中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-07-006

Video JET: Packet Loss-Resilient Video Joint Encryption and Transmission based on Media-Hash-Embedded Residual Data

Shih-Wei Sun, Jan-Ru Chen, Chun-Shien Lu, and Pao-Chi Chang

March 6, 2007 || Technical Report No. TR-IIS-07-006

http://www.iis.sinica.edu.tw/LIB/TechReport/tr2007/tr07.html

Video JET: Packet Loss-Resilient Video Joint Encryption and Transmission based on Media-Hash-Embedded Residual Data

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

[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan 115, ROC

[2]Dept. Electrical Engineering, National Central Univ., Chung-Li, Taiwan 320, ROC

[3]Dept. Communication Engineering, National Central Univ., Chung-Li, Taiwan 320, 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 creating packet loss. In order to achieve robust media encryption, the requirement of error resilience can be achieved with error-resilient media transmission. This study proposes a video joint encryption and transmission (video JET) scheme by exploiting media hash-embedded residual data to achieve motion estimation and compensation for recovering lost packets, while maintaining format compliance and cryptographic provable security. Interestingly, since video block hash preserves the condensed content to facilitate search of similar blocks, motion estimation is implicitly performed through robust media hash matching – which is the unique characteristic of our method. We analyze and compare the performance of resilience to (bursty) packet loss between the proposed method and forward error correction (FEC), which has been extensively employed to protect video packets over error-prone networks. The feasibility of our packet loss-resilient video JET approach is further demonstrated through experimental results.

keywords: (Selective) Encryption, Embedding, Error concealment, Error resilience, Media hashing, Motion estimation/compensation, Packet loss

*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. This paper was, in part, published in Proc. IS&T/SPIE: Visual Communications and Image Processing (EI127), Vol. 6077, San Jose, California, USA, 2006 [30].

I. Introduction

A. Background

Multimedia data transmitted over digital distribution networks can be secured by first line of defense technologies, i.e., cryptographic encryption. One main weakness of cryptographic encryption technologies is their 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 a cryptographic 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 [35] in that it provides passive protection only when copyright infringement needs to be solved. In view of these facts, this study focuses on error-resilient media encryption.

Multimedia encryption [34], [37], [38] needs to satisfy a number of requirements that, in some aspects, are conflicting, as briefly described in the following.

1. **Format compliance**: The encrypted video bitstream needs to be compatible with the syntax of coding standards so that a standard decoder can accept decrypted bitstream without needing specially designed decoders or additional information to enable decoding.
2. **Security**: Security is a cardinal requirement for encryption. Intuitively, cryptographic encryption is a good choice due to its provable security. However, considering 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.
3. **Complexity**: Modern applications need lower decryption complexity for use on low-powered consumer electronic devices. However, a higher secure encryption method needs complex operations.
4. **Robustness**: A practical media encryption scheme needs to be able to restrict error propagation and recover lost packets. In packet-switched networks, a single transmission error that cannot be recovered in the application layer 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.
5. **Coding efficiency**: The increased redundancy due to encryption should be limited in order not to considerably reduce coding efficiency.

Usually, the aforementioned requirements are conflicting, and proper trade-offs should be enforced depending on applications. In this paper, we focus on an issue of error resilience in video encryption that was seldom addressed in the literature. Due to the characteristic of packet-based networks in treating a packet with at least one erroneous bit unreliable, such a packet is dropped to create packet loss. Thus, error resilience of video encryption is regarded to be equivalent to error resilience of video transmission. In this paper, we investigate a novel media hashing-based error-

resilient video transmission scheme to meet the requirement of error-resilient video encryption, while preserving format compliance and provable security at the expense of an increase in computational complexity and a decrease in coding efficiency. In the following, related works about error-resilient video encryption and transmission will be reviewed.

B. Related Work about Error-Resilient Video Encryption

Tosun and Feng [31] proposed an error-resilient 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 a packet to create packet loss, it would be not enough to merely deal with bit errors.

Error-resilient mode in a video codec is also exploited to achieve error-resilient video encryption. Although the error-resilient mode of MPEG-4 associated with data partitioning is able to resist random packet loss, the solution of resistance to bursty packet loss in video encryption has not been studied in the literature. MPEG-4 video fine granularity scalability (FGS) was adopted for encryption in Zhu, *et al.* [39]. The access control at different levels is controlled based on encryption at different layers. The authors claim that the error resilience of their method was tested according to bit errors and packet losses in the enhancement layer under the assumption that the base layer is losslessly transmitted. When the base layer needs to be practically protected to satisfy error-free (transmission/encryption/etc), the expense of increasing bit rate is not discussed in their paper.

On the other hand, error-resilient video encryption can also be achieved with error-resilient video transmission, as described below.

C. Related Work about Error-Resilient Video Transmission

In the literature, the error control methods for video transmission can be roughly divided into five categories according to how and where the mechanism is operated.

1) Encoder-Level: 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 the decoded video can preserve a certain quality when errors are encountered. Among error resilience methods, inserting re-synchronization markers and data partitioning [20], [24] are able to efficiently separate errors from the video stream 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).

2) Transport-Level: The error control methods operated at the transport-level employ forward error correction coding (FEC) [2], [23] by adding redundant information coming from error-correction code (ECC) to protect the video stream. The penalties include: (i) the resultant video stream is not format-compliant, and (ii) the coding efficiency is sacrificed.

3) Decoder-Level: The error control methods that utilize some post-processing mechanisms on the decoder side to accomplish error recovery without needing to transfer redundant information are called error concealment (EC) with zero-redundancy [28], [32]. 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.

4) Data Hiding-based: Recently, data hiding technologies [3], [13], [15], [26], [29] have also been employed for correcting transmission errors. Song and Liu [29] 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 one slice and bursty errors are not considered. Shanableh and Ghanbari [26] 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. Chen, *et al.* [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, can still be detected.

5) Side Information-based: Recently, the side information concept [1], [25] 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 the decoder in estimating motion vectors. This work describes distributed source coding for sensor networks. Sehgal, *et al.* [25] exploited 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 was sacrificed.

D. 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 the strongest protections. For example, in H.264/AVC [5], [6] a normal slice can be partitioned into three parts, i.e., data partition (DP) A/B/C, each of which is encapsulated into an individual NAL packet. DP A comprises motion vectors, quantization parameters, and header information; DP B includes the transformed coefficients of intra-coded macroblocks and the corresponding block patterns (each indicating the relationship between a macroblock and its partitioned blocks); and DP C contains the residual data of inter-coded blocks and the corresponding block patterns. Traditionally, DP A is regarded to be the most important part, while DP C is the least important from the viewpoint of visual quality. In addition, when packet loss is detected, a conventional error concealment (EC) mechanism that can be adopted at the decoder for loss recovery is recommended. Among them, when the available partitions are B and C (i.e., motion vectors are lost), the recommended action is to drop data partitions B and C, and use the motion vectors of the neighboring lossless MBs around the lost MB for recovery.

However, this research shows that it is not feasible to protect motion vectors alone, because the recovered quality is still not good enough, in particular for video sequences with large motions. The residual data (belongs to DP C), even classified as low-priority streams in data partitioning, is indispensable to repair the high-frequency information so that the degraded quality can be further recovered. In addition, one finds that when motion vectors are lost, error concealment is a mechanism commonly used for loss recovery, which does not work well for video content with motions that cannot be neglected. In other words, when motion vectors are lost, there is not an efficient way to recover them. This issue has not been efficiently solved in the literature. In view of these facts, a new robust video joint encryption and transmission (video JET) method based on media-hash-embedded residual data is presented in this paper. The block hashes are extracted in a way similar to motion estimation and embedded into the residual data at the encoder, and, later, the hashes are extracted at the decoder for motion compensation. Specifically, when the packet used for motion estimation in DP A is lost, its block hash is used for block matching, and the best match is used to replace the lost data for motion compensation. In particular, if the recovered block is further combined with the residual data (in DP C), then the recovered data can be more similar to the lost data.

In this paper, we target the goal of robust video joint encryption and transmission. For commercial applications, high quality (bit-rate) video is acceptable for users who are authentic and own the authorized decryption keys. In addition, the high quality video is worthy of being protected by encryption technology. Therefore, despite the different priorities of data partitions A, B, and C, they all play the key roles in our study here.

The remainder of this paper is organized as follows. In the next section, why the error resilience problem in media encryption can be equivalent to that in media transmission is described. In Sec. III, the general principle of media hashing is discussed, and the proposed video hashing technique for motion estimation and compensation is described. In Sec. IV, the paper describes the proposed error-resilient video encryption and transmission scheme. In Sec. V, the analysis and comparison of error recovery between this method and forward error correction is described. Experimental results are given in Sec. VI, and conclusions are drawn in Sec. VII.

