

AN EFFICIENT AUTHENTICATION METHOD for H.264/AVC

J. Zhang, *Student Member, IEEE*, A.T.S. Ho, *Fellow, IEE*

Center for Information Security

School of Electrical and Electronic Engineering

Nanyang Technological University

Singapore 639798

Email: Jingzhang@pmail.ntu.edu.sg; etsho@ntu.edu.sg

Keywords: Hard authentication, video watermarking, H.264/AVC, best mode, tree-structured motion compensation.

Abstract

With the emerging technology of the H.264/AVC standard, highly efficient H.264/AVC standard is likely to find use in a wide variety of applications such as 3G Wireless and is destined to revolutionize video picture quality over wireless network. Thus, in order for authors and designers to protect their digital works, some form of security needs to be applied to the H.264/AVC data. Unlike video data of prior videoing designs, watermarking a H.264/AVC video is not an easy task, since many newly developed techniques become the new challenges for watermarking. This paper proposed a novel and efficient scheme to authenticate the H.264/AVC video. The scheme makes an accurate usage of the tree-structured motion compensation, motion estimation and Lagrangian optimization of the standard. Authentication information is embedded strictly based on the best mode decision strategy in the sense that if undergone any spatial and temporal attacks, the scheme can detect the tampering by the sensitive mode change. And the experimental results prove the effectiveness the algorithm against many transcoding and signal processing attacks.

1 Introduction

Digital multimedia Authentication techniques have witnessed a tremendous rise in interest over the past few years. By definition, authentication is a process where by an entity proves the identity to another entity [7]. In multimedia context, video authentication aims to establish its veracity in time, sequence and content. A video authentication system ensures the integrity of digital video, and verifies that the video taken into use has not been tampered. Video authentication is important in many applications such as surveillance, journalism and video broadcast, etc.

In the past, several techniques and concepts based on data hiding or steganography, have been introduced for temper detection in digital images and video. One class of authentication watermarks is *Hard Authentication* [3]. Hard Authentication rejects any modifications to multimedia content. The inserted watermark is so weak that any manipulations to the multimedia con-

tent disturbs its integrity. A simple approach referred as the Yeung-Mintzer Scheme [6] enabled single pixel authentication but only half of modified pixels on average can be detected. And the scheme's security depends critically on the secrecy of the watermark logo. Another approach is to partition a multimedia signal into two disjointed parts: a signature part and an embedding part. An authenticator such as a message authentication code or a digital signature is generated from the content of the signature part and is then embedded into the embedding part. One of the first fragile watermarking techniques was to insert key-dependent check sums of the seven most significant bits into the least significant bit (LSBs) of pseudo-randomly selected pixels as proposed in [8] and [9]. Lossless watermarking in [2] and [4] used a spatially additive, signal-independent robust watermarking to embed signal authentication data using a reversible modular addition. The watermark has to be robust enough to survive the reversible addition in the watermarking process so that for an unmodified watermarked signal the authentication data can be correctly recovered and the original watermark can be subsequently regenerated and recovered from the watermarked signal to recover the original signal. The amount of authentication data is typically constrained by the limited embedding capacity of the underlying robust watermark.

Although the concepts of hard authentication have been well studied, there are not many research works dealing with authenticating the state-of-the-art H.264/AVC standard. With the emerging technology of the H.264/AVC standard, H.264/AVC has achieved a significant improvement in the enhanced compression performance, providing, typically, a factor of two in bit-rate savings when compared with existing standards such as MPEG-2 video [11]. It also provides a network-friendly video representation which addresses conversational (video telephony) and non-conversational (storage and broadcast) applications [12]. However, whole new challenges and associated tradeoffs need to be considered for watermarking H.264/AVC data. One characterizing main difference is that since the highly compressed data almost contains no noise components to embed the watermark.

With the exhaustive investigation of the related techniques of the state-of-the-art H.264/AVC standard, we develop a novel and efficient watermarking algorithm for authentication. This algorithm works directly in the compressed domain. The

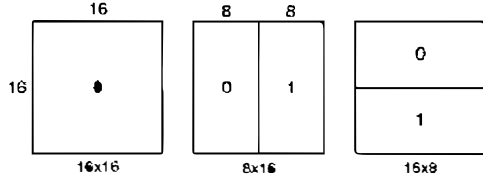


Figure 1: Macroblock partition modes: 16×16 , 8×16 , 16×8 , 8×8

scheme makes an accurate usage of the tree-structured motion compensation, motion estimation and Lagrangian optimization for mode decision. By making use of this feature to our advantage, a careful and detailed study was done on how this feature can be exploited in the implementation of a digital watermarking scheme. The authentication information is represented by a binary watermark sequence and embedded into video frames. And the experimental results prove the effectiveness the algorithm.

