Adaptive MPEG-2 Information Structuring

Pascal Frossard[a] and Olivier Verscheure[b]

[a]Signal Processing Laboratory
Swiss Federal Institute of Technology, Lausanne, Switzerland

[b]Institute for Computer communications and Applications
Swiss Federal Institute of Technology, Lausanne, Switzerland

September 20, 1998

ABSTRACT

This work addresses the optimization of TV-resolution MPEG-2 video streams to be transmitted over lossy packet networks. This paper introduces a new scene-complexity adaptive mechanism, namely the Adaptive MPEG-2 Information Structuring (AMIS) mechanism. AMIS adaptively modulates the number of resynchronization points (i.e the slice headers and intra-coded macroblocks) in order to maximize the perceived video quality assuming it is aware of the packet loss probability and the error concealment technique implemented in the decoder. The perceived video quality depends both on the encoding quality and the degradation due to data loss. Therefore, AMIS constantly determines the best compromise between the rate allocated to pure video information and the rate aiming at reducing the sensitivity to packet loss. Results show that the proposed algorithm behaves much better than the traditional MPEG-2 encoding scheme in terms of perceived video quality under the same traffic constraints.

1. INTRODUCTION

Audiovisual applications (e.g., video conferencing, video on demand, teleteaching, etc.) are foreseen as one of the major users of broadband networks (i.e. IP and ATM networks). At the heart of this revolution is the digital compression of audio and video signals. The biggest advantage of compression resides in data rate reduction which results in a decrease of transmission costs. The choice of the compression algorithm mostly depends on the available bandwidth or storage capacity and the features required by the application. The MPEG-2 standard,[1] a truly integrated audio-visual standard developed by the International Organization for Standards (ISO), is capable of compressing NTSC or PAL video into an average bit rate of 3 to 6 Mbits/s with a quality comparable to analog CATV.[2]

Several studies[3–5] have already been carried out on the MPEG-2 transmission over lossy networks area. However, work remains to be done to optimize multimedia applications so they can be offered at attractive prices. In other words, the user expects an adequate audio-visual quality at the lowest possible cost. From the user's viewpoint, in the case of video transmission over packet networks, both the encoding and the transmission processes affect the end-to-end quality of service. The most economic offering can thus only be found by considering the entire system and not by optimization of individual system components in isolation.[6]

In this work, we introduce our adaptive MPEG-2 information structuring (AMIS) algorithm. AMIS adaptively modulates the number of slice headers and intra-coded macroblocks in order to minimize the impact of data loss, and thus maximize the perceived video quality. AMIS computes the visual impact of hypothetical data loss in order to determine the most vulnerable locations in the bitstream. It is assumed that the encoding mechanism is aware of the packet loss probability and the error concealment technique implemented in the decoder.

The paper is organized as follows: Section 2 gives a brief overview of the MPEG-2 video communications area. Section 3 first starts by describing the experimental setup used throughout this paper and then presents some preliminary studies. The AMIS algorithm is presented in details in Sec. 4. Some comparative results are given in Sec. 5. Finally, concluding remarks are provided by Sec. 6.

Emails: {Pascal.Frossard, Olivier.Verscheure}@epfl.ch

2. MPEG-2 VIDEO COMMUNICATIONS

2.1. MPEG-2 Video Compression

An MPEG-2 video stream is highly hierarchically structured.[1]   The smallest entity defined by the standard is the *block* which is an area of 8×8 pixels of luminance or chominance. A *macroblock* (16×16 pixels) contains four blocks of luminance samples and two, four or eight blocks of chrominance samples, depending on the chrominance format. A variable number of macroblocks is encapsulated in an entity called a *slice* which shall start and terminate on the same line. Each *picture* is then composed of a variable number of slices.

The MPEG-2 video syntax defines three different types of pictures: intra-coded (I)*, predicted (P) and bidirectionally predicted (B). The use of these three picture types allows MPEG-2 to be robust to packet loss (I-pictures provide stop points for the error propagation) and efficient (B- and P-pictures allow good compression through motion estimation). All coding modes can even be chosen per macroblock which allows fine-tuned tradeoffs of robustness and efficiency.

Before being transmitted, the output of the video encoder goes through the MPEG-2 transport stream (TS) layer. Basically, the stream is segmented into variable-length packetized elementary stream (PES) packets and then subdivided into fixed-length TS packets (188 bytes). These packets are then encapsulated following the protocol of the underlying transmission network (e.g. ATM or Internet).

