Motion-Embedded Residual Error for Packet Loss Recovery of Video Transmission and Encryption

Shih-Wei Sun[1,3], Jan-Ru Chen[2,3], Chun-Shien Lu[3,*], Pao-Chi Chang[4], and Kuo-Chin Fan[2]

[1]Dept. Electrical Engineering, National Central Uni., Chung-Li, Taiwan, ROC
[2]Dept. Computer Sci. & Infor. Eng., National Central Uni., Chung-Li, Taiwan, ROC
[3]Institute of Information Science, Academia Sinica, Taipei, Taiwan, ROC
[4]Dept. Communication Engineering, National Central Uni., Chung-Li, Taiwan, ROC

ABSTRACT

Media encryption technologies actively play the first line of defense in securing the access of multimedia data. Traditional cryptographic encryption can achieve provable security but is unfortunately sensitive to a single bit error, which will cause an unreliable packet to be dropped to create packet loss. In order to achieve robust media encryption, error resilience in media encryption can be treated to be equivalent to error resilience in media transmission. This study proposes an embedded block hash searching scheme at the decoder side to achieve motion estimation and recover the lost packets, while maintaining format compliance and cryptographic provable security. It is important to note that the proposed framework is a kind of joint error-resilient video transmission/encryption and copyright protection.

Keywords: DES, (Selective) Encryption, Error concealment and resilience, Media hashing, Packet loss, Security, Transmission

1. INTRODUCTION

1.1. Background

Multimedia data transmitted in digital distribution networks can be secured by the first line of defense technologies, i.e., cryptographic encryption. The characteristics of cryptographic encryption technologies are their provable security and absolute fragility in that a single bit error will render the whole encrypted bitstream wrongly decrypted. However, with the advancement of the Internet and multimedia technologies, media data is usually compressed before transmission to save bandwidth and then transmitted in an error-prone or unreliable environment. The fragility of an encryption scheme will prohibit it from being used for protection or access control of media data. Therefore, multimedia encryption is different from fragile cryptographic encryption in that the former needs resilience to attacks. On the other hand, digital watermarking, a kind of data hiding technologies, plays the role of the second line of defense[26] in that it provides passive protection only when copyright infringement needs to be solved. In view of these facts, this study focuses on the error-resilient media encryption.

Multimedia encryption[25, 28, 29] is required to satisfy a number of conflicting requirements, as briefly described in the following.

A. **Format compliance**: The encrypted video bitstream needs to be compatible with the syntax of coding standards so that standard decoder can accept decrypted bitstream without needing special designed decoders or additional information to enable decoding.

B. **Security**: Security is a cardinal requirement for encryption. Intuitively, cryptographic encryption is a good choice due to its provable security. However, by concerning its fragility other encryption techniques (e.g., permutation, shuffling, or scrambling) without the side-effect of error propagation are useful at the expense of sacrificing strong security.

C. **Complexity**: Modern applications need lower decryption complexity on low-powered consumer electronic devices. However, higher secure encryption methods need complex operations to achieve high-degree security.

D. **Robustness**: A practical media encryption scheme needs to be able to restrict error propagation and recover

*Further author information: Send correspondence to Chun-Shien Lu, E-mail: lcs@iis.sinica.edu.tw, Telephone: +886 2 2788 3799 X 1513. The first two authors are Ph.D. students and contribute equally to this work.

Visual Communications and Image Processing 2006, edited by John G. Apostolopoulos, Amir Said,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6077, 60771B, © 2005 SPIE-IS&T · 0277-786X/05/$15

lost packets. In packet-switched networks, a single transmission error will make a packet unreliable and then dropped. How to recover the lost packets is known as the error control problem in the video communication community. Basically, error resilience in video encryption is equivalent to error resilience in video transmission. E. **Coding efficiency**: The increased redundancy due to encryption should be limited in order not to considerably reduce coding efficiency.

Usually, the afore-mentioned requirements are conflicting and proper trade-offs should be enforced depending on applications. In this paper, we will focus on the issue of error resilience in video encryption that was not properly addressed in the literature. Due to the characteristic of packet-based networks in treating a packet with at least one erroneous bit unreliable, such packet is dropped to create packet loss. Error-resilient video encryption is regarded to be equivalent to error-resilient video transmission. In the next section, we shall give a brief survey about the developments of robust video transmission methods. It is important to note that the framework of our method is adaptive to both error-resilient video transmission and encryption simultaneously.

1.2. Related Work

In the literature, the error control methods for video transmission can be roughly divided into five categories according to where and how the mechanism is operated. The first three categories can also be found in.[23]

1.2.1. Encoder-Level

The error control methods, operated at the encoder-level, are designed to enhance the resistance of encoded video streams to channel errors, and are usually called "error resilience" technologies. The design strategy is to suitably introduce redundant information to the encoded bit stream such that decoded video can preserve a certain quality when errors are encountered. Among error resilience methods, inserting re-synchronization markers and data partitioning[14, 16] is able to efficiently separate errors from video steam to deter error propagation. Error resilience can also be achieved based on sophisticated coding techniques such as layered coding(LC) or multiple-description coding (MDC).

1.2.2. Transport-Level

The error control methods operated at the transport-level employ forward error correction coding (FEC)[2, 15] by adding redundant information resulted from error-correction code (ECC) to protect the video stream. The penalty is the resultant video stream is not format-compliant and the coding efficiency is sacrificed.

1.2.3. Decoder-Level

The error control methods that utilize error concealment (EC) or post-processing mechanism at the decoder side to accomplish error recovery without needing to transfer redundant information is called error concealment with zero-redundancy.[19, 22] The error concealment mechanism needs to be triggered by means of syntax-based error detection. The detectable errors include (i) loss of synchronization (due to error-corrupted VLC parameters); (ii) syntax errors in codec; and (iii) errors in transport-level headers.

1.2.4. Data Hiding-based

Recently, data hiding technologies[9, 11, 20] have also been employed for correcting transmission errors. In,[20] Song and Liu proposed a motion vector (MV) protection scheme by embedding the MV parity bits of the current frame into its subsequent frame. The main drawbacks are that the number of lost slices within a frame is restricted to 1 and bursty errors are not considered. Shanableh and Ghanbari[18] proposed an error concealment scheme, which can be regarded as a data hiding-like method. Since motion vectors play a crucial role in the video decoding process, they exploited the inherent B-pictures property in a way that the concealment motion vectors can be restored if it is forced to be derived from the relationship between a pair of forward and backward motion vectors. In,[3] a fragile watermarking approach was proposed to better achieve error detection. The merit of exploiting the hidden watermark signals is that the errors, beyond syntax errors, remaining in the received video stream can still be detected.