The rest of this paper is organized as follows. Section 2 provides the reader with related techniques of the H.264/AVC standard. Section 3 gives a detailed explanation of the H.264 hard authentication algorithm. Analysis for reader's better understanding is in Section 4. Some experimental results of the developed algorithm are covered in Section 5. Section 6 concludes the paper with the outline of future work.

2 Investigation of Best Mode

2.1 Tree-structured Motion Compensation

Significant differences of the H.264/AVC standard from earlier standards include the support for a range of block sizes (from 16×16 down to 4×4) for prediction and fine sub-sample motion vectors (quarter-sample resolution in the luma component).

The luma component of each macroblock may be split up and motion compensated in four ways as shown in Figure 1. In cases where the 8×8 partition mode is chosen, each of the four 8×8 sub-macroblocks within the macroblock can be further split up in four ways as shown in Figure 2. This method of partitioning macroblocks into motion compensated sub-blocks of varying sizes is known as *tree-structured motion compensation* [12]. Therefore, the choice of partition mode significantly impacts the compression performance. In general, a large partition mode is appropriate for homogeneous areas of the frame and a small partition mode may be beneficial for detailed areas. Figure 3 shows four macroblocks with different partition modes.

The best mode I^* for a macroblock S is selected by minimizing the expression in Equation (1) within the constrained R and minimized D , using Lagrangian Optimization technique [1] of H.264/AVC, among all possible modes (denoted by \mathcal{I}):

$$I^* = \arg \min_{I_c \in \mathcal{I}} (D(S, I_c) + \lambda R(S, I_c)) \quad (1)$$

where λ denotes the predetermined Lagrangian multiplier for

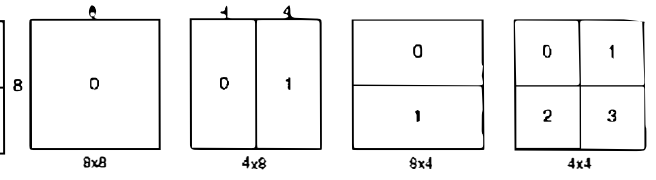


Figure 2: Sub-macroblock partition modes: 8×8 , 4×8 , 8×4 , 4×4

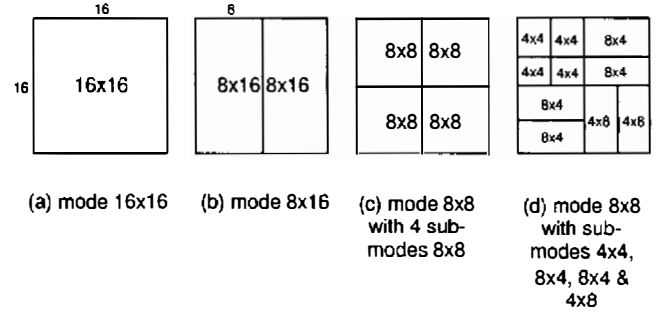


Figure 3: examples of partition modes using the tree-structured motion compensation

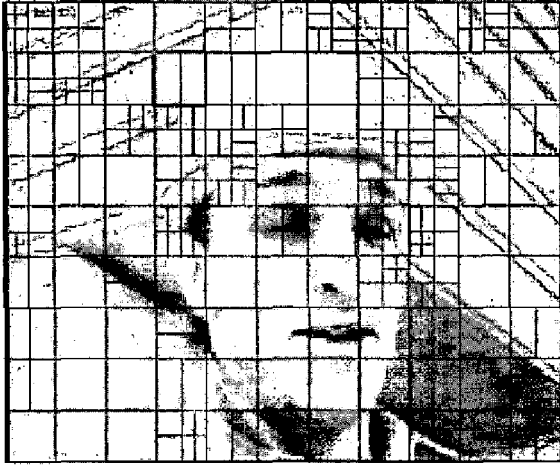
mode choice, and D and R represent the distortion and consumed bits for encoding the current mode I_c , respectively.

Figure 4(a) shows one frame of *Foreman* and the partition mode selection map. In areas where there is little change between the frames, a 16×16 partition mode is chosen and in areas of detailed motion, small partitions are more efficient.

