Watermark embedding algorithm.
Input:	Video sequence V, Watermark image *img*
Initialize	W1 = zeros(32,32), W = zeros(1,1024)
Begin:	W1 = Arnold(*img*)
	W = reshape(W1,1,1024)
	Encode V with H.264 encoding
	Extract chroma subblocks of all I frames in V
	Calculate the *C* of each subblock /*C = NNZ + Ef*/
	Select 1024 subblocks B_k with the largest *C* /*k = 1, 2, 3, ..., 1024*/
	Store the position of B_k in *D*
	Calculate S_i of B_k
	Select and record the modulation mode *M* corresponding to the minimum value in S_i
	By modulating the third non-zero coefficient of B_k to embed W(k), B'_k is obtained
	Put B'_k back to I frame
	Put I frame back in V
	Continue to encode V with H.264 encoding
Output:	Video with watermark *V*1, Subblock location *D*, Modulation mode *M*

3.2. Video Watermark Extraction Algorithm

The framework of the video watermark extraction algorithm is shown in Figure 4. Firstly, the video to be detected extracts all I frames in the decoding process, extracts the chroma subblock according to the recorded position in *D*, demodulates the chroma subblock according to the recorded modulation mode, extracts the watermark sequence, and obtains the watermark image after boosting and inverting the disorder. The pseudo code of watermark extraction is shown in Algorithm 2. The extraction methods corresponding to the four modulation methods are as follows:

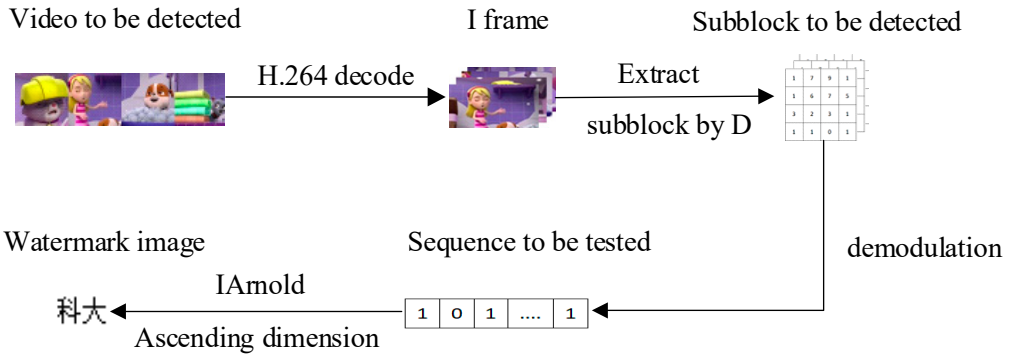


Figure 4. Watermark extraction algorithm framework.

The extraction method for modulation method 1 is:

$$w(k) = \begin{cases} 1 \; if \; q3_k \neq 1 \\ 0 \; if \; q3_k = 1 \end{cases} \tag{8}$$

where $w(k)$ denotes the *k*-th watermark value and $q3_k$ is the third non-zero quantization factor in the *k*-th subblock.

The extraction method for modulation method 2 is:

$$w(k) = \begin{cases} 1 \; if \; q3_k = 1 \\ 0 \; if \; q3_k \neq 1 \end{cases} \tag{9}$$

The extraction method for modulation method 3 is:

$$w(k) = \begin{cases} 1 \; if \; q3_k \neq -1 \\ 0 \; if \; q3_k = -1 \end{cases} \tag{10}$$

The extraction method for modulation method 4 is:

$$w(k) = \begin{cases} 1 \; if \; q3_k = -1 \\ 0 \; if \; q3_k \neq -1 \end{cases} \tag{11}$$

Algorithm 2:	Watermark extraction algorithm
Input:	Video to be detected v2, Subblock location D, Modulation mode M
Initialize:	w = zeros(1,1024), w1 = w2 = zeros(32,32)
Begin:	Decode v2 with H.264 decoding Extract all I frames in v2 Extract subblocks to be detected b_l in all I frames according to D /*l = 1, 2, 3, ..., 1024*/ Determine the demodulation algorithm according to the modulation mode M Demodulate b_l and extract w(l) w1 = reshape(w,32,32) w2 = IArnold(w1)/*IArnold is the inverse transformation of Arnold*/
Output:	The extracted watermark image w2

