Video Packet Categorization for Priority Delivery to Enhance End-to-End QoS Performance

Jin-Gyeong Kim, JongWon Kim* and C.-C. Jay Kuo

Integrated Media Systems Center and Department of Electrical Engineering
University of Southern California, Los Angeles, California 90089-2564
*Department of Information & Communication
K-JIST (Kwang-Ju Institute of Science & Technology) KwangJu, KwangJu, 500-712, KOREA

ABSTRACT

A data partitioning scheme and its associated prioritization method are proposed to enhance the end-to-end QoS performance in packetized video transmission. Data partitioning is performed for different loss sensitivity classes such as headers, motion vectors, DC coefficients, and AC coefficients. The bitstream in each class is packetized separately from other classes. A priority is assigned to each categorized packet according to its loss sensitivity, which is quantified by the expected distortion when the target packet is lost. The expected distortion is estimated with a corruption model that takes into account error concealment as well as error propagation. Prioritized packets are delivered through the DiffServ (DS) network with proper mapping to DS levels to enhance the end-to-end quality. The corruption model for the data-partitioned packet is verified by simulation, and quality enhancement is demonstrated with experimental results.

Keywords: Packet video, data partitioning, corruption model, network QoS, visual communication.

1. INTRODUCTION

The ever-increasing demand on multimedia communications via the wired/wireless Internet faces the challenge of packet loss as well as bandwidth fluctuation. The dependency between image frames makes the compressed video stream vulnerable even to a small number of lost packets. To address error resilience, the latest versions of ITU-T H.263+ [1] and ISO MPEG-4 have adopted a couple of options to alleviate the corruption of compressed video to error prone channels. Examples include layered representation (e.g. data partitioning), re-synchronization, error tracking, and error recovery options. The robustness issue against packet loss is more relevant to the Internet scenario, which is the main concern of this research.

Internet multimedia (especially packet video) applications have very diverse requirements on the network service. Recently, network architectures are geared towards providing different quality of services (QoS) while preserving network parameters such as loss, delay, and bandwidth. Thus, coordination between different priority packets and network service levels has arisen to the surface. The priority assignment to a video packet would be best if it can accurately represent its error propagation effect to the receiving video quality. A systematic solution is proposed in this work to assign the relative priority index (RPI) to each packet so that it can be used under the packet-level unequal error protection (UEP) framework, where a different level of protection for each packet is attempted. It can be realized at the transport end with different levels of forward error correction (FEC) and/or automatic repeat request (ARQ) for each packet [5]. Also, a proper priority in terms of RPI for each packet may be conveyed to the differentiated service (DiffServe) network to treat each packet differently by using the differentiated forwarding mechanism [6].

Data partitioning has higher coding efficiency than layered coding since it demands only the synchronization code overhead while layered coding requires the whole sub-structure to make bitstreams independent in syntax parsing. Data partitioning has been introduced either to split one bitstream into two or more layers as done in MPEG2 [2], or to isolate error propagation between bitstreams of different categories as done in MPEG4 [3]. Most previous work categorizes compressed bits according to their meaning such as headers, motion vectors, DCT coefficients, etc. These categorized bitstreams are prioritized by their importance. For instance, headers are most important, motion vectors are more important than DCT coefficients, DC coefficients are more important than AC coefficients, and so on. To quantify the importance of categorized bitstreams, the loss impact of each bitstream in the decoded sequence can be measured experimentally, then the value is used to determine the importance of each category.

There is however little work performed to quantify the importance of the partitioned individual packet. Even though the quantification task was performed in [8], the issue of differentiating packets within one category was not addressed. In [8], the average packet loss impact for categorized packets is pre-calculated with exhaustive computation. (Note that it is difficult to measure the packet loss impact due to the error propagation behavior of motion compensated prediction (MCP) existing in most video coding standards.) However, there exist limitations in this approach. First, the quantity can vary according to sequences and encoding parameters. Second, differentiation of individual packets within the same class is not possible based on this pre-determined category-based packet loss quantification scheme. As a result, prioritization can be inaccurate and the overall performance can be severely degraded.

To enhance end-to-end QoS quality, it is important to have a good quantification scheme to accurately reflect the packet loss impact with a moderate computational complexity. To increase the accuracy of packet loss impact estimation, we should take into account many factors such as sequence characteristics, encoding parameters, and dependency between packets. Besides accurate estimation, the computational complexity should be kept low for the practical consideration. In this research, we propose the use of the corruption model to estimate the packet loss impact with respect to data partitioned video packets. The corruption model was analytically developed to estimate macroblock-based (MB-based) error propagation in [7]. On one hand, the model takes into account encoding parameters and picture characteristics to increase the accuracy of the estimation. On the other hand, it allows a much lower computational complexity than actual calculation.

The goal of this research is to provide a packet categorization method for prioritized packet delivery over the DiffServ network. It is designed for data partitioned video packets. The proposed method can quantify packet importance for prioritization based on the end-to-end quality by partitioned classes as well as the estimated loss impact for an individual packet. The complexity and accuracy can be coordinated according to the need and the limitation of a specific application. Generally speaking, our method can be applied to error resilient video transmission through unreliable channels with UEP, where the unreliable channel can be the DiffServ network in the Internet or the wireless channel and the UEP can be done via adaptive channel coding.