Finally, it should be noted that almost all the entities defined by the MPEG-2 standard (e.g. slice, picture, TS, PES) are preceded by a header.

2.2. MPEG-2 Video Sensitivity to Data Loss

In an MPEG-2 video stream, data loss reduces the quality in relation to the importance of the lost information type: losses in headers affect the quality more than losses of DCT coefficients and motion vectors. The quality degradation depends also on the picture type of the lost video data because of the predictions used for MPEG-2.

Figure 1 shows how transmission losses map into visual information losses in different types of pictures. Data loss spreads within a single picture up to the next resynchronization point (e.g., picture or slice headers) due to the variable-length and differential coding within slices. This is referred to as spatial propagation. When loss occurs in a reference picture (I- or P- picture), the lost macroblocks will affect the predicted macroblocks in subsequent frame(s). This is known as temporal propagation.
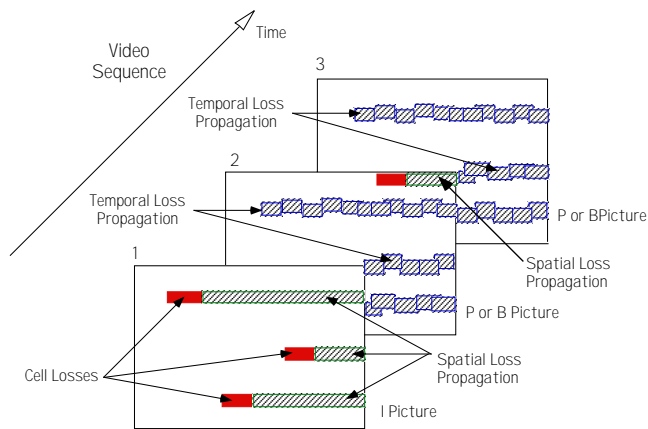

Figure 1. Data loss propagation in MPEG-2 video streams.

The impact that loss of header information may have is in general more important and more difficult to recover than the loss of pure video information (e.g. DCT coefficients, motion vectors). For instance when a frame header is lost, the entire frame is skipped since the decoder is not able to detect its beginning. If the skipped frame is a reference picture, the temporal error propagation may greatly reduce the perceptual quality. So, when a header is lost, in general, the whole underlying information is skipped. Some headers are thus more important than others.

---

*Intra-coded means aims at reducing the spatial redundancy only

Error concealment is generally used to reduce the impact of data loss on the visual information. The algorithms include, for example, spatial interpolation, temporal interpolation and early resynchronization. The MPEG-2 standard[1] proposes an elementary error concealment algorithm based on *motion compensation*. It estimates the vectors for the lost macroblock by using the motion vectors of neighbouring macroblocks in the affected picture (provided these have not also been lost). This improves the concealment of moving picture areas. There is however an obvious problem with lost macroblocks whose neighbours are intra-coded, since there have ordinarily no associated motion vectors. To get around this problem, the encoding can include motion vectors also for intra macroblocks [†]. Though error concealment may, in general, efficiently decrease the visibility of data loss, severe data loss may however still lead to annoying degradations in the decoded video quality.

The robustness of compressed MPEG-2 video may be dramatically reduced by judiciously inserting resynchronization points in the bit stream (i.e. slice headers to limit the spatial propagation and intra-coded macroblocks to stop the temporal propagation). However, the addition of extra slice headers and/or intra-coded macroblocks is not costless. Indeed, it reduces the amount of bits available to code pure video information under the same video traffic constraints (or, equivalently, it increases the bit rate to be sent throughout the network).

3. PRELIMINARY STUDY

In this section, we first describe the experimental setup that has been used throughout this work. The MPEG-2 Test Model v5 (TM5) framework is then depicted in terms of slice headers location and intra-coded macroblocks encoding (i.e. resynchronization points). The impact of adding extra resynchronization points in both constant bit rate (CBR) and variable bit rate (VBR) MPEG-2 streams is finally analyzed.