4. Experimental Results and Analysis

The experiments were conducted on Visual Studio2021 and MATLAB2016b platforms, using the H.264/AVC test model JM8.6 for video encoding and decoding, and programming in the C-language. The specific settings of the experimental parameters are shown in Table 1. In this paper, Hall.qcif, Mobile.qcif, Foreman.qcif, News.qcif, Stefan.cif and Bus.cif are selected as the test videos, among which, Stefan.cif is 90 frames, Bus.cif is 150 frames and all other videos are 300 frames.

Table 1. Experimental parameter configuration.

Parameter Type	Parameter Value
Frame rate	30 fps
GOP structure	IPPP
Coding grade	Baseline
Entropy coding type	CACVL
Quantitative parameters	20

4.1. Subblock Selection Rationality Experiments

In this paper, experiments were conducted on different subblock selection methods, and the total watermark experiments were conducted according to the subblocks selected by different methods. Figure 5 shows the extracted watermark BER (bit error rate) values for each group of experiments under the requantization attack, and the BER is calculated as in Equation (12). The combination of NNZ and Ef is 1 as the screening subblock condition in this paper, 2 is using only NNZ as the screening subblock condition, and 3 is using only Ef as the screening subblock condition. As can be seen from Figure 5, the BER values of

1 are mostly lower than those of 2 and 3, so the combination of NNZ and Ef for selecting subblocks results in better anti-requantization performance.

$$BER = \frac{1}{P \times Q}\left(\sum_{i=0}^{P-1}\sum_{j=0}^{Q-1} w(i,j) \oplus w'(i,j)\right) \tag{12}$$

where w is the original watermark, w' is the extracted watermark, P and Q are the size of the watermark, $\oplus$ is XOR operation.

