Book Chapter – Chapter 18 : Spread Spectrum Video Data Hiding, Interleaving and Synchronization, in Intelligent Watermarking Techniques, Editors : Jeng-Shyang Pan, Hsiang-Cheh Huang, and Lakhmi Jain, World Scientific Publishing Co., Spring 2004, pp. 515-558, ISBN 9-81238-757-9.

Chapter 18
Spread Spectrum Video Data Hiding, Interleaving and Synchronization

Yun Q. Shi, Jiwu Huang and Heung-Kyu Lee

Abstract In this chapter, some typical spread spectrum based video data hiding algorithms are presented, and two related research issues are addressed. One is to correct both random and bursts of errors using 3-D interleaving together with random error correction codes. Another is frame synchronization in hidden data detection.

1. Introduction

It is noted that by far copyright protection, in particular, the digital versatile disk (DVD) video copyright protection has been the driving force for the tremendous attention paid and efforts made to digital watermarking since 1990s. This does not come with surprise because digital video of high visual quality can be easily copied without distortion and quickly distributed to any place in the world through network and/or DVD distribution. In addition to DVD, video broadcasting and video on demanding (VOD) face the similar copyright protection issue. Consequently, owners of digital videos are hesitated to release their digital videos. It has been realized that encryption alone cannot resolve copyright protection and video watermarking becomes an effective measure to protect copyright after digital video has been decrypted.

There are some special requirements for video watermarking, which are listed below. 1) Imperceptibility, or difficult to notice the

distortion introduced by watermarking. Obviously, this is necessary for watermarked video to have commercial values. 2) Robustness, or difficult to remove. It is expected that even though the watermarking algorithm may be known to public unauthorized removal of watermark is impossible. In addition, common video manipulation such as video compression, noise addition, format conversions and geometric shift should not lead to failure of watermark detection. 3) Due to the fact that a video is a sequence of video frames presented at a certain frame rate, watermark detection should be fast and inexpensive. 4) The payload of video watermarking should satisfy certain requirement (say, equal to or larger than eight bits per detection interval for DVD application). 5) The probability of a false positive alarm (a watermark is detected while there was no watermark actually embedded) should be extremely small (say, less that 10^{-12} per detection).

Though video protection is the driving force of recent emerging research of digital multimedia watermarking, it appears that compared with image watermarking, video watermarking has been reported much less in the literature. In this chapter, we focus on spread spectrum video data hiding and some research issues that need to be resolved. We first present two algorithms that apply spread spectrum technique to video watermarking in uncompressed and compressed domains, respectively (Hartung and Girod 1998). Next, an algorithm in uncompressed spatial domain developed in what is known as Millennium system is introduced (Maes *et al*. 2000). It is not our goal to cover in this chapter all video watermarking algorithms, e.g., another algorithm in the compressed domain by Langelaar *et al*. (1998). It is noted that the scope of video data hiding is wider than that of video watermarking. Namely, the data hidden in a video sequence may be used for purposes other than copyright protection. Therefore, in the second part of this chapter, we address two issues related to video data hiding. One is correction of both random and bursts of errors occurred in stego-video. There, a newly developed multi-dimensional (M-D) inter-

leaving technique and investigations of applying this interleaving technique to enhance robustness of hidden data against bursts of errors (Shi and Zhang 2002, Shi *et al.* 2003) are presented. Another is resynchronization issue related to spread spectrum techniques. There, frame synchronization newly developed for video data hiding (Liu *et al*. 2002) is reported.

2. Spread Spectrum Video Watermarking in Uncompressed and Compressed Domains

Spread spectrum (SS) techniques, emerged since 1950s, have found wide applications in military communications systems due to its secrecy to, and robustness against interception from, an unauthorized person. It was first applied to image watermarking by Cox et al. (1997) and soon became the most popular one among numerous watermarking methods. The SS was applied to video watermarking by Hartung and Girod (1998).

2.1 Spread-Spectrum Technology

Because of its importance in image and video watermarking, in this section, we introduce the SS technology. A block diagram of SS digital communication system is shown in Figure 1. It is noted that an identical pseudorandom or pseudonoise (PN) binary sequence is generated and available at both transmitter and receiver. At the transmitter, it is used to spread the transmitted signal in spectrum, while it is used to despread the received signal in spectrum at the receiver. The PN sequence, which is independent of the information sequence and used to spread and despread information se-

quence at the transmitter and receiver, respectively, is the key feature of the SS technology.

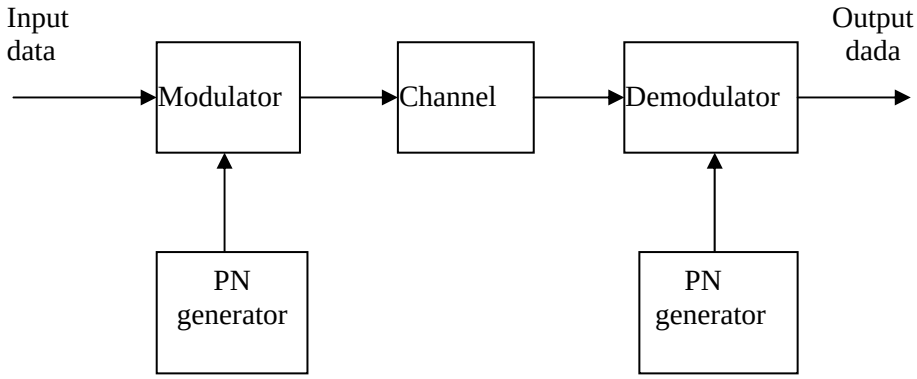


Figure 1. Block diagram of spread spectrum digital communication system.

Through popularly used direct-sequence SS technology, we explain how SS methods work. Considering an information sequence *is(t),*

$$is(t) = \sum_{n=-\infty}^{\infty} a_n p_{T_b}(t - nT_b) \qquad (1)$$

where $a_n = \pm 1$, $p_{T_b}(t)$ is a rectangular pulse of duration T_b, and T_b is the reciprocal of data rate of the baseband signal *is(t),* R_b. The binary PN code sequence *pn(t)* can be expressed as

$$pn(t) = \sum_{n=-\infty}^{\infty} b_n p_{Tc}(t - nT_c) \qquad (2)$$

where $b_n = \pm 1$, $p_{T_c}(t)$ is a rectangular pulse of duration T_c, and T_c is the reciprocal of data rate of the code sequence *pn(t),* R_c. The

rectangular pulse $p_{T_c}(t)$ is often referred to as a *chip*, and the rate of the code sequence *pn(t)*, R_c, is often called *chip rate*. The data rates, R_b and R_c, satisfy the following relation.

$$R_c > R_b \tag{3}$$

In spreading, depicted in Figure 1, we consider the product of *is(t)* and *pn(t)*. Note that $is(t) \cdot pn(t) = \pm 1$ for any given time moment *t*. In Figure 2, an example of *is(t)*, *pn(t)* and the product of the two are illustrated. Assume that the double-sideband suppressed carrier (DSB-SC) modulation is used and a sinusoid $A_c \cos(2\pi f_c t)$ is used as the carrier. The DSB-SC modulated signal is then

$$\text{mod}(t) = A_c \cdot is(t) \cdot pn(t) \cdot \cos(2\pi f_c t) \tag{4}$$

Because $is(t) \cdot pn(t) = \pm 1, \forall t$ the modulated signal is in fact a binary PSK signal. According to Figure 1, at the receiver side, we have the following despread signal.

$$despread(t) = A_c \cdot is(t) \cdot pn(t)^2 \cdot \cos(2\pi f_c t) \tag{5}$$

The conventional demodulation technique can now be applied to demodulate this despread signal and extract the information sequence.

A good summary of SS technology can be found, say, in (Proakis and Salehi 1994). The PN code is used at the transmitter to spread information sequence into a wide bandwidth for transmission and used at the receiver to despread the signal back to a narrow bandwidth. Since the channel interference normally occupies a wide bandwidth, the SS technology can reduce the interference power by a factor of R_c / R_b, which is referred to as SS processing gain τ_G. That is,

$$\tau_G = \frac{R_c}{R_b} \tag{6}$$

Often, the processing gain τ_G is an integer. In addition to channel interference reduction, secrecy is another major advantage brought by the SS technology. That is, since PN code is only known to the authorized receiver and unauthorized receivers cannot demodulate the transmitted signal. The price paid for achieving the processing gain and secrecy using SS technology is wide channel bandwidth and added computational complexity.