II. Problem Statement: Error-Resilient Video Joint Encryption and Transmission

A common way to achieve robustness of media encryption is to turn on the error concealment mode in a specific coding standard. 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. Therefore, one should consider error-resilient video encryption to be equivalent to error-resilient video transmission plus encryption. In other words, repairing the quality degradation caused by packet loss is the key achieving error-resilient video joint encryption and transmission.

In this paper, a new error-resilient video joint encryption and transmission scheme is presented, which exploits the embedded block hashes for macroblock matching at the decoder to search for the best target (motion estimation) and use it to recover (motion compensation) the lost packets. Since robust media hashing is considered (see Sec. III and [17]), partial content matching still permits one to find the desired target without certainly affecting the

capability of motion estimation and compensation. In contrast to motion vector embedding [29] 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. The above reasons explain the reason why block hash is embedded at the encoder for motion estimation and compensation at the decoder in this study.

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

III. Media Hash

A. General Principle

The media hash [4], [17], also known as the "digital signature" [12], [16] or "media fingerprint" [7], [18], has been widely used in many applications, including content authentication, copy detection, media recognition, error resilience, and distributed video coding [8], [9]. 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 [11], [33], a block hash-based matching metric for macroblock searching will be proposed in the next section. 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, $\mathcal{Z}$ is 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.

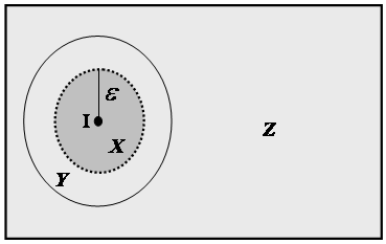


Fig. 1. **The Image Space. I is an element in the image space. $\mathcal{X}$ denotes the set of images modified from I that are still perceptually similar to I. $\mathcal{Y}$ denotes the set of images modified from I that are perceptually different from I. $\mathcal{Z}$ is the set of images that are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

Fig. 2 illustrates an example of extraction and comparison of video frame hashes that conforms to the media hash principle described above.

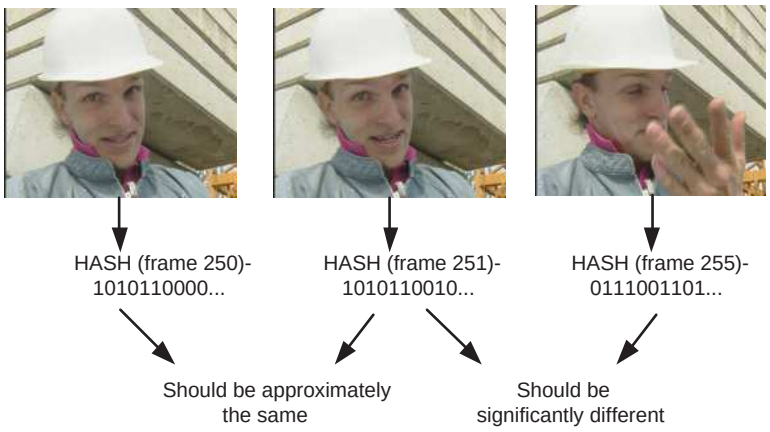


Fig. 2. **Illustration of video frame hash extraction and matching.**

B. Proposed Video Block Hashing

In this section, the proposed video block hash extraction technique is described for partitioned blocks that are generated from macroblock partitioning in H.264. For a partitioned block, it is further divided into several smaller blocks of size 4×4, which is the minimum block size in H.264 video coding. The hierarchical relationship among various types of "blocks" is shown in Fig. 3. Let $\mathbf{MB_{b,m,s,f}}$ denote the b-th partitioned block of the m-th macroblock

from the s-th slice of the f-th frame. In order to simplify notation, hereafter, this paper will use $\mathbf{MB_b}$ to represent $\mathbf{MB_{b,m,s,f}}$. For each 4×4 block, local DCT is performed. Let n_b denote the number of 4×4 blocks in $\mathbf{MB_b}$ and let $DCT_b^q(d)$ denote the d-th AC component of the q-th 4×4 block of $\mathbf{MB_b}$. To facilitate our discussion, the adopted notations are shown in Table I.

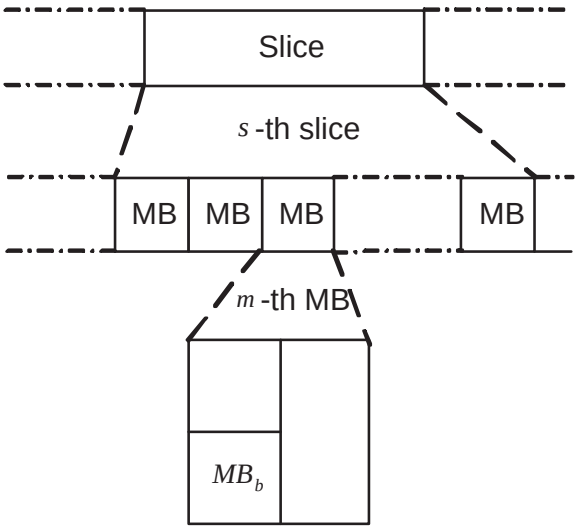


Fig. 3. **Hierarchical relationship among various types of blocks (MB: macroblock; $\mathrm{MB_b}$: partitioned block) and slice.**

Now, one can define the video block hash sequence, $\mathbf{MH_b}$, of $\mathbf{MB_b}$. Specifically, the video block hash proposed here consists of an edge hash, a sign hash, and a magnitude hash, as shown in Fig. 4. Among them, edge hash, $\mathbf{E_b}$, of $\mathbf{MB_b}$ is used to specify whether a partitioned block $\mathbf{MB_b}$ is smooth or not (edge-like); sign hash, $\mathbf{S_b}$, is used to represent the signs of AC coefficients in $\mathbf{MB_b}$; and magnitude hash, $\mathbf{M_b}$, specifies the magnitude differences for AC coefficient pairs in $\mathbf{MB_b}$.

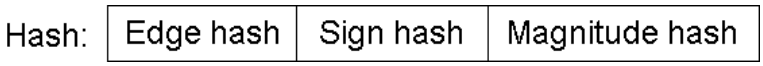


Fig. 4. **Format of a video block hash.**

In this method, the first n_s ($1 \leq n_s \leq 16$) subbands in the 4×4 DCT domain are considered for robust hash generation. For $\mathbf{E_b}$, its element is defined as:

$$E_b(i) = \begin{cases} 0, & \text{if } edge(DCT_b^q(d)) \text{ is edge-like;} \\ 1, & \text{if } edge(DCT_b^q(d)) \text{ is smooth,} \end{cases} \quad (1)$$

where $1 \leq q \leq n_b$, $1 \leq d \leq n_s$, and $edge(DCT_b^q(d))$ represents the edge detection result, which contains either a horizontal or vertical edge feature in a 4×4 block [10]. For $\mathbf{S_b}$, the sign hash bit is defined as:

$$S_b(i) = \begin{cases} 0, & \text{if } sgn(DCT_b^q(d)) \geq 0; \\ 1, & \text{if } sgn(DCT_b^q(d)) < 0, \end{cases} \quad (2)$$

TABLE I

NOTATIONS IN SEC.III-B.

$\mathbf{MB_b}$	the b-th partitioned block in a macroblock
m	the index of macroblock
s	the index of slice
f	the index of frame
n_b	the number of 4×4 blocks in a macroblock
n_s	the number of subbands
q	the index of 4×4 block
d	the index of the DCT AC coefficient in a $\mathbf{MB_b}$
i	the index of edge, sign, or magnitude hash bit
$\mathbf{DCT_b}$	the DCT coefficients of $\mathbf{MB_b}$
$\mathbf{E_b}$	the edge hash of $\mathbf{MB_b}$
$\mathbf{S_b}$	the sign hash of $\mathbf{MB_b}$
$\mathbf{M_b}$	the magnitude hash of $\mathbf{MB_b}$
$\mathbf{MR_b}$	the residual data of $\mathbf{MB_b}$
$\mathbf{RMB_b}$	the partitioned macroblock in the reference frame most similar to $\mathbf{MB_b}$
$\mathbf{MH_b}$	the media hash of $\mathbf{RMB_b}$

where $sgn(\cdot)$ denotes the sign of its argument. The element of $\mathbf{M_b}$ is defined as:

$$M_b(i) = \begin{cases} 0, & \text{if } sgn(|DCT_b^q(d)| - |DCT_b^{(q+1) \bmod n_b}(d)|) \geq 0; \\ 1, & \text{if } sgn(|DCT_b^q(d)| - |DCT_b^{(q+1) \bmod n_b}(d)|) < 0. \end{cases} \quad (3)$$

Finally, the video block hash sequence of a partitioned block is produced by concatenating the edge hash, sign hash, and magnitude hash together, as shown in Fig. 4. The block-based video hash constructed above is a binary sequence.