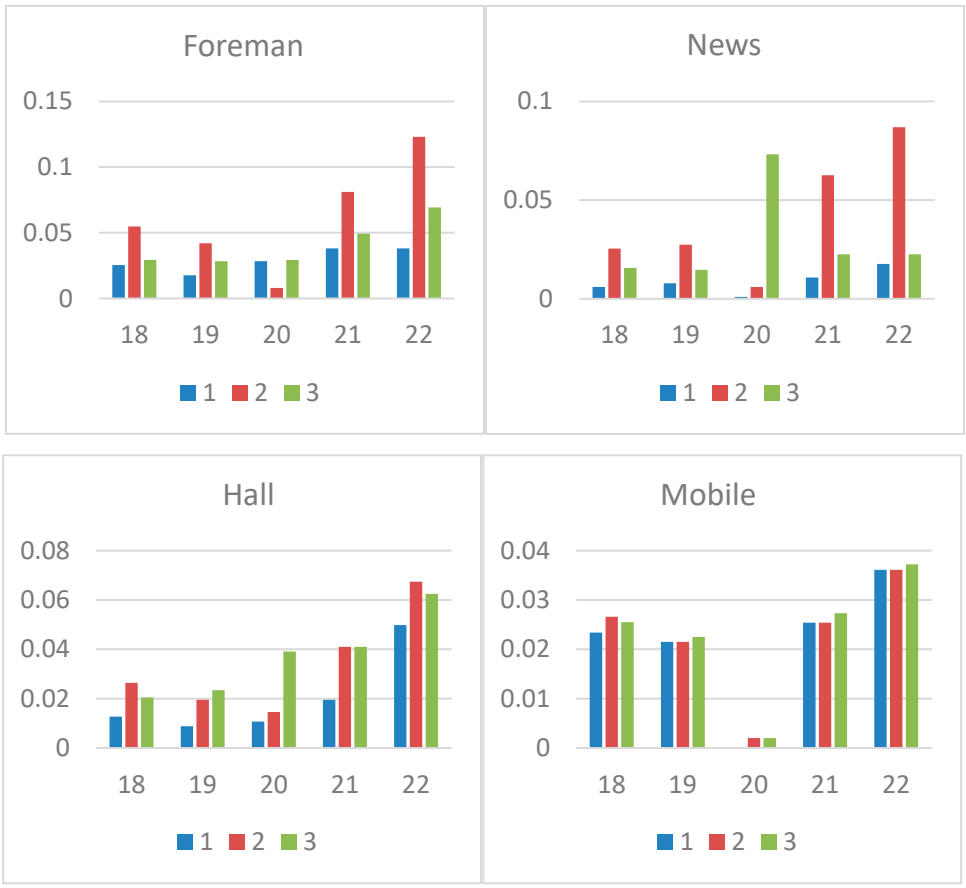


Figure 5. Requantization BER value under different screening conditions.

4.2. Invisibility Experiment

The algorithm in this paper is based on the video and watermark characteristics to select the modulation method that has the least modification to the video, thus greatly reducing the impact of the embedding of the watermark on the video quality. In this paper, we use PSNR (peak signal to noise ratio) and SSIM (structural similarity index) as video quality evaluation indexes to embed watermarks in six videos to experiment with the invisibility of the algorithm, and the results are shown in Table 2.

Table 2. PSNR and SSIM values of video after embedding watermark.

Experiment Video	PSNR1	PSNR2	SSIM
Foreman	35.6151	35.5184	0.9992
News	35.9164	35.7282	0.9974
Mobile	30.6107	30.5390	0.9991
Hall	36.0782	35.8611	0.9986
Stefan	34.6160	34.4690	0.9968
Bus	34.7719	34.7232	0.9993

In Table 2, PSNR1 is the PSNR value of the encoded video image and the original video image, and PSNR2 is the PSNR value of the video image and the original video image after embedding the watermark and encoding. It can be seen that the PSNR value of the video after embedding the watermark decreases very little compared with that before embedding, which is no more than 0.2 dB, and the SSIM values are all above 0.99, which indicates that the algorithm in this paper has good invisibility. Moreover, Stefan's experimental results show that although the video has only 90 frames, the SSIM is still above 0.99 when embedding the same size watermark as in other videos, which indicates that the algorithm from this paper has good invisibility for videos with fewer frames.

4.3. Anti-Recompression and Anti-Requantization Test

4.3.1. Anti-Recompression Test

Recompression and requantization are the two main attacks on the video, and experiments on both are conducted separately in this paper. In the anti-recompression experiment, the experimental video is recompressed with equal quantization values (the original quantization parameter is 20). The paper uses the normalized correlation coefficient NC (normalized correlation) and the false bit rate BER as evaluation metrics, and when the NC value is greater than 0.8, the correlation between the two watermarked images is higher and the algorithm is more robust [23]. The NC is calculated as follows:

$$NC = \sum_{i=0}^{P-1}\sum_{j=0}^{Q-1} w(i,j)\times w'(i,j) / \sum_{i=0}^{P-1}\sum_{j=0}^{Q-1} (w(i,j))^2 \times \sum_{i=0}^{P-1}\sum_{j=0}^{Q-1} (w'(i,j))^2 \quad (13)$$

where w is the original watermark, w' is the extracted watermark, P and Q are the size of the watermark.

Figure 6 shows the NC and BER values of the watermark extracted from the experimental video after the first, second and third recompressions, respectively. It can be seen that the NC values of the watermarks extracted from the experimental videos after three recompressions are above 0.98 and the BER are below 0.016, which indicates that the algorithm has a strong resistance to recompression.