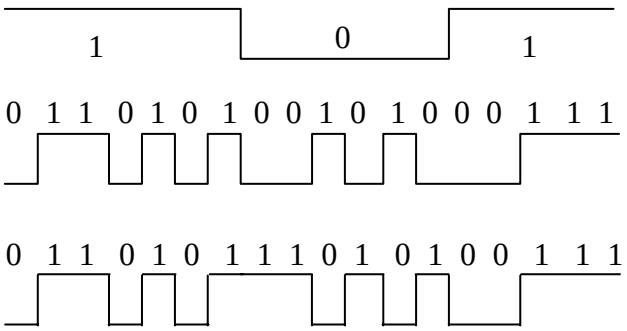


Figure 2. Information signal (the first row), PN sequence (the middle row) and their product (the bottom row).

2.2 SS-based Video Watermarking in Uncompressed Domain

In this case, watermark signal (a binary bit sequence) is the base-band information signal discussed in Section 2.1. It is denoted by

$$\{a_m, \quad a_m \in \{1,-1\}, \quad m \in I\} \tag{7}$$

The spreading of this information signal with a binary PN sequence is essentially the same as discussed in Section 2.1. The procedure, however, is seemingly different. That is, the watermark signal is first expanded into another binary sequence $\{b_i\}$ as follows.

$$\{b_i = a_m, \quad m \cdot \tau \le i < (m+1) \cdot \tau\} \tag{8}$$

where τ is an integer, equivalent to the processing gain τ_G or chip rate R_c when $R_b = 1$. Hence, we see that we in fact embed one information bit into τ pixels in a video frame. A binary PN sequence, $\{c_i, c_i = \{-1,1\}, i \in I\}$, is then used to multiply the expanded $\{b_i\}$ sequence. The SS watermark signal is now:

$$\{w_i = b_i \cdot c_i, \quad i \in I\} \tag{9}$$

By SS embedding in uncompressed domain, it is meant that the spread spectrum watermark signal multiplied by a scaling factor, denoted by α, which is possibly adapted to some local properties of the original video frame, is added into the pixel gray-level value of pixels of the original video frame, x_i, where *i* represents a sequential index of the pixel. Putting together, we have the following embedding formula.

$$x_i' = x_i + \alpha_i \cdot w_i \tag{10}$$

where the scaling factor α_i may be related to some local property of video frame.

The whole embedding process is exactly like what introduced in Section 2.1. In watermark signal detection, the same binary PN se-

quence is used to despread the received signal in order to retrieve the watermark signal like what discussed in Section 2.1. One implementation can be:

$$a_m = \sum_{i=m\cdot\tau}^{(m+1)\cdot m-1} x_i' \cdot c_i = \sum_{i=m\cdot\tau}^{(m+1)\cdot m-1} (x_i \cdot c_i + \alpha_i \cdot b_i \cdot c_i^{\,2})$$

$$= \sum_{i=m\cdot\tau}^{(m+1)\cdot m-1} (x_i \cdot c_i + \alpha_i \cdot b_i) \qquad (11)$$

where x_i' denotes the pixel gray value at the watermark detection. It can be shown that the sign of the above correlation sum determines the information bit a_m, i.e., positive sign represents a binary 1 and negative a binary 0. In (Hartung and Girod 1998), high-pass filtering of watermarked video sequence is applied. It is observed that this technique does not need to know the original video frames in watermark signal retrieval.

2.3 Synchronization in SS-based Video Watermarking

It is noted that pixels in each frame of a video is cascaded in a row-by-row or column-by-column manner. Then the whole video sequence is cascaded in a frame-by-frame manner. In this way, all pixels in a video sequence are cascaded into a 1-D sequence. Information bits are embedded into each pixel in the resultant 1-D sequence, which is depicted in Figure 3. In watermark signal detection, we need to do correlation for each group of pixels that are corresponding to one watermark bit. That is, watermark signal detection heavily replies on accurate synchronization. This is true in general for all SS technologies. The synchronization of all pixels in the video sequence, which are 3-D in nature, becomes a critical issue, and is not easy to handle.

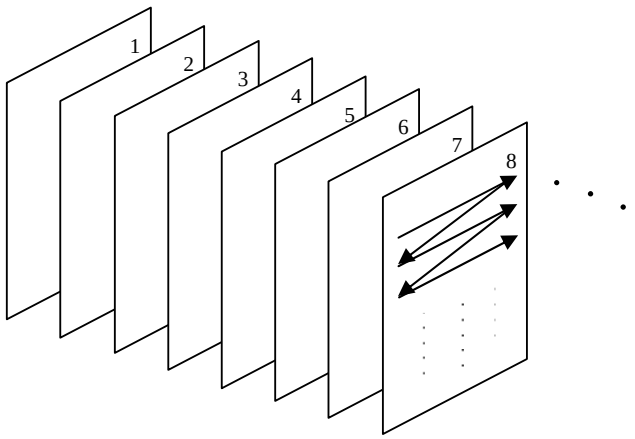


Figure 3. A 1-D pixel sequence of huge dimensionality formed from a 3-D video sequence.

2.3.1 Brute-force Search

Hartung and Girod proposed in their paper a technique, called sliding correlator. Namely, the window is sliding to find out the best position where the correlation achieves the maximum correlation sum. In fact, this is an exhaustive search for all possible shifts. Considering the huge size of the 1-D pixel sequence formed from a 3-D video, this sliding correlator technique requests high computation.

2.3.2 Embedding Side-information for Synchronization

In a newly developed technique by Liu *et al*. (2001), frame synchronization was achieved by embedding frame numbers as side-information into each video frame. It will be presented in Section 5 as a research subject.

2.4 SS-based Video Watermarking in Compressed Domain

It is known that almost all current international video compression standards including MPEG-1, 2, H.261 and H.263, the baseline mode of MPEG-4 are based on motion compensated prediction coding and block based discrete cosine transform (DCT) coding (Shi and Sun 1999). Video watermarking in compressed domain therefore makes sense. Hartung and Girod (1998) applied their technique, discussed in Section 2.2 to video watermarking in compressed domain as well. That is, they embed watermark signal in DCT coefficients using the spread spectrum method. Consequently, the SS based watermarking methods applied to uncompressed and compressed domains are compatible. A block diagram of forward motion estimation compensation is shown in Figure 4. Note that the DCT coefficients after quantization is zigzag scanned, run-length coded, and finally by Haffman coded.

As far as MPEG coded video is concerned the SS based video watermarking algorithm can be described as follows.

a. Within MPEG coded data, motion vectors and header information remain unchanged. Only the DCT coefficients (either in intraframe coding, or interframe coding) are used to embed watermark signal.

b. Watermark signal for each video frame is generated in the same way as used in watermarking in uncompressed domain, described in Section 2.2.

c. Thus generated watermark signal is arranged in a manner compatible to the MPEG data structure, say the hierarchical

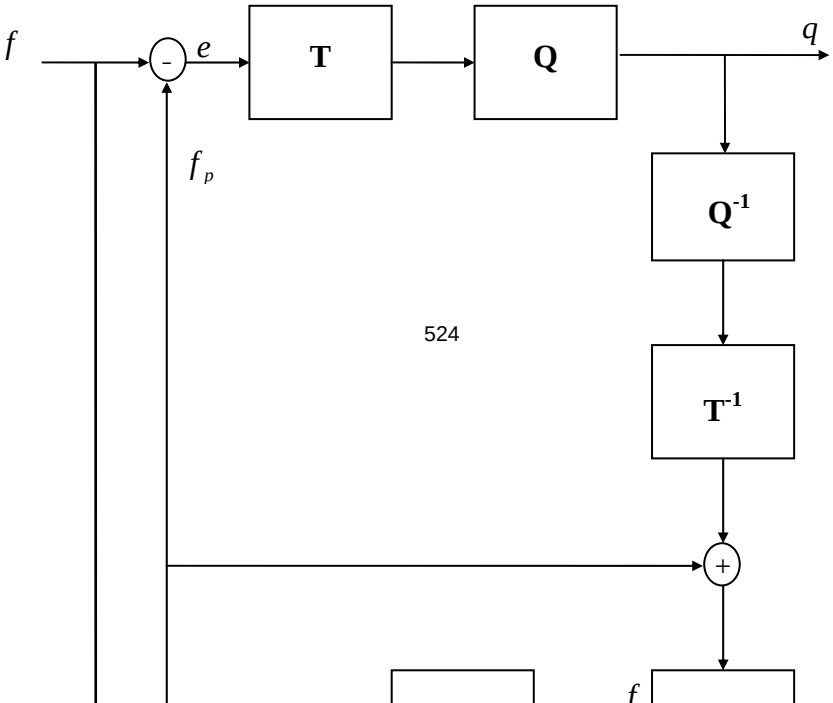


Figure 4. Forward motion estimation and compensation, T: transformer, Q: quantizer, FB: frame buffer, MCP: motion compensated predictor, ME: motion estimator, e: prediction error, f: input video frame, $f_{p:}$ predicted video frame, f_{r}: reconstructed video frame, q: quantized transform coefficients, v: motion vector.

structure (frame, macroblock and 8x8 block, refer to (Shi and Sun 1999)).

d. The watermark signal corresponding to one 8x8 block is DCT transformed.

e. The DCT coefficients generated from video sequence and the DCT transformed watermark signal are then added according to the way described in Section 2.2.