3.1. Experimental Setup

Our experimental testbed includes the following elements:

- An MPEG-2 software encoder composed of a TM5 video encoder[7] and a transport stream encoder. A 500-frame long sequence conforming to the ITU-R 601 format has been used. It includes five video scenes that differ in terms of spatial and temporal complexities. It has been encoded as interlaced video with a structure of 12 images per GOP and 2 B-pictures between every reference picture. Motion vectors have been generated for all macroblocks. Before being transmitted, the MPEG-2 video bitstream is encapsulated into 18800-byte packetized elementary stream (PES) packets and divided into fixed-length transport stream packets by the MPEG-2 system encoder.

- A model-based data loss generator. For this purpose, we used a two-state Markovian model (Gilbert model[8]) with which three parameters can be controlled: the packet size (PS), the packet loss ratio (PLR) and the average length of a burst of errors (ABL). In our simulations, we imposed a non-bursty ($ABL = 1$) TS packets ($PS = 188$ bytes) loss process and varied the packet loss ratio between 10^{-2} and 10^{-7}. It should be noted that the MPEG-2 encapsulation schemes defined for the transmission over both Internet and ATM networks produce fixed-length packets ($x \times 188$ where x is an integer greater or equal to 1).

- Video quality was evaluated by means of the MPQM tool[9] which proved to behave consistently with human judgments. The per-frame quality values produced by the MPQM tool are averaged throughout the sequence.

- An MPEG-2 software decoder composed of a TS decoder and a video decoder. The video decoder implements the motion compensated concealment technique briefly presented in the previous section.

3.2. MPEG-2 TM5 Framework

Slice headers: The MPEG-2 standard allows for building slices with a variable number of macroblocks. The only restriction is that a new slice shall start on every new line of macroblocks and that slices shall occur in the bitstream in the order in which they are encountered. The most widely accepted MPEG-2 TM5 implementation[7] limits to the minimum the number of slices per frame in respect to the standard. In this scenario, every frame of a TV-resolution PAL sequence (720*576 @ 25 fps) is composed of $\frac{720}{16} = 45$ slices. Every slice further encapsulates $\frac{576}{16} = 36$ macroblocks.

[†]Some MPEG-2 encoder chips automatically produce concealment motion vectors for all macroblocks.

Intra-coded macroblocks: The MPEG-2 standard does not specify when a macroblock might be intra-coded in a non-intra picture[‡]. The MPEG-2 TM5 implementation encodes a macroblock as intra based upon a per-macroblock activity metric. In other words, a macroblock is intra-coded in a non-intra picture when all the other coding modes suitable for the picture type would output a higher number of bits.

3.3. Extra Resynchronization Points

Slice headers: As previously mentioned, slice headers limit the spatial error propagation due to data loss.[10] However, the greater the number of slices, the bigger the overhead. Indeed, each new slice introduces a 5- to 6-byte length header and resets the differential coding of the DC values and motion vectors. Therefore, in OL-VBR encoding, extra slice headers increase the average bit rate (Fig. 2(a)). Moreover, we see that differential coding is not the predominant factor in comparison to the amount of header information added (SH stands for Slice Headers in Fig. 2(a)). Similarly, the addition of extra slice headers reduces the video quality in CBR encoding, since under a fixed bit budget, less bits may be used for pure video information. Fig. 2(b) illustrates this behavior. It is shown that the impact on quality increases when the encoding bit rate decreases. Furthermore, the addition of 1 or 2 slices per line of macroblocks is barely noticeable.
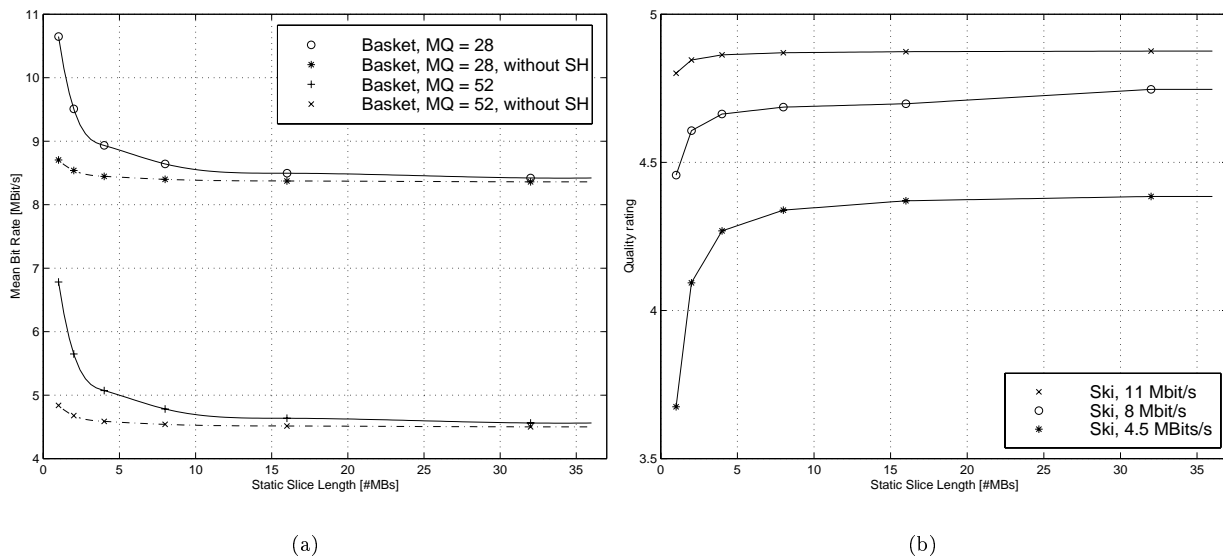