The rest of this paper is organized as follows. A packet delivery system with data partitioned video packets is described in Section 2. A data partitioning method for error resilience and its associated corruption model are presented in Section 3. An algorithm to estimate the distortion due to the loss of a packet is described in Section 4. Then, an optimization framework utilizing the RPI under UEP is introduced in Section 5. Experiments are performed to verify the proposed corruption model for error resilient ITU-T H.263+ video in Section 6. Finally, some concluding remarks are given in Section 7

2. SYSTEM OVERVIEW

A packetized video delivery system with the proposed corruption model is shown in Figure 1. It assumes the existence of a network that supports prioritized variable-rate delivery and the associated pricing mechanism. Since a video codec has several options to trade the compression efficiency for flexible delay manipulation, error resiliency, and network friendliness, the coordination framework has to provide an simplified interaction process between the video encoder and the target network. By utilizing the corruption model in the source coder and the network adaptation layer in an appropriate manner, the proposed delivery system can accommodate the demand of each packet to achieve the best end-to-end performance in adaptation to network fluctuating conditions. Under a given source and channel characteristics, efficient and optimal source and channel coding can be exploited under the coordination framework. The effectiveness of the system is however dependent on the accurate estimation of the distortion due to the channel error or packet drop , which is key investigation issue of this research.

The proposed delivery system consists of the video encoder, the packetizer, the proposed corruption model association module, the network adaptation module, the underlying delivery network, the de-packetizer, and the video decoder. Compressed video at the video encoder is packetized (maybe multiplexed with other media) at the packetizer. When the input video sequence is encoded, error resilient techniques can also be exploited. The resulting video packets are then associated with an index called RPI to account for its impact on video quality. In the DiffServ network, each RPI be mapped to a different DS level that controls the degree of loss, delay, and bandwidth. Successfully delivered packets are de-packetized, de-multiplexed and decoded for rendering. Generally, any reliable transmission scheme can be assumed as long as it supports the RPI-associated differentiation. For example, it can be applied to robust video transmission with adaptive forward error correction [5] or DiffServ packet forwarding [6].

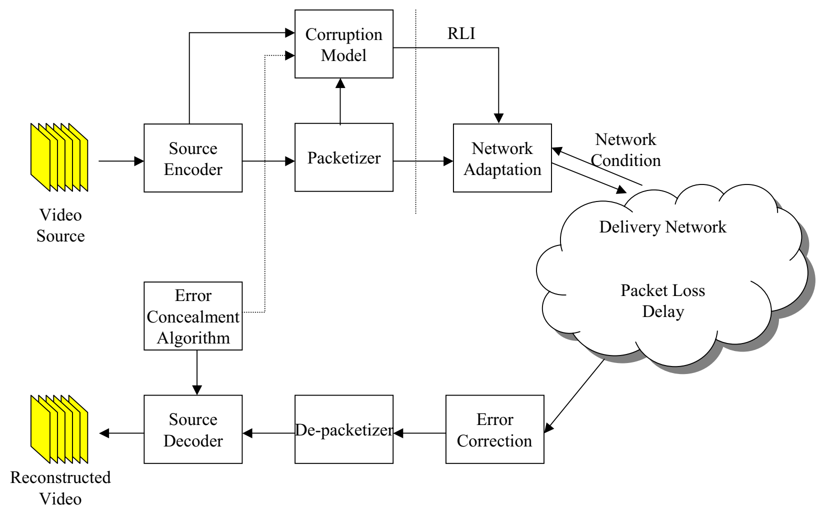

Figure 1. The packet video delivery system employing the RPI-based corruption model.

3. PACKET CATEGORIZATION WITH CORRUPTION MODEL

3.1. Data Partitioning

To apply unequal error protection to packets, packets should have different importance in their quality contribution. In traditional packetization schemes, such as group of block (GOB) in H.263 or the video packet in MPEG4, we do not see a wide range of importance differentiation. In the case of layered coding, it requires large redundancy to make separated bitstreams be syntactically independent. That is, each should have the synchronization code and the header information for picture and macroblocks. In contrast, data partitioned bitstreams are dependent on each other so that it does not require a duplicate header structure for picture and Macroblock, but a small amount of extra synch codes. As a results, data partitioned bitstreams still have high coding efficiency.

We adopt the data partitioning method similar to that specified by MPEG-2, where data partitioned packets are separated, *i.e.* each packet contains only data from one of different syntax categories. Syntax elements are classified into four classes: (1) picture headers, (2) Macroblock headers, (3) motion vectors and DC coefficients, and (4) AC coefficients. As shown in Fig. 2, encoded picture is partitioned into different classes. They are then packetized with a fixed size with data from the same class. In the packetization, since the picture header is too small to form an individual packet alone, we put picture headers and Macroblock headers together in one class so that one packet includes the picture header and Macroblock headers . Similarly, DC coefficients and motion vectors are packetized together. AC coefficients are packetized with themselves.

When a packet is decoded, codes are demultiplexed according to the syntax from a higher level to a lower one of partitioned bitstreams. For one picture decoding, the picture header is first decoded, and then the Macroblock header is decoded based on the decoded picture header. Thus, if the picture header is lost, the entire picture will be lost even though there are other lower level partition data. If the Macroblock header is decoded correctly, one can know how data follow for motion vector (MV) and AC coefficients. For instance, if the Macroblock header indicates an inter mode macroblock with 4MV, then there will be 4 motion vector differences and AC coefficient will follow if there is a coded block pattern included in the Macroblock header.