Indeed, we can see the SS based video watermarking algorithm applied to the compressed domain is compatible to the counterpart applied to uncompressed domain. Of course, however, there are some differences between techniques embedding in uncompressd and that in compressed domains.

2.4.1 Bit-rate Constraint of MPEG Coded Data

One difference is that in compressed domain we need to take special measure to prevent video bit-rate from increase. That is, SS based watermarking algorithm when applied in compressed domain may result in data increase, which is not desired and should be avoided. Let us consider a typical case, i.e., after motion compensated predictive coding, the residual errors are DCT transformed, the DCT coefficients are then quantized with some carefully constructed and selected quantization table. As a result, most of the quantized DCT coefficients may appear to be zero. Zigzag scan is carried out. Run-length coding is applied. Entropy coding, say, Hoffman coding is finally utilized to code the pair of run-length of zeros and the following non-zero DCT coefficient. Thus, one Huffman codeword represents a non-zero DCT coefficient, both its magnitude and its position. In data embedding, each Huffman codeword is decoded and then converted to a value via inverse quantization. On the other hand, the watermark signal is arranged into a 2-D array of the same dimensionality as the image, and split into 8 by 8 blocks. DCT is applied to each block. The corresponding DCT coefficient thus generated is added to the above-mentioned value. The sum is then quantized and Huffman encoded, resulting in a new codeword. This process may cause an increase of bit-rate. In order to prevent the bit-rate of marked data from increasing, the embedding in those coefficients, which will increase

bit-rate should not be pursued. That is, we do not embed watermark signal whenever it may cause bit-rate increase. Because DC coefficients of each block is encoded with a fixed-length codeword, all DC coefficients can be used to embed data. According to (Hartung and Girod 1998), typically about 10-20% of the DCT coefficients are altered for data embedding.

2.4.2 Drift Compensation

It is noted that motion compensated hybrid coding used in video coding standards is recursive in nature. Namely, motion estimation is conducted among neighboring frames. Therefore the "drift" caused by data embedding will be propagated and cause severe problem for video quality. This is a problem of directly modifying DCT-coefficients of an encoded video stream in compressed domain. In order to combat this type of drift, Hartung and Girod (1988) proposed to add a signal to compensate the drift due to data embedding. This signal is simply the difference of DCT coefficients before and after the data embedding.

3. DVD Video Copy Protection

As a result of tremendous efforts, there are two contending proposals on DVD video copy protection in the late 90s. One is known as "Galaxy" proposal jointly supported by Hitachi, IBM, NEC, Pioneer and Sony, another "Millennium" proposal, jointly supported by Philips, Macrovision, and Digimac. These two contenders in the standardization are currently pending. It is noted that the Galaxy and Millennium groups were merged into one group, called VWM (Video Watermarking) group, in April 2001. In this section, we describe the Millennium approach (Maes *et al*. 2000).

3.1 Fundamental Principles

As said that encryption alone is not sufficient for video copy protection. Namely, after decryption, which is inevitable in reality for video to be displayed, video data are vulnerable for illegal copying. Therefore, digital video watermarking will play a role in video copy protection.

3.1.1 Targeted Requirements

In this specific context, however, targeted requirements have to be determined carefully. It should be realistic, i.e., a trade-off between performance and cost. As discussed in Section 1, the Millennium proposal aimed at *fast detection* of watermark, i.e., within the declared DVD detection interval of 10 second. Therefore, highly complicated algorithms are not feasible. Instead, they have to be simple and yet effective. As far as *payload* is concerned, it targeted at equal or larger than eight bits per detection interval. For robustness, it aimed at common image processing such as compression, noise addition, format conversion, shift, etc. By *shift*, it is meant that the position of a video frame is varied by a small amount. This may be due to normal image handling or malicious attack. As to the latter, it is noted that it is easy and inexpensive for a hacker to shift video frame spatially even on a frame-by-frame basis. As a result, this shift becomes a problem that needs to be addressed. The probability of a *false alarm* was set as low as less than 10^{-12} per detection. It is worth noting that these requirements are carefully determined. They are compromised results at the current technological levels.

3.1.2 Embedding and Detection in Spatial Domain

The design of watermarking scheme followed the above-mentioned requirements. Because of the fast implementation requirement, the Millennium proposal did not choose to use embedding techniques

in transform domain. That is, it did not use the global DCT, the wavelet transform (WT), and the Fourier transform (FT). It did not use block-by-block transform techniques either (which can be implemented faster than the global transform counterparts) because possible spatial shift, discussed above, may fail detection easily. Consequently, the proposed technique was implemented exclusively in the spatial domain. Specifically, a watermark signal, $W = \{w_i\}$, is added to a video frame, $X = \{x_i\}$, thus generating a marked video frame, $X' = \{x_i'\}$, where w_i, x_i, and x_i' may be as defined in Section 2.2. Note that in (Maes *et al*. 2000) W is drawn from a Gaussian distribution with zero mean and unit standard deviation. The embedding is written below.

$$x_i' = x_i + \beta_i \cdot w_i \tag{12}$$

where the scaling factor β_i may be a product of two factors, one of which takes care of global scaling while another local scaling. Watermark detection is performed by spatial correlation. That is,

$$corr = \frac{1}{L}\sum_i x_i' \cdot w_i \tag{13}$$

where L is the total number of pixels involved in the correlation. The spatial correlation *corr* larger than a threshold indicates the existence of the watermark signal and otherwise the absence of the watermark signal.

3.1.3 Payload

In the Millennium proposal, a video sequence is treated a sequence of still video frames. Pixels of each frame are used to embed watermark signal, and the watermark signal is embedded repeatedly into each frame. This strategy is equivalent to first accumulating pixel gray level values and then multiplying the accumulated value

with the watermark signal, thus saving computation dramatically. It also makes algorithm more robust in watermark signal detection. However, this strategy leads to a low payload since one video sequence only contains one-bit information. Apparently, one-bit payload is not enough to meet the payload requirement. To increase payload, two ways are possible. One way is to embed more than one watermark sequence in each video frame. Another way is to embed a slowly time-varying watermark sequence. These two ways are not exclusive to one another. That is, combination of both ways can also be used to increase payload. It is reported in (Maes *et al*. 2000) that three or four may be the maximum number of watermark sequences that can be embedded into a video frame under the constraint of computational complexity and watermark imperceptibility.

3.1.4 Shift Invariance

As mentioned, the embedded watermark signal must be robust against spatial shit, because the spatial shift is likely to happen in practice and it will destroy the geographical synchronization that is a necessity for watermark detection.

The simplest and straightforward way to keep video watermarking shift invariant is brute-force search. That is, for each possible spatial shift, perform the spatial correlation; then find out the maximum correlation to determine the shift the video frame has experienced. This approach is, however, computational prohibitive due to the real-time watermarking requirement.

To dramatically reduce computational complexity, the Millennium approach introduced translational symmetry in the watermark signal W itself, expressed below.

$$w_{i+k} = w_i \tag{14}$$

where k is the amount of spatial shift that only assumes multiples of a pre-selected integer N with N chosen as 128. This is to say that the watermark signal is two-dimensionally periodic. Or in other words, the watermark signal is a tile of an $N \times N$ two-dimensional array of random numbers. It is clear that such a watermark signal, which possesses the translational symmetry, makes the search for maximum correlation much simpler in terms of computational complexity.

3.1.5 Synchronization

The shift invariance just discussed above takes care of geometric synchronization. In terms of temporal synchronization, the proposed Millennium approach embeds the same watermark signal to a group of video frames. In addition, watermark signal is formed from Gaussian random sequence. These measures achieve a more robust synchronization.

3.2 Performance

In this subsection, we present performance of the Millennium system reported in (Maes *et al*. 2000).

3.2.1 Real-time Implementation

Real-time watermark embedding has been implemented on TriMedia processor board and FPGA-based board. Real-time watermark detection has been performed on three different platforms, i.e., Silicon Graphics workstation, TriMedia and FPGA. These successes have demonstrated the feasibility of real-time video watermarking, a requirement set by Millennium system.

3.2.2 Robustness

Many experiments conducted have shown that video watermarking implemented by the Millennium system is robust against MPEG-2 compression down to 2.5 Mbps. MJPEG compression, D/A and A/D conversion, PAL conversion, noise addition, quantization, subtitling and logo insertion, cropping, frame erasure, speedups and transmission errors. It is noted that, however, there is no concrete robustness test performance available in public.