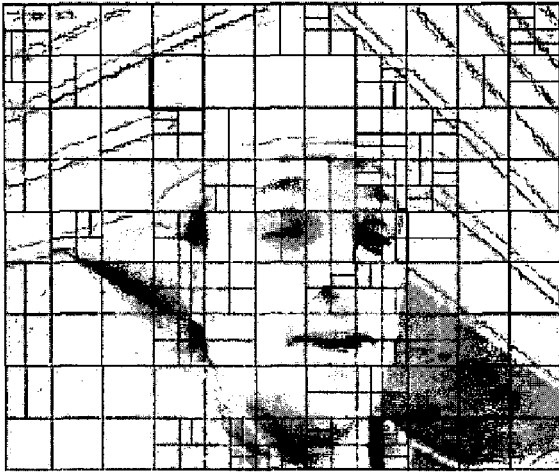
2.2 Partition Mode Change

For each macroblock, the H.264 encoder selects the best partition mode among all possible partition modes that minimizes the amount of information to be coded and sent. During transmission, once the coded stream undergoes any transcoding process (such as recompression, bit-rate change, and frame-rate change), the information may alter significantly. Thus the best partition modes for some macroblocks may be different. If more malicious signal processing are applied, more macroblocks may have different best modes.

Figure 4(b) shows the same frame as in Figure 4(a) and the new partition mode selection map after recompression. From these two figures, the best partition modes of many macroblocks are not the same. For example, for the fourth macroblock in the second row, due to recompression, its best mode changes from 8×16 to 16×16 . For the eighth macroblock in the fifth row, its original best mode is 8×16 . After recompression the new best mode is 8×8 . And each 8-by-8 sub-macroblock also has a best sub-mode: 8×8 , 4×8 , 8×8 , 4×8 . For some macroblocks, even if their best modes are the same, the information may be different. Thus the DCT coefficients may not be the



(a) mode map of the original coded B-frame



(b) mode map of the recompressed B-frame

Figure 4: *Foreman's* partition mode selection: (a) of the original coded B-frame, (b) of the recompressed B-frame.

same, when compared to the coefficients before recompression.

3 Proposed Authentication Method

3.1 Embedding

In a H.264/AVC video sequence, there are a total of five different types of slices: *I* (*Intra*), *P* (*Predicted*), *B* (*Bipredictive*), *SI* (*Switching I*) and *SP* (*Switching P*) [11]. Our algorithm is developed for the inter-predicted slices: *P*- and *B*-slices.

As mentioned before, the encoder needs to find the best partition mode for each macroblock, as different modes will produce different sets of bitrate and distortion to the video stream. The encoder will go through the motion estimation and compensation, transformation, quantization and entropy coding for all possible partition modes, and the Lagrangian optimization technique determines which partition mode has the lowest rate-

distortion related cost in Equation (1). Only when the minimum cost is attained, the encoder will allocate the corresponding partition mode as the best mode to the macroblock. Through careful observation of the mode decision scheme, it can be certain that in region where there is no motion (such as background), a partition mode of 16×16 is chosen by the encoder. In areas where there is a lot of detailed motion, smaller partition modes prove to be more efficient.

Therefore, by the use of the mode decision scheme of the encoder, we could implement our watermarking algorithm targeting at higher motion activities macroblocks with the best mode 8×8 (with four sub-modes chosen from 4×4 , 8×4 , 8×4 and 8×8). By choosing these smaller partition modes, it is difficult for the Human Visual System to detect the differences, or the distortions introduced by the watermark embedding scheme. Therefore, watermark components are only embedded in the macroblocks of best mode 8×8 with all possible sub-modes (4×4 , 8×4 , 8×4 and 8×8).

Looking at the iterative procedure of the mode decision for each macroblock, the watermark embedding process needs to follow the flow, to fulfil the watermarking targets. Therefore, although a period of the watermark sequence may be embedded in a macroblock, the macroblock best mode may not match what is required. Hence, the macroblock best mode is checked at the end of every encoding run of a macroblock to ensure that the watermark sequence is properly embedded. If the best mode is not what is needed, the embedding of the watermark sequence will have to be restarted.

For example, in one run of the iterative mode decision procedure for a macroblock S , if the current prediction mode I_c is 8×8 (with four sub-modes 4×4 , 8×4 , 8×4 and 4×8) (Figure 3(d)), the watermark components are embedded, starting with the 10th coefficient. As the coefficients represent the residual to code, all zero coefficients are avoided to prevent the video from getting badly distorted. The nonzero quantized DCT coefficients $D(u, v)$ are replaced by the watermark components w_i , $i \in N$ (N is the number of nonzero coefficients):

$$D(u, v) = w_i \quad (2)$$

where w_i is derived from the binary authentication coder as follows:

$$w_i = \begin{cases} 1, & \text{coder} = 0 \\ 2, & \text{coder} = 1 \end{cases} \quad (3)$$

In the H.264/AVC standard, the encoder will first carry out the motion estimation and mode decision for modes 16×16 , 16×8 and 8×16 , for the unitary macroblock, and compute the corresponding rate-distortion related costs in Equation (1). These functional blocks appear in the black dotted square in Figure 5. After big partition modes, the encoder will apply the motion estimation and mode decision for mode 8×8 with the sub-modes 4×4 , 8×4 , 8×4 and 8×8 , for four 8×8 sub-macroblocks, as shown in the red dotted square. The proposed authentication scheme embeds the watermark components in this phase, as shown in Figure 5. Suppose 6 watermark components are embedded, so the last watermark to be embedded is the 15th coefficient.

