Proceedings of the 6th WSEAS International Conference on Multimedia Systems & Signal Processing, Hangzhou, China, April 16-18, 2006 (pp25-30)

Implementation of Video Watermarking using Error Correcting Codes for MPEG2 video

*YUK YING CHUNG, *FANG FEI XU, *XIAOMING CHEN, **CHANGSEOK BAE
*School of Information Technologies, University of Sydney, NSW 2006, AUSTRALIA
** Post-PC Research Group, ETRI, 161 Gajeong Dong, Yuseong Gu, Daejeon, KOREA

Abstract: - Video watermarking is an important method of protecting the intellectual property copyright of the video media. It allows embedding of copyright information into the video pictures. In this paper a new digital video watermarking scheme with an **E**rror **C**orrecting **C**ode (***ECC***) is proposed. This watermarking scheme maximizes the watermark payload while minimizing the perceptual degradation of video quality caused by the embedded watermark by means of an appropriate choice of embedding position. Two error correcting codes, BCH(31,8) and Turbo (3,1) were implemented and compared. We found that BCH(31,8) achieved higher error correcting capability than Turbo (3,1) under the simulated noise tests.

Key-Words: - MPEG-2, error correction codes, digital watermark, DCT, video processing.

1 Introduction

The success of the Internet and popular digital recording storage devices, the promise of higher bandwidth and quality of service (QoS) for both wired and wireless networks have all made it possible to create, replicate, transmit, and distribute digital content in an effortless way. Therefore the protection and enforcement of intellectual property rights for digital media has become an important issue. *Digital watermarking* [2][3], which allows of embedding copyright information into the digital document has become increasingly indispensable.

Digital watermarking [2][3] can be applied to various types of digital documents such as image, video, audio and text. The video watermarking must be done under the triple contradictory constraints of *imperceptibility*, *robustness* and *capacity*. In other words a sufficient number of watermark bits should be embedded into the video images without causing noticeable distortion. The watermark should be correctly retrieved at the decoding stage, even after various types of image manipulation and other signal processing attacks.

In this paper a new DCT- based digital watermarking scheme for MPEG-2 [1] video is proposed and implemented. The system embeds a watermark into the quantized DCT coefficient during the MPEG-2 video encoding process. One watermark bit is embedded into the LSB of the DC coefficient in each DCT coefficient block of I-frames. This achieves the optimal tradeoff between watermark payload and distortion to video quality due to the embedded watermark bits. This watermark scheme can provide large capacity for watermark bitstream and has been proven to be perceptually invisible. In addition the watermark extraction process can be achieved without knowledge of the original watermark. **E**rror **C**orrecting **C**ode (***ECC***) [4][5] is employed to improve this watermarking scheme in terms of watermark robustness. With the intention to explore an effective way of applying coding methods for watermark protection purpose, different ways of applying error correcting codes were tested and investigated.

In this paper, Section 2 explains the details of the proposed video watermarking system. Section 3 gives an introduction to the ECC and how we can apply this coding to the video watermarking. Section 4 and 5 are the testing results and conclusion respectively.

2 The proposed video watermarking

The proposed video watermark embedding system and watermark extraction system are presented in Figure 1 and Figure 3.

2.1 Watermark embedding process

The first step of the watermark embedding process is the preprocessing of watermark information. Suppose the watermark information is a character string. It is first converted into its binary format using the ASCII value of each character. Since all ASCII values are in the range of 0 to 255, an 8-bit string is used to represent one character. For example, the character ‘a’ with ASCII value ‘97’ is converted into the bit string ‘01100001’. Then the error correcting coding may be applied on the original information bit string to produce the extended watermark bitstream.