The number of AC coefficients can be known from the end of block (EOB) code in the AC coefficient bitstream. After all necessary AC coefficients are decoded, the next Macroblock header is then decoded from the MB header bitstream. As shown in the figure, there is dependency between different categories. Due to the dependency, the higher level data are more important than the lower layer data. DC and MVs can be decoded without AC coefficients in the corresponding MB. Also AC coefficients can be decoded without DC and MVs, if the MB header is decoded correctly. The importance of MV and AC coefficients can be different according to their contents. Their dependency can be determined by the error concealment algorithm of the decoder. When a decoder can use received

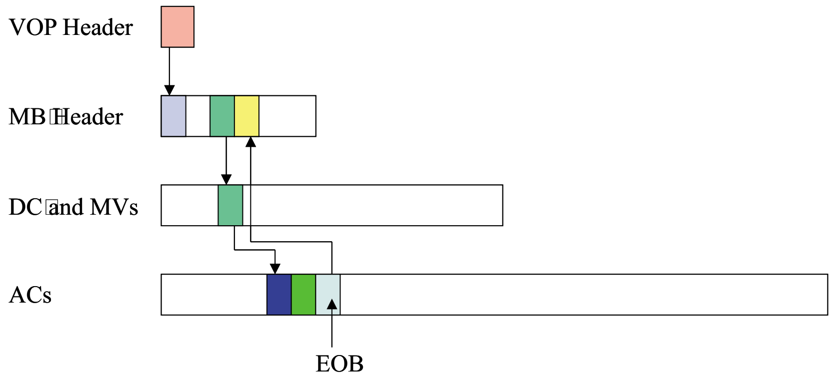

Figure 2. Classification for data partitioning

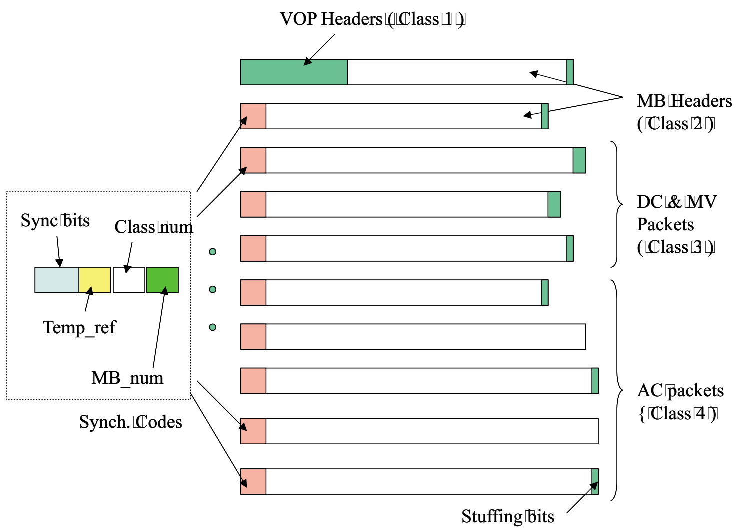

Figure 3. Packetization of a partitioned video bitstream

AC coefficients without MV coefficients, AC coefficients are not dependent on MV data. Otherwise, AC coefficients depend on the reception of MV data.

After data partitioning, packetization is performed for each partitioned data. In order to isolate decoding of each picture from others, we do not allow a packet to contain data from different picture. On the other hand, we make the picture header data belong to MB header data to avoid a very small packet. The packet size is roughly equal. However, this rule is not strictly applied. If the remaining data amount in one category is too small to match the size, it is added to the previous packet to form a larger packet. We construct each packet to keep MB boundary so that each new packet starts with the byte aligned MB data. The remaining bits in a packet are filled with stuffing bits. If the data amount of one partition is larger than the allowed packet size, it is split into multiple packets with synchronization codes. In addition to the synchronization code, the frame counter, the class number, and the MB number are inserted as shown in Fig. 3.

In the case of packet loss, the decoder should be able to use received data as many as possible to improve reconstructed video quality. Packet loss may make it difficult because a lost packet can cause the synchronization problem. The decoder should know which macroblocks correspond to the received paceket and which class the packet contains. For this purpose, the class number and the MB number follow the synchronization code. Temporal reference can be inserted for heavier packer loss or mis-ordered packet delivery that may cause mis-assignment of packets to an incorrect frame. In this work, we adopt 17 bits for the synchronization code, 4 bits for temporal reference, 2 bits for the class number and 9 bits for the MB number. Therefore, 4 bytes are added for the synchronization purpose.

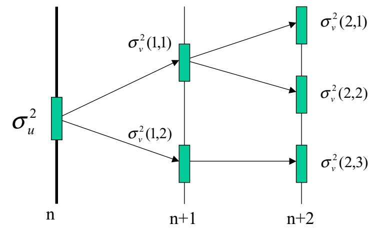

Figure 4. Error propagation of a lost MB.

3.2. Macroblock-based Corruption Model

The distortion can be quantified with the mean square error (MSE) between the decoded sequence in the decoder and the reconstructed sequence in the encoder. To calculate the MSE, it is required to decode the sequence from an impaired bitstream, which does not include data that belong to the lost packet. The MSE can be calculated by comparing the impaired sequence and the decoded sequence without packet loss (called the reference decoded sequence). We can perform this calculation for every packet. However, this would require a huge amount of computation. To avoid that, we propose a distortion estimation method based on a corruption model. With this model, we can estimate the packet loss impact with reasonable accuracy and moderate complexity.

There is dependency between packets from different categories. There are four cases in decoding a Macroblock in the decoder. Case 1 is when all data are available. Case 2 is when the header data (picture and MB header) and motion vectors are available but no coefficient data are available. Case 3 is when the header data and coefficient data are available but no motion vector. Case 4 is when no data for the macroblock is available. For each loss case, *i.e.* Cases 2 to 4, the decoder should reconstruct the MB as much as possible with the provided information. There are many ways to minimize the distortion for the loss of macroblock data. One way is error concealment, where the reconstructed MB is chosen to be the concealed MB. The impaired macroblock can lessen its distortion with the concealed macroblock. With the pre-estimated concealed macroblock, we can estimate the loss impact of each data categorized by proposed scheme. From Cases 2, 3 and 4, the coefficient loss impact, the motion vector loss impact and the header loss impact can be estimated, respectively.