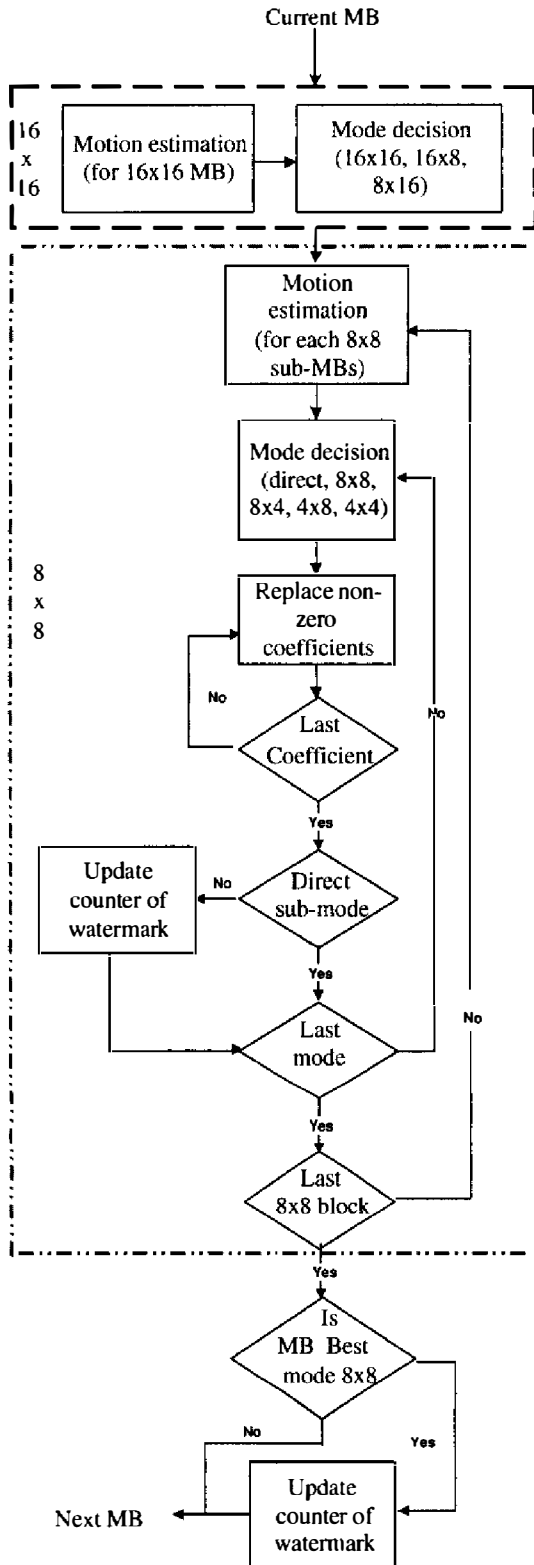


Figure 5: Flow Diagram of Watermark Embedding Process (Enclosed in the Red Dotted Line is the embedding Process).

After embedding watermark components w_i into all the available modes (denoted by \mathcal{I}), the best mode I^* for the marked macroblock S_m is selected by minimizing the expression in Equation (1):

$$I^* = \arg \min_{I_c \in \mathcal{I}} (D(S_m, I_c) + \lambda R(S_m, I_c)) \quad (4)$$

where λ , D and R have the same meanings as in Equation (1). After all possible modes are tested for the minimum rate-distortion cost of the current macroblock, the encoder is checking whether the partition mode 8×8 with the 4 sub-mode 4×4 , 8×4 , 8×4 and 4×8 has the minimum cost. In case this is the best partition mode I^* , a counter of watermark sequence will be updated. And the beginning watermark component for the next macroblock will be the 16th component. However, if the rate-distortion cost is not minimal, the watermark sequence will have to redo the embedding. That is, the algorithm will have to embed the watermark components started from 10th coefficient again, when the encoder checks the rate-distortion cost of the next partition mode. Hence, the algorithm will be more complex and the sequence to embed needs to be scrutinized.

3.2 Extraction

In the decoding process of the video stream, the retrieval of the watermark will be performed. Compared to the embedding process, the watermark extraction is rather straight forward.