3.3 Concluding Remarks

In conclusion, we would like to point out that the framework of the Millennium system, reported in (Maes *et al*. 2000), has provided us with a picture of the state-of-the-art in digital watermarking for DVD video copy protection. Based on the targeted requirements, design of watermarking system was carefully carried out with various signal processing techniques. These requirements, in fact coming from compromised decision, aiming at fulfillment of most critical and practical and yet feasible requirements with the current technologies. Consequently, the system works well with respect to these requirements. For the issues for further investigation, including copy control, interested readers are referred to (Maes *et al*. 2000).

4. Interleaving to Combat Random and Bursts of Errors in Video Data Hiding

So far what we have discussed in this chapter are concerned with video watermarking, in particular, for DVD video copy protection. However, video data hiding has a wider scope. That is, video data hiding may be used for other applications such as covert communications, annotation, control, authentication, data security and other purposes. Therefore, it is necessary to discuss some issues, which

may or may not relate to video copy protection. In this and the next sections, two of these issues will be addressed. In this section, we present our initial investigation on interleaving to combat bursts (clusters) of errors, which occurred to video with hidden data. In the next session, we present our initial work on synchronization, which is related to video data hiding for both copy protection and for applications other than copy protection.

It is well-known that robustness is one of the basic requirements for imperceptible data hiding in some applications. Error correction codes (ECC) have been adopted to improve the robustness of watermark signal in (e.g., Huang *et al*. 1998, Huang and Shi 2002). As shown next, however, ECC is only suitable to correct random errors, and will not be efficient for correction of bursts of errors. When cropping or random rows/columns removal, also known as jitter attack, takes place in a stego-image, bursts of errors do occur in watermarked images. Frame loss and 3-D error clusters are typical bursts of errors that can occur to watermarked video sequences. Transmission error may be another source of bursts of errors. When bursts of errors occur, how to extract and detect the hidden data correctly becomes a challenge. While using ECC alone to correct these bursts of errors is not efficient, surprisingly, combating bursts of errors using interleaving, a common tool used in communication systems, has been neither recognized nor addressed so far in the data hiding community.

In this section, we first introduce philosophy of interleaving, followed by the t-interleaved array technology for multi-dimensional (M-D) interleaving. We further point out why this technology does not fit the image and video data hiding well. Then, the novel successive packing (SP) approach to 2-D/3-D interleaving is discussed. Finally, we present our initial investigation on applying 2-D/3-D successive packing interleaving techniques to combat bursts of errors occurred in marked still image/video sequence. The experimental results demonstrate that the robustness of hidden data inside

still image/video sequences against bursts of errors is significantly improved by using 2-D/3-D SP interleaving followed by ECC.

4.1 Introduction

In data manipulation and transmission, errors may be caused by a variety of factors including noise corruption, limited channel bandwidth, and interference between channels and sources. It is well known that many ECCs have been developed to correct errors in order to ensure data fidelity. In doing so, redundancy has been added to ECC, resulting in what is known as random error correction codes. There are basically two different types of ECCs. One is known as block codes, another convolutional codes. The frequently used block codes are often denoted by a pair of two integers, i.e., *(n, k)*, and one block code is completely defined by 2^k binary sequences, each is an *n*-tuple of bits, known as codeword. Note that for simplicity only binary code symbols are considered in this section. Specifically, consider the commonly used BCH codes, which are one kind of block codes, named after the three inventors Bose, Chadhuri, and Hocquenghem (Bose and Chadhuri 1960, Hocquenghem 1959). The notation BCH (31,6) indicates that there are at most 2^6 distinct messages, each represented by six bits and encoded by a codeword consisting of 31 bits in this BCH code. Instead of six bits, 31 bits are used to represent an input symbol, implying an added redundancy. According to channel coding theory, the minimum Hamming distance between any two different codewords in the BCH (31,6) code is 15, and the error correction capability of the code is seven. In other words, a 31-bit codeword in the BCH (31,6) code can be correctly decoded as long as there are no more than seven error bits regardless of the bit error positions within the codeword. That is why random ECC refers to its ability to correct *random* bit errors within a codeword.

4.1.1 ECC Alone Cannot Correct Bursts of Errors Efficiently

Bursts of errors are defined as a group of *consecutive* error bits in the one-dimensional (1-D) case or *connected* error bits in M-D cases. In this sense, we can see that the channel has memory. One example can be several consecutive transmitted error bits in a mobile communication system caused by a multipath fading channel. Another example can be an area formed by many connected error bits in a 2-D barcode. In (Wicker 1995), a bursty channel is defined as a channel over which errors tend to occur in bunches, or "bursts," as opposed to the random patterns associated with a Bernoulli-distributed process. Therefore, a random error correction code, when applied, may not be powerful enough to correct the bursts of errors and at the same time it may be a waste for other occasions (1-D case) when there are no bursts of errors, or regions (M-D case) where there are no bursts of errors. For instance, consider a case in which there is one burst of errors consisting of 60 consecutive bits. Obviously, the BCH (31,6) code is not be able to correct this burst of errors. One may think of using a more powerful BCH code to combat this burst of errors. For example, BCH (255, 9) seems to be a suitable candidate since it can correct 63 errors in a 255-bit codeword. Indeed, this error burst consisting of 60 consecutive error bits can be corrected by this powerful BCH code. However, for vast majority of time, the error correction capability at the expense of high redundancy (each codeword now consists of 255 bits) has been wasted. This example demonstrates that using random ECC to combat bursts of errors is not efficient.

4.1.2 The *t*-Interleaved Array Approach to M-D Interleaving

Although some codes, including Fire codes (e.g., Imai 1973), suitable for correcting bursts of errors have been developed, they are not efficient for random error correction. In most practical systems, unfortunately, both types of errors may exist. By far, interleaving before applying random ECCs is a most frequently used and efficient way to combat bursts of errors and random errors. In this sub-

section, first M-D bursts of errors and optimality of interleaving are defined. Afterwards, the *t*-interleaved array M-D interleaving techniques (Blaum *et al*. 1998) are presented.

2-D and M-D Bursts of Errors The scenarios, where M-D error bursts may occur, include magnetic and optical (say, holographic) data storage, charge-coupled devices (CCDs), 2-D barcodes, and information hiding in digital images and video sequences. In particular, it is worth mentioning that in the holographic recording a laser beam illuminates a programmable spatial light modulator thereby generating an object beam, which represents a 2-D page of data. An entire page of data can be retrieved all at once, thus achieving a very high data rate. Therefore, the reliability issue of M-D information has arisen as an important task, having both theoretical and practical significance.

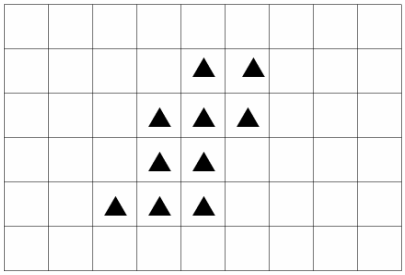


Figure 5. A 2-D burst of errors of size 10.

Instead of defining a burst of errors as a rectangular area or a circular area, Blaum et al. defined a 2-D burst of errors as an arbitrarily-shaped, connected area (Blaum *et al.* 1998). Consider Figure 5, where all the code symbols (assigned to the elements of the 2-D array) marked with triangles form a 2-D error burst. Note that all of these symbols are connected to each other and the connectivity here is constrained to the horizontal and vertical directions, referred to as 4-connection. This definition can be generalized to the M-D case. The size of a burst is defined as the total number of code

symbols contained in the burst. Hence, the size of the error burst in Figure 5 is 10.

8	1	2	3	4	5	6	7
3	4	5	6	7	8	1	2
6	7	8	1	2	3	4	5
1	2	3	4	5	6	7	8
4	5	6	7	8	1	2	3
7	8	1	2	3	4	5	6
2	3	4	5	6	7	8	1
5	6	7	8	1	2	3	4

(a) Interleaving degree is 8.

1	2	3	4	5	6	7	8
5	6	7	8	9	10	1	2
9	10	1	2	3	4	5	6
2	3	4	5	6	7	8	9
5	6	7	10	1	2	3	4
8	1	2	9	4	5	10	7
3	10	5	6	7	8	1	2
6	7	8	1	2	3	4	9

(b) Interleaving degree is 10.

Figure 6. Two 4-interleaved arrays with different interleaving degrees.