For each case except for Case 1, the reconstructed macroblock with the condition could have error from the non-impaired macroblock. This is the first distortion due to the packet loss. Let us define this error as the initial error. For each macroblock, three different initial errors can be calculated based on the cases defined above. The initial errors can propagate when the macroblock is referenced to in the next frame. The propagating error in the next frame can propagate furthermore to successive frames. All errors initiated by the initial error can be considered as the loss impact of the first macroblock. The loss impact as we defined with MSE of the impaired sequence is equivalent to the sum of MSE of the impaired macroblock due to the initial error in the decoded sequence.

The calculation of the propagated error is much more complex due to the complexity of the motion compensated prediction codec. However, a simplified analytical model was proposed and applied to the macroblock based corruption model in [4]. The brief description is as follows.

Fig. 4 shows typical error propagation behavior. If a packet is lost, information for macroblocks that belong to the packet is not available so that decoder cannot properly reconstruct the macroblocks. Then, initial error $u(x,y)$ is introduced. x and y is spatial position of the error. The introduced initial error is propagated to the next frames due to motion compensated prediction in the decoder process. The propagation errors $v_i(x,y)$ can be multiple depending on the encoding mode of macroblocks in the next frames, where i is number of the propagated errors in different spatial and temporal position. In order to quantify the loss impact, we use mean square error of the errors that are caused by the macroblock loss. With an assumption of zero mean of $u(x,y)$ and $v_i(x,y)$, the error energy can be calculated with variances σ_u^2 and $\sigma_{v_i}^2$s.

Thus, the total error energy due to a MB loss in a sequence can be written as

$$\sigma^2 = \sigma_u^2 + \sum_{m=1}^{M}\sum_{j=1}^{N} W(m,j)\cdot\sigma_v^2(m,j)\,, \qquad (1)$$

where M is the size of the estimation window and N is the total number of MB in a frame, respectively. And m indicates the frame distance between the lost macroblock and the corrupted macroblock. $W(m,j) \cdot \sigma_v^2(m,j)$ term is the estimated distortion of j_{th} macroblock in the m_{th} frame due to the loss of current macroblock. Also, we have

$$\sigma_v^2(m.j) = \frac{\sigma_u^2}{1+\gamma_{m,j}}, \quad \text{and} \quad \gamma_{m,j} = \frac{\sum_{k=1}^{m} \sigma_{f_{k,j}}^2}{\sigma_g^2}, \tag{2}$$

where $\gamma_{m,j}$ is the error transfer factor for macroblock (m,j) w.r.t current macroblock. $\sigma_{f_{k,j}}^2$ is filter strength of error transition to macroblock (k,j) from macroblock in frame $k-1$. The corruption model derived above can be viewed as an MB-level extension of the statistical error propagation model given in [9].

According to the corruption model, the propagation behavior differs for a different initial error due to its frequency characteristics. Thus, the propagation error should be calculated for different cases of packet loss. For a data partitioned bitstream, a compressed macroblock data is partitioned into different packets and delivered in a different way. Therefore, there are possibilities that some codes are missed as described above. In each case, the initial error can be different. That is, the energy of the initial error and the frequency characteristics of the error can vary for different error cases. If the frequency characteristics are different, the propagation error will also be different because decaying factor γ is not the same. Thus, for one macroblock, we can have 3 different initial error energies and frequency characteristics, which are $(\sigma_{u_2}^2, \sigma_{g_2}^2)$, $(\sigma_{u_3}^2, \sigma_{g_3}^2)$, and $(\sigma_{u_4}^2, \sigma_{g_4}^2)$ for cases 1,2, and 3, respectively.

Note also that the initial errors depend on the error concealment algorithm. For each error case, the decoder can do its best to conceal the artifact of the macroblock with the received information. Different concealment macroblocks lead to different initial errors. Finally, Eq. (1) can be extended to

$$\sigma_i^2 = \sigma_{u_i}^2 + \sum_{m=1}^{M}\sum_{j=1}^{N} W(m,j) \cdot \sigma_{v_i}^2(m,j),\ i = 2,3,4, \tag{3}$$

where

$$\sigma_{v_i}^2(m.j) = \frac{\sigma_{u_i}^2}{1+\gamma_{m,j}}, \quad \text{and} \quad \gamma_{m,j} = \frac{\sum_{k=1}^{m} \sigma_{f_{k,j}}^2}{\sigma_{g_i}^2}, i = 2,3,4. \tag{4}$$

As shown in Eqs. (3) and (4), the same $W(m,j)$ and $\sigma_{f_{k,j}}^2$ are applied to all cases. This means, different errors propagate in the same path and are applied with the same prediction filter. With the proposed model, the loss impact of each component of the encoded macroblock can be quantified with the mean square error so that they can be compared for prioritization.

4. RPI GENERATION WITH THE PROPOSED CORRUPTION MODEL

In this section, an algorithm to calculate the propagation error will be described. RPI can be calculated from the error energy caused by macroblock loss. The error energy can be expressed as the variance of the error when the mean of the error is assumed zero. The variance is the sum of variances of the initial error and the propagation error as derived in the previous section. In order to calculate the variance, Eq. (3) is used for each macroblock. The variance of the propagation error is a sum of the variances of errors that are referenced to the lost macroblock in the direct or the cascade way.

In order to find corrupted macroblocks and calculate the variance of the propagation error, all macroblocks in the estimation windows should be examined in the forward direction to see if the macroblock is referenced or not. However, this method requires a very large amount of computation. Instead of finding corrupted macroblocks in the forward direction, the reference macroblocks of an encoding macroblock can be searched in the backward direction. In order to trace the error propagation in the backward direction, the encoding parameters for macroblocks in the reference frames should be stored for the estimation window period. The parameters are: the macroblock coding mode, motion vectors, the variance of the propagation error.