1.2.5. Side Information-based

Recently, side information concept[1, 17] has been addressed for robust video transmission. Aaron *et al.*[1] proposed to extract frame hashes and transmit them using an independent channel to help decoder in estimating motion vectors. This work describes distributed source coding for sensor networks. Sehgal *et al.*[17] proposed the idea of peg frame in H.264, which is used to play the role of a reference frame in order to prohibit from error propagation. However, video frames were restricted to refer peg frames only in motion estimation, thus the coding efficiency would be sacrificed.

1.3. Our Observations

Unequal error protection (UEP) is usually adopted for video transmission in that significant data is given more protections than insignificant data. In this situation, motion vectors are usually regarded to be the most important data and should be given strong protections. For example, in H.264/AVC a normal slice can be partitioned into three parts, i.e., data partition A/B/C, each of which is encapsulated into an individual NAL packet. Data Partition (DP) A comprises motion vectors, quantization parameters, and the header information, DP B includes transformed coefficients of I-blocks and Intra coded block patterns (CBPs), and DP C contains Inter CPBs and transformed coefficients of P-blocks. Traditionally, DP A is regarded to be the most important part while DP C is less important. In addition, when partition loss is detected, four actions that can be adopted at decoder for loss recovery are recommended. Among them, when the available partitions are B and C (i.e., motion vectors are lost), the recommended action is to drop DPs B and C, and use motion vectors of the spatially above MB row for each lost MB.

However, we observe that it is not feasible to protect motion vectors only because the recovered quality is still not enough, in particular for the video sequences with large motions. The residual errors are indispensable to repair the high-frequency information so that the degraded quality can be further recovered. In addition, we can find that when motion vectors are lost, error concealment (EC) is recommended for loss recovery, which does not work for videos with large motions. In other words, when motion vectors are lost, there is not an efficient way to recover them. In view of these facts, we present a new error-resilient video transmission method based on motion-embedded residual error recovery in this paper. The block hashes are extracted and embedded into the residual data at encoder and the hashes are extracted at decoder for motion estimation and compensation. Specifically, when the packet in DP A is lost, the block hash is used for block matching (motion estimation) and the best match is used to replace the lost data for motion compensation. In particular, if the recovered block is combined with the residual error (DP C), then the recovered data can be more similar to the lost data. That is the main contribution of the proposed method.

The remainder of this paper is organized as follows. In the next section, we describe why the error resilience problem in media encryption can be equivalent to that in media transmission. In Sec. 3, the general principle of media hashing is discussed and the proposed scheme is described. In Sec. 4, we describe how to embed block hash so that the block hash matching can be performed at decoder for motion estimation and compensation. Experimental results are given in Sec. 6 and conclusions are drawn in Sec. 7, respectively.

2. PROBLEM STATEMENT: ROBUSTNESS IN MEDIA ENCRYPTION CONSIDERED TO BE EQUIVALENT TO ERROR RESILIENCE IN VIDEO TRANSMISSION

A common way to achieve robustness of media encryption is to turn on the mode of error concealment associated with coding standards. However, the inherent capability of error concealment is rather restricted because the surrounding motion vectors of a lost packet and its own motion vector are often inconsistent. In,[21] Tosun and Feng proposed an error-resilience encryption scheme, which recovers errors up to the bit-level. However, in view of the fact that packet-based networks treat a packet with at least one erroneous bit unreliable and drop such packet to create packet loss. In such a case, it will be not enough to merely deal with bit errors. We, therefore, consider error-resilient video encryption to be equivalent to error-resilient video transmission. In other words, to repair the quality degradation caused by packet loss is the key to achieve error-resilient video encryption. We present a new error-resilient video encryption scheme, which exploits the embedded block hashes for macroblock matching at decoder to search for the best target (motion estimation) and use it to recover (motion compensation)

the lost packet. In addition to error resilience, the other requirements of media encryption, including security and format compliance, are also taken into considerations. To our knowledge, a technology about robust video encryption against packet loss beyond traditional error control mechanisms has not been found in the literature.

Since robust media hashing is considered (see Sec. 3 and[13]), partial content matching still permits to find the desired target without being certain to affect the capability of motion estimation and compensation. However, when motion vectors are embedded for recovering the lost macroblocks, even a few bit errors may render the extracted motion vector significantly different from the embedded one so as to affect the recovery performance. In addition, the size of a motion vector may be too large to be totally embedded. However, if hiding capacity is limited, partial hash matching is still possible to find the best match. Based on the above concerns, that's why block hash is embedded at encoder for motion estimation and compensation at decoder in this study.

3. MEDIA HASH

3.1. General Principle

The media hash,[4,13] also known as the "digital signature"[8,12] or "media fingerprint",[5,6] has been used in many applications, including content authentication, copy detection, media recognition, and error resilience. Referring to the image space shown in Fig. 1, let $\mathbf{I}$ denote an image, and let $\mathcal{X}$ denote the set of images that are modified from $\mathbf{I}$ by means of content-preserving operations (e.g., filtering, compression, and geometric distortions) and are defined as being perceptually similar to $\mathbf{I}$. Although perceptual similarity is still an ill-posed concept,[7,24] we will propose a block hash-based matching metric in the next section for macroblock searching. We further use $\mathcal{Y}$ to denote those images that are modified from $\mathbf{I}$ but can hardly be recognized as originating in $\mathbf{I}$. For example, severe noise adding and severe cropping are two representative attacks that can generate the elements of $\mathcal{Y}$. In addition, we denote using $\mathcal{Z}$ a set which contains all the other images that are irrelevant to $\mathbf{I}$ and its modified versions. Consequently, $\{\mathbf{I}\} \cup \mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ is a case that forms an entire image space.

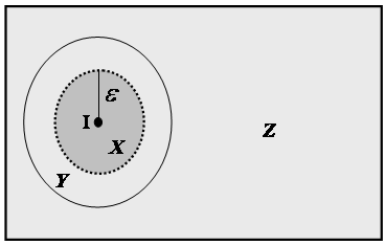


Figure 1. The Image Space. I is an element in the image space. $\mathcal{X}$ denotes the set of images modified from I that are still perceptually similar to I. $\mathcal{Y}$ denotes the set of images modified from I that are perceptually different from I. $\mathcal{Z}$ is the set of images that are irrelevant to I.