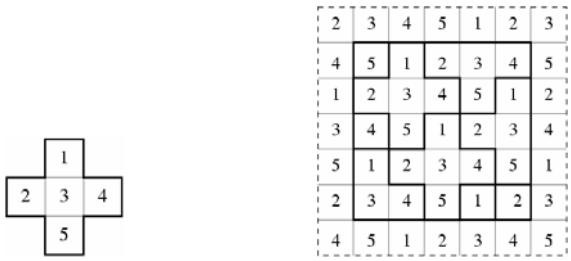
t-Interleaved Array, Interleaving Degree and Optimality Blaum et al. (1998) introduced the concept of the *t*-interleaved array. Consider the two 4-interleaved arrays shown in Figure 6. By a 4-interleaved array, it is meant that no matter how we choose four 4-connected elements in the array we always have these four elements marked with distinct numbers. Assuming that in Figure 6 all elements in the 2-D array denoted by the same number form one codeword, we can then conclude that whenever a burst of errors of size four takes place within the 4-interleaved 2-D array each codeword will encounter at most one error. If further assuming that the code has a one-random-error-correction capability, we can see that the error burst can be corrected. Through this discussion, it is observed that a *t*-interleaved array together with a random error correction code having one-random-error-correction capability can combat an error burst of size *t*. Without loss of generality, only one error burst is discussed in this subsection.

A close look at Figure 6 (a) and (b) reveals that there are a total of eight and 10 distinct numbers, i.e., eight and 10 distinct codewords

in (a) and (b), respectively. The total number of distinct codewords is referred to as the *interleaving degree*. Optimality of interleaving is achieved if the interleaving degree reaches its lower bound. The lower bound for correcting arbitrarily-shaped error bursts has been proved to be: $t^2/2$ if t is even and $(t^2+1)/2$ if t is odd (Blaum *et al.* 1998). Thus, for a 4-interleaved array, the lower bound of the interleaving degree is equal to eight. Therefore the code depicted in Figure 6 (a) is optimal, while that in Figure 6 (b) is not. It has been shown that optimality can be guaranteed for the 1-D and 2-D cases; however, this is not always true for the 3-D case (Balum *et al.* 1998, Golomb and Welch 1970).

Basic Ideas and Algorithms The key idea of the Blaum et al. approach is based on the Lee-spheres and the close tiling. Linking the Lee spheres with the odd burst sizes and creating some spheres for the even burst sizes, Blaum et al. used these spheres as fundamental building blocks to construct interleaved arrays via closely titling. Lee sphere of radius 1 is shown in Figure 7 (a), where one can see that the maximum *4-distance* (only considered in either horizontal or vertical directions) from the central element to any other elements in the sphere is equal to one. This Lee sphere can be used as a fundamental building block to construct (closely tile) a 3-interleaved array as shown in Figure 7 (b). Note that, in Figure 7 (b), there is a square array of 5×5 enclosed by solid lines and formed by several whole and partial Lee spheres of radius 1 (also bounded by solid lines). It is further noted that the 3-interleaved arrays shown in Figure 7 (b) can in turn be used as the building block to closely tile 3-interleaved array of larger size. This can be verified by the observing that the five central elements in the first row in Figure 7 (b): 3,4,5,1,2 with dashed lines repeat themselves in the last row within the solid line square. The same is true for the five in the bottom row, the left-most column, and the right-most column of Figure 7 (b), respectively. By closely titling, it is meant that the translated building blocks are used to construct a larger

block, which is the union of the translated building blocks and there is no overlapping among the translated building blocks in the process. (For more information on these two concepts, i.e., the Lee sphere and close tiling, interested readers may refer to (Golomb and Welch 1970).) Blaum et al. have shown that if one labels each element in the fundamental building block with a distinct number and uses the building block to closely (meaning no uncovered elements) tile (meaning no overlapping between blocks) a large enough 2-D area, then one can produce a *t*-interleaved array. In this interleaved array, each element in any arbitrarily-shaped, connected subset consisting of *t* elements is labeled with a distinct number. All numbers of the same kind form a codeword. Consequently, the error burst of size *t* can be corrected by one-random-error-correction codes.



(a)Lee sphere of radius 1 (b) 3-interleaved array

Figure 7. 3-interleaved array and its fundamental building block, Lee sphere of radius 1.

Comments and Discussions Though it can effectively spread arbitrarily-shaped 2-D burst errors of size *t*, the above characterization of the technique (Blaum *et al*. 1998) does reveal some of its limitations on the other hand. Firstly, the technique is based on the size of a burst of errors, *t*. For combating bursts of errors of size *t* equal to a specific t_0, one needs to implement the algorithm with a set of parameters to construct an interleaving code. When the size *t* increases, i.e., $t>t_0$, one needs to implement the algorithm with a new

set of parameters to construct another interleaving code. That is, the interleaved array constructed for a specific t_0 may not be able to correct a burst of errors of size t as $t>t_0$. Since in reality, e.g., in the application of 2-D barcodes, *the size of error bursts may not be known exactly a priori*, the implementation of the technique may become cumbersome and ineffective.

Secondly, when the actual size of a burst, t, is less than t_0, with which the interleaving algorithm is applied, the technique is no longer optimal. This can be justified as follows. As mentioned, the optimality means that the interleaving degree reaches its lower bound. And in the 2-D case the interleaving degree, associated with an interleaving scheme designed for some burst size t in (Blaum *et al*. 1998), is guaranteed to reach its lower bound. Furthermore it is known that the lower bound of the interleaving degree is a monotonically increasing function of the burst size t. Specifically, the lower bound is $t^2/2$ for even t and $(t^2+1)/2$ for odd t. Therefore, with respect to the implementation of the interleaving scheme designed for a burst size t_0, when the actual size of an error burst, t, is smaller than t_0, the achieved interleaving degree with t_0 is larger than the lower bound that corresponds to t. That is, the interleaving scheme designed for a burst size t_0 is not optimal for a smaller bust size, t.

In many applications, *the size of a given 2-D or M-D array is known*. For instance, a digital (watermarked) image may be known to have a size of 512 by 512 pixels. Under the circumstances, one may wonder if it is possible to develop a 2-D interleaving technique, which is optimal for all (if possible) or (at least) for many of the possible error burst sizes. Therefore, it can be implemented only once for a given 2-D array. Motivated by these observations, a novel 2-D interleaving technique, called successive packing approach, has been proposed (Shi and Zhang 2002).

4.2 2-D/3-D Successive Packing Interleaving

Given that digital images, video frames, charge-coupled devices (CCDs), and 2-D bar-codes are all in the form of 2-D arrays, without loss of generality, square arrays of $2^n \times 2^n$ are considered here. The utilization of $2^n \times 2^n$ arrays will be further justified later.

4.2.1 2-D Codewords and 1-D Sequence of Code Symbols

In general, the codewords in the 2-D case are of 2-D in nature. 1-D codewords, either row-type, or column-type, or other-type, can be considered as special cases of 2-D codewords. The successive packing technique is able to handle 2-D codewords because all the code symbols in the 2-D codewords are first linked into a 1-D sequence of code symbols. Without loss of generality, the quartering indexing scheme is described below for illustrative purposes. That is, a square array of $2^n \times 2^n$ is viewed as consisting of four quadrants, each quadrant itself consisting of its own four quadrants; the process repeats itself until it reaches a level where all four quadrants are of 2×2. This is referred to as 2-D successive doubling, as shown in Figure 7. These 2×2 arrays are the fundamental structure. When the quartering indexing scheme is applied, each code symbol, assigned to an element of the array has a pair of subscripts. The first subscript represents the index of the 2×2 array in which the code symbol is located, while the second subscript indicates the index of the code symbol within the 2×2 array. To convert the quartering index, $s_{i,j}$, into the 1-D index, s_k, we apply the following operation: k=4i+j.

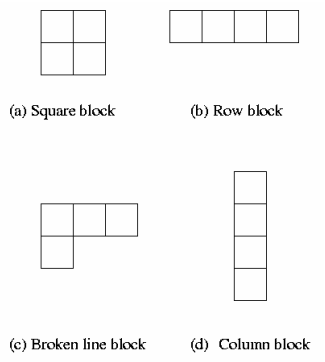


Figure 8. Four different types of 2-D codewords having four code symbols.

Quartering indexing is not the only choice for the proposed interleaving technique. Actually, codewords can be of any shape. Several shapes of a codeword consisting of four code symbols are shown in Figure 8. Obviously, for any given shape of 2-D codewords, it is always possible to label the code symbols into a 1-D sequence with a possibly more complicated bookkeeping scheme.

4.2.2 The Successive Packing Algorithm

Now we present the successive packing interleaving technique in the 2-D case in a general and compact way, which allows straightforward generalization to the M-D case. The 2-D interleaving using the successive packing proceeds as follows. Consider a 2-D array of $2^n \times 2^n$ for 2-D interleaving.

When $n = 0$, i.e., an array of 1×1 is considered for interleaving, the interleaved array is the original array itself. That is,

$$S_1 = [s_0] \tag{15}$$

where s_0 represents the element in the array, and S_1 the array. Note that the subscript in the notation S_1 represents the total number of elements in the interleaved array. Hence, when *n = 1*, i.e., for a 2×2 array, the interleaved array is denoted by S_4; when *n = 2*, the interleaved array is S_{16}. In general, for a given *n*, the interleaved array is denoted by $S_{2^{2n}}$.