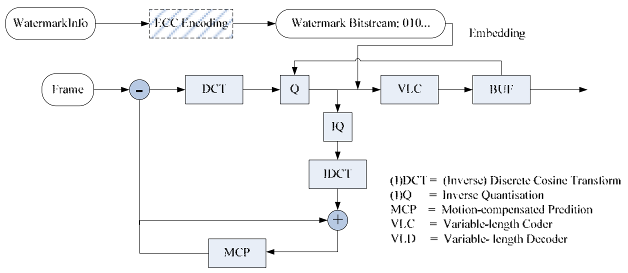

Figure 1 The block diagram of watermark embedding system with MPEG-2 encoder

In the proposed watermarking scheme the watermark is embedded into each I-frame of video sequence bit by bit when encoding the image sequence to MPEG-2 video. This watermarking process does not alter the motion vector information or any of the critical side information. During the MPEG-2 video encoding process in Figure 1, each 8×8 block of pixels of the I frames are transformed into 8×8 block of DCT coefficients and then quantized. For each quantized DCT coefficient block, one bit of the watermark is embedded in the *least significant bit* (LSB) of the quantized DC coefficient by changing the value of the LSB to the value of the embedded bit. A schematic diagram of this process is given in Figure 2.

In the proposed system, the LSB of quantized DC coefficient is chosen to embed a watermark bit. First, DCT concentrates most energy of the original pixel block into a few low frequency coefficients during the MPEG-2 video encoding process. After quantization, many mid- or high- frequency DCT coefficients are set to zero. The entropy coding in the next stage of these zero coefficient are not be separately coded so as to reduce the bit rate of video bitstream. If the watermark bits are embedded in the mid or high frequency coefficients, these zero-coefficients may become non-zero ones, which results in the increase of bitrate which influences the compression ratio. Since the DC coefficient, also called the zero-frequency coefficient, always has non-zero value, the watermark bit can be embedded into it without increasing the bitrate of compressed video bitstream.

Second, a watermark embedded in the compressed video can be considered as noise that causes distortion to video quality. Since the human visual system (HVS) is more sensitive to noise in low frequency range where most energy concentrates, we must make sure the DC coefficient of a DCT coefficient block will not be changed too much after embedding the watermark bit. Changing the LSB only is a natural choice to minimize the change to the DC coefficient.

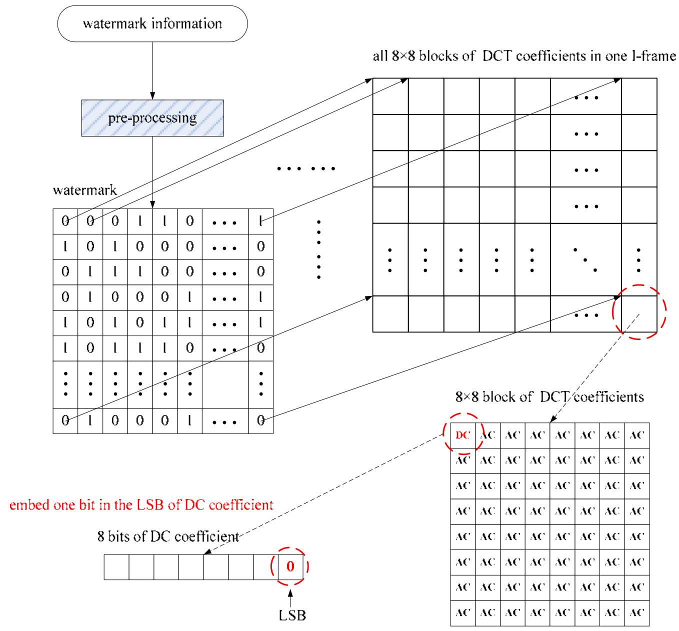

Figure 2 The schematic diagram of watermark embedding process

2.2 Watermark Extracting Process

The watermark extraction process is the inverse of watermark embedding process. When decoding MPEG-2 video to image sequence, the watermark is retrieved bit by bit from the LSB of DC coefficient of each DCT coefficient block before de-quantization. The extracted watermark bits are then manipulated to recover the watermark information. A block diagram of the watermarking extraction process is shown in Figure 3.