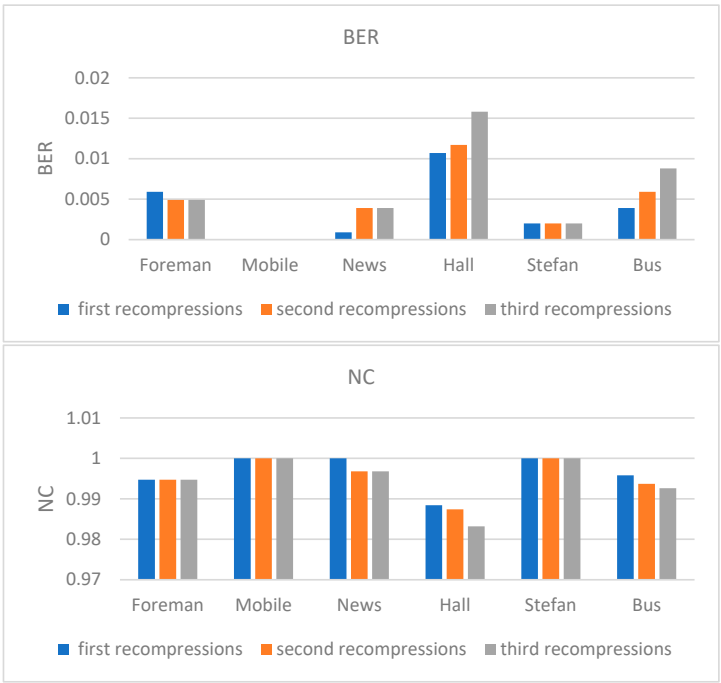


Figure 6. BER and NC values under recompression attack.

4.3.2. Anti-Requantization Attack

In this paper, experiments are performed on videos under different QP (quantization parameter) for the heavy compression attack. In the experiment, the initial QP is 20, and then a requantization attack with QP from 16 to 23 is performed to calculate the NC and BER values for the extracted watermarked images. The results are shown in Figure 7, the abscissa is the QP value, and the error rates of the extracted watermarks after requantization are all below 0.14, indicating that this algorithm can resist the requantization attack well, especially when the QP is reduced.