In order to represent the condensed essence of an image for perceptual similarity measurement, a hash function is usually employed. Conventionally, a cryptographic hash function, H^c, is used to map an image $\mathbf{I}$ as a short binary string, $H^c(\mathbf{I})$. One of the most important properties of cryptographic hashing is that it is collision-free, which means that it is hard to find two different images that can be transformed to produce the same hashes. Let $z \in \mathcal{Z}$, and let z and $\mathbf{I}$ be distinct. The collision-free property of cryptographic hashing will yield $H^c(\mathbf{I}) \neq H^c(z)$. Furthermore, let $x \in \mathcal{X}$; then, cryptographic hashing will yield $H^c(\mathbf{I}) \neq H^c(x)$. This implies that cryptographic hashing inherently produces totally different hash sequences if the media content has been modified.

However, this characteristic is too restricted to be suitable for multimedia applications since multimedia content permits acceptable distortions. As a result, it is necessary to develop a media hashing function, H^m, that can provide error-resilience. The error-resilience property of media hashing is defined as follows. It is said that x ($\in \mathcal{X}$) is successfully identified as having been modified from $\mathbf{I}$ if $d(H^m(\mathbf{I}), H^m(x)) \leq \epsilon$ holds, where $d(\cdot,\cdot)$ indicates a Hamming distance function. In other words, if two images are perceptually similar, their corresponding hashes must be highly correlated. In addition, the desired media hash function still needs to possess the collision-free property, like cryptographic hashing, except that the distance measure is changed to

$d(H^m(\mathbf{I}), H^m(x)) > \epsilon$. On the other hand, it is insignificant whether y ($\in \mathcal{Y}$) can be identified as having been modified from $\mathbf{I}$ or not because y is severely degraded from $\mathbf{I}$ and they are perceptually dissimilar in terms of similarity measurement. It should be noted that the traditional cryptographic hash function is a special case of the media hash function in that its ϵ value is set to 0. As a whole, the main idea behind media hashing is to develop a robust hash function that can identify perceptually similar media contents and possess the collision-free property.

3.2. Proposed Video Block Hashing

In this paper, the proposed video macroblock hash extraction method is described in the following. For blocks partitioned from a macroblock (MB) MB_b, a piece of representative and robust information is created. First, a block is divided into several smaller blocks of size 4×4, which is the minimum block size in H.264 video coding. For each 4×4 block, local DCT transformation is performed. Let N_b denote the number of 4×4 blocks in a partitioned block, and let $DCT_b^k(m)$ denote the m-th AC component. In addition, we define a difference sequence $\mathbf{d_b}$ as

$$d_b(k) = |DCT_b^k(m)| - |DCT_b^{(k+1) \ mod \ N_b}(m)|. \tag{1}$$

And the hash bits, $r_b(k)$'s, are assigned as follows.

$$r_b(k) \quad = \quad \begin{cases} 1, & \text{if } d_b(k) > 0 \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $r_b(k)$ is an element of a feature sequence $\mathbf{r_b}$. From our empirical observations,[13] the DC coefficients are not selected because they are not helpful for capturing identifiable features.

4. PROPOSED ERROR-RESILIENT VIDEO TRANSMISSION AND ENCRYPTION METHOD

The block diagrams of the proposed method at the encoder and decoder sides are, respectively, depicted in Figs. 2(a) and (b). In addition, the data partitioning mode of H.264 is turned on. As shown in Fig. 2(a), media hashes are extracted based on the macroblock partitioned blocks in the DCT domain and embedded into the quantized coefficients of residual data in DP C with the rate-distortion optimization. In addition, after quantization is performed, some important data is selected for light-weight encryption. In order to achieve format compliance, encryption is conducted before entropy coding. For the decoder shown in Fig. 2(b), the decoding and decryption processes are the inverse processes at the encoder. In the following, each component will be specifically described.

4.1. Basic Structure of H.264 and its Light-Weight Encryption

In this paper, H.264 is adopted as the video codec for video encryption and transmission. The basic encoding/decoding unit in H.264 is the slice, which contains several macroblocks. In order to deal with the problem of error-resilient encryption and transmission, we propose to encrypt and transmit the video sequence based on the basic unit, slice, in the packet-based networks.

In the proposed method, only the sign bits of DC components in I-frames and the sign bits of MVs[§] in P-frames are selected for encryption. The selections are about the concern of computational overhead so that the proposed method is a kind of selective encryption[†] in the sense that important information is encrypted. With the selected data, the famous cryptographic encryption mechanism 3-DES is applied to perform encryption. Our method possesses many advantages. First, the bit-rate is not increased[‡] since the data selected for encryption does not affect coding efficiency. Second, encryption is conducted based on a video slice (packet), therefore, transmission

[§]If the number of sign bits of MVs is not enough for block-based encryption, some sign bits of DCT components will be used.

[†]Selective encryption is not suitable for military applications because even slight but un-clear information cannot be revealed.

[‡]Since block hashes are embedded for error resilience, the final bit-rate will be increased due to embedding.

