Multimed Tools Appl (2010) 47:461–476
DOI 10.1007/s11042-009-0333-5

A new unequal error protection scheme based on FMO

Jhong-Yu Shih · Wen-Jiin Tsai

Published online: 4 August 2009

Abstract This paper presents a novel scheme for video transmission over error-prone networks. The proposed scheme exploits the error resilient features of H.264/AVC and employs an unequal error protection (UEP) approach to protect effectively the streams. A novel algorithm is proposed to classify macroblocks into slice groups using the explicit mode of Flexible Macroblock Ordering (FMO). A rate distortion optimized algorithm is used to assign unequal amounts of FEC protection to the resulting slice groups according to their importance. In addition, a Converged Motion Estimation (CME) is proposed to further improve the proposed UEP scheme. The idea behind the CME is to make the macroblocks be referenced in a skewed manner, such that highly important macroblocks are converged on only a few and the use of redundancy for error protection is efficient. Simulation results show the superiority of the proposed method over other approaches for transmission of H.264/AVC streams.

Keywords Unequal error protection · UEP · Flexible macroblock ordering · FMO · Converged motion estimation · Error resilience

1 Introduction

Transmission of video over error-prone channels, such as mobile networks, presents many challenges because the bit-error rate can be quite high [12]. Associating the video bit-stream with forward error correction (FEC) for error detection and correction can overcome the problem. In the video sequence, some data are higher important than others. For example, the picture headers are much more important than the data block because, once lost, the entire picture cannot be reconstructed well. If these important data can be protected with much more FEC, then they can be received with a much lower uncorrected error rate and a better video quality can be obtained. However, favoring the higher important data with

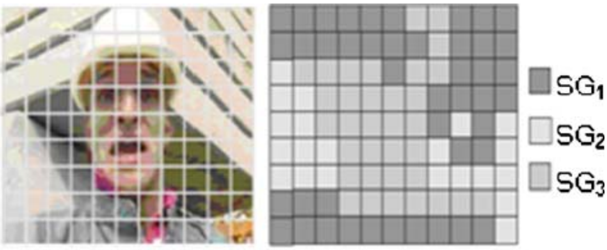
J.-Y. Shih · W.-J. Tsai (✉)
Department of Computer Science, National Chiao-Tung University, Hsinchu, Taiwan, Republic of China
e-mail: wjtsai@cs.nctu.edu.tw

more protection implies that less important data will be less protected. This is the concept of the so-called "*Unequal Error Protection* (UEP)".

Different UEP schemes ponder the criteria of importance from different kinds of viewpoints, for example, frame based [3], layer based [13], data partition based [8], and slice based [2, 10]. Since the frame based and the data partition based schemes cannot provide UEP at a fine granular level such as MBs and the layer based scheme only applies to scalable streams, we chose to base our UEP method on the slice based scheme. For slice-based *UEP*, each frame is divided into slices with different importance and each of the slices is given protection according to its importance and will be transported separately. Robust transmission of H.263 streams was examined in [5]. A packetization method of slices and an UEP algorithm for joint optimization of macroblock coding parameters and selection of FEC codes was presented. An algorithm which adaptively classifies the data packets of MPEG2 encoded video streams into two Quality of Service classes was proposed in [4].

Flexible Macroblock Ordering (FMO) is an error resilient tool in H.264/AVC. In FMO mode, macroblocks can be assigned to any slice group (SG) in a frame, so the flexibility allows them to be transmitted in a non-scanning order. Since each SG is independently decodable, one can scatter macroblocks spatially into different SGs. Thus, if one of the SGs gets missed, its missing macroblocks can be surrounded by correctly received macroblocks. Hence, an appropriate error concealment method can recover the lost macroblock efficiently. In general, error concealment is well performed using H.264/AVC dispersed FMO mode. However, it is a fixed pattern applied on every frame without considering the importance of the individual MB. In [2, 10], the UEP schemes with H.264/AVC explicit FMO mode are proposed, as shown in Fig. 1, where the MBs of similar estimated distortion are assigned to the same SGs. In [13], they used the explicit FMO mode with two SGs. The most important MB was moved to one of the SGs first. However, since this will affect the distortion estimation on its neighboring MBs, the importance of the neighboring MBs must be recalculated accordingly. The MB assignment and the update of the MB importance are performed repeatedly till an optimal solution is found. In [2], the expected length of error propagation (ELEP) is used to describe the effect of packet loss on the video quality. Initially, each SG has equal number of FEC packets, and then FEC packets are moved among these SGs repeatedly till the minimal distortion is achieved. In [10], they used three SGs and two thresholds, *Th* and *Tl*. If the importance of a MB is greater than *Th*, it is assigned to the high important SG; if that is less than *Tl*, to the low important SG; otherwise, to the middle important SG. However, the two thresholds are not easily determined.

The proposed scheme is based on macroblock classification and UEP of H.264/AVC streams. Instead of using an exhausted algorithm to determine the optimal solution of assigning MBs to SGs as used by many methods described above [2, 8, 13], this paper



(a) The original frame. (b) The explicit FMO mode.

Fig. 1 Macroblock classification [10]

presents a novel method, which provides a new variation of the dispersed FMO mode and uses the k-means clustering algorithm to classify macroblocks into slice groups. The RS protection of each slice group is determined using a rate-distortion optimized algorithm. In addition, we propose a motion estimation technique, called Converged Motion Estimation, which further improves the efficiency of UEP by making MBs be referenced in a skewed manner. The remainder of this paper is organized as follows. Section 2 gives an introduction to UEP related works, and Section 3 describes the proposed UEP scheme and the CME is presented in Section 4. Section 5 shows the experimental results, and Section 6 concludes this paper.

2 Related works

2.1 Forward error correction

Forward error correction (FEC) is a channel coding technique used to recover data from packet losses in the transmission. The type of FEC used depends on the requirements of the application and the nature of the channel. The most commonly studied erasure codes are based on RS codes [6], which have good erasure correcting properties and are widely used in practice [12, 15]. In this work, we consider RS codes, but the basic framework could easily be applied to other codes.