The procedure is carried out successively. Given an interleaved array S_i, the interleaved array of S_{4i} can be generated according to

$$S_{4i} = \begin{bmatrix} 4 \times S_i + 0 & 4 \times S_i + 2 \\ 4 \times S_i + 3 & 4 \times S_i + 1 \end{bmatrix} \tag{16}$$

where the notation of $4 \times S_i + k$ with k=0, 1, 2, 3 represents a 2-D array that is generated from S_i. This indicates that $4 \times S_i + k$ has the same dimensionality as S_i. Furthermore, each element in $4 \times S_i + k$ is indexed in such a way that its subscript equals to four times of that of the corresponding element in S_i plus *k*. By the corresponding element, we mean the element occupying the same position in the 2-D array. It appears that S_{4i} is derived from S_i by *packing* S_i four times. This explains why the term successive packing is used.

According to the above rule, we have

$$S_4 = \begin{bmatrix} 4 \times S_1 + 0 & 4 \times S_1 + 2 \\ 4 \times S_1 + 3 & 4 \times S_1 + 1 \end{bmatrix} = \begin{bmatrix} s_0 & s_2 \\ s_3 & s_1 \end{bmatrix} \tag{17}$$

Similarly, we have S_{16} as follows.

$$S_{16} = \begin{bmatrix} 4 \times S_4 + 0 & 4 \times S_4 + 2 \\ 4 \times S_4 + 3 & 4 \times S_4 + 1 \end{bmatrix} = \begin{bmatrix} s_0 & s_8 & s_2 & s_{10} \\ s_{12} & s_4 & s_{14} & s_6 \\ s_3 & s_{11} & s_1 & s_9 \\ s_{15} & s_7 & s_{13} & s_5 \end{bmatrix} \tag{18}$$

The resemblance between the successive packing interleaving and the fast Fourier transform (FFT) is observed. Firstly, the successive doubling mentioned before is also used in FFT. Secondly, after the successive doubling, what is left here is a 2×2 basis array, which is expressed in (3) and depicted in Figure 9. This 2×2 basis array is the counterpart of the basic butterfly computation structure used in FFT. Thirdly, both techniques work on a group of data whose dimensionality is an integer power of two to facilitate utilization of digital computers.

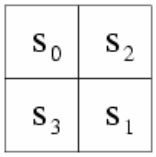


Figure 9. 2×2 basic array.

4.2.3 Main Results

It has been proved in (Shi and Zhang 2002) that in a 2-D interleaved array of $2^n \times 2^n$, A, generated with the successive packing technique, any square error burst of $2^k \times 2^k$ with $1 \le k \le n-1$ and any rectangular error burst of $2^k \times 2^{k+1}$ or $2^{k+1} \times 2^k$ with $0 \le k \le n-1$ can be spread so that each element in the burst falls into a distinct block in the de-interleaved array, where the block size, K, is 2^{2n-2k} for the burst of $2^k \times 2^k$, and $2^{2n-2k-1}$ for the burst of $2^k \times 2^{k+1}$ or $2^{k+1} \times 2^k$. This indicates that, if a distinct code symbol is assigned to each element in a block and all the code symbols associated with the block form a distinct codeword, then this technique guarantees that the error burst can be corrected with a one-random-error-correction code, provided the code is available. (Note that the code capable of correcting one code symbol error within a codeword of two code symbols does not exist in reality. Therefore,

though the error burst of $2^{n-1} \times 2^n$ or $2^n \times 2^{n-1}$ can be effectively spread in the de-interleaved arrays as described above, they in fact cannot be corrected with a one-random-error-correction code.) Furthermore, the interleaving degree equals to the size of the burst error, hence minimizing the number of codewords required for an interleaving scheme. In other words, the interleaving degree obtained by the successive packing interleaving is indeed the lower bound (Shi and Zhang 2002). In this sense, the successive packing interleaving technique is optimal. If a coding technique has a strong random-error-correction capability, say, it can correct one error in every codeword of size eight, then any error burst of $2^{n-1} \times 2^{n-2}$ or $2^{n-2} \times 2^{n-1}$ can be corrected. If a code, on the other hand, has a weaker random-error-correction capability, say, it can only correct one random error within a codeword of size 64, then only smaller error bursts, i.e., any burst of $2^{n-3} \times 2^{n-3}$ in the interleaved array can be corrected by the successive interleaving.

4.3 Simulation Results

In this subsection, we report our initial investigation on applying 2-D/3-D successive packing interleaving techniques to combat bursts of errors occurred in still images/video sequences data hiding. The experimental results demonstrate that the robustness of hidden data inside still images/video sequences against bursts of errors is significantly improved by using 2-D/3-D SP interleaving followed by ECC.

4.3.1 Applying 2-D SP Interleaving to Enhance Robustness of Still Image Data Hiding

Consider the "Lena" image ($256 \times 256 \times 8$) shown in Figure 10 (a), a widely used image for image processing experiments. The data hiding is carried out in the block discrete cosine transform (DCT) domain. First, the image is split into non-overlapped blocks of 8×8 pixels each. Then, DCT transform is applied to each block. The

largest three AC DCT coefficients are used to embed bits. If the bit to be embedded is “1”, the coefficient is added by a quantity Δ (e.g., $\Delta = 6$ is empirically used in our simulations). If the bit is “0”, the coefficient is subtracted by Δ. We use six bits to represent one symbol. The ECC used in our simulations is the BCH (31,6) code. Hence, 99 symbols are embedded. The image is *scanned* three times. Each scan embeds 1/3 of the total bits (about 1024 bits) in an AC coefficient having the same position within each block. These 1024 bits are first interleaved using the 2-D SP interleaving technique, and are then embedded into the AC coefficients. For the data extraction, parts damaged by the error burst are replaced with “0”s. Without 2-D interleaving, we simply embed bits block by block, say, from left to right, from top to bottom through out the whole image. In each block, we embed three bits. The experimental results are shown in Figures 11.

(a) Original Lena image. (b) Marked Lena image.

Figure 10. The original and marked Lena images.

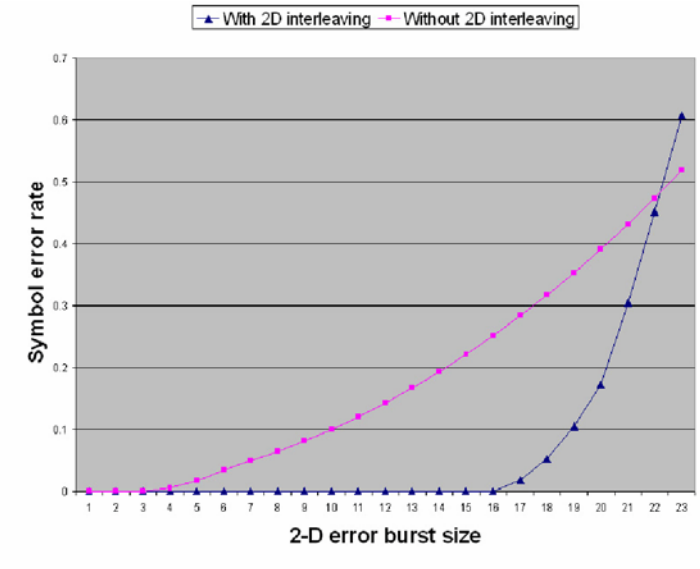


Figure 11. Test results of square error bursts.

In Figure 11, the horizontal axis represents the size of the 2-D error burst. For example, size 2 means that the error burst is a square area of 2×2 blocks (each block is of 8×8 pixels). That is, the square error burst has a size of 16×16 pixels. The vertical axis stands for the symbol error rate (SER). In the simulation, we consider all of the possible positions of the error burst. Then, the arithmetic average of SERs corresponding to the bursts occupying all possible positions is reported in the figure. As shown in the figure, without interleaving, the SER emerges when the error burst size is larger than four. With 2-D SP interleaving, the SER is still zero even when the error burst size is 16. This indicates that even when a quadrant of the marked Lena image has been in error, the SER is still zero, implying significant improvement of robustness. Note that when the error burst size is larger than 22, the SER with interleaving will be larger than that without interleaving. In this case, almost half of the image has been damaged. The SER for both

algorithms with and without interleaving has been almost 50%, practically rendering both algorithms useless in this case.

4.3.2 Applying 3-D SP Interleaving to Enhance Robustness of Video Sequence Data Hiding

A video sequence is a set of successive frames. Each frame is a 2-D image. Here we consider two types of bursts of errors that may occur to the data embedded in a video sequence. The first type of error bursts is frame loss. Frame loss may occur in video transmission, especially when the video is transmitted through a bursty and noisy channel. The second type of error bursts is what is known as 3-D error bursts. Since there is a high correlation among the successive frames, a 2-D error burst sometimes leads to the errors approximately in the same location of the succeeding frames, thus causing a 3-D error burst.