During decoding a macroblock, if the best mode is 8×8 , all the nonzero quantized DCT coefficients (level values of entropy coding) are extracted to form the watermark sequence for authentication in a 1-D order. If the best mode is not 8×8 , the extraction process skips to the next macroblock till the end of the frame. And the extracted watermark sequence is compared to the original authentication watermark information to check if the video has been tampered.

4 Analysis

4.1 Computational Complexity

Initial suspect is the time involved. On the surface, it seems that the time needed will increase. But, the extra time needed is nothing compared to the time taken to encode a frame. This is because, during the encoding process, much of the time is taken up by the motion estimation and motion compensation. The number of computations involved in these functions occupies the majority of the encoding time. Hence, there is no need to consider the time factor.

4.2 Watermarking Capacity

The watermark capacity per frame is mainly determined by (a) the motion activity, (b) the distortion introduced by watermark and (c) the *Direct/Skip* prediction modes in B-/P-slices.

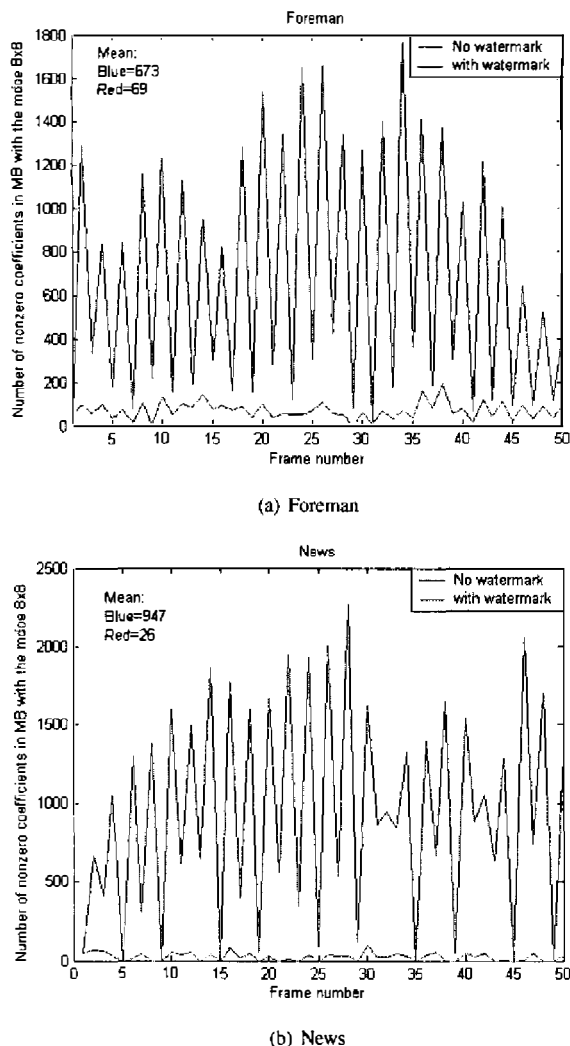


Figure 6: The number of the nonzero coefficients in the MBs with the mode 8×8 without(blue) and with(red) watermark.

All the watermark components are embedded in the macroblock with the best mode 8×8 . The nonzero coefficients in the remaining macroblocks are untouched. Therefore, the watermark capacity is the number of the nonzero coefficients of the macroblocks with the best mode 8×8 . Figure 6 show the capacity of video *Foreman* and *News*. The blue lines are the number of the nonzero coefficients without watermark, and the red lines are the number of the nonzero coefficients with watermark.

For the video sequence with significant motion activities, many macroblocks are allocated with the mode 8×8 . Therefore, a significant amount of suitable nonzero coefficients are produced. While if the video contains modest motion activities, the number of suitable nonzero coefficients decrease. Less watermark components can be embedded into these video sequences. This is proved in Figure 6 by comparing the capacity of two different video sequence with watermark (red lines).

The mean value of the capacity of *Foreman* (69 per frame) is almost 3 times to the capacity of *News* (26 per frame).

The distortion introduced by the watermark embedding process also affects the watermark capacity. More distortion is caused, less suitable nonzero coefficients are produced. This is because that the other bigger partition modes (such as 16×8) may produce the minimal rate-distortion cost. For example, in Figure 6(a), the potential watermark capacity mean per frame without watermark (the blue line) is approximately 9.8 times to the capacity mean with watermark (the red line). Thus, decreasing the watermark distortion (such as by using watermark gain adaptive to the local characteristics) can increase the watermark capacity.