An RS code is represented as RS(n, k), where k is the number of source symbols and (n–k) is the number of parity symbols. The protection capability of an RS code depends on the block size and the code rate. The block length, n, is usually determined based on the end-to-end system delay constraints, and the code rate of an RS(n, k) is defined as k/n. An RS(n, k) decoder can correct up to (n–k)/2 errors or up to (n–k) erasures, regardless of which symbols are lost. Since the errors in the wired channels are typically in the form of packet erasures, an RS(n, k) code applied across packets can correct up to (n–k) lost packets. Thus, the block unrecoverable probability, $p_{\mathrm{unrecoverable}}$, (i.e., the probability that at least one of the original k packets is in error) is

$$p_{unrecoverable} = \sum_{i=(n-k)+1}^{n} \binom{n}{i} p_{ch}^{i} (1 - p_{\mathrm{ch}})^{n-i} \tag{1}$$

where p_{ch} is the packet erasure probability before error recovery.

2.2 End-to-end distortion

A number of methods have been proposed to estimate end-to-end distortion. The problem was originally considered for optimizing intra-inter decisions to combat temporal error propagation. In [16], the authors suggest a recursive optimal per-pixel estimate (ROPE) for optimal intra/inter mode selection. In ROPE, the expected distortion for any pixel is calculated recursively as follows. Let f_n^i denote the original pixel value at location i in frame n and $\tilde{f}_n^i$ the reconstruction of the same pixel at the decoder. The expected MSE distortion d_n^i of the pixel at that location can then be written as

$$\mathrm{E}\left[d_n^i\right] = \mathrm{E}\left\{\left(f_n^i - \tilde{f}_n^i\right)^2\right\} = \left(f_n^i\right)^2 - 2f_n^i \mathrm{E}\left\{\tilde{f}_n^i\right\} + \mathrm{E}\left\{\left(\tilde{f}_n^i\right)^2\right\} \tag{2}$$

At the encoder, the value f_n^i is known and the value $\tilde{f}_n^i$ is a random variable. So the expected distortion at each location can be determined by calculating the first and second moment of the random variable $\tilde{f}_n^i$.

If we assume the encoder uses full pixel motion estimation and frame copy error concealment, each lost pixel is reconstructed by copying the pixel at the predicted location k in the previous frame. The predicted location is determined by the median motion vector of the three MBs that are nearest to the lost pixel i. We assume a Bernoulli-independent packet loss model [14] where the probability that any packet is lost is independent of any other packet. We further assume that the packet lost rate, denoted by p, is available at the encoder. The respective recursion formulae of ROPE are as follows.

Pixel in an intra-coded MB

$$\mathrm{E}\{\tilde{f}_n^i\}(\mathrm{I}) = (1-p)\left(\tilde{f}_n^i\right) + p(1-p)\mathrm{E}\{\tilde{f}_{n-1}^k\} + p^2\mathrm{E}\{\tilde{f}_{n-1}^i\} \tag{3}$$

$$\mathrm{E}\left\{\left(\tilde{f}_n^i\right)^2\right\}(\mathrm{I}) = (1-p)\left(\tilde{f}_n^i\right)^2 + p(1-p)\mathrm{E}\left\{\left(\tilde{f}_{n-1}^k\right)^2\right\} + p^2\mathrm{E}\left\{\left(\tilde{f}_{n-1}^i\right)^2\right\} \tag{4}$$

Pixel in an inter-coded MB

$$\mathrm{E}\{\tilde{f}_n^i\}(\mathrm{P}) = (1-p)\left(\hat{e}_n^i + \mathrm{E}\{\tilde{f}_{n-1}^j\}\right) + p(1-p)\mathrm{E}\{\tilde{f}_{n-1}^k\} + p^2\mathrm{E}\{\tilde{f}_{n-1}^i\} \tag{5}$$

$$\begin{aligned}\mathrm{E}\left\{\left(\tilde{f}_n^i\right)^2\right\}(\mathrm{P}) &= (1-p)\mathrm{E}\left\{\left(\hat{e}_n^i + \tilde{f}_{n-1}^j\right)^2\right\} + p(1-p)\mathrm{E}\left\{\left(\tilde{f}_{n-1}^k\right)^2\right\} + p^2\mathrm{E}\left\{\left(\tilde{f}_{n-1}^i\right)^2\right\} \\ &= (1-p)\left(\left(\hat{e}_n^i\right)^2 + 2\hat{e}_n^i\,\mathrm{E}\{\tilde{f}_{n-1}^j\} + E\left\{\left(\tilde{f}_{n-1}^i\right)^2\right\}\right) \\ &\quad + p(1-p)\mathrm{E}\left\{\left(\tilde{f}_{n-1}^k\right)^2\right\} + p^2\mathrm{E}\left\{\left(\tilde{f}_{n-1}^i\right)^2\right\}\end{aligned} \tag{6}$$

The inter-coding notation above assumes that pixel i is predicted from pixel j in the previous frame. The prediction error e_n^i is quantized to the value $\hat{e}_n^i$.

The above recursions are performed at the encoder in order to calculate the expected distortion at the decoder. In this paper, we exploit this result to estimate the importance of macroblocks, called *impact factor (IF)*. Specifically, the *IF* of a macroblock i in frame n is defined as follows.

$$\mathrm{IF}\left(m_n^i\right) = E\left[D\left(m_n^i\right)\right] = \sum E\left[d_n^j\right], \text{ for all the pixels } d_n^j \text{ located in } m_n^i \tag{7}$$

3 Proposed method

This section presents a detailed description of our UEP method. In Fig. 2, we illustrate how the proposed UEP method encodes the bit-stream. The encoder module first encodes the bit-stream. After encoding, the impact factor calculation module calculates the impact factors of all MBs using Eq. 7. The MB classification module then assigns MBs into SGs

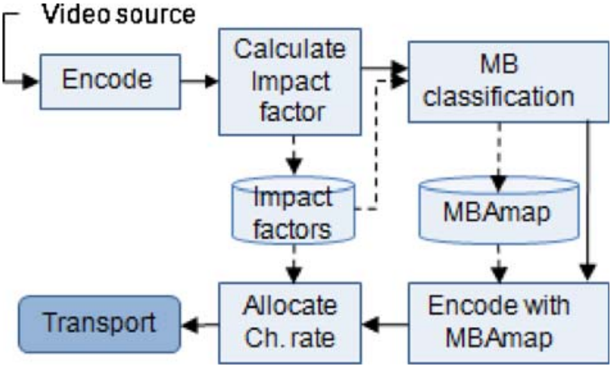


Fig. 2 Flowchart of the proposed scheme

according to the impact factors and generates an MB Allocation map (MBAmap). After this, the encoder uses the MBAmap to encode the stream again. Finally, the channel-rate-allocation-module allocates FEC to the stream, and transports it to the network.