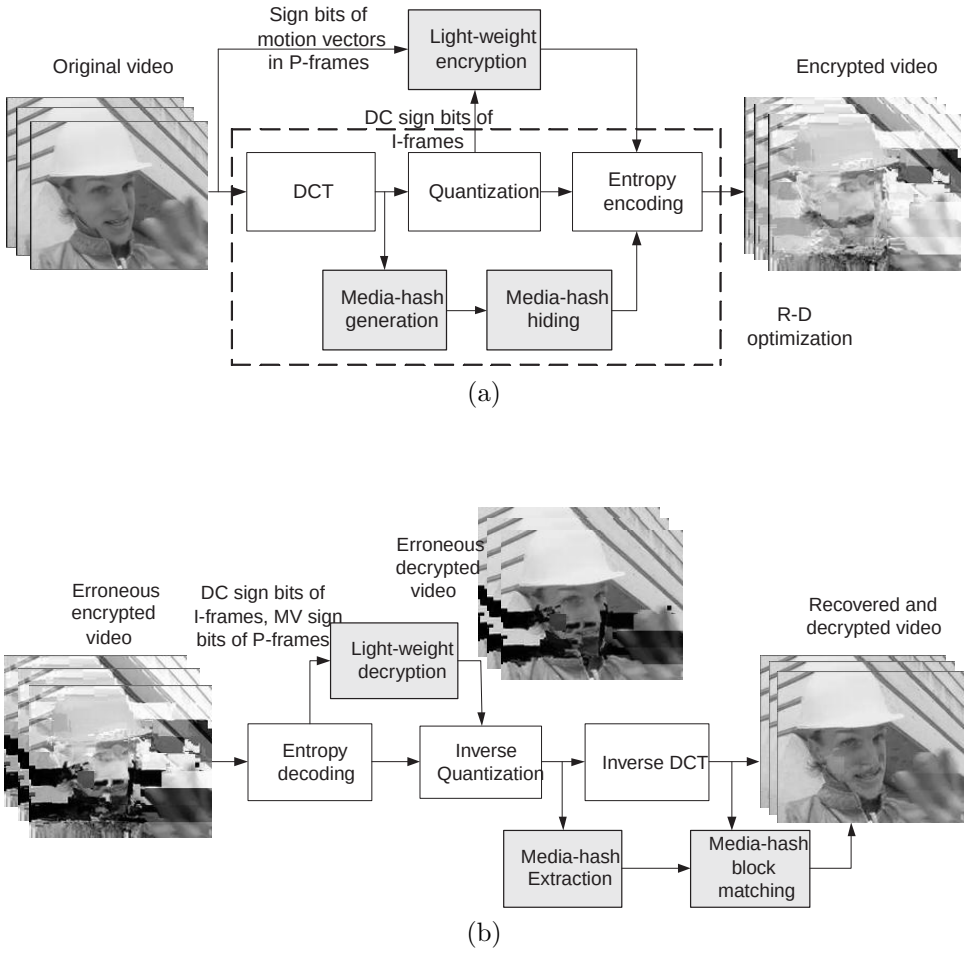


Figure 2. Block diagrams of the proposed error-resilient video transmission and encryption at the encoder side (a) and decoder side (b).

errors propagated among different video packets are avoided. Third, the encrypted format is compatible to video codecs in that the encrypted video can be decoded without needing additional decryption information. Fourth, the security level is guaranteed up to the provable security provided by 3-DES. Finally, error resilience with recovery of packet loss is investigated and is the focus of this study.

4.2. Media Hash Hiding at Encoder

The media hashes of 16×16 macroblocks are extracted in the DCT domain using the method described in Sec. 3. In this paper, the media hash of each macroblock is embedded into the residual data located in the same macroblock position of the current frame. The residual data selected as the hiding places are the non-zero quantized AC coefficients. During the data hiding, the rate-distortion optimization process also helps to achieve the optimized mode selection. The modified quantized AC coefficients are taken into considering for R-D optimization to maximize the compression performance. Let $MH_{j,k,l}(i)$ denote the i-th hash bit of the j-th macroblock from the k-th slice of the l-th frame. The slice hash $\mathbf{MH_{k,l}}$ is embedded into the quantized slice residual data $\mathbf{SR_{k,l}}$, which means the slice residual data of k-th slice of the l-th frame.

An odd-even data hiding technique is applied for hash bit embedding. Let $SR_{k,l}(m)$ be the m-th non-zero

DCT coefficient of the slice residual data $\mathbf{SR_{k,l}}$.

$$SR^h_{k,l}(m) = \begin{cases} sgn(SR_{k,l}(m))(|SR_{k,l}(m)|+1), \\ \text{ if } SR_{k,l}(m) \; mod \; 2 \neq MH_{j,k,l}(i); \\ SR_{k,l}(m), \text{ otherwise,} \end{cases} \quad (3)$$

where $sgn(\cdot)$ denotes the sign of its argument, and $SR^h_{k,l}(\cdot)$ denotes stego data. The principle of our embedding is to enlarge the magnitude of a DCT coefficient with the aim that the DCT coefficients are not easy to become zero after compressions so as not to affect hash extraction.

Although the macroblock hash $\mathbf{MH_{j,k,l}}$ is ideally of length $N_b \times 16$ bits, not all bits can be guaranteed to be embedded if the number of non-zero DCT coefficient of $\mathbf{SR_{k,l}}$ is smaller than $N_b \times 16$. Before embedding, the length, $NZ_{k,l}$, of $\mathbf{SR_{k,l}}$ is calculated. Let $NZ_{k,l}$ be the number of non-zero coefficients of the k-th slice from the l frame. The data hiding capacity of a MB is $MBC = min(N_b \times 16, NZ_{k,l})$.

It is important to note that the macroblock hash $\mathbf{MH_{j,k,l}}$ here does not extracted from the macroblock $\mathbf{MB_{j,k,l}}$. On the contrary, at encoder the macroblock that is most similar to $\mathbf{MB_{j,k,l}}$ is searched through motion estimation, and then the hash of that found macroblock is extracted as $\mathbf{MH_{j,k,l}}$, which is embedded using the method described above. The goal is that when $\mathbf{MB_{j,k,l}}$ is lost, we can guarantee to find the best match to recover it through $\mathbf{MH_{j,k,l}}$, which is useful to retrieve the result that is consistent with that obtained from motion estimation.

4.2.1. Analysis of Distortions Caused by Hash Embedding

As indicated in Eq. (3), hash bits are embedded into the quantized residual data. Let Δ_s be the uniform quantization parameter for a subband s. It is reasonable to assume that the mean quantization distortions, E^s_Q, is uniformly distributed over the interval Δ_s. We have

$$E^s_Q = \frac{1}{\Delta_s} \int_{\frac{-\Delta_s}{2}}^{\frac{\Delta_s}{2}} x^2 dx = \frac{\Delta_s^2}{12}, \quad (4)$$

which is well-known in the video codec community. In addition, when embedding is further performed, the distortions introduced for quantized residual data in the k-th slice of the l-th frame, i.e., $SR^h_{k,l}(m) - SR_{k,l}(m)$, is also uniformly distributed over the interval Δ_s. According to the embedding rule Eq. (3), we can derive

$$E(SR^h_{k,l}(m) - SR_{k,l}(m)) = \frac{1}{2}\frac{\Delta_s^2}{12} + \frac{1}{2}\frac{1}{\Delta_s}\int_{\frac{-3\Delta_s}{2}}^{\frac{-\Delta_s}{2}} x^2 dx = \frac{7\Delta_s^2}{12}. \quad (5)$$

By comparing the above two equations, we can find that the distortions increased due to embedding in a subband s is $\frac{\Delta_s^2}{2}$.

4.3. Media Hash Extraction and Matching at Decoder

When packet loss occurs during encrypted video transmission, the embedded block hashes are extracted for block matching to achieve error recovery, as shown in Fig. 2(b).

Assume that the MVs of the k-th slice of the l-th frame is lost and the residual data of the k-th slice of the l-th frame is error free. The MB capacity MBC can be calculated, as described in Sec. 4.2. Next, the media hash bits can be extracted from the slice residual data as

$$MH_{j,k,l}(i) = SR_{k,l}(m) \; mod \; 2, \quad (6)$$

where $SR_{k,l}(m)$ denotes the m-th non-zero DCT coefficient of the slice residual data $\mathbf{SR_{k,l}}$.