Let $MH_{b,m,s,f}(i)$ denote the i-th hash bit of the b-th partitioned block of the m-th macroblock from the s-th slice of the f-th frame. Similarly, $MH_b(i)$ is used to represent $MH_{b,m,s,f}(i)$ to simplify notation. For a partitioned block, $\mathbf{MB_b}$, its hash, $\mathbf{MH_b}$, will be embedded into the corresponding residual data, $\mathbf{MR_b}$ of $\mathbf{MB_b}$.

However, for this media hash-based error-resilient video transmission scheme, we have to stress that the block hash, $\mathbf{MH_b}$, is not extracted from the partitioned block, $\mathbf{MB_b}$, in this paper. On the contrary, the partitioned block, $\mathbf{RMB_b}$, in the reference frame that is most similar to the partitioned block, $\mathbf{MB_b}$, (as depicted in Fig. 5) is searched by means of motion estimation. Then, the hash of the found reference partitioned block, $\mathbf{RMB_b}$, is extracted as $\mathbf{MH_b}$, which will be embedded into $\mathbf{MR_b}$. The key is that when $\mathbf{MB_b}$ is lost, one can guarantee the ability to find the best match from a "reference frame" for recovery purpose by means of hash matching-based motion compensation. More specifically, if $\mathbf{MH_b}$ is extracted from $\mathbf{MB_b}$ and embedded into $\mathbf{MR_b}$, then when $\mathbf{MB_b}$ is lost, it is difficult to find a "best match" to the extracted block hash (due to loss of $\mathbf{MB_b}$) that is guaranteed to be similar to $\mathbf{MB_b}$. On the contrary, if $\mathbf{MH_b}$ is extracted from $\mathbf{RMB_b}$ in the reference frame that is most

similar to the lost block $\mathbf{MB_b}$, then finding a block similar to $\mathbf{MB_b}$ can be better guaranteed as long as the corresponding block in the reference frame is not lost either. Fig. 6 illustrates an example of the above strategy. In addition, when the best matched partitioned block is found, the residual data of the non-lost packet in DP C can be further added to better reconstruct the partitioned block because the effect of error propagation in the temporal domain can be reduced.

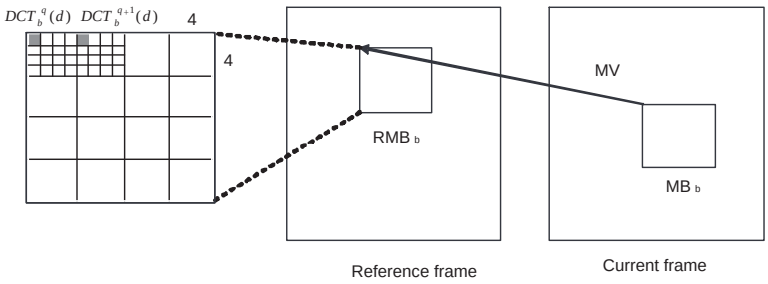


Fig. 5. **Search of partitioned blocks in reference frames for robust media hash generation.**

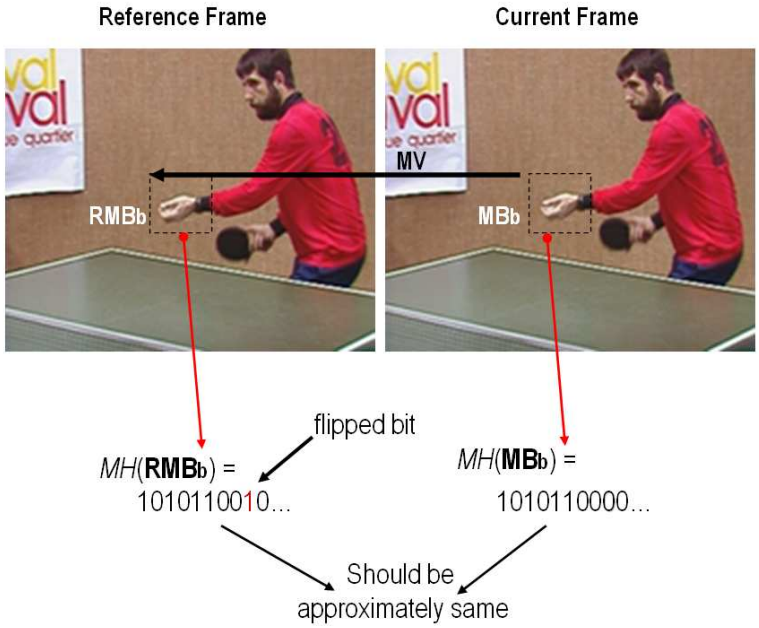


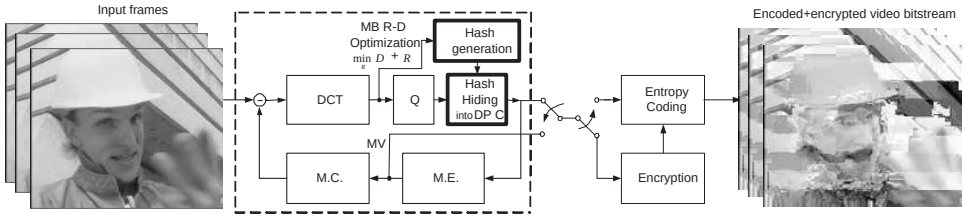
Fig. 6. **The relationship between $\mathbf{MB_b}$ and $\mathbf{RMB_b}$, and their corresponding media hash bits. If $\mathbf{MB_b}$ is lost, then the hash of $\mathbf{RMB_b}$ embedded in the residual data of $\mathbf{MB_b}$ can be extracted to find $\mathbf{RMB_b}$ for approximate recovery of $\mathbf{MB_b}$ because $\mathbf{MB_b}$ and $\mathbf{RMB_b}$ look perceptually similar.**

IV. Proposed Error-Resilient Video Joint Encryption and Transmission Method

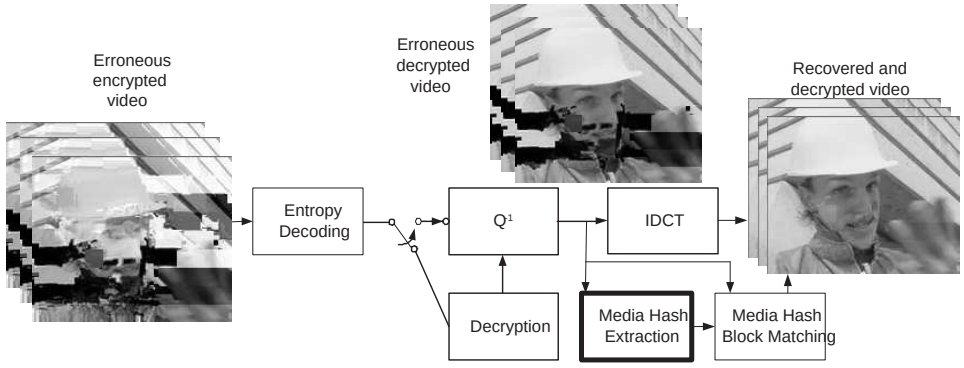
The block diagrams of the proposed video JET method at the encoder and decoder sides are, respectively, depicted in Figs. 7(a) and (b). The new coding standard H.264 is adopted with its data partitioning mode turned on. As shown in Fig. 7(a), media hashes are generated based on the macroblock partitioned blocks in the DCT

domain (Sec. III-B) and embedded via rate-distortion optimization [21] into the residual data in DP C. After hash embedding is performed, some important data is selected for light-weight encryption using 3-DES. In order to achieve format compliance, encryption is conducted before entropy coding. On the decoder side, the decoding and decryption processes shown in Fig. 7(b) are the inverse operations of the encoder.

This method possesses many advantages. First, the bit-rate after encryption is performed is not increased[‡] since the data selected for encryption does not affect coding efficiency at all. Second, encryption is conducted based on a video slice (encapsulated as a packet), thus, transmission error propagation among different video packets are avoided. Third, the encryption format is compatible to video codecs in that the encrypted video can be decoded without needing additional information. Fourth, the achievable security is guaranteed up to the provable security provided by 3-DES. Finally, a new (bursty) packet loss-resilient technique is proposed for video joint encryption and transmission. In the following, each component will be specifically described.



(a) Encoder



(b) Decoder

Fig. 7. **Block diagrams of the proposed error-resilient video JET method: (a) Encoder; (b) Decoder.**

A. Basic Structure of H.264 and Light-Weight Encryption

The basic encoding/decoding unit in H.264 is a slice, which is composed of several macroblocks. Therefore, the basic unit for encryption should be carefully selected by considering different video encoding units, including packets, slices, data partitions, etc. Basically, the encryption unit can have one of three different types, as depicted

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

in Fig. 8. In Fig. 8(a), the important parts of the whole video packets are encrypted into only one unit. In this situation, even one bit error or one packet loss will lead to totally incorrect decryption. In Fig. 8(b), video packets and encryption units are de-synchronized, i.e., an encryption unit is not composed of exactly integer number of packets. Once a video packet is lost, not only is the lost packet erroneously decrypted, but also the corresponding neighboring video packet is erroneously propagated. For example, whenever encryption unit 1, containing packet 1 and partial packet 2, is lost, the remaining packet 2 in encryption unit 2 cannot be correctly decrypted. In Fig. 8(c), the size of a video packet is exactly equal to that of an encryption unit, which implies that the number of the encryption units is equal to the number of the video packets. When video packets are lost, either an error concealment or error resilience technique can be employed to recover the lost packets. In addition, the encrypted bits in the lost video packets won't be propagated thereby affecting the decryption of other video packets. Therefore, the encryption unit shown in Fig. 8(c) is suitable for packet video joint encryption and transmission, and is adopted here.