3.1 Macroblock classification

After obtaining the impact factors of all the MBs, the most obvious method for MB classification is to sort the MBs based on their impact factors and then sequentially assign them to SGs of equal size. We call this method *simple UEP*. However, this method may result in the important MBs becoming connected. In Fig. 1, if SG_1 is lost, the error concealment procedure becomes ineffective since the MBs around the lost MB are also lost. We describe how to solve this problem in subsection 3.1.1.

The fixed-size SGs and the variable-size SGs are two options for SG-based UEP. Fixed-size SGs, while simple, are not efficient for UEP. Consider the case where we assume that there are 99 MBs in the frame and among them, 15 MBs contain very important data. With three SGs of a fixed-size (i.e., 33 MBs for each SG), not only would these 15 MBs be assigned to the most important SG (say SG_1), but some MBs which are not important would also be assigned to it. This would result in SG_1 requiring more FEC than expected. We describe how to solve this problem in subsection 3.1.2.

3.1.1 Variation of the dispersed mode

The dispersed mode (checkerboard-like) is classified as type 1 of the slice group map in H.264/AVC. Error concealment is effective using this mode because when a SG is lost, it can be concealed from other SGs. However, the dispersed mode cannot take advantage of the UEP method because of its fixed pattern, making MBs of different importance become part of the same SG.

Our MB classification scheme first adopts the dispersed mode with two SGs (say SG_1 and SG_2) and then splits them further into more SGs according to the impact factors of the MBs. In Fig. 3, assume that we want to classify MBs into three different levels of importance. Six SGs will then be used, with three SGs coming from SG_1 (all odd blocks) and the other three coming from SG_2 (all even blocks). It is obvious that any MB belonging to SG_{11}, SG_{12}, and SG_{13} must have adjacent MBs coming from SG_{21}, SG_{22}, as well as SG_{23}, and vice versa. Since no connected MBs are assigned to the same SG, the error concealment is assumed be effective. The procedure for splitting a SG obtained from the dispersed mode into more SGs according to the impact factor is described next in Section 3.1.2.

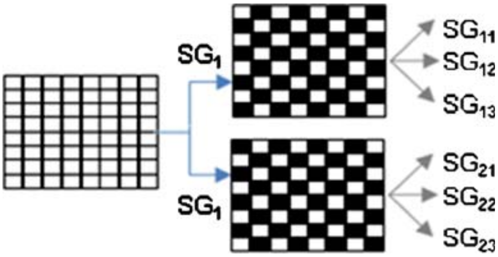


Fig. 3 Variation of the dispersed mode

3.1.2 K-means clustering

In order to make the application of FEC more efficient, it is better to group the MBs with similar impact factors into the same SG. Thus, we simply formulate the MB classification problem as a clustering problem and use the k-means algorithm [7] to solve it. K-means clustering follows a simple and easy procedure to classify a given data set into a certain number of clusters (assuming k clusters) where each cluster is distanced as far away as possible from each other. Here, we regard each MB as a single data point and the impact factor as its position in the coordinate axis of the algorithm. For example, in Fig. 3, assume that there are 14 MBs in SG_1 with impact factors sorted in a descending order as {89, 84, 81, 59, 55, 51, 48, 46, 42, 22, 21, 14, 8, 6}. After applying k-means clustering with k=3, we will have three subsets: SG_{11}={89, 84, 81}, SG_{12}={59, 55, 51, 48, 46, 42} and SG_{13}= {22, 21, 14, 8, 6}. The k-means algorithm automatically groups together macroblocks that need similar protection levels. Taking into account the error concealment efficiency, the proposed algorithm applies k-means on all-even and all-odd blocks separately, and then generates the integrated MBAmap accordingly.

3.2 Channel rate allocation

Since our algorithm organizes SGs by grouping together macroblocks that have similar impact factors, we use the same protection level for all the macroblocks belonging to the same SG. Let l_i denote the protection level assigned to SG_i, the ith slice group of a GOP, and (n_i , k_i) denote the RS parameters associated with this protection level. Since the RS block length is usually determined based on the end-to-end system delay constraints, we assume it is constant (say N) over the entire GOP. So, we have n_i = N for all the possible i in the GOP. We consider the use of RS codes for the application-layer FEC and apply the RS coding across video packets. This is due to the fact that some MAC implementations such as 802.11 [1, 9] discard the whole MAC frame in the event of an error. The erroneous frame at the receiving MAC is never passed on to the higher layer. If RS coding is applied within a single packet at the application layer, the erroneous packet will not be available for error correction at the application layer. Therefore, RS coding at the application layer is applied across packets. The packet formation of a slice group after RS encoding is illustrated in Fig. 4, where the slice group SG_i is blocked into source blocks of length k_i bytes; and each source block is expanded into channel codewords of length N > k_i bytes using an (N, k_i) RS code such that the channel code rate k_i/N provides erasure protection commensurate with the importance of SG_i. Finally, the length-N channel codewords are packetized into N packets by putting the i^{th} byte in each channel codeword into the i^{th} packet, resulting in k_i source packets and (N–k_i) RS packets. This packetization scheme has

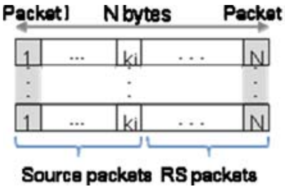


Fig. 4 Packet formation of a slice group

the property that if any $k \geq k_i$ out of the N packets are received, then the slice group SG_i can be recovered, otherwise SG_i is lost.