For the sake of convenience, let us denote a macroblock that is being encoded as the *current* macroblock. The reference macroblock of a current macroblock can be placed across adjacent encoded macroblocks in the reference frame. The encoded macroblock in the reference frame that has a part of the reference macroblock is denoted as a *contribution* macroblock of the current macroblock. The reference macroblock can be segmented as shown in

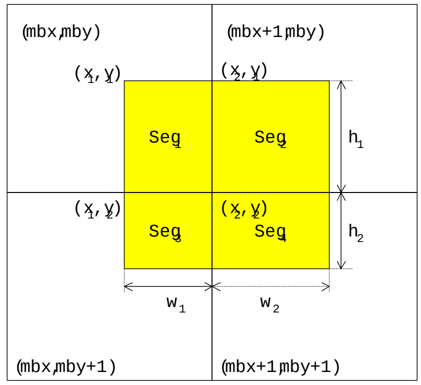

Figure 5. Illustration of the reference macroblock and four contribution macroblocks.

Figure 5. mbx and mby are macroblock's horizontal and vertical positions, respectively. (x_1, y_1), (x_2, y_1), (x_1, y_2), and (x_2, y_2) are top-left positions of possible segments. Each segment is placed in a contribution macroblock. The shaded area is the reference macroblock of the current macroblock. In this case, there are four contribution macroblocks positioned in (mbx, mby), $(mbx + 1, mby)$, $(mbx, mby + 1)$, and $(mbx + 1, mby + 1)$. The dependency weighting W can be calculated by $w1 * h1$, $w2 * h1$, $w1 * h2$, and $w2 * h2$ for four contribution macroblocks, respectively. All the contribution macroblocks of the current macroblock can be found in the estimation window by tracing backward using stored macroblock encoding parameters.

When each macroblock is encoded, the following procedure is executed. After one macroblock is encoded, encoding parameters of the macroblock are stored for the future use. For the initial error calculation, error concealment is performed for the current macroblock in the same way as the decoder does. The error concealment is performed for each error case. The difference of the concealment macroblock and the decoded macroblock is the initial error. The variances of initial errors, $\sigma_{u_i}^2$, $i = 2, 3, 4$ are calculated and stored.

To take into account the error propagation behavior, the filter strength σ_g^2 from the power spectral density (PSD) of the initial error should be used. However, measuring the PSD and σ_g^2 is complex. Instead of a direct calculation of these values, the transition factor γ_i is calculated for each possible filter strength. In this paper, we assume that only the half-pel motion prediction filter is used. For each horizontal, vertical, and diagonal direction half-pel prediction, the filter is applied to the initial error. Then, the variance of output data is calculated. We can get transition factors:

$$\gamma_i = \frac{\sigma_{u_i}^2}{\sigma_{v_i}^2} - 1, \quad i = 2, 3, 4, \tag{5}$$

where $\sigma_{u_i}^2$ is the variance of the initial error and $\sigma_{v_i}^2$ is the variance of the filtered data for each case. We can calculate transition factors by applying different filters. The transition factors are: γ_{i_h} for the horizontal half-pel filter, γ_{i_v} for the vertical half-pel filter, and $\gamma_{i_{hv}}$ for the diagonal filter. These values are stored instead storing $\sigma_{g_i}^2$ and σ_f^2.

After that, recursive calculation of the propagation error is performed if the encoding mode of the current macroblock is not intra-coded. The filter type can be determined from the half-pel flag of the motion vector. The coordinates of the reference macroblock is then calculated. One can trace error propagation recursively with parameters until the depth of trace in terms of the number of frames reaches zero. While the tracing is performed, the number of usage of each filter is counted from the current macroblock to each contribution macroblock. Let the number of usage be denoted by n_h, n_v, and n_{hv}, for the horizontal, the vertical, and the diagonal half-pel filters, respectively. Then, the transition factor from contribution macroblocks to the current macroblock can be calculated as

$$\gamma_i = n_h \cdot \gamma_{i_h} + n_v \cdot \gamma_{i_v} + n_{hv} \cdot \gamma_{i_{hv}}. \tag{6}$$

Based on γ_i and the stored initial error σ_u^2 of the contribution macroblock, the propagation error from the contribution macroblock to the current macroblock can be calculated as

$$\sigma_{v_i}^2 = W \cdot \frac{\sigma_{u_i}^2}{1 - \gamma_i}. \tag{7}$$

This is accumulated in the error propagation energy for the contribution macroblock. Next, the reference macroblock of the contribution macroblock is found with the stored motion vector of the contribution macroblock. These are recursively performed.

If the parameter *depth* is larger than zero and the contribution macroblock is encoded in the inter-mode, the trace is continued with recalculated parameters. This is performed for all segments. Finally, after one frame is encoded, the sum of variances of the initial error and the propagation error of each macroblock in the earliest reference frame in the estimation window is reported as the MSE of the error due to the macroblock loss.

4.1. Analysis of Computational Complexity

The number of operations for recursive calculation depends on the number of branches, which in turn depend on the motion vector. If a reference block is across four macroblocks, four new branches will be created. In the worst case, every subblock creates four new subblocks, then the nth frame has $4n$ nodes. It requires $O(4n)$ operations. However, the area will be decreases when a new branch is created. The probability of four new branches is proportional to the area.

Let us suppose T operations are required to calculate the propagation error from contribution macroblocks to the current macroblock, and the estimation window size is E. Then, the number of operations is $4(n-1)T$ per each macroblock ($n < E$) and this value is very small compared to motion estimation (e.g. for 16x16 half pel search range, 256x256 operations). Therefore, our corruption model can be used for practical applications such as the on-line streaming system.

5. CORRUPTION MODEL BASED RPI AND COORDINATION FOR NETWORK ADAPTATION

5.1. Corruption Model Based RPI

For packet video applications, the priority assignment for each packet in terms for loss and delay should reflect the influence of each packet to end-to-end video quality. In the case of delay, classification of video streams depends more on the application context (e.g., video conferencing or video on demand (VOD)) rather than video contents within a stream. Thus, we focus on the packet loss priority assignment in the network adaptation module and assign a relative priority index (RPI) to each packet.