In addition to the motion-compensated macroblock partition modes described before, a P-slice macroblock may also be coded in the *SKIP* mode [11]. If a macroblock has motion characteristics that allow its motion to be effectively predicted from the motion of neighboring macroblocks, and it contains no or very few non-zero quantized transform coefficients, then it is flagged as skipped. For this mode, neither a quantized prediction residual signal, nor a motion vector or reference index, has to be coded and transmitted. B-slices also have a special prediction mode - *Direct mode*. In this mode, no prediction error signal is coded and transmitted. It is also referred to as *B-slice SKIP mode* and can be coded very efficiently, in a similar way to the *SKIP* mode in P-slices. In this situation, no watermark components are embedded, thus, the capacity decreases. This has been considered in the proposed authentication scheme, as shown in Figure 5.

4.3 Sensitivity against spatial tempering

An important and inherent feature of our algorithm is that the embedding location selection is tightly dependent on the mode decision scheme. In our method, since the embedding location selection is based on the best mode of macroblocks (8×8), a small change in the content of a frame can lead to new mode allocation. That is, the original best mode 8×8 may change to other big partition modes or even direct/skip mode, and the locations of suitable nonzero coefficients change in the video. Even one change of one macroblock's best mode affects the correct extraction of the embedded watermark in this macroblock. This change is propagated from this period of extracted watermark to the following watermark components, which eventually destroys the extraction synchronization of the entire watermark sequence.

4.4 Sensitivity against temporal tempering

Another main feature of the H.264/AVC made use of in our algorithm is that the standard supports *Multi-picture Motion-compensated Prediction* [10]. That is, more than one previously coded pictures (≤ 5) can be used as a reference for motion-compensated prediction. Even previously coded B-pictures can be references. Therefore, if any of the frame in the video sequence changes (content-wise or position-wise), it is reflected not only in itself, but also in all the sequent frames.

For instant, any alteration such as dropping and reordering of the K_{th} frame is reflected on the K_{th} frame, just as mentioned in the previous sub-section. That directly changes the sequent frames which take the K_{th} frame as the prediction reference. Eventually all the sequence frames are affected directly or indirectly. So our algorithm is very secure against all the temporal attacks.

5 Experiments and Results



(a) Original



(b) Watermarked

Figure 7: The original and watermarked *Foreman*.

5.1 Test Conditions

The proposed watermarking technique has been integrated into the H.264 JM-9.0 reference software [5]. The video sequences: *Foreman*, *Stefan*, *Coastguard*, *Mobile*, *Bus*, *Football* and *News* are used in the experiments. All video clips are coded in CIF format (352×288 pixels) at the frame rate 30 frames/s at the bit-rate 512 kbit/s. The GOP structure comprises IBPBP..., compliant to the *Main Profile* of H.264/AVC. A binary water-

mark sequence is used as the authentication information for our experiments.

5.2 Fidelity after Watermarking

Figure 7 shows the frame samples from the unmarked and marked video clips with PSNR(dB). The unmarked and marked video clips are reconstructed from the compressed data without and with watermarks, respectively. The average PSNR values are computed by comparing the reconstructed video frames to the original raw video frames.

On average, the watermark insertion leads to a decrease of approximately 0.04 dB for all video clips coded at 512 kbits/s, as shown in Table 1. In the experiments, no visible artifacts can be observed in all of the test video sequences.

Table 1: The average PSNR comparison of the unmarked and watermarked frames.

PSNR(dB)	Unmarked	Watermarked	Difference
Foreman	39.4	39.5	0.1
Mobile	37.4	37.4	0.0
News	41.0	41.1	0.1
Bus	37.7	37.8	0.1
Football	39.2	39.2	0.0
Stefan	38.8	38.8	0.0
Coastguard	37.4	37.4	0.0

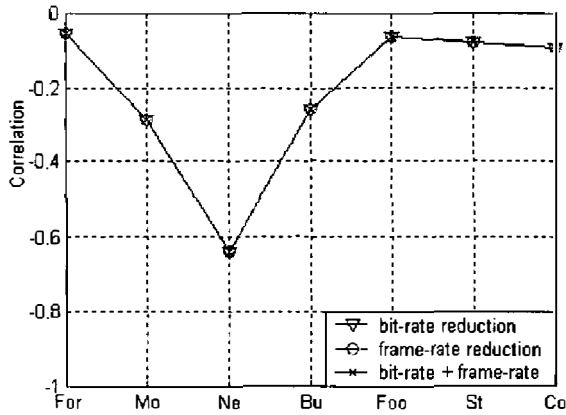
5.3 The Results of Authentication Tests

Two categories of video attacks have been applied to the marked video to test the sensitivity of the authentication algorithm: transcoding and common signal processing processes.