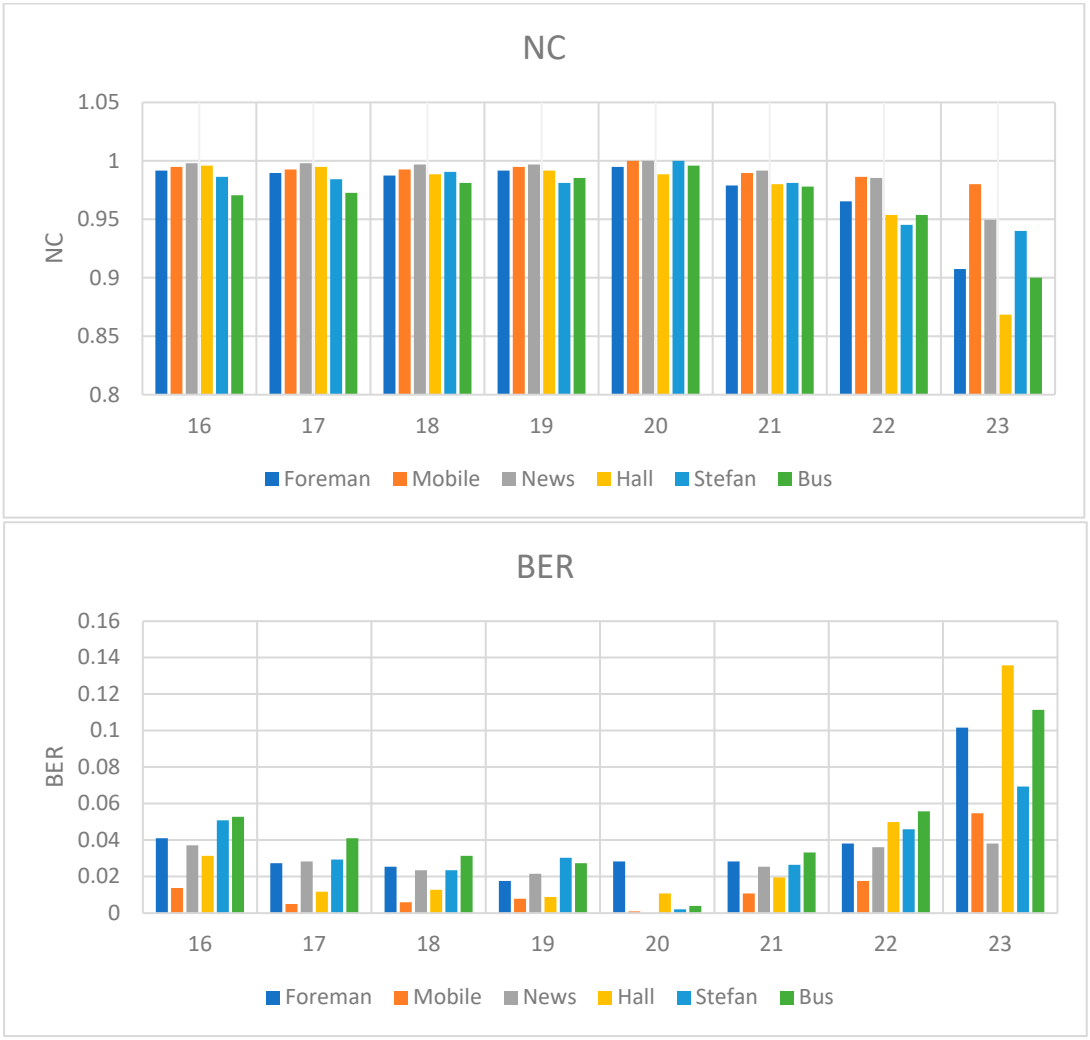


Figure 7. BER and NC values under requantization attack.

4.4. Bit Increase Rate Experiment

Because the watermark will modify the data in the video during the embedding process, the size of the video will change and affect the transmission of the video in the network. Therefore, this paper uses the bit increase rate index to measure video network affinity. The smaller the bit increase rate, the better video affinity can be obtained. The bit increase rate is calculated as follows:

$$BIR = ((BIR_{water} - BIR_{org})/BIR_{org}) \times 100\% \quad (14)$$

where, BIR_{water} is the bit rate after the video is embedded with the watermark, and BIR_{org} is the bit rate before the video is embedded with the watermark.

Figure 8 shows the bit increase rate of the proposed algorithm, all of which are lower than 0.04, indicating that the proposed algorithm has good real-time performance.

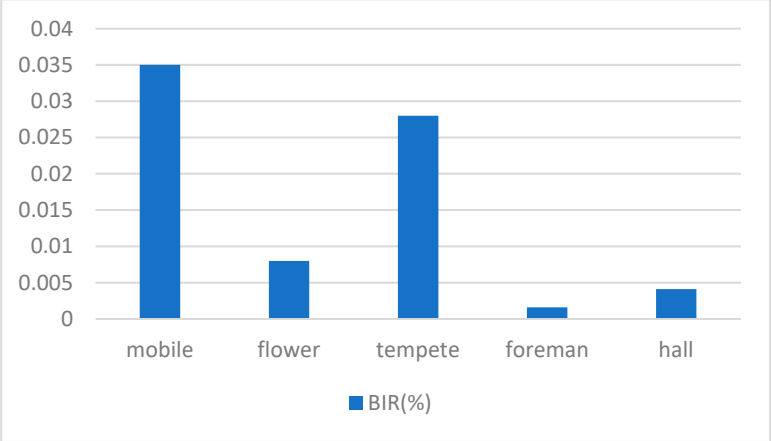


Figure 8. Experimental results of bit increase rate.

4.5. Complexity Experiment

Considering the computational complexity of the algorithm in this paper, the time required for encoding and decoding before and after embedding the watermark was measured. The experimental results comparing the encoding and decoding time of the original video and the encoding and decoding time of the video with watermark are shown in Table 3. It can be seen that before and after embedding the watermark, the time difference of encoding is no more than 5.262 s, and the time difference of decoding is no more than 5.093 s, and the complexity is low.

Table 3. Time complexity experiment results.