In this work, we proposed a data partitioning scheme to differentiate packet importance. And, we developed corruption model to estimate packet loss impact of the data partitioned packet. The estimated packet loss impact for data partitioned packet has wide range so that it allows differentiated delivery system to fully utilize the differentiation ability to enhance End-to-End visual quality.

5.2. RPI-based Coordination for Network Adaptation

With the RPI-based corruption-model for each packet, a coordinated effort to deliver packetized video over QoS networks is investigated in this section. In particular, the packet-based protection framework with UEP is realized. This framework incorporates both the end-to-end video performance and pricing. The end-to-end video performance is measured in either objective quantity (e.g. the peak signal to noise ratio (PSNR) value) or subjective quality. In our approach, the impact on visual quality due to the loss of packet is represented by RPI (independent or dependent RPI).

In order to evaluate the total impact of the loss of $MB_{n,i}$, the weighted error variances for MBs of subsequent frames should be summed. Because the initial error can sustain over a number of frames without converging to zero, we have to limit frames to be evaluated under an acceptable computational complexity. Fortunately, when the intra-MB refresh technique is used at the encoder, the propagated error energy converges to zero within a fixed number of frames. Thus, in general, a pre-defined number of frames is sufficient to estimate the total impact of the MB loss. This defines an estimation window for the corruption model. The appropriate estimation window might be determined based on the strength of intra-MB refresh. As a result, the total energy of errors due to a MB loss in a sequence can be written as

$$E(i) = 10 \log\left(\sum_{n=-\infty}^{\infty} MSE(n,i)\right), \tag{8}$$

where

$$MSE(n,i) = \frac{1}{N} \sum_{(x,y)\in I} |\hat{R}_n^i(x,y) - R_n(x,y)|^2 \tag{9}$$

where $MSE(n,i)$ is Mean Square Error of n_{th} frame when i packet is lost. And (x,y) is coordinate of a frame, $R_n(x,y)$ is n_{th} frame of reconstructed video sequence when there is no packet loss is introduced. $\hat{R}_n^i(x,y)$ is n_{th} frame of reconstructed video sequence when i_{th} packet is lost. The $E(i)$ can be easily categorized to RLI value.

Given RPI-assigned packets, the network adaptation task can be formulated as follows. The loss impact of each packet expressed in terms of RPI is first categorized (i.e., normalized and quantized) among K categories according to the significance of the packet from the perspective of quality degradation . In fact, the actual categorization process may vary according to the employed network adaptation module and the underlying network service. Under the network delivery scenario, the resulting quality degradation depends on both the categorized RPI and delivery mechanism. The delivery mechanism for each packet is again categorized into level q among a total of Q levels anticipating price p according to q. For example, in the DiffServ network, DS level indicates different forwarding with which a certain level of QoS is assured.

Thus, the total quality degradation of video with N packets can be expressed as

$$QD = \sum_{i=1}^{N} QD(k(i), q(i)). \tag{10}$$

Since the total cost P for N packets are limited and each packet i costs $p_{q(i)}$, the optimal assignment of $\vec{q} = (q(1), q(2), ..., q(N))$ can be found by minimizing the total quality degradation. That is,

$$\min_{\vec{q}} QD = \min_{\vec{q}} \sum_{i=1}^{N} QD(k(i), q(i)), \tag{11}$$

subject to

$$\sum_{i=1}^{N} p_{q(i)} \leq P. \tag{12}$$

We can solve the problem by finding the service level mapping $\vec{q}$ that minimizes the Lagrangian formula

$$J_i(\lambda) = QD(k(i), q(i)) + \lambda \cdot p_{q(i)}. \tag{13}$$

The solution depends on $QD(k(i), q(i))$ and $p_{q(i)}$. The Lagrangian formulation of this problem is illustrated in [20]. The mapping function from categorized RPI to the quality delivery mechanism and the pricing strategy affects the mapping solution. With these two determined, the cost to quality degradation function can be derived for each packet. The solution is then to set each $p_{q(i)}$ equal to the price at which the slope $-\lambda$ line intersects the quality degradation curve. Since the total cost P is the sum of all packet costs, by adjusting λ, the cost-constraint (12) has to be met.

6. EXPERIMENTAL RESULTS

The proposed corruption model was first verified with simulations. Then, the QoS mapping was performed based on RPI that is calculated with the proposed corruption model. Simulations were carried out with the QCIF Foreman sequence, encoded by the H.263+ encoder at 30 fps and 300 kbps. The Intra-MB refresh was employed to increase robustness, and the proposed data partitioning scheme was applied to the encoded bitstream. Classes 1 and 2 were packetized together, class 3 and class 4 were packetized separately. The packet size was around 100 bytes. Evaluation of the impact of packet loss was performed for each packet. The error energy was estimated for each MB by using the proposed corruption model for data partitioned packets. In this simulation, the decoder performed error concealment for lost macroblocks. Temporal replacement was used for error concealment to reduce packet dependency which is significant in the case of motion compensated error concealment.

6.1. Verification of Proposed Corruption Model

The distortion capturing capability of the proposed model was verified by comparing the error propagation behavior over time. Fig. 6 shows the error propagation behavior of a single packet loss from Class 3 and Class 4. The actual MSE was measured by comparing with the reconstructed video without packet loss. Note that the estimated MSE with calculation matches the actual loss closely. In the case of AC coefficient loss (the left of Fig. 6), the higher frequency component decays rapidly over time due to the prediction filtering effect while motion vector loss introduce a largee DC error which lasts longer.