In order to attain better performance from the proposed UEP scheme, our strategy for the channel rate allocation method aims at optimizing the code rates k_i /N for each slice group in order to minimize the overall distortion for a given transmission rate. Here, we present a rate-distortion optimized solution that seeks the best protection level for each SG. Our objective is to seek the vector of the RS coding parameters, K={k_1, k_2, ..., k_h}, that minimizes the expected end-to-end distortion of the corresponding GOP, where h is the total number of SGs in the GOP. The expected distortion of a GOP is given by:

$$\mathrm{E}[\mathrm{D}(\mathrm{GOP})] = \sum\nolimits_{i=1}^{h} \mathrm{E}[\mathrm{D}(SG_i)] \tag{8}$$

where $\mathrm{E}\left[\mathrm{D}\left(SG_j\right)\right] = \sum \mathrm{E}\left[\mathrm{D}\left(m_j\right)\right]$ for all the macroblocks m_j belonging to SG_i , and the E[D(m_j)] is defined in Eq. 7. However, since SG_i has a protection level l_i = (N, k_i), the packet-loss rate p in Eqs. 3–6 must be substituted with p_{l_i} below:

$$p_{l_i} = \sum\nolimits_{j=(N-k_i)+1}^{h} \binom{\mathrm{N}}{j} p_{\mathrm{ch}}^{j} (1-p_{\mathrm{ch}})^{\mathrm{N}-j} \tag{9}$$

where p_{l_i} is the probability that SG_i cannot be correctly decoded by the RS decoder and p_{ch} is the packet erasure probability. The channel rate allocation problem is formulated as:

$$\min\ \mathrm{E}[\mathrm{D}(GOP)]\ \text{subject to}\ \mathrm{R}(GOP) < \mathrm{R}_{\max} \tag{10}$$

where

$$\mathrm{R}(\mathrm{GOP}) = \sum\nolimits_{i=1}^{h} \mathrm{R}(SG_i) = \sum\nolimits_{i=1}^{h} \mathrm{R}_{\mathrm{source}}(SG_i) \times \frac{\mathrm{N}}{k_i} \tag{11}$$

where R(SG_i) is the total bitrates of SG_i. This includes source bitrates, denoted by $\mathrm{R}_{\mathrm{source}}$($SG_i$), and the redundancy bitrates with RS parameter (N, k_i). The above constrained minimization problems (10) and (11) are naturally recast in the standard Lagrangian formulation as:

$$\min\ \mathrm{J}(\mathrm{GOP}) = \min\{\mathrm{D}(\mathrm{GOP}) + \lambda \mathrm{R}(\mathrm{GOP})\} \tag{12}$$

By appropriately choosing the Lagrange multiplier λ, the above problem (10) can be solved within a convex-hull approximation by solving (12). The search for an appropriate choice of λ can be carried out using the bisection algorithm or a fast convex search technique which is not discussed here. The optimal K then leads to the optimal rate distribution between the source and FEC rates. Here, we present the solution to the minimization problem using a trellis-based dynamic programming algorithm [11], as shown in Fig. 5, where stage i corresponds to the decision made about the best RS coding parameter k up to the i^{th} slice group (say SG_i) in the GOP. The slice groups are numbered in

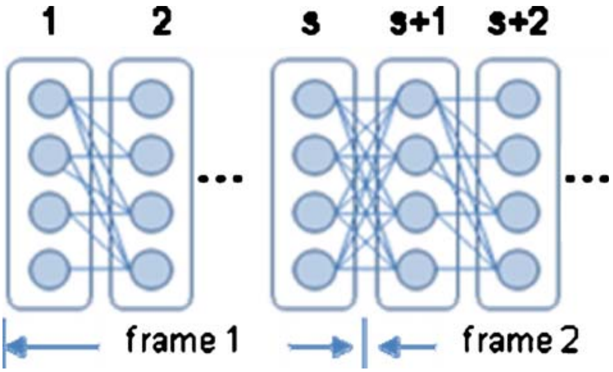


Fig. 5 The trellis used to find the optimal coding parameter *k* for all the slice groups in a GOP

a descending order of importance in each frame whereas the slice groups in the earlier frames of the GOP are numbered before those in the later frames. In Fig. 5, each node at stage *i* represents a particular choice of *k* for the slice group SG_i . The metric of this trellis diagram is the Lagrangian cost J. For the first stage, the Lagrangian cost is computed for all possible choices of parameter k_1. Then, for each node at stage *i*, the algorithm evaluates all the paths leading to that node from any admissible coding parameter k_{i-1} for the previous slice group, and stores the one that produces the minimum Lagrangian cost:

$$\mathrm{J}(\mathrm{k_i}) = \begin{cases} \min\{\mathrm{E}[\mathrm{D}(SG_i)] + \lambda\mathrm{R}(SG_i)\} & \text{if } i = 1 \\ \min\{\mathrm{J}(k_{i-1}) + \mathrm{E}[\mathrm{D}(SG_i)] + \lambda\mathrm{R}(SG_i)\} & \text{if } i \neq 1 \end{cases} \tag{13}$$

In Fig. 5, for each node at the stages s+1, 2s+1, ..., (g-1)s+1. (where s is the number of slice groups per frame and g is the number of frames in a GOP), all the paths from the previous stage are admissible. However, for each node at the other stages, only paths leading to that node from the small coding parameters in the previous stage are admissible. That is, the *k* assigned to SG_i should be less than or equal to the *k* assigned to SG_{i-1} if SG_i is not the first slice group (i.e., the most important one) of any frame. This is based on the assumption that the optimal solution shall give lower protection levels to less important slice groups. This offers considerable computational savings. The winning node in the last stage, corresponding to the last slice group in the GOP, determines the optimal path in the trellis and the solution K={k_1, k_2, ..., k_h} for the GOP. With the known benefits of dynamic programming, the complexity of the trellis only grows linearly with the number of slice groups.

4 Converged motion estimation

Since the more important MBs are typically protected more than the less important MBs, the use of FEC would be efficient if the more important MBs are converged on only a few MBs. Based on this idea, we propose a Converged Motion Estimation (CME), which sets out to achieve a skewed reference pattern to MBs. The CME changes the general concept of motion estimation. Instead of choosing the MB with the smallest Mean Square Error (MSE) as a reference, the MB located in a highly protected area is chosen if the increase in the MSE is acceptable.