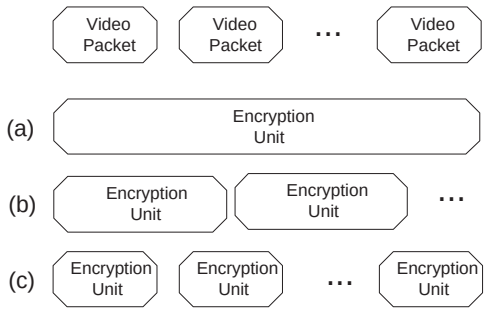


Fig. 8. **Different types of encryption unit: (a) all-in-one encryption; (b) desynchronization between video packets and encryption units; and (c) synchronization between video packets and encryption units.**

The goal of the proposed method is for commercial applications. Therefore, the visual quality of an encrypted video has been partially destroyed, the intended receivers need to pay to decrypt the light-weight encrypted video stream. As for applications such as military or confidential video conferencing that require higher security, the proposed light-weight video encryption method is not suitable because even slight but un-clear information is not allowed to be revealed.

Based on the above concerns, only the sign bits of DC components in I-frames and the sign bits of MVs[§] in P-frames are selected for encryption. Thus, the proposed method is a kind of selective encryption [27] in the sense that only important information is encrypted while saving computational overhead. With the selected data, the well-known cryptographic encryption mechanism 3-DES is applied to perform encryption.

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

B. Media Hash Hiding at Encoder for Error Resilience

The media hashes of reference partitioned blocks, $\mathbf{RMB_b}$'s, are extracted in the DCT domain using the technique described in Sec. III-B and are embedded into the residual data of their corresponding $\mathbf{MB_b}$s. The residual data, selected as the hiding carrier, is the non-zero AC coefficients. During the data hiding process, the rate-distortion optimization mechanism is employed to achieve the optimized mode selection and guaranteed coding efficiency, as depicted in Fig. 7(a).

Here, a simple odd-even data hiding technique is applied for hash bit embedding. Let $\mathbf{MR_b}$ be the residual data corresponding to $\mathbf{MB_b}$. Let $MR_b(i)$ and $MR_b^h(i)$ be, respectively, the i-th residual data (i.e., DCT coefficients) in $\mathbf{MR_b}$ before and after embedding. They are related as

$$MR_b^h(i) \;=\; \begin{cases} sgn(MR_b(i))(|MR_b(i)|+1), \text{ if } MR_b(i)\ mod\ 2 \neq MH_b(i); \\ MR_b(i), \text{ otherwise.} \end{cases} \tag{4}$$

The principle of this embedding is to enlarge the magnitude of the DCT coefficient with the aim that only the non-zero AC coefficients are selected as the hiding carrier, which is the prior information for the purpose of hash embedding and extraction.

If the number of non-zero DCT coefficients in $\mathbf{MR_b}$ is smaller than the size of the hash $\mathbf{MH_b}$, then not all hash bits can be embedded. In this study, the embedding capacity with respect to $\mathbf{MB_b}$ is, thus, calculated as the minimum value between the number of non-zero coefficients in $\mathbf{MR_b}$ and the size of the hash $\mathbf{MH_b}$. If the capacity is not large enough for embedding all of the hash bits of a macroblock, this proposed method cannot provide the best effort for error resilience. Hence, the high bit-rate video stream, containing more non-zero residual data, is the target that this method is better to be applied to. Recall that for commercial applications that are the focus of this paper, high quality (bit-rate) video is acceptable for users who are authentic and own the authorized decryption keys. Thus, high quality videos are worthy of being protected by encryption technology.

1) Analysis of Distortions Caused by Hash Embedding: As indicated in Eq. (4), hash bits are embedded into the residual data. Let Δ_s be the uniform quantization parameter for a DCT subband s ($1 \leq s \leq n_s$). It is reasonable to assume that the expectation of the quantization distortions, E_Q^s, is uniformly distributed over the interval Δ_s. One has

$$E_Q^s = \frac{1}{\Delta_s}\int_{\frac{-\Delta_s}{2}}^{\frac{\Delta_s}{2}} x^2 dx = \frac{\Delta_s^2}{12}, \tag{5}$$

which is well-known in the video codec community. In addition, when embedding is further performed, the distortion introduced for residual data in the b-th partitioned block, i.e., $MR_b^h(i) - MR_b(i)$, is also uniformly distributed over the interval Δ_s. According to the embedding rule in Eq. (4), one can derive

$$E(MR_b^h(i) - MR_b(i)) = \frac{1}{2}\frac{\Delta_s^2}{12} + \frac{1}{2}\frac{1}{\Delta_s}\int_{\frac{-3\Delta_s}{2}}^{\frac{-\Delta_s}{2}} x^2 dx = \frac{7\Delta_s^2}{12}. \tag{6}$$

By comparing Eqs. (5) and (6), one can find that the distortions increased due to embedding in a DCT subband s is $\frac{\Delta_s^2}{2}$.

C. Media Hash Extraction at Decoder

When packet loss occurs during encrypted video transmission, the embedded block hashes are extracted for block hash matching-based motion compensation to achieve error recovery, as shown in Fig. 7(b). Assume that the MVs of $\mathbf{MB_b}$ are lost but its residual data is still error free due to data partitioning employed in this method. The media hash bits can be extracted from the residual data of a slice as

$$MH_b(i) = MR_b(i) \; mod \; 2. \tag{7}$$

D. Two-Stage Hash Matching at Decoder

To facilitate discussion here, the adopted notations are shown in Table II. When the reference partitioned block hash $\mathbf{MH_b}$ is extracted, the block hash matching process is performed to search for the most similar block and use it to recover the lost block. In some applications, if high-complexity is permitted, then a two-stage hash matching mechanism proposed below is employed to improve accuracy of hash matching. In this study, so-called two-stage hash matching contains the first-stage matching process, which is a forward hash matching process searching similar blocks from the reference frames ahead of the current frame, and the second-stage matching process, which is a backward hash matching process searching the frames subsequent to the current frame. Two-stage hash matching can overcome a disadvantage due to possibly limited hiding capacity provided in the residual data, in that a one-stage (i.e., forward) hash matching process may find a large number of candidates, which make the final selection of the target block difficult. Fig. 9 depicts a scenario of two-stage hash matching.

TABLE II

NOTATIONS IN SEC.IV-D.

$\mathbf{MB_{b_p}}$	a reference block in a reference frame ahead of the current frame
$\mathbf{MH_{b_p}}$	the media hash of $\mathbf{MB_{b_p}}$
$\mathbf{SMB_{b_p^1}}$	the set of candidates found in the first-stage hash matching
$\mathbf{MB_{b_p^1}}$	an element of $\mathbf{SMB_{b_p^1}}$
$\mathbf{MH_{b_p^1}}$	the media hash of $\mathbf{MB_{b_p^1}}$
$\mathbf{MB_{b_{n_i}}}$	a block in the backward referencing frames
$\mathbf{MH_{b_{n_i}}}$	the media hash of $\mathbf{MB_{b_{ni}}}$
$\mathbf{MH_b^c}$	the concatenated media-hash used in the second-stage matching
$\mathbf{MH_{b_p^1}^c}$	the vector of media hash bits with common positions between $\mathbf{MH_{b_p^1}}$ and $\mathbf{MH_b^c}$
$\mathbf{SMB_{b_p^2}}$	the set of candidates found in the second-stage hash matching

More specifically, in the first-stage hash matching process, the hash sequences extracted from the blocks of the search windows in the reference frames are utilized to compare with that of the lost block stored in the residual data. In Fig. 9, $\mathbf{MB_{b_p}}$ indicates a reference block in a reference frame ahead of the current frame. Let $\mathbf{MH_{b_p}}$ be the hash extracted from $\mathbf{MB_{b_p}}$. If the bit error rate (BER) resulted from hash matching, $BER(\mathbf{MH_b}, \mathbf{MH_{b_p}})$, is less than a threhold Th_b, then all such blocks are collected as a set, $\mathbf{SMB_{b_p^1}} = \{\mathbf{MB_{b_p}} | BER(\mathbf{MH_b}, \mathbf{MH_{b_p}}) < Th_b\}$,

which forms a candidate list. In $\mathbf{SMB_{b_p^1}}$, its elements $\mathbf{MB_{b_p^1}}$'s will be further sieved out via the second-stage matching process.