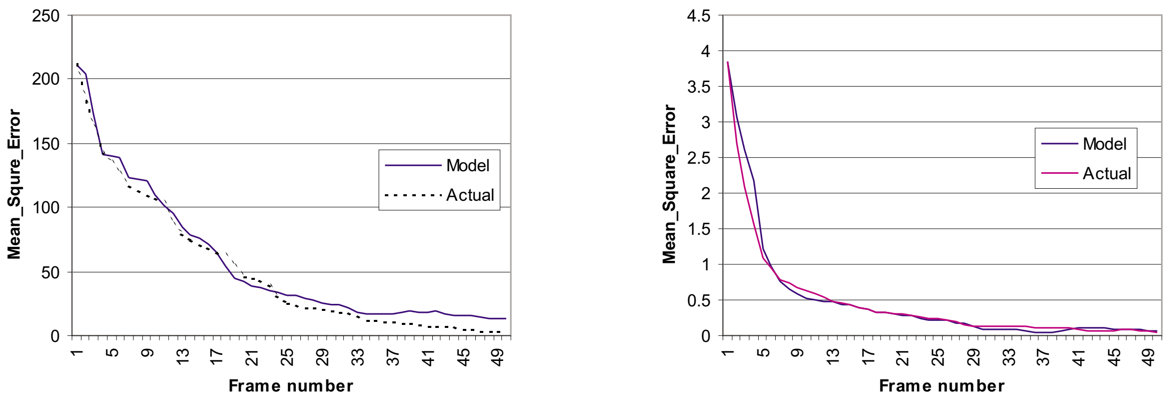

Figure 6. The packet loss effect of a packet of Class 3 (right) and Class 4 (left).

Packets from the same class can have a different packet loss impact. The distribution of packets of each class is given in Fig. 7. The packet loss impact was estimated with the proposed corruption model. The loss impact of Class 4 packets is relatively small compared to that of other class packets. However, some packets from Class 4 have a higher packet loss effect, which is comparable to that of Class 3 packets and Class 1&2 packets. The packet loss impact of Class 3 and Class 4 has a similar distribution. The motion vector loss gives a big impact on the decoded sequence as much as the header data loss.

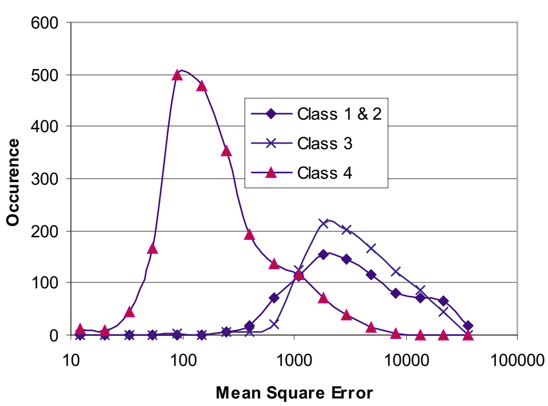

Figure 7. MSE distribution for packet classes.

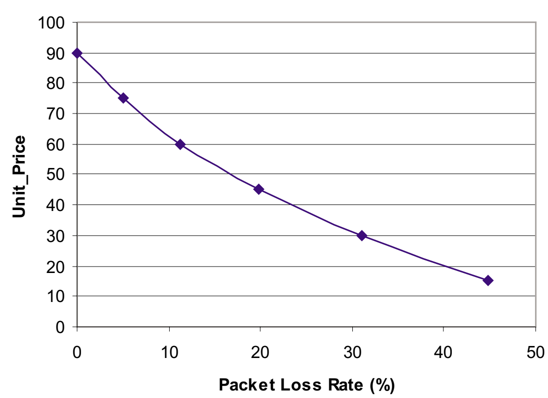

Figure 8. The packet loss rate for the unit packet cost.

6.2. RPI-based Coordination of Network Adaptation

The RPI-based coordination of network adaptation was evaluated under the proposed video delivery framework given in Fig. 1. The test sequence and encoding parameters were the same as those given in the previous section. The network delivered prioritized packets with different reliability property corresponding to the service level, which was rated according to the packet loss rate. For coordination, the Lagrangian-based mapping method was implemented to satisfy the given cost-constraint.

With the proposed corruption model, dependent RPI was calculated for all 3998 data partitioned packets of the Foreman sequence. RPI was then categorized into 6 different levels. Each level had a different packet loss rate and the unit cost for transmitting a packet. The relationship between the unit cost and the corresponding packet loss rate is shown in Fig. 8.

For performance comparison, the video packet(VP) scheme was used as the reference. VP has the synchronization code and the header information. It consisted of 3702 packets, which is smaller than that of data partitioning. As a result, a higher price can be assigned than that of the data partitioned packet. Given the total cost and the pricing mechanism, Optimal QoS mapping is applied to data partitioned packets with the Lagrangian method. However, VP is assigned to a level which has the packet loss rate corresponding to the same total cost as DP packets.

Fig. 9 shows the average packet loss rate for the total cost. VP has a lower packet loss rate at the same total cost in the pricing mechanism. The end-to-end quality is given in Fig. 10. The figure shows that the differentiated DP has a higher packet loss rate, but provides higher end-to-end quality for a noisy channel. The advantage of the RPI assignment with data partitioning is obvious.

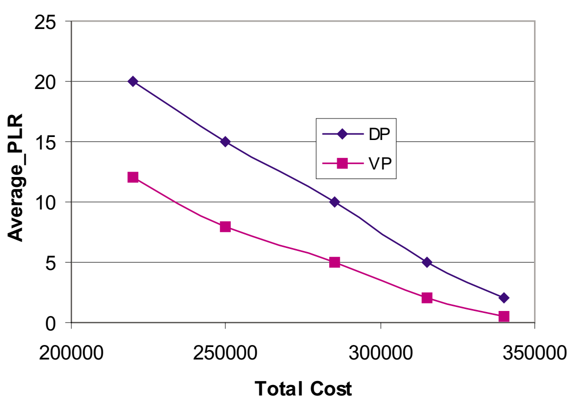

Figure 9. The total cost versus the average packet loss rate.