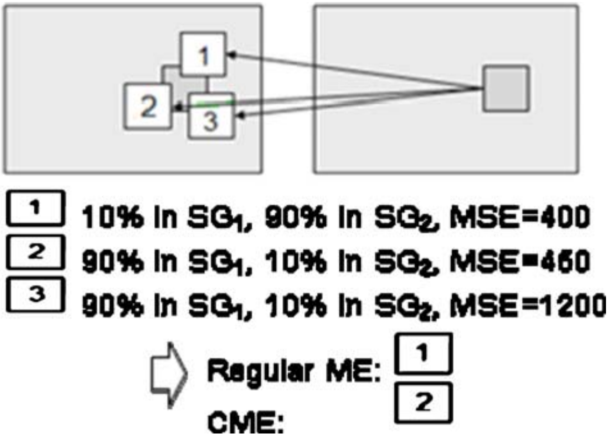
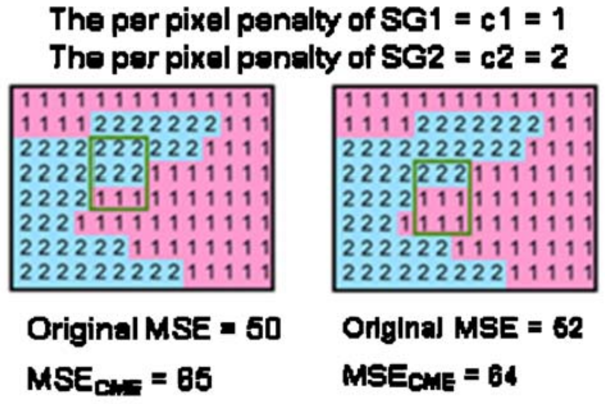

1
2
3
1 10% in SG1, 90% in SG2, MSE=400
2 90% in SG1, 10% in SG2, MSE=450
3 90% in SG1, 10% in SG2, MSE=1200
Regular ME: 1
CME: 2

Fig. 6 A sketch map of the converged motion estimation

In Fig. 6, there are three candidates of reference MBs, with MSE equal to 400, 450, and 1200, respectively. In the general motion estimation, the one with the smallest MSE (MSE=400) will be chosen. Assuming that SG_2 is less protected in comparison to SG_1, choosing the candidate with MSE=400 may cause a greater distortion because it has 90% of its pixels in SG_2, which is less protected. In this case, CME prefers to choose the candidate with MSE=450 because it has 90% of its pixels located in SG_1 although it has a higher MSE. The candidate with MSE=1200 is not considered because of its large MSE even though it has 90% of its pixels in SG_1.

To realize this concept, we modify the definition of the MSE by including a penalty to the pixels that are referred to, which is specified as follows:

$$MSE_{CME} = MSE + \sum_{i=1}^{s} (n_i \times c_i) \tag{14}$$


The per pixel penalty of SG1 = c1 = 1
The per pixel penalty of SG2 = c2 = 2
Original MSE = 50
MSE_CME = 85
Original MSE = 62
MSE_CME = 64

1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 2 2 2 2 2 2 2 1 1 1
2 2 2 2 2 2 2 2 2 2 1 1 1 1
2 2 2 2 2 2 2 1 1 1 1 1 1 1
2 2 2 2 1 1 1 1 1 1 1 1 1 1
2 2 2 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 2 2 2 2 2 2 2 1 1 1
2 2 2 2 2 2 2 2 2 2 1 1 1 1
2 2 2 2 2 2 2 1 1 1 1 1 1 1
2 2 2 2 1 1 1 1 1 1 1 1 1 1
2 2 2 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 1 1 1 1 1

(a) The more important block. **(b)** The less important block.

Fig. 7 An example of CME

Table 1 Parameters used in the experiments

Parameters	Values
Video sequences	Forman, Coastguard, stefan
Group of pictures	I P P P P ......
GOP size	15 frames
Frame format and rate	QCIF (176×144 pixels), 30 fps
Number of SGs	6
The per pixel penalty in CME	ci = i, for i = 1 to 6, where ci is for $\mathrm{SG_i}$, and the SG is numbered in descending order of impact factors
Overall bit rate	Forman: 340 kbps, Coastguard: 360 kbps, Stefan: 980 kbps

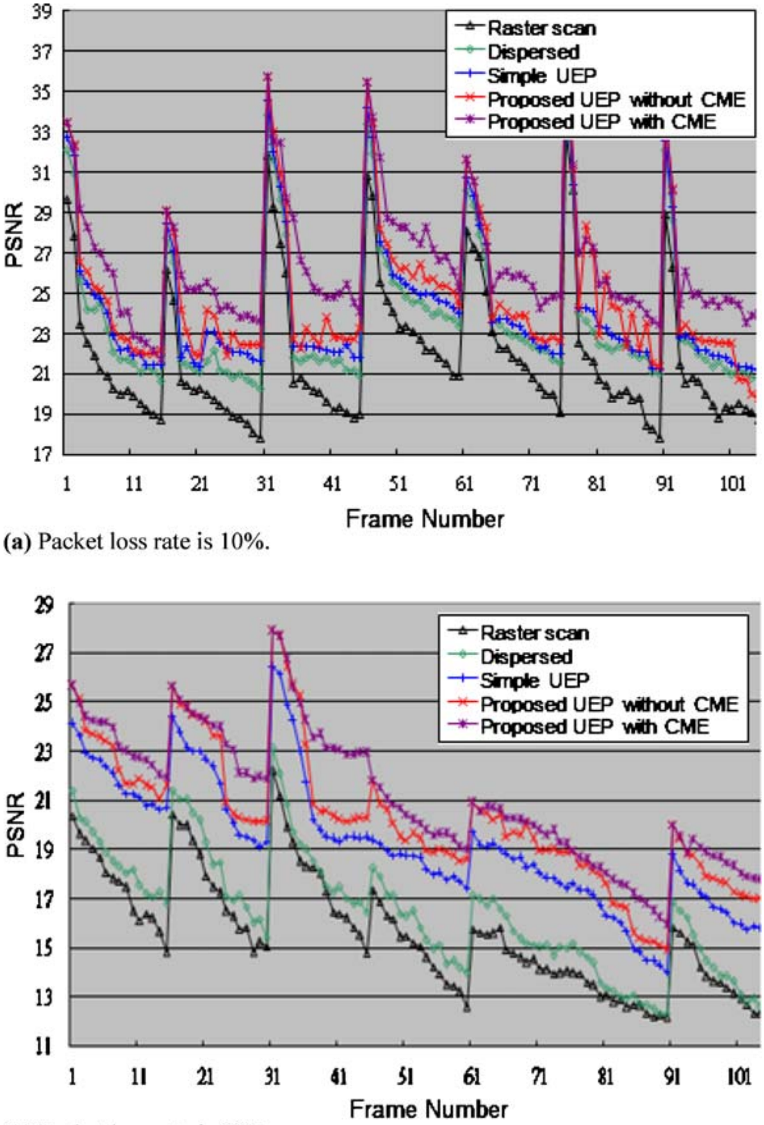
where s is the number of SGs in the reference frame; c_i is the per pixel penalty for the ith SG; and n_i is the number of pixels located in the ith SG for the reference MB currently under consideration. The pixels in the SG with a low impact factor are assigned a large c_i, meaning that reference to these pixels will result in a higher penalty and thus a large $\mathrm{MSE_{CME}}$. On the other hand, the pixels in the SG with a high impact factor will be assigned a small c_i. Figure 7 shows an example where the penalty of pixels in $\mathrm{SG_1}$ is assigned as 1, and that in $\mathrm{SG_2}$ is assigned as 2, meaning that $\mathrm{SG_1}$ is more important than $\mathrm{SG_2}$. In order to simplify this illustration, we assume a macroblock contains 3×3 pixels. In Fig. 7(a), assume that the original MSE is equal to 50, but with the proposed CME, a penalty which is equal to 15 will be added to it. Therefore, the $\mathrm{MSE_{CME}}$ is equal to 65. For the same reason, $\mathrm{MSE_{CME}}$ in Fig. 7(b) is 64 given the original MSE was 52. As a result, in this example, the reference block in Fig. 7(b) has a smaller $\mathrm{MSE_{CME}}$ and will thus be chosen for prediction even though it has a larger original MSE. Using $\mathrm{MSE_{CME}}$, the motion estimation method tends to select MBs in important SGs; as a result, the MB reference pattern will be more skewed compared with using the MSE.