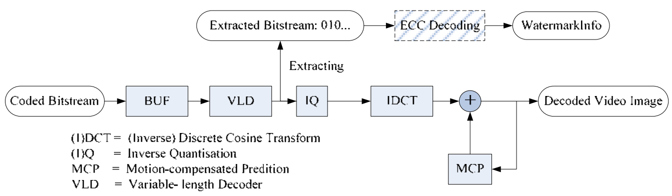

Figure 3 The block diagram of watermark extraction system with MPEG-2 decoder

During the watermarking extraction process we can implement simple error detection and recovery of the watermarked video bitstream if the original watermark bitstream or watermark information is available. After extracting all the watermark bits from one frame, we can compare the extracted watermark with the original watermark. If there is any difference between these two sets of data, some error must have occurred. Then we can locate the corrupted block by tracing the watermark bit which has error and replacing that block by its neighboring block. However when neighboring blocks are quite different from each other, the error block cannot be recovered if we use the neighboring block to conceal the error. The error recovery scheme will fail when a large area of video has been corrupted. In this case a damage neighboring block may be used to recover the corrupted block, which may lead to much worse video quality.

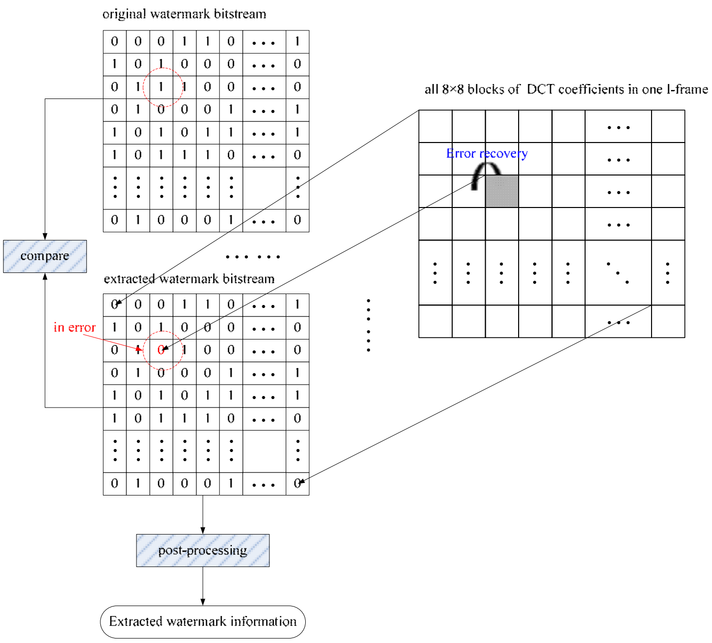

Figure 4 Schematic diagram of watermark extraction process

3 Error Correcting Codes (ECC) for video watermarking

In the proposed video watermarking system we have used the **E**rror **C**orrection **C**odes (***ECC***) [4][5] to improve the robustness of the system. In this section we investigate the effectiveness of using different error correcting codes in protecting codes of the watermark information.

Two Error Correcting codes [4][5] BCH(31, 8) and Turbo code(3, 1) were implemented and tested. The testing watermark data used is plain text "abcdefgh". Using 1 byte/8 bits to represent each character, the binary sequence for the watermark is

01100001, 01100010, 01100011, 01100100,
01100101, 01100110, 01101111, 01101000

The length of the binary sequence is 64 bits, containing eight 8-bit symbols.

3.1 Implementation of ECC Algorithms

We have implemented and compared two error correcting codes. The error correcting capacity of these two ECC algorithms will be analysised in this section. The implementation of these two ECC algorithm: BCH(31,8) and Turbo(3,1) are described as follows:

3.1.1 BCH(31, 8)

Each 8-bit symbol of the watermark data is individually BCH encoded. BCH(31, 8) is a shortened BCH(31, 11) code over GF(2^5), which can correct up to 11 bit errors anywhere while the below code length is 31 bits. There are actually 11 "data bits'" and 20 (31-11=20) "check bits" within the code. The first 8 bits of the 11 data bits contains the bit sequence of one watermark symbol, and the remaining 3 bits are fixed to '0'. Therefore, this BCH(31, 11) code can be regarded as a BCH(31, 8) code with t = 11, where t refers to the error correcting capacity.