When media block hash $\mathbf{MH_{j,k,l}}$ is extracted, the block hash matching process is performed to search for the most similar block and use it to recover the lost block. The concept sounds like motion estimation traditionally performed at encoder, but media hash matching is applied here instead. The major difference is that the media hash, the condensed representation of a block, is exploited to search the best match.

In the block hash matching process, we first define the search range, which concerns about the trade-off between recovery and complexity. Let the lost packet, $\mathbf{MB_{j,k,l}}$, be located at the j-th macroblock of the k-th slice in the l-frame. Here, the search range is defined to be the set, Ψ, of positions covered by the set, Φ, of macroblocks neighboring to $\mathbf{MB_{j,k,l}}$. Both Φ and Ψ are defined as:

$$\begin{aligned} \Phi &= \{\mathbf{MB_{j_w,k_w,l_w}} || j| \leq j_w, |k| \leq k_w, |l| \leq l_w\}, \\ \Psi &= \{(x,y) | (x,y) \text{ is a position in the MB} \in \Phi\}. \end{aligned} \tag{7}$$

It is important to note that we cannot merely use the macroblocks belonging to Φ for hash matching because their starting positions are multiples of 16. In order to obtain accurate motion estimation, we must use the positions in Ψ as a starting pixel of a macroblock for hash matching. We further let Φ^{ψ} denote the macroblocks whose starting pixel belongs to Ψ. Thus, we have $\Phi^{\psi} \supset \Phi$.

With the search range, the block hash is extracted from each macroblock belonging to Φ^{ψ}, and compared with the extracted hash $\mathbf{MH_{j,k,l}}$ by calculating the Hamming distance. For those macroblocks ($\in \Phi^{\psi}$) with Hamming distances smaller than a threshold, they are put into a list of candidates that will be further considered to find the best match. With a list of candidates, a side-match strategy is exploited to choose the final target for recovery. Yes) are applied for comparison.

5. ANALYSIS OF MEDIA HASH MATCHING RESILIENCE VS. FORWARD ERROR CORRECTION

In this section, comparisons of error resilience between the proposed method and forward error correction (FEC) are analyzed. In our method, the hash bits extracted from a certain block are hidden into the corresponding residual data. The block hash hiding provides the side information to recover the lost packets at the expense of increasing little bit rate. On the other hand, FEC based on well-known error correction coding (ECC) is able to recover the lost packets only if the error rate is less than a pre-determined threshold. In order to fairly compare their performance, the increased bit rate has to be kept the same.

5.1. Recovery Capability of Our Method

Since our method uses embedded block hashes of residual data for motion estimation and compensation, we will discuss its error resilience depending on the characteristic of a block, i.e., inter-block or intra-block. For inter-blocks, let the hiding capacity be n_{mr}, which means the number of non-zero residual coefficients. It is said that bit collision happens when the detected bit is identical to the original hash bit. Let the bit collision probability be p_{cmr} and let the number of correctly detected bits be n_d bits. The probability of n_d hash bits detected to be identical to the original block hash bits is

$$p(n_d) = \binom{n_{mr}}{n_d} \cdot {p_{cmr}}^{n_d} \cdot (1 - p_{cmr})^{n_{mr} - n_d}, \tag{8}$$

which also indicates the probability of Hamming distance $(1 - \frac{n_d}{n_{mr}})$. If the Hamming distance is equal to zero, i.e., $n_d = n_{mr}$, the probability is computed as

$$p(n_d = n_{mr}) = \binom{n_{mr}}{n_{mr}} \cdot {p_{cmr}}^{n_{mr}} \cdot (1 - p_{cmr})^{n_{mr} - n_{mr}} = {p_{cmr}}^{n_{mr}}. \tag{9}$$

For intra-blocks, let the hiding capacity be n_{ma}, which means the number of non-zero residual coefficients. Let the bit collision probability be p_{cma}. If the original hash bits can be completely found from a pool of searched intra-blocks, i.e., $n_d = n_{ma}$, the probability is computed as

$$p(n_d = n_{ma}) = \binom{n_{ma}}{n_{ma}} \cdot {p_{cma}}^{n_{ma}} \cdot (1 - p_{cma})^{n_{ma} - n_{ma}} = {p_{cma}}^{n_{ma}}. \tag{10}$$

According to the characteristics of video codec, the number of non-zero residual errors in intra-blocks is much larger than that in inter-blocks, i.e., $n_{ma} > n_{mr}$. In addition, the probability of bit collision in inter-blocks is much larger than that in intra-blocks based on the inherent temporal nature of video sequences, i.e., $p_{cmr} > p_{cma}$. As a result, we can derive

$${p_{cmr}}^{n_{mr}} > {p_{cma}}^{n_{ma}} \Longleftrightarrow p(n_d = n_{mr}) > p(n_d = n_{ma}). \tag{11}$$

The result of Eq. (11) indicates that a larger set of candidate blocks is easier extracted in inter-blocks than in intra-blocks. This also implies that the desired target is hidden among a set of candidate blocks. When the set of candidate blocks is generated from a short block hash, they are not really similar and the final winner may not be the desired target. In other words, the recovery capability in intra-blocks is higher than in inter-blocks. In addition, when the true block can be searched, its carrier, i.e., the corresponding residual data, can be further added to recover the lost block in terms of PSNR improvement.

5.2. Recovery Analysis of FEC

The recovery capability of FEC is related to three parameters, (n, k, t), where the original data has k bits and is expanded to n bits, where $n - k = 2t$ bits are the overhead used for recovery. Due to consideration of packet loss in video transmission, the unit of FEC is set to "packet" instead of "bit" in the remainder of this paper. Let the additional overhead generated from FEC be the set O_t. Let the number of packets generated after FEC in a video frame and the packet loss rate be p_l. The number of packet errors n_{pe} is calculated to be $n \times p_l$. For a video frame, we describe three cases of packet recovery in the following.