Using CME to encode a frame only requires that its reference frame has multiple SGs with different per pixel penalties assigned. Since CME has no prerequisite information on the current frame, it is applied on the first encoding block in Fig. 2, where the impact factor calculation is performed after it.

5 Experimental results

The proposed UEP scheme for transmission of H.264/AVC streams was evaluated for three standard QCIF sequences “Foreman”, “Coastguard”, and “Stefan” at 30 fps. Group of Pictures (GOPs) of IPPP... structure were considered for all the sequences. The video sequences were encoded and decoded using JM 12.1 with the motion estimation modified to support the proposed CME. The parameters used are shown in Table 1.

In our experiments, “*Raster scan*” and “*Dispersed*” are used for comparison. Both of them are standard defined in H.264 and use Equal Error Protection (EEP) with the same overall bit rate. For comparison, we add the “*Simple UEP*” described in the Section 3.1. The Simple UEP does not adopt the dispersed mode and the k-means algorithm, but it uses our channel rate allocation scheme to have unequal protection according to impact factors. In order to evaluate the effects of CME, we have made a comparison between the “*Proposed UEP method with CME*” and the “*Proposed UEP method*” which did not use CME.

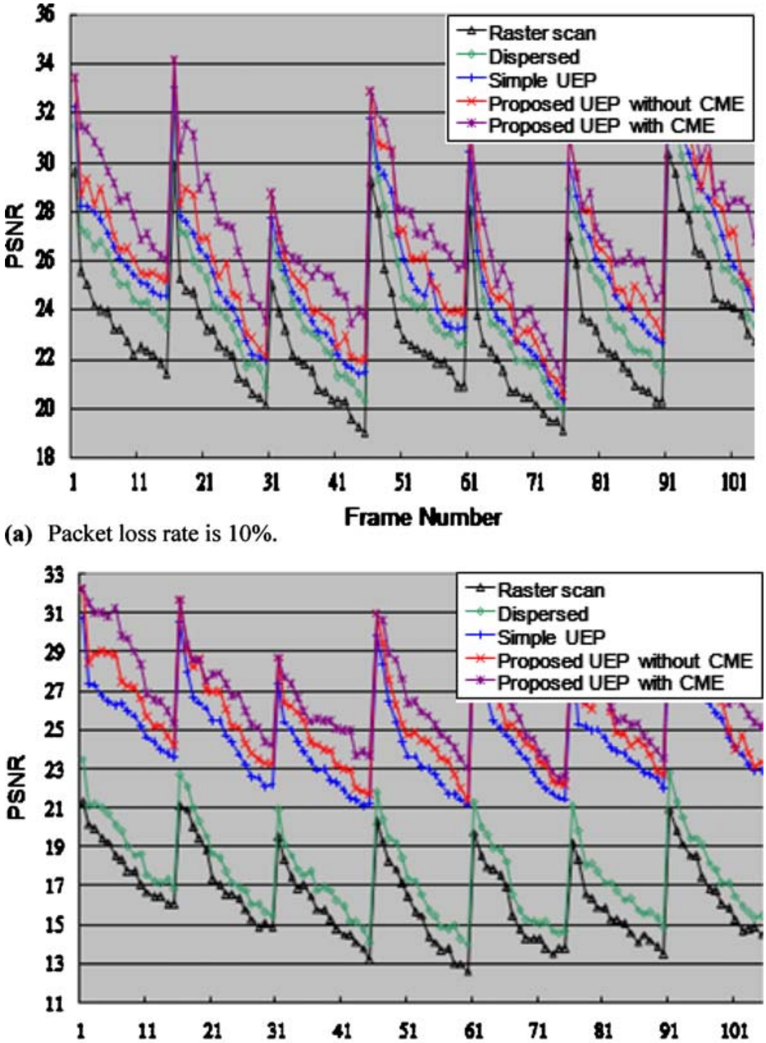


(a) Packet loss rate is 10%.

(b) Packet loss rate is 20%.

Fig. 8 The PSNR when the packet loss rate reaches **a** 10% **b** 20% in Forman

In our experiments, if a MB is not received correctly, the decoder first checks if the adjacent MBs are inter-coded. If at least three inter-coded MBs are available, the median motion vector of them is calculated and used to associate the MB in the previous frame for error concealment. Otherwise, spatial error concealment is performed by using adjacent MBs interpolation. The results of Forman, Coastguard, and Stefan are presented as shown in Figs. 8, 9 and 10, respectively. For each video sequence, two kinds of packet loss rates, 10% and 20%, are used. From the results it is observed that the efficiency of using packet loss rate 20% is better than that of using 10% because EEP cannot handle a high packet loss