By this method a 31-bit BCH code is generated for each symbol of the watermark. So there would be a total of 248 bits if the watermark consists of 8 symbols, as shown in Figure 5. Figure 6 lists the 31-bit BCH code for each watermark symbol. Since the BCH(31, 8) code we implemented here is systematic, each codeword can be divided into three parts: 0^{th} to 19^{th} are the check bits, 20^{th} to 27^{th} are the bit sequence of one watermark symbol (shown in bold font), and the last 3 bits are redundant 0s. At the decoder, each 31-bit sub BCH code is passed to the BCH decoder for decoding, resulting in eight original watermark symbols.

11001111110110000110011000010001110001110 1
00001000000110001000011111000011101100010 01
10001100010111011010100111100011001000001 0
10000010000100111001100101000100011001111 1
10110000011001100001001011100101010101001 10
01110000000101010110110010001101000000

Figure 5 BCH(31, 8) codeword sequence

Watermark symbol	**31-bit BCH codeword**
‘a’	11001111110110000110**01100001**000
‘b’	11100011101000010000**01100010**000
‘c’	11111000011101100010**01100011**000
‘d’	10111011010100111100**01100100**000
‘e’	10100000100001001110**01100101**000
‘f’	10001100111111011000**01100110**000
‘g’	10010111001010101010**01100111**000
‘h’	00001010101101100100**01101000**000

Figure 6 31-bit BCH codeword for each watermark symbol

3.1.2 Turbo(3,1)

The code rate of this turbo code is 1/3. The encoder structure of this turbo code and its component convolutional code are shown in Figure 7 and Figure 8, respectively.

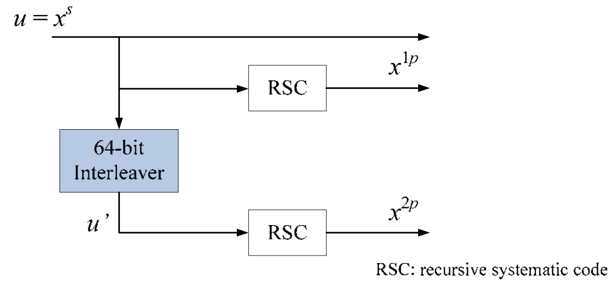

Figure 7 Encoder structure of the Turbo (3,1)

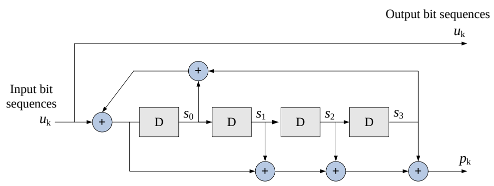

Figure 8 Encoder structure of the component convolutional code

Based on the above encoder structure, the trellis diagram of Turbo (3, 1) can be constructed as Figure 9 below.