For these two types of errors, again, we conducted our simulation in a way similar to that used for image data hiding. The testing video sequence has 32 frames. Each frame is a gray image of 256×256. We split each frame into non-overlapped blocks of 8×8 pixels each. As a result, we have $32 \times 32 \times 32$ blocks within the entire sequence. DCT transform is applied to each block and the data bits are embedded into the three largest AC DCT coefficients in each block. Again, six bits are used to represent one symbol and the BCH (31,6) code is used. Data embedding is carried out in three scans. In each scan, we embed $32 \times 32 \times 32$=32,768 bits (equivalent to 1,057 symbols) in an AC coefficient having the same position in each block. These 32,768 bits are first interleaved using the 3-D SP interleaving technique, which is a straightforward extension from 2-D SP to the 3-D case (Zhang et al. 2002, Elmasry 1999), before being embedded into the AC coefficients. In the data extraction, error parts are filled with “0”s. Without 3-D interleaving, we simply embed bits block by block, say, from left to right, from top to bottom, and from front to rear. In each block, we em-

bed three bits. Figures 12-13 show the simulation results, demonstrating that 3-D interleaving can greatly improve the robustness of data hiding.

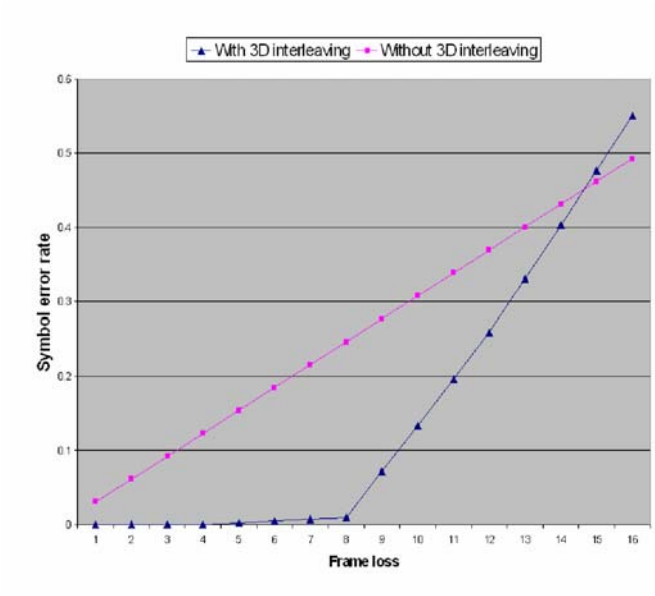


Figure 12. Test results of video frame loss.

In Figure 12, the horizontal axis denotes the number of lost frames (that are consecutive). From this figure, it is seen that when eight frames are lost, the SER with interleaving is almost zero while that without interleaving is about 25 percentages. Note that when 15 frames are lost, the error rate with interleaving will be higher than that without interleaving, in which case almost half of the 32 frames are lost. The SER for both algorithms with and without interleaving has been almost 50%, implying that the hidden data have been severely damaged.

In Figure 13, the horizontal axis is the size of the 3-D error burst. For example, size 2 means that the error burst is a cubic volume of

$2 \times 2 \times 2$ blocks (each block is of 8×8 pixels). That is, the cubic error burst has a size of 16×16 pixels in two consecutive frames. With interleaving, the SER is still zero even when the error burst size is 16 (one eighth of the blocks are lost), while without interleaving the SER is more than 12%. Clearly, the significant improvement on the robustness of hidden data against 3-D bursts of errors has been achieved.

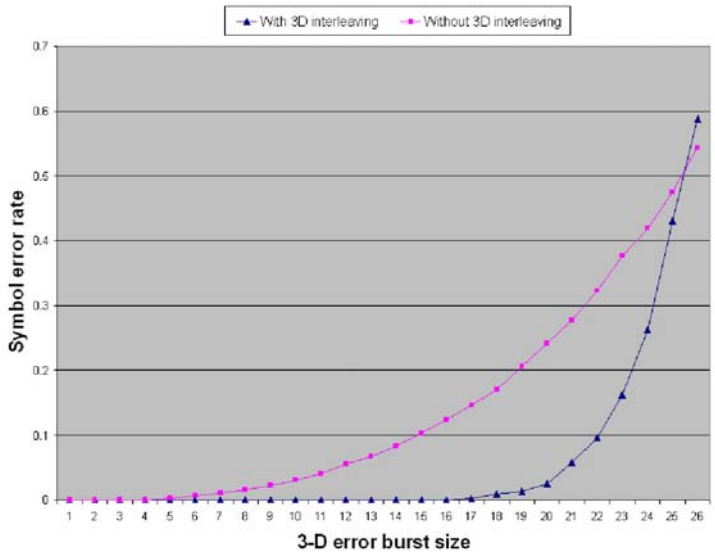


Figure 13. Test results of 3-D error burst.

In summary, from our initial investigation of applying 2-D/3-D SP interleaving techniques to enhance the robustness of hidden data in still images/video sequences, it is shown that in either still image or in video sequence data hiding, the SP interleaving can greatly enhance the robustness of hidden data against bursts of errors. Therefore, 2-D/3-D successive packing interleaving can play a promising role in enhancement of robustness in image/video data hiding. It is

expected that the 2-D interleaving that can make 2-D data more robust and reliable can also find important applications in the areas such as 2-D bar-codes and holographic storage.

5. Frame Synchronization in Video Data Hiding

As discussed in Section 2.2, pixels in each frame of a video are cascaded in, say, a row-by-row or column-by-column manner, then the whole video sequence is cascaded in a frame-by-frame manner so that all pixels in a video sequence are cascaded into a huge 1-D sequence. In spread spectrum video data hiding, we need to do correlation for each group of pixels that are corresponding to one bit in data extraction. This indicates that data detection heavily relies on synchronization. This is true in general for all spread-spectrum technologies. The synchronization of all pixels in the video sequence, which are 3-D in nature, becomes a critical issue, and is not easy to handle. It is also discussed that brute-force search for best matching is not an efficient to achieve synchronization.

5.1 Embedding Side-information for Synchronization

It is known that for coherent modulation digital communication systems, a three-level synchronization: phase, symbol and frame synchronization is required. For noncohenrent modulation, a different three-level synchronization: frequency, symbol and frame synchronization is necessary (Sklar 1988). Therefore embedding some side-information is a way to achieve synchronization. In (Liu *et al*. 2001), frame synchronization was achieved by embedding frame numbers as side-information into each video frame.

5.1.1 Introduction to the Algorithm by Liu *et al*. (2001)

This algorithm embeds watermark signal and frame number into uncompressed domain, specifically, in discrete wavelet transform (DWT) domain, a three-level DWT using Daubechies 97 technique. A number of features of this video watermarking techniques are listed below.

1) Instead of embedding watermark signal into high frequency subbands, the watermark signal and side-information are embedded into the LL subbands in order to achieve stronger robustness.

2) In order to combat possible bursts of errors and random errors that may occur in watermarked video sequences, the newly developed, efficient 3-D interleaving technique (Shi and Zhang 2002) and with an error correction code, BCH (61,8), are utilized.

3) The security of the watermark signal is enhanced by modulating the watermark with a random sequence.

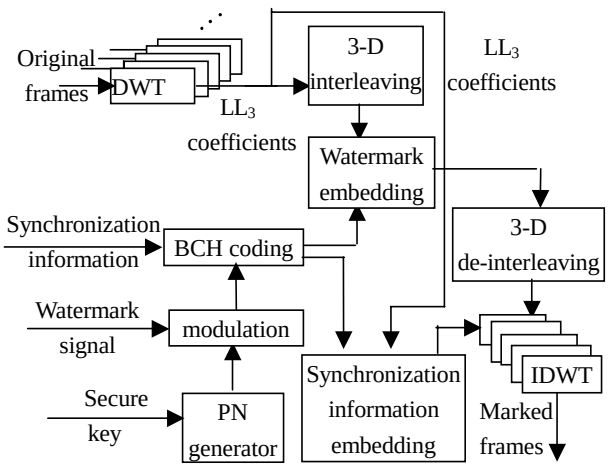


Figure 14. Block diagram of the proposed algorithm.

4) Frame number is embedded into each frame as side-information to achieve frame synchronization. The watermark signal is embedded into the centered portion of the LL_3 sub-band in the DWT domain of a video frame. The rest four corners can be used to embed the frame number. Considering the phenomenon that people pay more attention to the centered area of the image while less to the boundary especially to the four corners of the image, the synchronization information is embedded with extra strength, compared with watermark signal.

A block diagram of the proposed algorithm is shown in Figure 14.