In the second-stage hash matching process, the search range in the temporal domain is from the current frame to the next few frames with the aim of exploiting the backward referencing characteristic to further determine the best match from $\mathbf{SMB_{b_p^1}}$ for error recovery. Given the reference block $\mathbf{MB_{b_{n_i}}}$ in the backward referencing frames, as shown in Fig. 9, the size of the overlap area represents its contribution in describing partial similarity between $\mathbf{MB_b}$ and $\mathbf{MB_{b_{n_i}}}$. Thus, backward block hash matching can help to further sieve out the blocks from $\mathbf{SMB_{b_p^1}}$ that are dissimilar to $\mathbf{MB_b}$ and keep the most similar blocks for final selection.

Based on the above discussions, a practical implementation of second-stage hash matching is described as follows. First, a so-called concatenated media hash is represented as:

$$\mathbf{MH_b^c} = \bigcup_{\mathbf{i}} (\mathbf{MH_{b_{n_i}}} | \mathbf{MB_b}), \tag{8}$$

where $\mathbf{MH_{b_{n_i}}}$ represents the media hash of $\mathbf{MB_{b_{n_i}}}$ and $(\mathbf{MH_{b_{n_i}}}|\mathbf{MB_b})$ represents the common hash bits in the overlap area between $\mathbf{MB_b}$ and $\mathbf{MB_{b_{n_i}}}$. In this way, both $\mathbf{MB_b}$ and $\mathbf{MB_{b_{n_i}}}$ (for all i) have the hash bits in common, which will be collected, as indicated in Eq. (8). The overlap area in middle picture of Fig. 9 indicates the positions of $\mathbf{MH_b^c}$. Similarly, let $\mathbf{MH_{b_p^1}^c}$ represent the vector of media hash bits with common positions between $\mathbf{MH_{b_p^1}}$ (an element of $\mathbf{SMB_{b_p^1}}$) and $\mathbf{MH_b^c}$. Let BER_{min} be the minimum value of $BER(\mathbf{MH_b^c}, \mathbf{MH_{b_p^1}^c})$ for all $\mathbf{MH_{b_p^1}^c}$. The candidate blocks in $\mathbf{SMB_{b_p^1}}$ that are considered to be very similar to the lost block $\mathbf{MB_b}$ are determined by:

$$\mathbf{SMB_{b_p^2}} = \{\mathbf{MB_{b_p^1}} | BER(\mathbf{MH_b^c}, \mathbf{MH_{b_p^1}^c}) = BER_{min},\ \forall \mathbf{MH_{b_p^1}^c}\}. \tag{9}$$

Since the number of elements in $\mathbf{SMB_{b_p^2}}$ may be larger than 1, an edge detection [22] based side-match strategy is further exploited to choose the final target for recovery. If the lost $\mathbf{MB_b}$ is judged to be edge-like, which is described by the edge hash in $\mathbf{MH_b}$, then the edge orientation calculated from each candidate block in $\mathbf{SMB_{b_p^2}}$ is used for matching with the boundary blocks neighboring to the lost block and the one with the best match is chosen as the final target. If the lost block is determined to be smooth, a conventional side-match process (depending only on spatial interpolation) is applied for choosing the final target. Fig. 10 illustrates an example of candidate list obtained after performing two-stage hash matching. We can observe that the candidate blocks are highly correlated to the original content of the corrupted block.

After describing the two-stage hash matching process, one must now define the "search range," which concerns the trade-off between error resilience and complexity. Let the lost macroblock $\mathbf{MB_{m,s,f}}$ be located at the m-th macroblock of the s-th slice in the f-frame. Here, the search range is defined to be the set, Ψ, of positions covered by the set, Φ, of macroblocks neighboring to $\mathbf{MB_{m,s,f}}$. Both Φ and Ψ are expressed as:

$$\begin{aligned} \Phi &= \{\mathbf{MB_{m_s,s_s,f_s}} | |m_s - m| \leq m_w, |s_s - s| \leq s_w, |f_s - f| \leq f_w\}, \\ \Psi &= \{(x,y) | (x,y) \text{ is a pixel position of the macroblock belonging to } \Phi\}, \end{aligned} \tag{10}$$

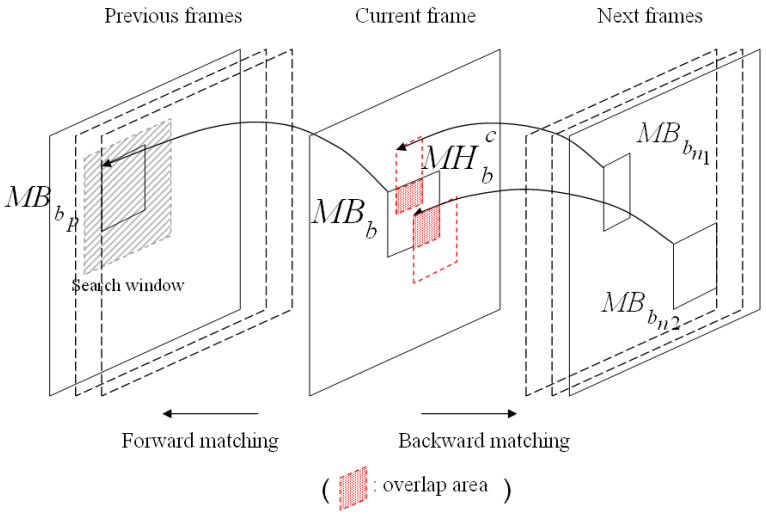


Fig. 9. **Illustration of two-stage hash matching.**

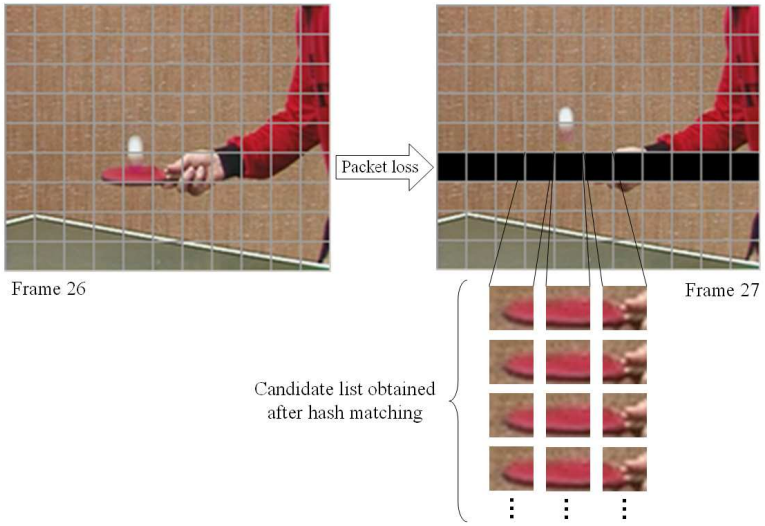


Fig. 10. An example of candidate list obtained after hash matching.

where m_w, s_w, and f_w define the search range with respect to m, s, and f. In this work, the size of the search range is simply defined to be the one used for motion estimation at the encoder. It should be noted that larger search range is beneficial to find the desired target with a longer motion vector corresponding to the lost macroblock at the expense of spending more computational time.

On the other hand, in order to obtain accurate motions, we must use the positions in Ψ as a starting pixel of a block for hash matching. We further let Φ^{ψ} denote the blocks whose starting pixel belongs to Ψ. Thus, we have $\Phi^{\psi} \supset \Phi$. With the search range, the block hash is extracted from each block belonging to Φ^{ψ} and compared with the extracted hash $\mathbf{MH_b}$ by calculating their Hamming distance, as stated above.

The starting point for block matching is based on the position of the lost partitioned block because the assumption that even when MVs in a packet are lost, the corresponding residual data may still be free from loss due to data

partitioning is employed in our method. Once residual data is lost, baseline error concealment will finally be applied.

V. Analysis of Error Recovery between Our Method and Forward Error Correction

In this section, comparison of error resilience between the proposed method (video JET) and forward error correction (FEC) is analyzed because FEC has been extensively employed to protect the video packets transmitted over error-prone networks. In the proposed method, hash bits extracted from a macroblock are hidden in the corresponding residual data. The block hash hiding provides the side information to recover the lost packets at the expense of increasing 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.

A. Error Resilience of Video JET

Let n_{mh} be the number of partitioned blocks in a macroblock. It is said that "media hash collision" happens when the BER calculated between the media hash extracted from a candidate block and the embedded media hash is less than a threshold Th_b. Let the media hash collision probability be p_{cmh}. Thus, the probability of n_d partitioned block hash detected to be identical to the original partitioned block hash in a macroblock is assumed to obey binomial distribution:

$$p(n_d) = \binom{n_{mh}}{n_d} \cdot {p_{cmh}}^{n_d} \cdot (1 - p_{cmh})^{n_{mh} - n_d}. \tag{11}$$

In addition, if all partitioned blocks are detected to be identical to the original partitioned blocks, the probability is:

$$p(n_d = n_{mh}) = {p_{cmh}}^{n_{mh}}. \tag{12}$$

B. Error Resilience 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 memory overhead used for error resilience. Due to consideration of packet loss in video transmission, the correction unit of FEC is set to "packet" instead of "bit" in the remainder of this paper. Let the additional memory overhead generated from FEC be the set O_t. Let the number of packets generated after FEC in a video frame be n and let the packet loss rate be p_l. The number of lost packets, n_{pe}, is calculated to be $n \times p_l$. For a video frame, FEC-based packet recovery can be operated in three cases described below.