Case 1: If $n_{pe} < t$ holds, then all lost packets can be completely recovered. The probability of case 1 can be calculated as

$$p(n_{pe} < t) = \sum_{i=0}^{t-1} \binom{n}{i} p_l^i (1 - p_l)^{n-i}. \tag{12}$$

Case 2: If $2t \geq n_{pe} \geq t \wedge$ erroneous packets all belong to the overhead O_t, then the original data still can be completely recovered. One can calculate the probability of case 2 as

$$p(n_{pe} \geq t, \text{erroenous packets} \in O_t) = \sum_{i=t}^{2t} \binom{2t}{i} p_l^i (1 - p_l)^{2t-i}. \tag{13}$$

Case 3: The packet loss is beyond the correction capability of FEC leading to k packets totally lost. Under this situation, the probability of occurrence is

$$p(\text{FEC fails}) = 1 - p(n_{pe} < t) - p(n_{pe} \geq t, \text{erroenous packets} \in O_t). \tag{14}$$

Under the first two cases, FEC can provide perfect recovery capability with the probability $p(n_{pe} < t) + p(n_{pe} \geq t, \text{erroenous packets} \in O_t)$. Since the packets are totally lost using FEC when case 3 is encountered, the decoder uses the standard error concealment (EC) mechanism to recover the lost packets. In this study, the conventional EC provided by the H.264 JM software is used. We assume that the potential blocks used for error concealment are error-free. When a P- or B-slice is lost, the blocks in the lost packet are recovered by using the motion vectors (MVs) of their neighboring inter-blocks based on the assumption that the lost block carries consistent MVs with the neighboring macroblock partitions (resulted from encoding).

Now, the recovery capability of FEC is analyzed as follows. Based on the characteristics of H.264, let the number of MVs of an original (error-free) inter-macroblock be n_{fr}. The correct MV detection is defined that the MV found by EC is identical to the original (error-free) macroblock. We further let the correct inter-macroblock detection probability be p_{cfr} and the number of correctly detected motion vectors be n_d. The probability of all the MVs of a inter-macroblock detected to be identical to the original MVs is

$$p(n_d = n_{fr}) = \binom{n_{fr}}{n_{fr}} \cdot {p_{cfr}}^{n_{fr}} \cdot (1 - p_{cfr})^{n_{fr} - n_{fr}} = {p_{cfr}}^{n_{fr}}. \tag{15}$$

Let the original (error-free) intra-macroblock has n_{fa} MVs. Let the correct intra-macroblock detection probability be p_{cfa} and the number of correctly detected motion vectors be n_d. The probability of all the MVs of a intra-macroblock detected to be as identical to the original MVs is

$$p(n_d = n_{fa}) = \binom{n_{fa}}{n_{fa}} \cdot {p_{cfa}}^{n_{fa}} \cdot (1 - p_{cfa})^{n_{fa} - n_{fa}} = {p_{cfa}}^{n_{fa}}. \tag{16}$$

For intra-blocks, it is reasonable to assume that no exact the same MVs exist in the neighboring blocks so that $n_{fa} = 0$ hold. Therefore,

$$p(n_d = n_{fa}) = {p_{cfa}}^{n_{fa}} = {p_{cfa}}^0 = 1, \tag{17}$$

and

$$1 > {p_{cfr}}^{n_{fr}} \Leftrightarrow p(n_d = n_{fa}) > p(n_d = n_{fr}). \tag{18}$$

No matter how small $p(n_d = n_{fr})$ is, it is easy for inter-blocks to find the correct MVs for recovery than intra-blocks. We can conclude that for FEC the recovery capability in inter-blocks is better than that in intra-blocks.

5.3. Our Method vs. FEC

In order to fairly compare the performance between our method and FEC, the bit-rate increase should be kept nearly the same. By calculating the probabilities shown in Eq . (12) and Eq.(13), respectively, it is found that the probability for FEC to achieve perfect error protection is sufficient small. For example, in a QCIF video sequence, the FEC parameter (n, k, t) is selected as $(15, 11, 2)$ with the bit-rate increase 36.36%. Let p_l be 0.5. The sum of probabilities for Case 1 and Case 2 is 0.0592, and the probability for Case 3 is 0.9408. This indicates that in most situations it is reasonable to compare error recovery capability only under Case 3. In the following, we conduct comparisons under Case 3.

When blocks with inter-coding are considered, $n_{mr} > n_{fr}$ and $p_{cmr} \approx p_{cfr}$ hold mostly. Thus, the relationship between Eq.(9) and Eq.(15) can be derived as

$${p_{cmr}}^{n_{mr}} < {p_{cfr}}^{n_{fr}}, \tag{19}$$

which implies that our method outperforms FEC.

Furthermore, for the situation of error propagation, if $n_{fr} \rightharpoonup 0$, then ${p_{cfr}}^{n_{fr}} \rightharpoonup 1$ for FEC. For our method, n_{mr} is still large enough due to the robustness of proposed media hashing. Therefore, ${p_{cmr}}^{n_{mr}} << {p_{cfr}}^{n_{fr}} = 1$ can be derived, which implies that our method outperforms FEC.

On the other hand, for consideration of intra-blocks, we can derive

$${p_{cma}}^{n_{ma}} < {p_{cfa}}^{n_{fa}} = 1 \tag{20}$$

based on Eq.(10) and Eq.(17), which indicates that our method outperforms FEC for blocks with intra-coding.

Based on the above derivations, we can conclude that our method can achieve stronger error resilience for both intra- and inter-coding blocks.

6. EXPERIMENTAL RESULTS

Some experiments were conducted to validate the capability of the proposed error-resilient video transmission and encryption method. The H.264 codec was adopted for video compression and 3-DES was adopted for video encryption due to its probable security. A number of standard video sequences (i.e., *Foreman*, *Table tennis*, and *Stefan*) of size 176×144 in the QCIF format were used for joint compression and encryption. The GOP structure of length 15 is defined to be "IPP...P," which contains 1 I frame and 14 P frames. After performing our method, the increased bit rates are within the range of $3\% \sim 8\%$, which are all smaller than 10% and are considered acceptable. Then, the compressed and encrypted video sequences were transmitted over error-prone networks. Here, the two-state Markov chain model[27] was used to simulate bursty packet loss[10] with parameters determined as follows: average burst length, $L_b = 4$, and packet loss rate, $p_l = \{0.01, 0.02, 0.05\}$. The performance of our method in error resilience is also compared with the fundamental mechanism, i.e., error concealment (EC), associated with H.264.