Experiment Video	Original Encoding	Watermarking Encoding	Difference	Original Decoding	Watermarking Decoding	Difference
Foreman	340.237	345.499	5.262	123.944	125.172	1.228
News	299.304	302.826	3.522	118.118	120.398	2.28
Mobile	359.185	361.152	1.967	126.979	121.886	5.093
Hall	315.482	317.878	2.396	135.158	132.640	2.518

4.6. Comparison Experiments with Similar Algorithms

We also conducted comparison experiments with an algorithm from the literature [24,25]. The experimental parameter settings are consistent with the comparison literature, and the data values of the comparison literature are taken from the original paper. The paper [24] is the first quantization coefficient that embeds the watermark into the subblock by selecting the subblock for embedding the watermark based on the video texture information and motion vector entropy. The paper [25] achieves the embedding of the watermark by modifying the positive and negative signs of the non-zero DCT coefficients of the chromaticity subblock. The algorithm in this paper differs from the comparison algorithm in that the modulation can be determined according to the video and watermark characteristics after selecting the subblock to achieve a more robust and invisible video watermark.

4.6.1. Invisibility Comparison

The initial QP from the literature [24] is 20, and the initial QP from the literature [25] is 28. In this paper, when performing the invisibility comparison experiment, the initial QP was consistent with the comparison literature. Tables 4 and 5 show the invisibility experimental results of the three algorithms, and it can be seen that the SSIM value of this

algorithm is higher than that of the literature [24,25], which indicates that this algorithm has better invisibility.

Table 4. The invisibility comparison between the algorithm in this paper and the algorithm in [24].

Experiment Video	SSIM of Ref. [24]	SSIM of Ours
Mobile	0.9951	0.9991
Stefan	0.9845	0.9968
Foreman	0.9941	0.9992
Bus	0.9901	0.9993
Hall	0.9763	0.9986

Table 5. The invisibility comparison between the algorithm in this paper and the algorithm in [25].

Experiment Video	SSIM of Ref. [25]	SSIM of Ours
Mobile	0.988	0.9967
News	0.970	0.997
Tempete	0.988	0.9962
Flower	0.972	0.9994
Waterfall	0.981	0.9979

4.6.2. Comparison of Recompression and Requantification

Figure 9 shows the results of the gravimetric comparison experiment with the literature [24]. In the literature [24], the initial QP in the gravimetric experiment is 20, and the value range of QP in the second compression is [16,23]. It can be seen from Figure 9 that the BER value of this algorithm is lower than that of the literature [24] and has better robustness.

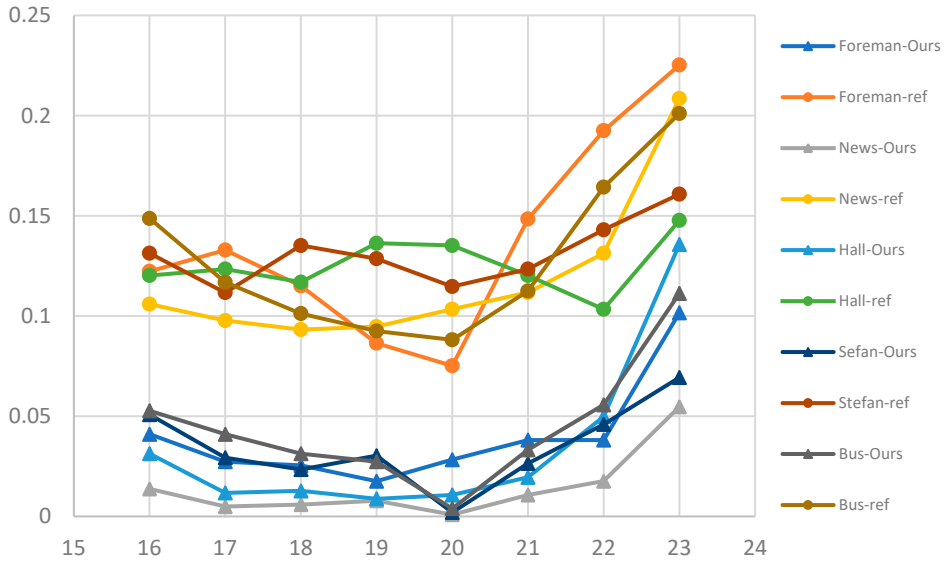


Figure 9. Comparison of BER values between the algorithm in this paper and the algorithm in the literature [24] under the recompression attack.