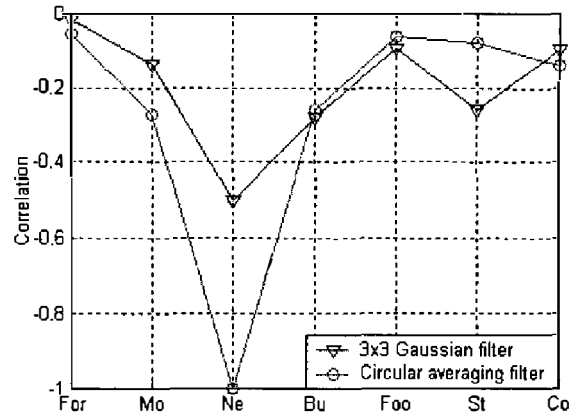
In the first group, we utilized the bit-rate reduction, frame-rate reduction, frame reordering, and frame replacing. During watermarking in the H.264/AVC encoding process, the Lagrangian Optimization technique targets the overall consuming bit-rate at 512 kbits/s. When transcoding is applied after watermarking, the bit-rates have been reduced to approximately the 1/2 of the original bit-rates. During re-encoding with the bit-rate reduction, most high frequency components which represent detailed texture will be discarded. To applying the transcoding of frame-rate reduction, the frame-rate is changed from 30 to 15 frames/s and the video is re-compressed.

After decoding the marked bitstream, common signal processing attacks are applied to the raw video frame by frame, including 3×3 Gaussian low-pass filtering, circular averaging filtering, unsharpened contrast enhancement, additive Gaussian noise (mean=0, variance= 0.001), cropping (central 85% remains) and rotation (-1 degree). Then the attacked raw video is coded again. Thus, there are actually two attacks applied each time: decoding and re-encoding, and common signal processing.

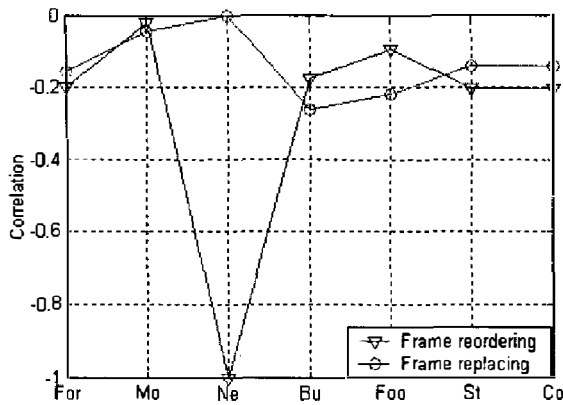
The authentication security of our watermarking algorithm is



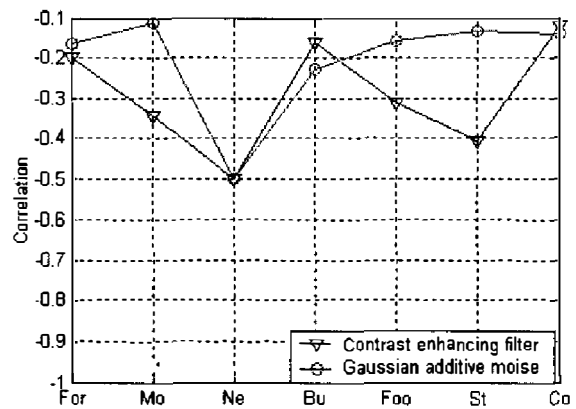
(a) Bit-rate/frame-rate/bit-rate + frame-rate reduction



(a) 3 × 3 Gaussian filter and Circular averaging filter



(b) Frame Reordering and replacing



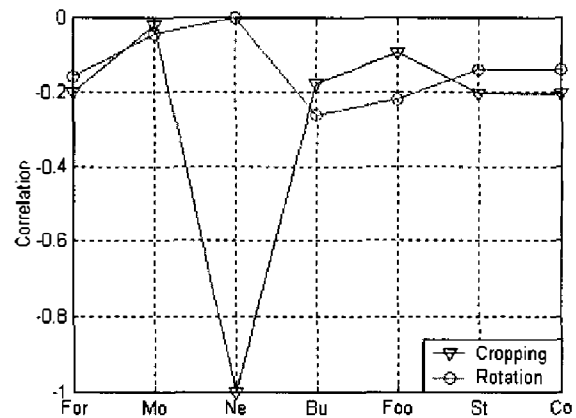
(b) Contrast enhancing filter and Gaussian Noise

Figure 8: Sensitivity under transcoding on *Foreman*, *Mobile*, *News*, *Bus*, *Football*, *Stefan* and *Coastguard* (denoted by *For*, *Mo*, *Ne*, *Bu*, *Foo*, *St* and *Co* in the horizontal axis) (512 kbit/s, CIF-size).

represented by the sensitivity against attacks. The standard normalized correlation values are measured and shown in Figure 8 and 9, with the dynamic range from 1 to -1 . From these two figures, almost all the correlation values are less than 0. For *News*, our algorithm is significantly sensitive to the transcoding and common signal processing attacks. The correlation values even reach -1 when the video undergoes frame reordering, circular averaging filter and cropping(85%).