In the first experiment, we demonstrate the quality loss due to block hash embedding. Fig. 3 shows the comparison of PSNR values between an (encryption and embedding)-free decoded video, a decrypted and decoded video (hash embedded), and a decoded and encrypted video. We can observe that when an encrypted video is not decrypted, its quality is considerably inferior to the other two. In addition, the objective quality measured in

terms of PSNR is degraded due to hash embedding. However, if we consider the multi-function of embedded data for purposes of protection, authentication, and error resilience, we believe that it is worth paying such a cost. In addition, to check the visual effects of selective encryption, a pair of unencrypted and encrypted frames is shown in Fig. 2(a). As we can see from *Foreman* in Fig. 2(a) that much visual information (e.g., eyes, mouth, nose, and fingers) cannot be seen from the encrypted frame. This is because significant motions are contained in the *Foreman* sequence and the sign bits of motion vectors were selected for encryption. Although little background information can also be revealed, by considering the significant degraded quality this encrypted video without decryption loses its commercial value. On the other hand, Fig. 4 shows the visual quality comparison between a pair of cover and stego frames to illustrate the distortion introduced by hash embedding. As we can see from this typical example, the embedding effect is hard to be perceived subjectively. In this case, only 1.5 dB quality degradation can be measured objectively. However, the cost of increasing bit rate can be compensated by using the embedding data for joint error resilient video transmission, encryption, and copyright protection.

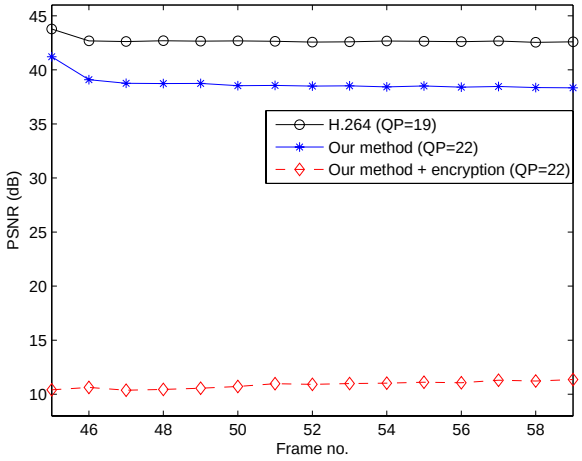


Figure 3. PSNR comparisons between a decoded *Table tennis* video without encryption and embedding, a (decoded+decrypted) video with embedding, and a (decoded+encrypted) video. The bit rate of the *Table-Tennis* sequence at encoder is set to be 120 Kbps.

(a) (b)

Figure 4. Visual quality comparison between (a) cover video frame and (b) stego video frame.

In the second experiment, we compare the ability of resilience to packet loss between our hash assisted macroblock searching method and the error concealment method. The final recovery results, obtained by averaging from 10 results obtained with different keys for loss simulations, are shown in Fig. 5. In the beginning, packet loss has not occurred and our method exhibits lower PSNRs than EC due to data hiding. Once packet losses happen, our method starts to outperform EC. The improved PSNRs range from 1.34 to 2.53 dB. Moreover, in Fig. 6

we show two pairs of video frames recovered based on our method and error concealment of H.264 for visual verification. It is obvious to find that the macroblocks recovered using our method show smooth and natural visual information, while discontinuous visual information can be observed using traditional error concealment.

From these results, some hints are obtained that are helpful to further improve error resilience. That is, we can approximately incorporate data hiding-based macroblock hash matching and error concealment for error resilience. According to our observations, block hash assisted motion estimation is rather helpful for macroblocks with large motions, while error concealment is very useful to recover lost packets of small motions without wasting bit rates for embedding.

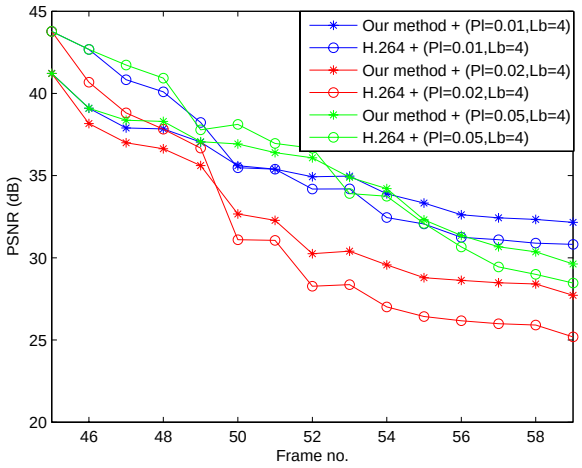


Figure 5. Comparison of resilience to packet loss between our hash assisted macroblock matching method and the error concealment technique of H.264 with respect to the *Table tennis* video. Thus, we got a hint from these results that integrating both hash matching and error concealment can yield the best error recovery.

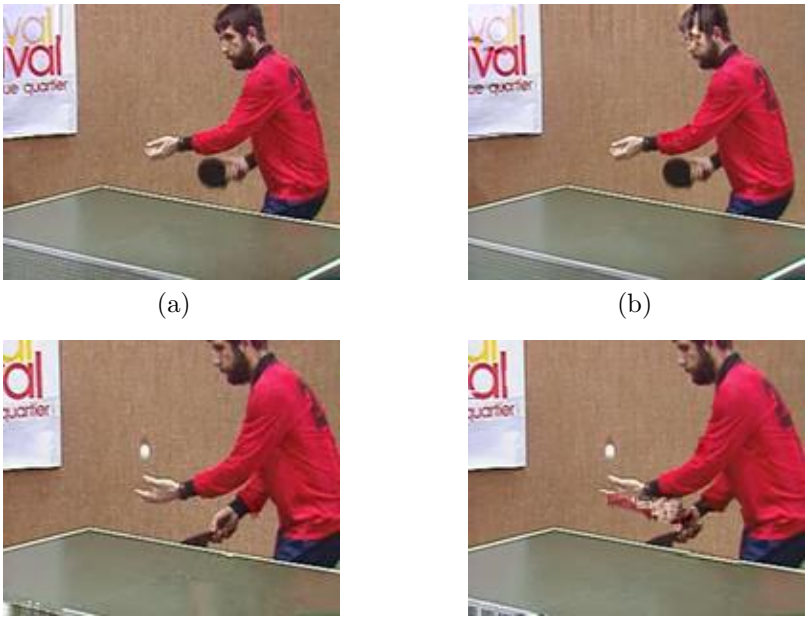
7. CONCLUSIONS

This study proposes a solution to robust video transmission and encryption with errors tolerated up to the degree of motion compensation. We investigate a macroblock hash embedding scheme at encoder and exploit the extracted hashes for macroblock matching at decoder to achieve estimation and compensation of motion vectors. Since the embedded macroblock hashes are available at the decoder, more information is helpful for resisting errors in a non-blind manner. In addition, our method can be used for error-resilient video transmission and encryption simultaneously. These constitute the major contribution of this study. On the other hand, it is worth mentioning that this framework is a kind of joint error-resilient video transmission/encryption and copyright protection.

Acknowledgment: This research was supported by the National Science Council under NSC grants 94-2213-E-001-027 and 94-2422-H-001-007.