Case 1: If $n_{pe} < t$ holds, then all lost packets can be completely recovered. Under this situation, the recovery probability corresponding to case 1 can be calculated as

$$p(n_{pe} < t) = \sum_{i=0}^{t-1} \binom{n}{i} p_l^i (1 - p_l)^{n-i}. \tag{13}$$

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

$$p(2t > n_{pe} \geq t \wedge \text{lost packets} \in O_t) = \sum_{i=t}^{2t} \binom{2t}{i} p_l^i (1-p_l)^{2t-i}. \quad (14)$$

Case 3: Packet loss occurs beyond case 1 and case 2 so that k packets are totally lost. Under this situation, the probability is calculated as

$$p(n_{pe} > 2t \text{or FEC fails}) = 1 - p(n_{pe} < t) - p(2t \geq n_{pe} \geq t \wedge \text{lost packets} \in O_t). \quad (15)$$

From the above discussions, one can see that FEC can provide perfect recovery capability with probability $p(n_{pe} < t) + p(2t \geq n_{pe} \geq t, \text{erroneous packets} \in O_t)$. When case 3 is encountered, lost packets cannot be recovered by means of FEC as the decoder uses the standard error concealment mechanism to recover the lost packets. In this study, the conventional EC provided by the H.264 JM software [6] is used.

Now, the recovery capability of EC when FEC fails can be analyzed as follows. Based on the characteristics of H.264, let the number of MVs in an original (error-free) macroblock be n_{mv}. The correct MV detection is defined that the MV found by EC is similar to that in the original (error-free) macroblock if their Euclidean distance[¶] is less than a threshold Th_m. We further let the correct MV detection probability be p_{cmv} and let the number of correctly detected motion vectors in a macroblock be n_d. Similarly, the probability that all the MVs of a macroblock detected to be similar to the original MVs is

$$p(n_d = n_{mv}) = {p_{cmv}}^{n_{mv}}. \quad (16)$$

C. Video JET vs. FEC

By calculating the probabilities shown in Eq. (13) and Eq. (14), respectively, it is found that the probability of FEC to achieve perfect error protection is sufficiently small. For example, in a QCIF video sequence, let the FEC parameter, (n, k, t) based on Reed-Solomon (RS) code be selected as $(11, 9, 1)$, which has an increase in bit-rate of 22%. Also, let the average bursty length be 4 [14]. Under these circumstances, more than half of bursty lost packets cannot be recovered by FEC, therefore, Case 3 happens. On the other hand, if the FEC-based approach would like to correct up to 4 bursty lost packets, RS$(17, 9, 4)$ should be used, which generates an increase of bit-rate, 88.9%, which is sufficiently large. These pieces of evidence indicate that in most situations it is reasonable to compare error recovery capability between the proposed method and FEC under Case 3 only.

In the proposed method, in order to have more media hash hiding capacity, n_{mh} is designed as small as possible. However, in general H.264 implementation, n_{mv} is usually large to have higher coding efficiency. On the other hand, in the proposed method, p_{cmh} is generally high because the embedded media hash can describe the condensed essence of the lost block and use it to search for similar ones from the reference frames. Nevertheless, p_{cmv} depends

[¶]In order to provide analytic comparison between the hash-based method and EC of H.264, the Euclidean distance measured between a pair of motion vectors in EC corresponds to the bit error rate measured between a pair of hash sequences in the proposed method.

on motion vector consistency between the lost block and its neighboring lossless blocks. Once MV consistency does not exist, p_{cmv} would be very small. To summarize the above discussions, $n_{mh} < n_{mv}$ and $p_{cmh} > p_{cmv}$ mostly hold in practice. As a result, by comparing Eq. (12) and Eq. (16) one gets

$$p_{cmh}{}^{n_{mh}} > p_{cmv}{}^{n_{mv}}, \tag{17}$$

which implies that the proposed method outperforms FEC in that visually similar blocks are easier to be found for the purpose of error recovery. Experimental comparison between the proposed method and FEC will be provided in Sec. VI to further verify this analytic result.

VI. Experimental Results

A. Experimental Setup

Several experiments were conducted to validate the capability of the proposed error-resilient video JET method. The H.264 codec [6] was adopted for video compression and 3-DES was adopted for video encryption due to its provable security. A number of standard video sequences of frame size 176×144 in the QCIF format were used for experiments. Among them [19], the *Table-tennis* sequence, belonging to video Class C, contains high spatial detail and medium amount of movement or vice versa, whereas the *Foreman* sequence, belonging video Class B, contains medium spatial detail and low amount of movement or vice versa (It is also known that still video sequences, e.g., Akiyo, belong to Class A). The GOP structure of length 15 is defined to be "IPP...P," which contains 1 I frame and 14 P frames.

The error-prone network was simulated by means of a two-state Markov chain model [36] with bursty packet loss [14], which contains two states, "Reception State" and "Loss State." Each packet was assigned with one of the two states. Let P_{LL}, P_{LR}, P_{RL} and P_{RR} denote the transition probabilities, where R denotes "Reception State" and L denotes "Loss State." For example, P_{LR} is the transition probability from the "Loss State" to the "Reception State." The average bursty length, L_b, and packet loss rate, P_l, are, respectively, defined as:

$$\begin{aligned} P_l &= \frac{P_{RL}}{P_{LR} + P_{RL}}; \\ L_b &= \frac{1}{P_{LR}}. \end{aligned}$$

The bursty packet loss [14] with parameters determined as $L_b = 4$ and $P_l = 0.05$ were adopted.

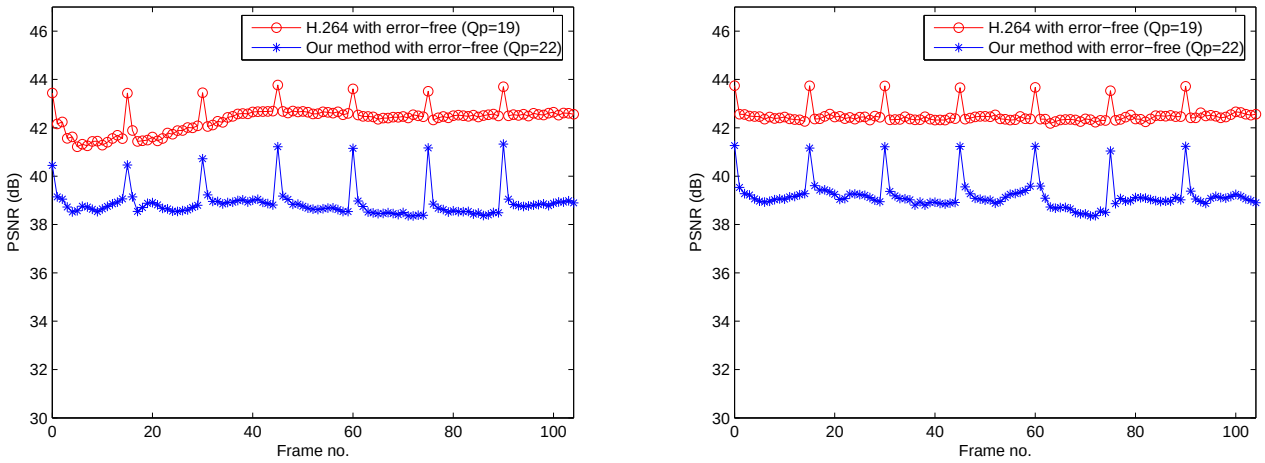
In this paper, we focus on recovery of lost motion vectors and assume that no errors occur in I frames. As a result, I-frame can be set to start from the "Reception state." In this implementation, the mode of data-partition in H.264 was turned on, which contains motion vectors (DP A), intra-blocks (DP B), and residual data (DP C). Among them, the residual data in DP C was adopted as the hiding carrier for media hash hiding. The error resilience of the proposed method was also compared with the fundamental mechanism, i.e., error concealment associated with the advanced coding standard, H.264 (case 3 of FEC). Since not only the MV inconsistent blocks but also error-propagated blocks will be encountered, in this method the blocks that are perceptually similar to the lost block

are the targets to be found to guarantee packet loss recovery. Therefore, a higher threhold $Th_b < 0.5$ used in the first-stage hash matching is set in order to not leave out possible candidates. With the assistance of the two-stage hash matching process, dissimilar blocks can be eliminated.