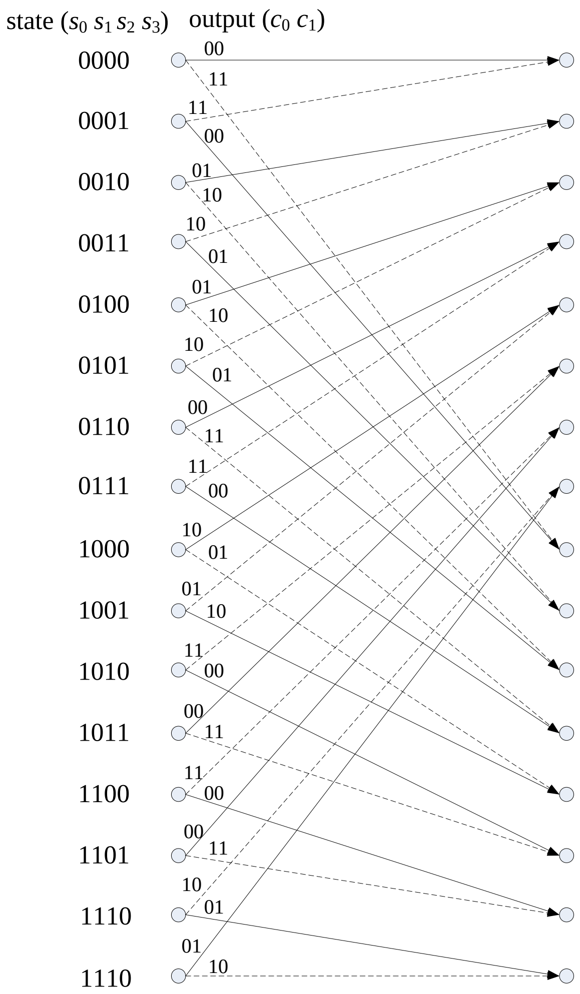

Figure 9 Trellis diagram of Turbo (3, 1)

Using the binary sequence of watermark data “abcdefgh” as input, the encoded watermark bitstream contains192 (64×3=192) bits. The Turbo (3,1) codeword sequence is shown below.

000111100011011000010111000100
01100000000000110010000100110001
00110111011010001011100110101 0
00000010001001100000000110010 11
00001001110010011111010100101 1
11100110001011111010011011010 0
1001001101110

4 Testing results

In this section seven cases of noise were simulated and tested. In order to compare the error correcting capacity of these two error correcting codes, the **W**atermark **C**orrecting **R**ate (**WCR**) of each error correcting code was measured under different **B**it **E**rror **R**ate (***BER***). BER is defined as

$$\text{BER} = (\boldsymbol{e} / \boldsymbol{n}) \times 100\%$$

where ***e*** is the number of bits in error in the extracted watermark bitstream before ECC decoding, and ***n*** is the total number of bits in the extracted watermark bitstream before ECC decoding. WCR is defined as:

$$\text{WCR} = (\boldsymbol{w} / \boldsymbol{m}) \times 100\%$$

where ***w*** is the number of correct bits in the extracted watermark data after ECC decoding, and ***m*** is the total number of bits in the extracted watermark data after ECC decoding.

The tests were conducted under the following seven cases of simulated noise:

Case no.	Description
Case 1	Set the value of ¼ part of Y components in each I-frames to 0
Case 2	Set the value of all Cr components in each I-frames to 0
Case 3	Set the value of all Cb and Cr components in each I-frames to 0
Case 4	Set the value of ¼ part of Y components and all Cb and Cr components in I-frames to 0
Case 5	Set the value of all Y components in I-frames to half of the original value
Case 6	Set the value of all Y, Cr components in I-frames to half of the original value
Case 7	Set the value of all Y, Cb and Cr components in I-frames to half of the original value

ECC type	BCH (31,8)			Turbo (3,1)		
	BER (%)	**WCR (%)**	**Extracted Watermark Data**	**BER (%)**	**WCR (%)**	**Extracted Watermark Data**
Case 1	6.6484	100	abcdefgh	5.5556	95.125	abcdefgc
Case 2	7.6613	100	abcdefgh	8.5859	95.3125	abcdefgc
Case 3	14.5161	95.3125	ab defgh	15.6566	95.3125	abcdefgc
Case 4	20.5645	76.5625	A `&Eh	21.2121	76.5625	$B#„äf§"
Case 5	33.0968	66.6250	æþsd5CêX	33.6019	49.8750	µ_□ úN_KW
Case 6	41.4597	59.2812	æþ3b!KúZ	42.0673	50.3437	±_ÿúN_KD
Case 7	49.6774	51.0938	Æ¿1f![úZ	50.3981	50.8437	ÿåÓl–Ã

Figure 10 BER and WCR comparison of two types ECC under the seven cases of simulated noise

According to the results in Figure 10 we find that BCH (31,8) can achieve a higher error correcting capacity than Turbo (3,1). When the BER is around 10% or lower, BCH(31, 8) can always recover all the watermark information bits in errors and obtain the right watermark information. When the BER is around 15%, the WCR after BCH decoding still remains above 95% and the extracted watermark information is almost correct.