Table 6 shows the results of comparative experiments with the literature [25]. In the literature [25], the initial QP of the anti re-compression experiment is 28, the initial QP of the requantification experiment is 24, and the QP of the second compression is 22. From Table 6, it can be seen that the algorithm has better performance of anti-compression and quantification than the literature [25].

Table 6. Comparison between the algorithm in this paper and the algorithm in [25] under the recompression and requantization attack.

Experiment Video	NC Value under Recompression Attack		BER Value under Recompression Attack		BER Value under Requantification Attack	
	Ref. [25]	Ours	Ref. [25]	Ours	Ref. [25]	Ours
Mobile	0.98	0.9968	0.03	0.0029	0.14	0.0225
News	0.92	1	-	-	-	-
Tempete	0.94	0.98	0.1	0.0137	0.13	0.0205
Flower	0.97	0.9989	-	-	0.14	0.0225
Waterfall	0.91	0.9926	0.17	0.0078	-	-

5. Conclusions

This paper presents an anti-recompression video watermarking algorithm based on H.264/AVC. The algorithm optimizes subblock selection and watermark embedding with the proposed modulation method to minimize the number of subblock modifications. In experiments, the SSIM of the algorithms are above 0.99, the NC values are greater than 0.9, BER is 0.0271—lower than the comparison algorithms. The experimental results show that the proposed algorithm can resist the recompression and requantification attacks well, and the video quality changes little. Compared with other similar methods, it has more advantages.

The algorithm in this paper can be used to solve the security and copyright problems in the current video compression domain, and can be used to identify and protect the copyright of images in videos. At present, the information such as subblock position and modulation method is also needed in the watermark extraction. It is a non-blind watermark. Future research continues on blind extraction of watermarks and examination of algorithm performance under noise, clipping and other image attacks.

Author Contributions: D.F. the first author, proposed and completed the algorithm to guide the experiment and data analysis. Revising and improving the paper. H.Z. the second author, proposed and improved algorithms, and was responsible for experiments and data sorting. Wrote this paper. C.Z. the third author, assisted in experiments and data processing, and was responsible for drawing tables, etc. H.L. the forth author, assisted in the collaborative paper, improved the chart formula, and checked the whole paper. X.W. the corresponding author, supported the proposal and improvement of algorithms, and guided the ideas and schemes of experiments and data analysis. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the following projects: Scientific research project of National Language Commission (YB135-125).

Data Availability Statement: The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Wang, Y.F.; Zhou, Y.M.; Qian, Z.X.; Li, S.; Zhang, X.P. Review of robust video watermarking. *J. Image Graph.* **2022**, *27*, 27–42.
2. Liby, J.J.; Jaya, T. HSV model based data hiding in video for watermark applications. *J. Intell. Fuzzy Syst.* **2021**, *41*, 2731–2742. [CrossRef]
3. Amiri, M.D.; Amiri, A.; Meghdadi, M. HVS-based scalable video watermarking. *Multimed. Syst.* **2019**, *25*, 273–291. [CrossRef]
4. Di, F.; Zhang, X.; Kang, W.; Zhao, H.; Lv, Y. Video watermarking algorithm based on NSCT, pseudo 3D-DCT and NMF. *Sensors* **2022**, *22*, 4752.
5. Zhang, S.; Guo, X.; Xu, X.; Li, L.; Chang, C.-C. A video watermark algorithm based on tensor decomposition. *Math. Biosci. Eng.* **2019**, *16*, 3435–3449. [CrossRef] [PubMed]
6. Sun, W.; Zhao, H.; Zhang, X.; Sun, Y.; Liu, X.; Lv, X.; Fan, D. Zero-watermarking algorithm for audio and video matching verification. *Aims Math.* **2022**, *7*, 8390–8407. [CrossRef]