B. Error Resilience of The Proposed Method

In the first experiment, the quality loss due to block hash embedding was studied. Fig. 11 shows the comparison of PSNR values between a decoded (without data embedding) video, a decoded+embedded video. The video bit-rate was 960 Kbps. It is not surprising to note that the objective quality, measured in terms of PSNR, is degraded due to hash embedding. However, this cost is paid for in protection of videos transmitted over error-prone networks.

For subjective quality evaluation of selective encryption, a typical pair of unencrypted and encrypted frames is shown in Fig. 7(a). As one can see from the right part of *Foreman* in Fig. 7(a) much of the visual information (e.g., eyes, mouth, nose, and fingers) cannot be seen in 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 loses its commercial value.



(a) *Table-tennis* sequence (b) *Foreman* sequence

Fig. 11. PSNR comparisons before (H.264) and after (our method) data hiding: (a) *Table-tennis* sequence. (b) *Foreman* sequence. The bit rate is set to be 960 Kbps at encoder.

On the other hand, Figs. 12 and 13 show the visual quality comparison between a pair of cover and stego frames to illustrate the visual distortion introduced by hash embedding. As one can see from these examples, the embedding effect is hard to be perceived subjectively. After performing the proposed method, the increased bit rates are within the range of $20\% \sim 30\%$ in P-frames. In this case, $3 \sim 4$ dB quality degradation can be measured objectively. The reason is that all non-zero coefficients in DP C are embedded with the hash bits and that macroblock partitions are also modified to yield more hiding carriers. However, the cost of increasing bit rate can be compensated for by using

the embedding information for the purpose of error resilient video joint encryption and transmission considered here.

In the second experiment, the capability of resilience to packet loss between the hash-assisted macroblock searching method and the error concealment method was investigated. The final recovery results, obtained by averaging from 30 runs with different embedding keys for packet loss simulations, are shown in Figs. 14 for two different video sequences. In the beginning, packet loss has not occurred and the proposed method exhibits lower PSNRs than EC due to data hiding. Once packet losses happen, the proposed method gradually outperforms error concealment. For *Table-Tennis* video, the improved PSNR values, which range from 1 to 4 dB, are rather significant, as shown in Fig. 14(a). In Fig. 15, we show two pairs of video frames recovered based on the proposed method and error concealment of H.264 for visual verification. One can see that the visual quality of Figs. 15 (a) and (c) obtained by the proposed method are significantly better than that of Figs. 15 (b) and (d) obtained by H.264. The reason is that hash length is large enough (corresponding to the fact the hiding carrier is large enough in the Table-tennis video) to find a candidate block most similar to the lost block.

For the *Foreman* video, the proposed method obtains most of its improvement at the end of GOPs due to efficient suppression of error propagation, as shown in Fig. 14(b). Fig. 16 further shows the visual quality comparison between the proposed method and EC of H.264. By comparing Figs. 16(a) and (b), the visual quality of Fig. 16(a) is degraded more than Fig. 16(b) because (i) the blocks' motions are similar to their neighbors' motions (see later explanation in Fig. 18, *Foreman*, GOP 4: the average MVs matching probability is 0.86 for $Th_m = 20$) such that the error concealment mechanism in H.264 can recover the lost macroblocks well (e.g., frames 45-59); (ii) the non-zero coeffieicnts in DP C are not enough (corresponding to the fact the hiding carrier is not large enough for the Foreman video) to hide the hash bits at the encoder. On the other hand, the proposed method outperforms H.264's error concealment when the neighboring blocks' correlations are small (see later explanation in Fig. 18, GOP 5: the average MVs matching probability is 0.46 for $Th_m = 5$), as shown in Figs. 16(c) and (d).

From these experimental results, it is obvious that the recovery of macroblocks using the proposed method usually shows smooth and natural visual information, while discontinuous effects can be observed using traditional error concealment. In addition, some hints are obtained that are helpful to further improve error resilience. That is, one can approximately incorporate data hiding-based macroblock hash matching and error concealment for purpose of error resilience. According to our observations, block hash assisted motion estimation is rather helpful for macroblocks with inconsistent motions in the neighboring blocks, while error concealment is very useful to recover lost packets with motions consistent with their neighboring blocks without wasting bit rates for embedding.

C. Verification of Analytic Results

In this section, we will provide experimental evidence to further verify analytic comparison between the proposed method and the benchmark, FEC. For the experiments described above, when FEC is applied for error recovery, the average number of motion vectors in a macroblock for each GOP is shown in Fig. 17 and the average probability

of motion vector matching is shown in Fig. 18. For the proposed method, the average number, n_{mh}, of partitioned blocks in a macroblock for each GOP is shown in Fig. 19 and the hash matching probability is shown in Fig. 20. From the numerical results above, we can derive $p_{cmh} \approx 1$ and $n_{mh} = 1.03$ for the proposed method. On the other hand, if the threshold used for measuring motion similarity in terms of Euclidean distance is set to a higher value, $Th_m = 20$ (see Fig. 18), then $p_{cmv} = 0.72$ and $n_{mv} = 3.20$ can be derived for EC of H.264. Therefore, the relationship of between ${p_{cmh}}^{n_{mh}}$ and ${p_{cmv}}^{n_{mv}}$ is derived to be ${p_{cmh}}^{n_{mh}} = 1 > {p_{cmv}}^{n_{mv}} = 0.35$, which implies that this study's analytic result is identical to the experimental result.

VII. Conclusions

This paper proposes a solution to robust video joint encryption and transmission with errors tolerated up to the level of bursty packet loss. We investigated a macroblock hash embedding scheme at the encoder and exploited the extracted hashes for macroblock matching at the decoder to achieve estimation and compensation of motion vectors. Since the embedded macroblock hashes are available at the decoder, the research shows that such information is helpful for resisting errors in a non-blind manner. It is also worth noting that, since video block hash preserves the condensed content to facilitate the search of similar blocks, motion estimation is implicitly performed through robust media hash matching – that is the unique characteristic of this method. In particular, the advantage of the proposed method is that even if the number of lost packets in a frame is larger than one (as opposed to the recovery limitation of [29]), this method can still provide effective error recovery. In addition, the proposed method can be used for error-resilient video encryption and transmission simultaneously. These constitute the major contributions of this study.

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*, Vol. 5, pp. 3097-3100, 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.

[4] J. Fridrich, "Visual Hash for Oblivious Watermarking," *Proc. SPIE: Security and Watermarking of Multimedia Contents II*, Vol. 3971, pp. 286-294, 2000.

[5] "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC," in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050, 2003.

[6] *http://iphome.hhi.de/suehring/tml/download/old_jm/jm96.zip*.

[7] *IEEE Int. Conf. on Multimedia and Expo*: special session on Media Identification, June 2004.

[8] L. W. Kang and C. S. Lu, "Low-Complexity Wyner-Ziv Video Coding Based on Robust Media Hashing," *Proc. IEEE Int. Workshop on Multimedia Signal Processing*, Canada, 2006.

[9] L. W. Kang and C. S. Lu, “Wyner-Ziv Video Coding with Coding Mode Hiding-based Motion Compensation,” *Proc. IEEE Int. Conf. on Image Processing*, USA, 2006.

[10] M. Lee, S. Nepal, and U. Srinivasan, “Role of Edge Detection in Video Semantics,” *Proc. ACS Conferences in Research and Practice in Information Technology*, Vol. 22, pp. 59-68, 2003.

[11] B. Li, E. Chang, and C. T. Wu, “DPF – A Perceptual Distance Function for Image Retrieval,” *Proc. IEEE Int. Conf. on Image Processing*, Vol. 2, pp. II-597-II-600, 2002.

[12] 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.

[13] C. Y. Lin, D. Sow, and S. F. Chang, “Using Self-Authentication-and-Recovery Images for Error Concealment in Wireless Environments,” *SPIE ITCom/OptiComm*, Vol. 4518, p. 267-274, 2001.

[14] 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.

[15] C. S. Lu, “Wireless Multimedia Error Resilience via A Data Hiding Technique,” *Proc. 5th IEEE Int. Workshop on Multimedia Signal Processing*, pp. 316-319 , 2002.

[16] 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.

[17] 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, Vol. 11, No. 2, pp. 159-173, 2005.

[18] *IEEE Int. Workshop on Multimedia Signal Processing*, special session on Media Recognition, 2002.

[19] *http://www.ece.cmu.edu/∼ece796/mpeg4.pdf*

[20] 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.

[21] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Processing Magazine*, Vol. 15, pp. 23-50. Nov. 1998.

[22] W. Pongpadpinit and A. Pearmain, “Recovery of Motion Vectors for Error Concealment Based on an Edge-Detection Approach,” *IEE Proc.-Vis. Image Signal Process.*, Vol. 153, No. 1, pp. 63-69, Feb. 2006.

[23] 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.

[24] 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.

[25] 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, pp. 249-258, Apr. 2004.