The test results in Figure 10 show that the Turbo(3,1) can not obtain 100% correct information under seven cases simulated noise test . The WCR after turbo decoding remains at about 95% when the BER is around 15% or lower, and drops to around 75% when BER reaches 20%. The poor performance is partially due to the small size of the component interleaver. For turbo code, in order to achieve the good error correcting capacity, the size of the component interleaver in a turbo codec needs to be very large. However in this test, the size of the interleaver is set to be equal to the number of bits in the watermark data before turbo encoding, which is only 64 when using the plain test "abcdefgh" as watermark data.

5 Conclusion

In this paper a new digital video watermarking scheme based on error correcting code for MPEG2 video was proposed and implemented. The proposed watermark scheme was developed under the triple contradictory constraints of imperceptibility, robustness and capacity. As to the watermark embedding capacity, since one watermark bit can be embedded in the LSB of the DC coefficient of one 8×8 DCT coefficient block, the maximum number of bits that can be embedded in one I-frame is exactly the same as the number of 8×8 blocks. Given a video sequence with standard image size 352×288, more than two thousand watermark bits can be embedded in just one I-frame. Therefore the proposed watermarking scheme can support large watermark payload.

A new video watermarking system based on the proposed watermarking scheme was implemented and tested. Experiments were conducted on this system to show the effect of embedded watermark on the video quality. From the experimental results in section 2, it can be seen that there is no perceptible difference between the watermarked and unwatermarked video images. In other words, the watermark is perceptually invisible. The PSNR of watermarked and unwatermarked frames was also computed as the measure of the quality of the video image. The computation results show that the embedded watermark bits only slightly reduce the PSNR of decoded video images. The proposed watermarking scheme causes little distortion to the video quality in terms of PSNR.

In this paper we combined the watermarking scheme with different error correcting coding schemes to improve the performance in terms of watermark robustness. Two error correcting codes: BCH(31,8) and Turbo (3,1) were implemented with digital watermark technique. The effectiveness of these error correcting codes in protecting watermark were investigated. According to section 4, we have implemented seven cases of simulated noise to test this proposed video watermarking system. To compare the error correcting capacity of these error correcting codes, the watermark correcting rate (WCR) of each error correcting code was measured under different bit error rate (BER). From the experimental results in section 4, we can see that the BCH(31, 8) has the highest error correcting capacity. When the BER is around 15%, the WCR after BCH decoding still remains above 95% and can extract the watermark information with almost perfect accuracy.

References:

[1] T.Y.Chung, M.S.Hong, Y.N.Oh, O.H.Shin, S.H.Park, Digital watermarking for copyright Protection of MPEG2 compressed video, IEEE Trans. On Consumer Electronics, 1998, pp.895-901

[2] F. Hartung and B. Girod, “Digital Watermarking of MPEG-2 Coded Video in Bitstream Domain”, *Proc. IEEE Internatioinal Conference on Acoustics, Speech and Signal Processing, ICASSP-97*, Vol.4, No.4, pp.2621-2624

[3] F. Hartung and B. Girod, “Digital Watermarking of Raw and Compressed Video”, *Digital Compression Technologies and Systems for Video Communications, SPIE Proceedings Series*, Vol.2952, Oct. 26-29, 1996, pp.205-213

[4] Robert H. Morelos-Zaragoza (2002), *The Art of Error Correcting Coding*, John Wiley & Sons Ltd, Baffins Lane, Chichester, West Sussex, P019 lUD, England

[5] C.Berrou, A. Glavieux and P.Thitimajshima, “Near Shannon limit error – correcting coding and decoding: Turbo codes”, *Proc. 1993 Int. Conf. Comm.*, pp.1064-1070