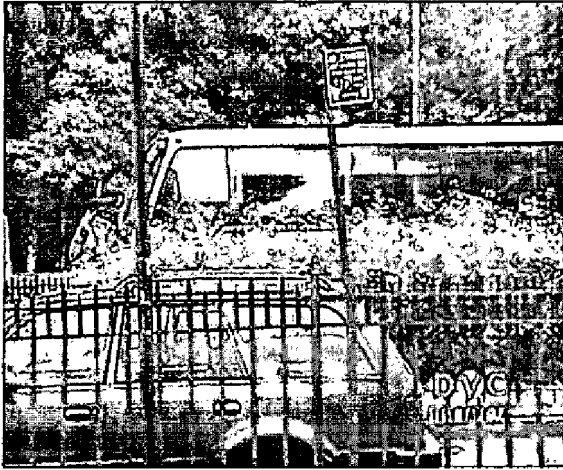
One special attack, the cutting and pasting attack is also applied to test the security of our authentication system. For example, the 8×48 square of *DVC* characters in the bottom right corner of the *Bus* frame is cut and replaced as shown in Figure 10. Even though the square is so small (compared to the frame size 352×288), our algorithm shows the significant sensitivity at the average correlation value equal to -0.261 .

6 Conclusions



(c) Cropping and Rotation

Figure 9: Sensitivity under common signal processing on *Foreman*, *Mobile*, *News*, *Bus*, *Football*, *Stefan* and *Coastguard* (denoted by *For*, *Mo*, *Ne*, *Bu*, *Foo*, *St* and *Co* in the horizontal axis) (512 kbit/s, CIF-size).



(a) Watermarked



(b) undergone cutting and pasting

Figure 10: The watermarked and the attacked *Bus* under cutting and pasting attack.

The proposed hard authentication algorithm performed well in terms of sensitivity against transcoding and common signal processing. The watermarked H.264/AVC video clips maintained the good visual quality and almost the same Bit-rate. However our algorithm lacks the ability to provide further information necessary to characterize the attack. Therefore, the future work will focus on enhancing the proposed algorithm by localizing the attacked areas.

References

[1] A. Joch, F. Kossentini, H. Schwarz, T. Wiegand and G.J.Sullivan, "Performance comparison of video coding standards using Lagrangian coder control," *Proc. IEEE*

Int. Conf. Image Processing, vol. 4, pp. IV-3728-IV-3731, (2002).

[2] C.W. Honsinger, P.W. Jones, M. Rabbani and J.C. Stoffel, "Lossless recovery of an original image containing embedded data," *U.S. Patent*, no. 6,278,791, (2001).

[3] B.B. Zhu, M. D. Swanson and A. H. Tewfik, "When Seeing Isn't Believing," *IEEE Signal Processing Magazine*, (2004).

[4] J. Frodroch, M. Goljan and R. Du, "Invertible authentication," in *Proc. SPIE, Security watermarking of Multimedia Contents*, vol. 3971, pp. 197-208, San Jose, CA, (2001).

[5] K. Hühning, E.d., H.264/AVC Joint Model 9.0 (JM-9.0) Reference Software. [Online]. Available: ftp://imtc.org/jvt-exports/reference_software/.

[6] M.M. Yeung and F. Mintzer, "An Invisible Watermarking technique for image verification," in *Proc. IEEE Int. conf. Image Processing*, pp. 680-683, (1997).

[7] P.K. Atrey, W.-Q. Yan, E.-C. Chang and M. S. Kankanhalli, "A Hierarchical Signature Scheme for Robust Video Authentication using Secret Sharing," in *Proc. the 10th International Multimedia Modelling conference*, (2004).

[8] P.W. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Trans. Image Processing*, vol. 10, no. 10, pp. 1593-1601, (2001).

[9] S. Walton, "Information authentication for a slippery new age," *Dr. Dobbs J.*, vol. 20, no. 4, pp. 18-26, (1995).

[10] T. Wiegand and B. Girod, "Multi-Frame Motion-Compensation Prediction For Video Transmission," *In the 6th meeting, Awaji, JP, Island, Dec. 2002*, Kluwer Academic Publisher, USA, (2001).

[11] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and compression of video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688-703, (2003).

[12] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560-576, (2003).