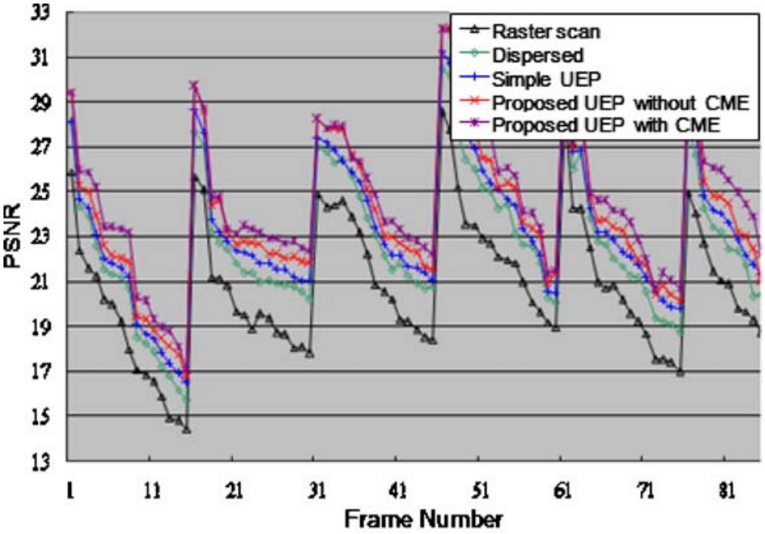


(a) Packet loss rate is 10%.

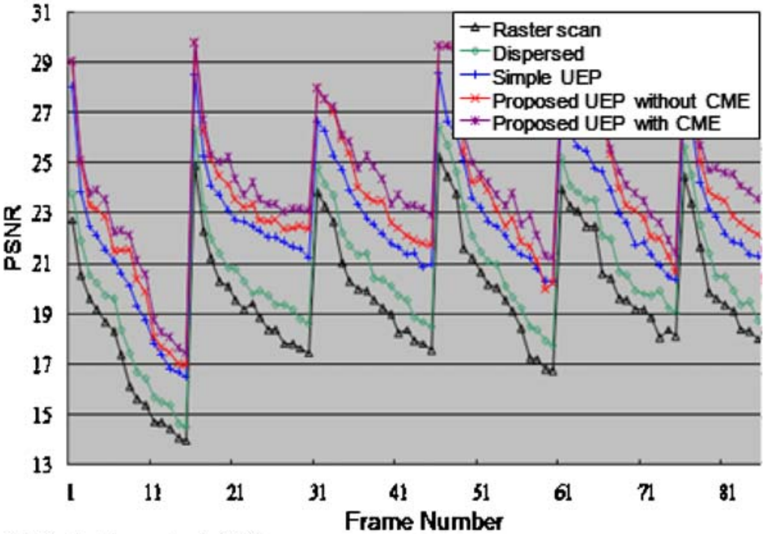
(b) Packet loss rate is 20%.

Fig. 9 The PSNR when the packet loss rate reaches **a** 10% **b** 20% in Coastguard

rate. We also observed that the Dispersed performs better than Raster scan because error concealment is more efficient in Dispersed than Raster scan. Simple UEP outperforms Raster Scan and Dispersed mode in all cases, showing that the impact factors we used can correctly estimate the importance of MBs and thus the unequal protection takes effect. We also observed that the proposed UEP method (without CME) is superior to the simple UEP, showing that using k-means clustering and the variation of the Dispersed mode can improve the overall quality. The reason is that the proposed variation of the Dispersed mode makes no connected MBs in the same slice groups and therefore, is good for error concealment method which results in better performance. The performance gain is about 2.2 dB, and it is even higher if CME is used, which indicates that making energy converged on few MBs will further improve the UEP



(a) Packet loss rate is 10%.

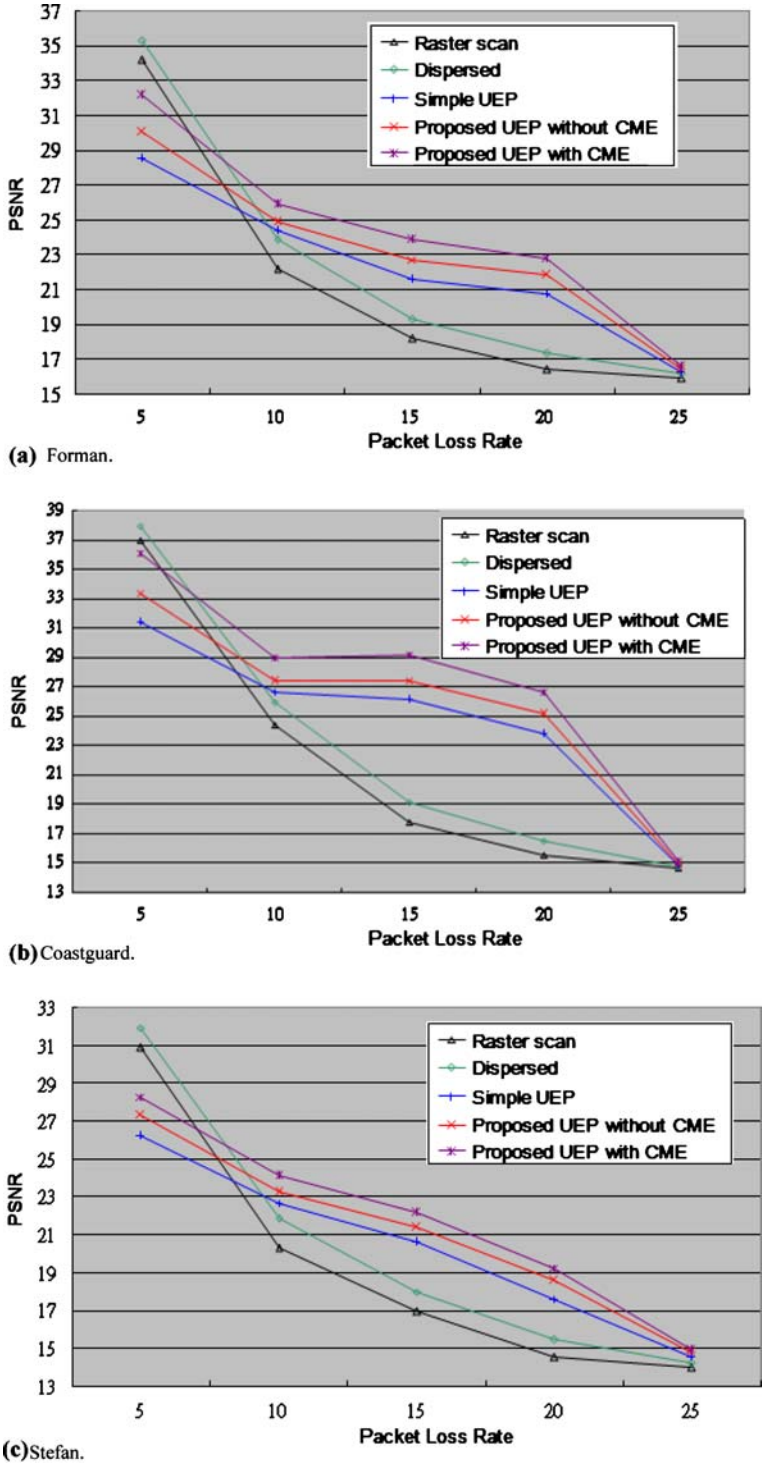


(b) Packet loss rate is 20%.