REFERENCES

1. A. Aaron, S. Rane, and B. Girod, "Wyner-Ziv Video Coding with Hash-based Motion Compensation at the Receiver," *Proc. IEEE Int. Conf. on Image Processing*, 2004.
2. E. Ayanoglu, R. Pancha, A. R. Reibman, and S. Talwar, "Forward Error Control for MPEG-2 Video Transport in a Wireless ATM LAN," ACM/Baltzer Mobile Networks and Applications, Vol. 1, No. 3, pp. 245 258, 1996.
3. M. Chen, Y. He, and R. L. Landgelijk, "A Fragile Watermark Error Detection Scheme for Wireless Video Communications," *IEEE Trans. on Multimedia*, Vol. 7, No. 2, pp. 201-211, 2005.



(a) (b) (c) (d)

Figure 6. Visual quality comparison between (a) the video frame recovered using our method, $p_l = 0.05$; and (b) the video frame recovered using error concealment, $p_l = 0.05$; (c) the video frame recovered using our method, $p_l = 0.01$; and (d) the video frame recovered using error concealment, $p_l = 0.01$. We note from (b) and (d) that some poor recovered results are perceived obviously.

4. J. Fridrich, "Visual Hash for Oblivious Watermarking," *Proc. SPIE: Security and Watermarking of Multimedia Contents II*, 2000.
5. *IEEE Int. Conf. on Multimedia and Expo*: special session on Media Identification, June 2004.
6. *IEEE Int. Workshop on Multimedia Signal Processing (MMSP)*, special session on Media Recognition, 2002.
7. B. Li, E. Chang, and C. T. Wu, "DPF – A Perceptual Distance Function for Image Retrieval," *Proc. IEEE Int. Conf. on Image Processing*, 2002.
8. C. Y. Lin and S. F. Chang, "A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation," *IEEE Trans. on Circuits and Systems for Video Tech.*, Vol. 11, No. 2, pp. 153-168, 2001.
9. C. Y. Lin, D. Sow, and S. F. Chang, "Using Self-Authentication-and-Recovery Images for Error Concealment in Wireless Environments," *SPIE ITCom/OptiComm*, Denver, CO, Vol. 4518, 2001.
10. Y. J. Liang, J. G. Apostolopoulos, and B. Girod, "Analysis of Packet Loss for Compressed Video: Does Burst-length Matter?" *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing* , 2003
11. C. S. Lu, "Wireless Multimedia Error Resilience via A Data Hiding Technique," *Proc. 5th IEEE Int. Workshop on Multimedia Signal Processing*, US Virgin Islands, USA, 2002.
12. C. S. Lu and H. Y. Mark Liao, "Structural Digital Signature for Image Authentication: An Incidental Distortion Resistant Scheme," *IEEE Trans. on Multimedia*, Vol. 5, No. 2, pp. 161-173, 2003.
13. C. S. Lu and C. Y. Hsu, "Geometric Distortion-Resilient Image Hashing Scheme and Its Applications on Copy Detection and Authentication," *ACM Multimedia Systems Journal*, special issue on Multimedia and Security. (published online, Oct. 17, 2005)
14. I. Moccagatta, A. Soudagar, J. Liang, and H. Chen, "Error-Resilient Coding in JPEG-2000 and MPEG-4," *IEEE Journal on Selected Area in Communications*, Vol. 18, No. 6, pp. 899 914, 2000.
15. R. Puri, K. Ramchandran, K. W. Lee, and V. Bharghavan, "Forward Error Correction (FEC) Codes Based Multiple Description Coding for Internet Video Streaming and Multicast," *Signal Processing: Image Communication*, Vol. 16, pp. 745-762, May 2001.

16. K.C. Roh, K.D. Seoa, and J.K. Kim "Data Partitioning and Coding of DCT Coefficients Based on Requantization for Error-Resilient Transmission of Video," *Signal Processing: Image Communication*, Vol. 17, pp. 573 585, 2002.
17. A. Sehgal, A. Jagmohan, and N. Ahuja, "Wyner-Ziv Coding of Video: An Error-Resilient Compression Framework," *IEEE Trans. on Multimedia*, Vol. 6, No. 2, Apr. 2004.
18. T. Shanableh and M. Ghanbari, "Loss Concealment Using B-Pictures Motion Information," *IEEE Trans. on Multimedia*, Vol. 5, No. 2, 2003.
19. S. Shirani, F. Kossentini, and R. Ward, "A Concealment method for Video Communications in an Error-Prone Environment", IEEE Journal on Selected Areas in Communications, Vol. 18, NO. 6, pp. 1122 1128, June 2000.
20. J. Song and K. J. R. Liu, "A Data Embedded Video Coding Scheme for Error-Prone Channels," *IEEE Trans. on Multimedia*, Vol. 3, No. 4, 2001.
21. A. Tosun and W. C. Feng, "On Error Preserving Encryption Algorithms for Wireless Video Transmission," *Proc. ACM Conf. on Multimedia*, 2001.
22. S. Tsekeridou and I. Pitas, "MPEG-2 Error Concealment Based on Block-Matching Principles," *IEEE Trans. on Circuits and System for Video Technology*, Vol. 10, No. 4, pp. 646 658, June 2000.
23. Y. Wang, J. Ostermann, Y. Q. Zhang (Editors), Video Processing and Communications, Prentice Hall, 2002.
24. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. on Image Processing*, Vol. 13, No. 4, pp. 600-612, 2004.
25. J. Wen, M. Severa, W. Zeng, M. H. Luttrell, and W. Jin, "A Format-compliant Configurable Encryption Framework for Access Control of Video," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 12, No. 6, pp. 545-557, 2002.
26. X. Xu, S. Dexter, and A. M. Eskicioglu, "A Hybrid Scheme of Encryption and Watermarking," *IS&T/SPIE Symposium on Electronic Imaging 2004, Security, Steganography, and Watermarking of Multimedia Contents VI Conference*, San Jose, CA, January 19-22, 2004.
27. P. Yin, M. Wu, and B. Liu, "A Robust Error Resilient Approach for MPEG Video Transmission over Internet," *Proc. SPIE: Visual Communication and Image Processing*, 2002
28. W. Zeng and S. Lei, "Efficient Frequency Domain Selective Scrambling of Digital Video," *IEEE Trans. on Multimedia*, Vol. 5, No. 1, pp. 118-129, 2003.
29. W. Zeng, X. Zhuang, and J. Lan, "Network Friendly Media Security: Rationales, Solutions, and Open Issues," *Proc. IEEE Int. Conf. on Image Processing*, 2004.