5.1.2 Experimental Results

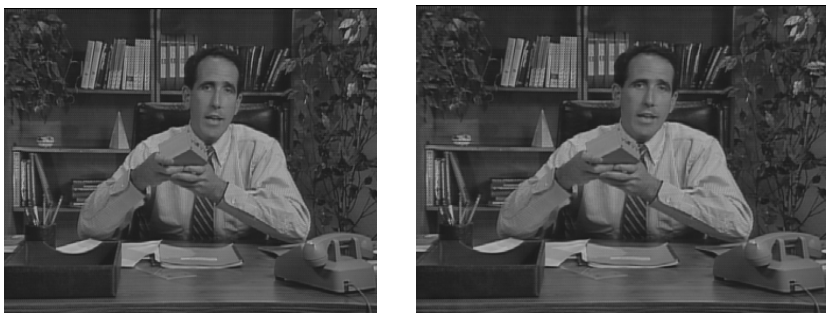
It was reported that the algorithm has been tested on various CIF video sequences, including "Salesman," "Mobile," and "Paris." Total of 96 frames in these sequences are used to embed data. An 1140-character string can be hidden. 128 bytes (including the redundant bits in EEC) can be embedded into one CIF video frame. Note that this payload is much larger than that in the DVD video copy protection, discussed in Section 3. The experimental results with the "Salesman", including invisibility and the robustness to some attacks such as frame loss, MPEG-2 coding, and rescale are shown in Figures 15, 16, and 17, respectively. Similar results have been obtained with other video sequences.

Figure 15 demonstrates the invisibility of the hidden data with the proposed algorithm, where 15 (a) is one of the original frames of the "Salesman" sequence, 15 (b) is the data hidden frame. The embedded data are perceptually invisible when we compare two video frames. It was reported that the experiments also showed that the dynamic invisibility of the watermark could be guaranteed when the marked frames are played. The tests of the robustness of the data hidden video sequence against consecutive and random dis-

crete frame loss were carried out. The comparison between the performance of with the 3-D interleaving and without the 3-D interleaving was made. The frame loss rate is defined as

$$fr = number\ of\ lost\ \ frames / number\ of\ total\ \ frames\,. \qquad (19)$$

Figure 16 (a) and (b) demonstrate the robustness of the embedded data against consecutive and random discrete frame loss, respectively. It is observed that the robustness with the 3-D interleaving is much stronger than that without interleaving when $fr < 0.5$. The experiments also demonstrate that the algorithm achieves almost zero byte error rate when the frame loss rate is about 0.4, no matter how the frames are lost, randomly or consecutively.

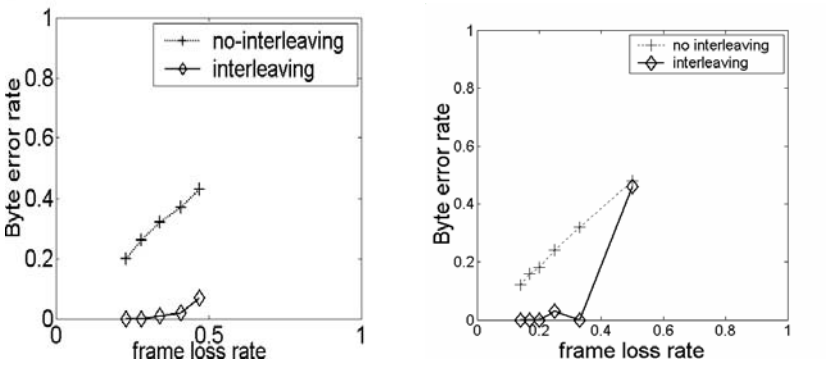


(a) Original video frame (b) Marked frame (PSNR=49.23 dB)

Figure 15 Demonstration of invisibility.

Robustness against MPEG-2 compression at different compress ratio (CR) is tested. Figure 17 is the performance curve of the robustness to MPEG-2 coding. The algorithm can correctly detect the embedded character string when CR=11.6, and data rate is 2.7Mb with the PSNR of video frames equal to 41.12 dB. The byte error rate is less than 1%, when CR is within 13.81. In addition, we test

the robustness to rescale (scale 2.0) attack. The embedded information can be detected error-freely.



(a) Consecutive frame loss (b) Random frame loss

Figure 16. Robustness against frame loss.

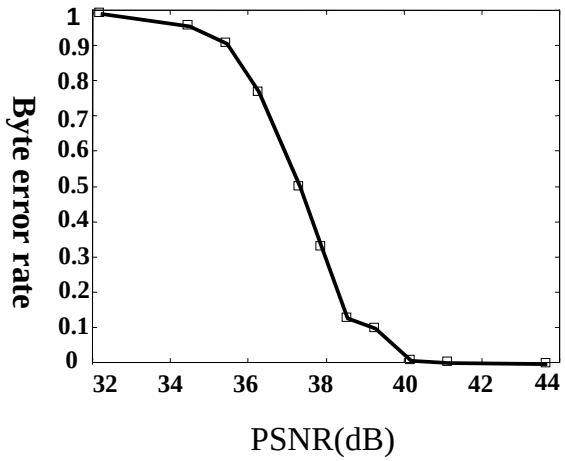


Figure 17 Robustness against MPEG-2 coding

Acknowledgments

This work was supported in part by New Jersey Commission of Science and Technology via NJCWT and NJWINS, New Jersey Commission of Higher Education via NJ-I-TOWER, and NSF via IUCRC; by NSF of China (69975011, 60172067, 60133020), "863" Program (2002AA144060), NSF of Guangdong (013164), Funding of China National Education Ministry; and by the KOSEF through the AITrc in Korea.
.

References

M. Blaum, M., J. Bruck and A. Vardy (1998), "Interleaving schemes for multidimensional cluster errors," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 730-743.

Bose, R. C., D. K. Ray-Chaudhuri (1960), " On a class of error correcting binary group codes," *Inform. Control*, vol. 3, pp. 68-79.

Cox, I. J., J. Kilian, T. Leighton and T. Shamoon (1997), "Secure spread spectrum watermarking for multimedia," *IEEE Trans. on Image Processing*, vol. 6, no. 12, pp. 1673-1687.

Elmasry, G. F. (1999), "Detection and robustness of digital image watermarking signals: A communication theory approach," Ph.D. dissertation, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, August 1999.

Golomb, S. W., and L. R. Welch (1970), "Perfect codes in the Lee metric and the packing of polyominnoes," *SIAM J. Appl. Math*., vol. 18, no. 2, pp. 302-317.

Hartung, F. and B. Girod (1998), "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, no. 3, pp. 283-301.

Hocquenghem, A. (1959), *Codes Correcteurs d'Erreurs*, Chiffres, vol. 2, pp. 147-156.

Huang, J., G. Elmasry and Y. Q. Shi (1998), "Power constrained multiple signaling in digital image watermarking," *Proceedings of 1998 IEEE Workshop on Multimedia Signal Processing*, pp. 388-393, Los Angeles, CA.

Huang, J. and Y. Q. Shi (2002), "Reliable information bit hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 916-920.

Imai, H. (1973), "Two-dimensional Fire codes," *IEEE Transactions on Information Theory*, vol. 19, no. 6, pp. 796-806.

Maes, M, T. Kalker, J. M. G. Linnartz, J. Talstra, G. F. G. Depovere and J. Haitsma (2000), "Digital watermarking for DVD video copy protection," *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 47-57.

Langelaar, R. L. Lagendijk and J. Biemond, (1998), "Real-time labeling of MPEG-2 compressed video," *J. Visual Commun. Image Representation*, vol. 9, no. 4, pp. 256-270.

Liu, H, N. Chen, J. Huang, X. Huang and Y. Q. Shi (2002), "An robust DWT-based video watermarking algorithm," *Proceedings of IEEE International Symposium on Circuits and Systems*, Phoenix, AZ.

Shi, Y. Q. and H. Sun (1999), Image and Video Compression for Multimedia Engineering, CRC Press, Boca Raton, FL.

Shi, Y. Q., and X. M. Zhang (2002), "A new two-dimensional interleaving technique using successive packing," *IEEE Transactions on Circuits and Systems, Part I: Fundamental Theory and Application,* vol. 49, no. 6, pp. 779-789.

Shi, Y.Q., Z. C. Ni, J. Huang, N. Ansari, and W. Su (2003), "A successive 2-D/3-D interleaving to enhance robustness of image/video data hiding," *IEEE International Symposium on Circuits and Systems*, Bangkok, Thailand. (accepted)

Sklar, B. (1988), *Digital Communications*, PTR Prentice Hall, Englewood Cliffs, New Jersey.

Wicker, S. B. (1995), *Error Control System for Digital Communication and Storage*, Prentice-Hall, Inc, Englewood Cliffs, NJ.

Zhang, X. M., Y. Q. Shi and S. Basu, (2002), "On successive approach to multidimensional interleaving," *Fifteenth International Symposium on Mathematical Theory of Networks and Systems* (MTNS02), University of Notre Dame.