7. Asikuzzaman, M.; Mareen, H.; Moustafa, N.; Choo, K.K.R.; Pickering, M.R. Blind camcording-resistant video watermarking in the DTCWT and SVD domain. *IEEE Access* **2022**, *10*, 15681–15698. [CrossRef]
8. Rajpal, A.; Mishra, A.; Bala, R. A novel fuzzy frame selection based watermarking scheme for MPEG-4 videos using Bi-directional extreme learning machine. *Appl. Soft Comput.* **2019**, *74*, 603–620. [CrossRef]
9. Li, Y.; Wang, H. Robust H.264/AVC video watermarking without intra distortion drift. *Multimed. Tools Appl.* **2019**, *78*, 8535–8557. [CrossRef]
10. Ghasempour, M.; Ghanbari, M. A low complexity system for multiple data embedding into H.264 Coded video bit-Stream. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 4009–4019. [CrossRef]
11. Zhou, Y.; Wang, C.; Zhou, X. An Intra-Drift-Free robust watermarking algorithm in high efficiency video coding compressed domain. *IEEE Access* **2019**, *7*, 132991–133007. [CrossRef]
12. Kaur, G.; Kasana, S.S.; Sharma, M.K. An efficient authentication scheme for high efficiency video coding/H.265. *Multimed. Tools Appl.* **2019**, *78*, 21245–21271. [CrossRef]
13. El-Shafai, W.; El-Rabaie, S.; El-Halawany, M.M.; El-Samie, F.E.A. Security of 3D-HEVC transmission based on fusion and watermarking techniques. *Multimed. Tools Appl.* **2019**, *78*, 27211–27244. [CrossRef]
14. Yi, Y.; Feng, G. A Video Zero-watermarking algorithm against recompression coding for 3D-HEVC. *J. Signal Process.* **2020**, *36*, 778–786.
15. Yang, J.; Zheng, G. A fragile video watermarking algorithm based on motion vectors. *Electron. Des. Eng.* **2017**, *25*, 153–156.
16. Sun, J.; Jiang, X.; Liu, J.; Zhang, F.; Li, C. An anti-Recompression video watermarking algorithm in bitstream domain. *Tsinghua Sci. Technol.* **2021**, *26*, 22–30. [CrossRef]
17. Liu, K.; Li, C.; Tian, L.; Zhang, Y.; Song, H. A video watermarking algorithm against re-compression based on H.264/AVC. *Comput. Appl. Softw.* **2018**, *35*, 163–168.
18. Li, C.; Yang, Y.; Liu, K.; Tian, L. A semi-Fragile video watermarking algorithm based on H.264/AVC. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 8848553. [CrossRef]
19. Xue, Y.; Zhou, J.; Zeng, H.; Zhong, P.; Wen, J. An adaptive steganographic scheme for H.264/AVC video with distortion optimization. *Signal Process. Image A Publ. Eur. Assoc. Signal Process.* **2019**, *76*, 22–30. [CrossRef]
20. Sun, Y.; Wang, J.; Huang, H.; Chen, Q. Research on scalable video watermarking algorithm based on H. 264 compressed domain. *Optik* **2021**, *227*, 165911. [CrossRef]
21. Nguyen, T.S. Reversible Data Hiding Scheme Based on Coefficient Pair Mapping for Videos H. 264/AVC without Distortion Drift. *Symmetry* **2022**, *14*, 1768. [CrossRef]
22. Tian, L.; Dai, H.; Li, C. A semi-fragile video watermarking algorithm based on chromatic Residual DCT. *Multimed. Tools Appl.* **2020**, *79*, 1759–1779. [CrossRef]
23. Di, F.; Sun, W.; Zhao, H.; Kang, W. Audio and video matching zero-watermarking algorithm based on NSCT. *Complexity* **2022**, *2022*, 3445583.
24. Ding, H.; Tao, R.; Sun, J.; Liu, J.; Zhang, F.; Jiang, X.; Li, J. A compressed-domain robust video watermarking against recompression attack. *IEEE Access* **2021**, *9*, 35324–35337. [CrossRef]
25. Dai, H.; Tian, L.; Li, C. Video watermarking algorithm against recompression based on chromatic DCT. *Comput. Eng. Des.* **2019**, *40*, 1218–1224.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.