Fig. 10 The PSNR when packet loss rate reach **a** 10% **b** 20% in Stefan

performance because FEC can be utilized more efficiently. Since there are no motion vectors in I frame, CME cannot be performed in I frame Thus CME cannot improve our proposed UEP method at I frame in a GOP as shown in each result.

Next, we evaluate the results when different packet loss rates are used. Figure 11 shows the PSNR result as a function of packet loss rates, ranging from 5% to 25%. When the packet loss rate is 5%, EEP is better than UEP because the protection rate is good enough for EEP to handle most errors, but UEP cannot protect less important data well in that situation. With the increase of packet loss rate, the efficiency of UEP is more obvious because UEP methods make the quality degrades gradually as packet loss increases. When



(a) Forman.

(b) Coastguard.

(c) Stefan.

Fig. 11 The average PSNR when the packet loss rate equals to 5~25%

the packet loss rate reaches 25%, the measured PSNRs of all methods are almost equal because the overall FEC is not high enough to handle the high error rate no matter whether UEP or EEP is used.

To sum up, the overall results of the Simple UEP show that our macroblock classification scheme correctly estimates the importance of macroblocks and, therefore, achieves a better performance than the dispersed mode (EEP) and Raster Scan mode (EEP). The results of the proposed UEP method without CME show the superiority of our macroblock assignment scheme, which is based on the variation of the dispersed mode and the k-means clustering algorithm. The results of the proposed UEP method with CME show that CME can further improve the performance.

6 Conclusion

We have proposed a scheme which adopts the impact factor, the variation of dispersed mode, the k-means algorithm to provide unequal protection for different data in a video stream. We also proposed a new motion estimation method, called the Converged Motion Estimation, to make MBs be referenced in a skewed manner so the use of FEC is efficient. The overall results show the superiority of the proposed UEP method and the use of CME achieves an even better performance. We are currently investigating how to determine the parameter c_i in CME and analyzing how CME affects the overall source rate.

Acknowledgements This work was supported in part by grants from the National Science Council of Taiwan under Contracts NSC 96-2221-E-009-162-MY2.

References

1. IEEE 802.11 (1999) Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications
2. Im SK, Pearmain AJ (2005) An efficient data classification scheme with the H.264 flexible macroblock ordering. Proc. IEEE 5th International Conference on Information, Communication, and Signal-Processing (ICICS 2005), 1135–1139
3. Marx F, Farah J (2004) A novel approach to achieve unequal error protection for video transmission over 3G wireless networks. Signal Process Image Commun 19(4):313–323
4. Masala E, Quaglia D, Martin JCC (2001) Adaptive picture slicing for distortion-based classification of video packets. Proc IEEE Workshop on Multimedia Signal Processing, Cannes, France, 111–116
5. Masala E, Yang H, Rose K, Martin JCD (2004) Rate-distortion optimized slicing, packetization and coding for error resilient video transmission. Proc DCC Data Compression Conference, Snowbird, UT, USA, 182–191
6. Rizzo L (1997) Effective erasure codes for reliable computer communication protocols. Comput Commun Rev 27(2):24–36
7. Selim SZ, Ismail MA (1984) K-means-type algorithms. IEEE Trans PAMI-6:81–87
8. Stockhammer T, Bystrom M (2004) H.264/AVC data partitioning for mobile video communication. Proc IEEE Int Conference Image Processing, Singapore, 545–548
9. Supplement to IEEE 802.11 (1999) Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications high-speed physical layer in the 5 GHz band
10. Thomos N, Argyropoulos S, Boulgouris NV, Strintzis MG (2005) Error-resilient transmission of h.264/AVC streams using flexible macroblock ordering. IEEE Integration of Knowledge, Semantics and Digital Media Tech
11. Thomos N, Boulgouris NV, Strintzis MG (2005) Wireless image transmission using turbo codes and optimal unequal error protection. IEEE Trans Image Process 14(11):1890–1901
12. var der Schaar M, Krishnamachari S, Choi S, Xu X (2003) Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs. IEEE J Sel Areas Commun 21:1752–1763

13. Wang Y, Fang T, Chau LP, Yap KH (2006) Two-dimensional channel rate allocation for SVC over error-prone channel. Proc. IEEE International Symposium on Circuits and Systems (ISCAS 2006), 21–24
14. Yajnik M, Sue Moon, Kurose J, and Towsley D (1999) Measurement and modeling of the temporal dependence in packet loss. Proc. IEEE 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99), NY, USA, 1:345–352
15. Zhai F, Eisenberg Y, Pappas TN, Berry R, Katsaggelos AK (2004) Rate-distortion optimized product code forward error correction for video transmission over IP-based wireless networks. Proceedings of the IEEE International Conf. on Acoustics, Speech, and Signal Processing (ICASSP'04), Montreal, QUEBEC, Canada
16. Zhang R, Regunathan SL, Rose K (2000) Video coding with optimal inter/intra-mode switching for packet loss resilience. IEEE J Sel Areas Commun 18(6):966–976

Jhong-Yu Shih received the M.S. degrees in computer science from National Chiao-Tung University (NCTU), Taiwan, R.O.C., in 2007. He is with MediaTek Inc. as a researcher in the multimedia processing unit. His research interests include video codec firmware development.

Wen-Jiin Tsai received the Ph.D. degree in 1997 in computer science from National Chiao-Tung University (NCTU), Taiwan, R.O.C. She is an Assistant Professor at the Department of Computer Science of NCTU, Taiwan, R.O.C. Before joining NCTU in 2004, she was with Zinwell Corporation as a Senior R&D Manager for 6 years. Her research interests include video coding, video streaming, error-concealment, and error resilience techniques.