[26] T. Shanableh and M. Ghanbari, “Loss Concealment Using B-Pictures Motion Information,” *IEEE Trans. on Multimedia*, Vol. 5, No. 2, pp. 249-258, 2003.

[27] C. Shi and B. Bhargava, “A fast MPEG video encryption algorithm,” *Proc. ACM Conf. on Multimedia*, pp. 81-88, 1998.

[28] 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.

[29] 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, pp. 415-423, 2001.

[30] S. W. Sun, J. R. Chen, C. S. Lu, P. C. Chang, and K. C. Fan, “Motion-Embedded Residual Error for Packet Loss Recovery of Video Transmission and Encryption,” *Proc. IS&T/SPIE: Visual Communications and Image Processing* (EI127), Vol. 6077, pp. 452-465, 2006.

[31] A. Tosun and W. C. Feng, “On Error Preserving Encryption Algorithms for Wireless Video Transmission,” *Proc. ACM Conf. on Multimedia*, pp. 302-308, 2001.

[32] 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.

[33] 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.

[34] 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.

[35] 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*, Vol. 5306, pp. 725-736. 2004

[36] 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*, Vol. 4671, pp. 103-111, 2002.

[37] 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.

[38] W. Zeng, X. Zhuang, and J. Lan, “Network Friendly Media Security: Rationales, Solutions, and Open Issues,” *Proc. IEEE Int. Conf. on Image Processing*, Vol. 1, pp. 565-568, 2004.

[39] B. B. Zhu, C. Yuan, Y. Wang and S. Li, “Scalable Protection for MPEG-4 Fine Granularity Scalability,” *IEEE Trans. on Multimedia*, Vol. 7, No. 2, pp. 222-233, 2005.

(a) cover video frame 46

(d) stego video frame 46

(b) cover video frame 52

(e) stego video frame 52

(c) cover video frame 59

(f) stego video frame 59

Fig. 12. Visual quality comparison of *Table-Tennis* between (a-c) cover video frame and (d-f) stego video frame.

(a) cover video frame 61

(d) stego video frame 61

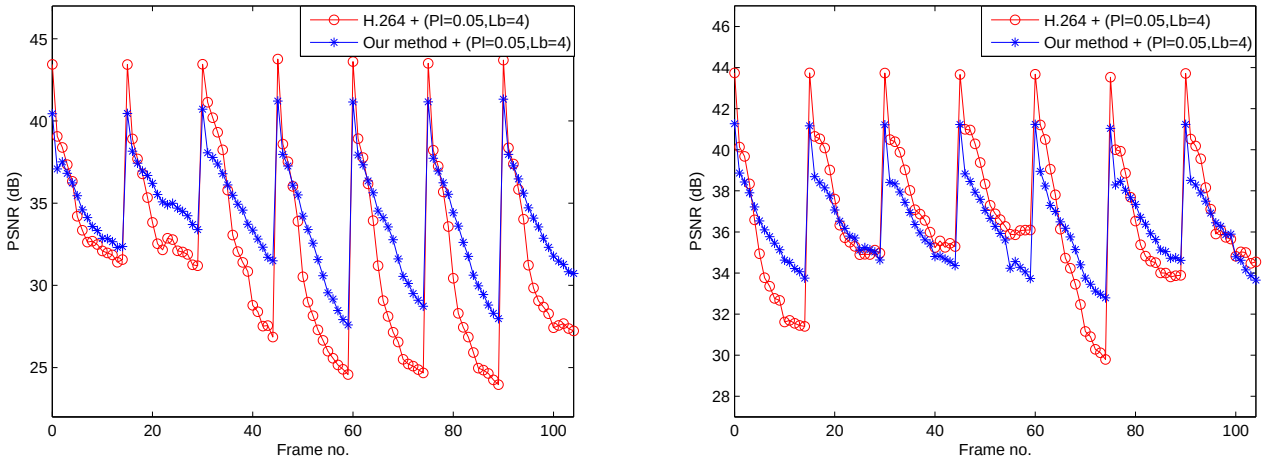
(b) cover video frame 67

(e) stego video frame 67

(c) cover video frame 74

(f) stego video frame 74

Fig. 13. Visual quality comparison of *Foreman* between (a-c) cover video frame and (d-f) stego video frame.



(a) Table-tennis (b) Foreman

Fig. 14. Comparison of error resilience between the proposed method and H.264 error concealment for (a) Table-tennis video and (b) Foreman video.

(a) Proposed method (frame 52) (*PSNR=33.63 dB*)

(b) H.264 EC (frame 52) (*PSNR=25.10 dB*)

(c) Proposed method (frame 59)(*PSNR=29.05 dB*)

(d) H.264 EC (frame 59) (*PSNR=22.94 dB*)

Fig. 15. Visual quality comparison of *Table-Tennis*: (a) the video frame reconstructed using the proposed method (frame 52); and (b) the video frame reconstructed using error concealment (frame 52); (c) the video frame reconstructed using the proposed method (frame 59); and (d) the video frame reconstructed using error concealment (frame 59). One notes from (b) and (d), when compared, respectively, with (a) and (c), that some poorly recovered results are obviously perceived.

(a) Proposed method (frame 55) (*PSNR=31.59 dB*) (b) H.264 EC (frame 55) (*PSNR=37.05 dB*)

(c) Proposed method (frame 70) (*PSNR=32.74 dB*) (d) H.264 EC (frame 70) (*PSNR=30.45 dB*)

Fig. 16. Visual quality comparison of *Foreman*: (a) the video frame reconstructed using the proposed method (frame 55); and (b) the video frame reconstructed using H.264 error concealment (frame 55); (c) the video frame reconstructed using the proposed method (frame 70); and (d) the video frame reconstructed using H.264 error concealment (frame 70).

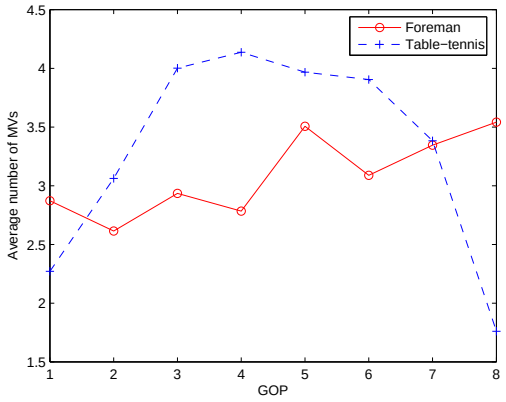


Fig. 17. The average number (n_{mv}) of motion vectors in a macroblock for different GOPs.

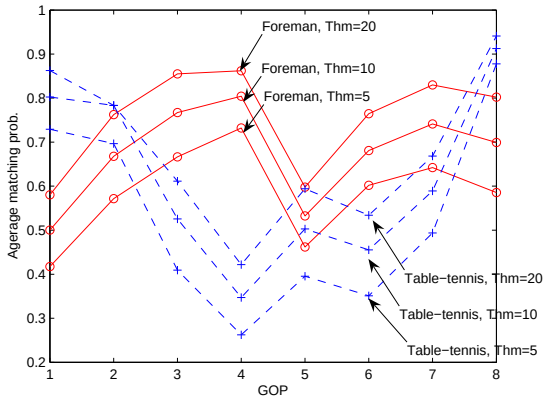


Fig. 18. The motion vectors matching probabilities (p_{cmv}) in different GOPs.

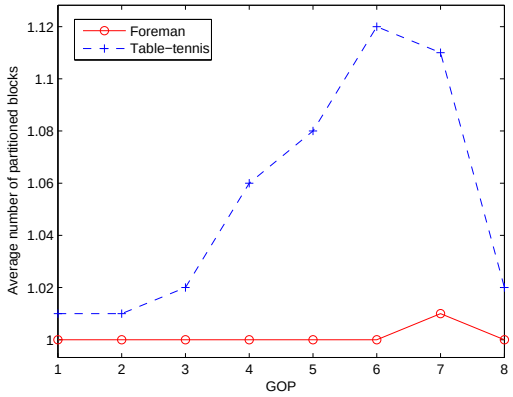


Fig. 19. The average number of partitioned blocks or hash sequences (n_{mh}) in a macroblock for different GOPs.

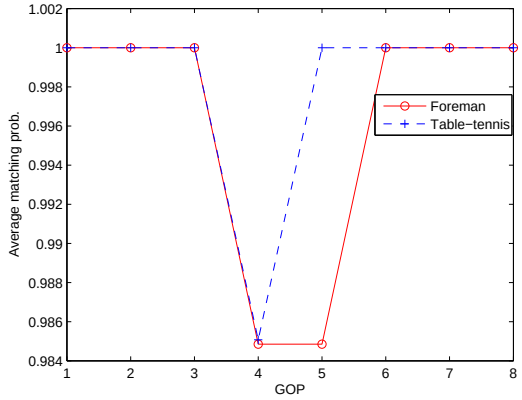


Fig. 20. The media hash matching probabilities (p_{cmh}) in the candidate pool of different GOPs.