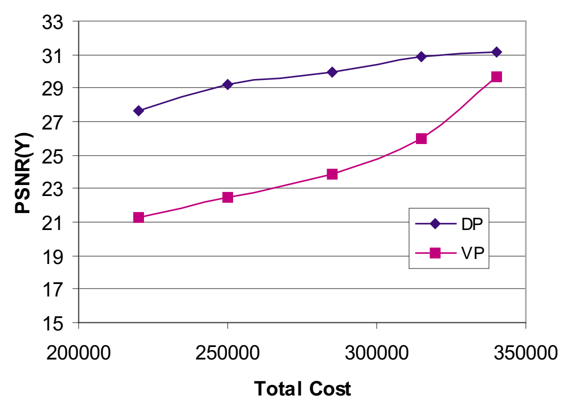

Figure 10. Performance comparison for data partitioned and video packets.

7. CONCLUSION

We proposed a corruption model for data partitioned video packets. The data partitioning scheme provides the capability to differentiate the packet importance, which was accurately estimated with the proposed corruption model. The resulting estimation approximates the real loss impact within a narrow margin while requiring a small computational overhead. When applied in association with the RPI assignment, the corruption model-based RPI satisfies the requirement of the proposed coordinated packetized video delivery and provides a reasonable performance improvement.

REFERENCES

1. ITU-T Recommendation H.263 Version 2 (H.263+), *Video coding for low bit rate communication*, Jan. 1998.

2. ISO/IEC 13818, "Information Technology Generic Coding of Moving Pictures and Associated Audio Information," Part 2: Video, 1996.
3. Motion Pictures Experts Group, "Overview of the MPEG-4 standard", ISO/IEC JTSC1/SC29/WG11 N2459, 1998.
4. Jin-Gyeong Kim, JongWon Kim and C.-C. Jay Kuo, "Coordinated packet level protection employing source and channel redundancy for robust video transmission," Conference on Visual Communications and Image Processing, Part of the Symposium on Electronic Imaging 2001, San Jose, CA., 21-26 January 2001
5. W. Kumwilaisak, J. Kim and C.-C. J. Kuo, "Reliable Wireless Video Transmission via Fading Channel Estimation and Adaptation," in *Proc. WCNC 2000*, Sept. 2000.
6. J. Shin, J. Kim and C.-C. J. Kuo, "Content-based packet video forwarding mechanism in differentiated service networks," in *Proc. Packet Video Workshop 2000*, May 2000.
7. J.-G. Kim, J. Kim and C.-C. J. Kuo, "On the corruption model of loss propagation for relative prioritized packet video," in *SPIE Proc. Applications of Digital Image Processing XXIII*, July 2000.
8. Gero Base, Jurgen Pandel, Sebastian Purreiter, Thomas Stockhammer, "Data partitoing for packcet oriented H.26L - a network friendly interface", ITI-T Study Group 16, Q.15/SG16 Osaka, 16-18 May, 2000
9. N. Farber, K. Stuhlmuller, and B. Girod, "Analysis of error propagation in hybrid video coding with application to error resilience," in *Proc. IEEE ICIP '99*, Oct. 1999.
10. G. Reyes, A. R. Reibman, and S.-F. Chang, "A corruption model for motion compensated video subject to bit errors," in *Proc. Packet Video Workshop '99*, Apr. 1999.
11. R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra mode switching for packet loss resilience," *IEEE J. Select. Areas Communications*, vol. 18, no. 6, June 2000.
12. M. H. Willebeek-LeMair, Z.-Y. Shae, and Y.-C. Chang, "Robust H.263 video coding for transmission over the Internet," in *Proc. INFOCOM '98*, March 1998.
13. Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *Proceedings of the IEEE*, vol. 86, no. 5. May 1998.
14. B. Girod, "Feedback-based error control for mobile video transmission," *Proceedings of the IEEE* , vol. 87, no. 10, Oct. 1999.
15. J. Kim, W. Kumwilaisak, and C.-C. J. Kuo, "Cross-validation of proposed data partitioning annex for enhanced error resilience," *ITU-T standardization Sector Q.15/SG16* , Document Q15-G-23, Feb., 1999.
16. ITU-T, *Video codec test model near-term version 10 (TMN10),* Q.15/SG16, Document Q15-D-65, Apr. 1998.
17. C.-S. Kim, R.-C. Kim, and S.-U. Lee, "An error detection and revovery algorithm for compressed video signal using source level redundancy," in *IEEE Trans. on Image Processing*, vol. 9, no. 2, Feb. 2000.
18. U. Horn, K. Stuhlmuller, M. Link, and B. Girod, "Robust Internet video transmission based on scalable coding and unequal error protection," in *Signal Processing: Image Communication*, vol. 15, no. 1-2, Sept. 1999.
19. D. Wu, Y. T. How, and Y.-Q. Zhang, "Transporting real-time video over the Internet: Challenges and approaches," in *Proceedings of the IEEE*, vol. 88, no. 12, Dec. 2000.
20. A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," in *IEEE Signal Processing Magazine*, Nov. 1998.
21. L. Zhang, D. Chow and C. H. Ng, "Cell loss effect on QoS for MPEG video transmission in ATM networks ," in *IEEE ICC '99*, 1999.
22. Wuttipong Kumwilaisak, JongWon Kim, and C.-C. Jay Kuo, "Video transmission over wireless fading channels with adaptive FEC," in*Picture Coding Symposium* 2001, Seoul, Korea, Apr. 2001.