(a)   (b)

Figure 2. Slice length impact on (a) OL-VBR and (b) CBR encoding for different bit rates.

Intra-coded macroblocks: Intra-coded macroblocks stop the temporal propagation of areas damaged by packet loss. But, again, the greater the number of intra-coded macroblocks, the higher the overhead. The amount of overhead generated is however not easy to quantify. Indeed, it depends on the encoding complexity of each extra macroblock encoded in intra mode.

3.4. Problem Formulation

It has been shown that extra slice headers and intra-coded macroblocks decreased the video quality in CBR encoding (increased the average bit rate in OL-VBR encoding) while increasing the robustness of the bitstream to data loss. There is thus a trade-off between the encoding quality and the bitstream robustness. Moreover, the efficiency of adding extra resynchronization points in the bitstream strongly depends on the content type of the corresponding protected video area. Indeed, the insertion of resynchronization points where the impact of data loss would not affect the video quality (under a given error concealment technique) leads to a suboptimal scenario.

[‡]Macroblocks of an I-picture are obviously all intra-coded

In the following section, we first describe the distortion metric that will be used to predict the impact an hypothetical data loss would have on a given video area. We then study how a packet loss both spatially and temporally propagates throughout the sequence. It will help our algorithm (AMIS) to improve the robustness of the most vulnerable bitstream areas. We finally describe AMIS, which adaptively modulates the number of both slice headers and intra-coded macroblocks according to the expected packet loss probability and the error concealment technique implemented in the decoder.

4. AMIS: ADAPTIVE MPEG-2 INFORMATION STRUCTURING

4.1. Distortion Metric

The distortion metric chosen in this paper is one of the most commonly used metric, the mean luminance difference. In a coarse approximation, it corresponds to the simplest metric correlated with human perception (under the assumption that the viewer stands far enough from the monitor).[11]

In MPEG-2, the error propagates spatially within slices, and temporally between adjacent pictures (see Sec. 2.2). These two phenomenons are distinct and therefore need a different distortion metric definition. For spatial error propagation, the distortion metric computes the impairments between a correctly decoded macroblock and its substitute after loss (i.e. the macroblock obtained after using the error concealment technique). In this case, the mean luminance difference can be expressed as follow

$$\delta_s(i) = \frac{1}{256} \sum_{p=1}^{256} M_i(p) - \frac{1}{256} \sum_{p=1}^{256} \hat{M}_i(p), \tag{1}$$

where M_i and $\hat{M}_i$ represent the ith macroblock in respectively the correctly decoded frame and the concealed frame. The index p is the pixel position in the corresponding macroblock.

For temporal error propagation, on the other hand, the distortion is caused by loss in previous reference frames, and not in the current frame any more. The perceptual relevance of a macroblock should report of the visual difference between a correctly decoded area and the same area damaged by temporally propagated impairments. The mean luminance difference can then be expressed as follows:

$$\delta_t^k(i) = \frac{1}{256} \sum_{p=1}^{256} M_i(p) - \frac{1}{256} \sum_{p=1}^{256} \tilde{M}_i^k(p), \tag{2}$$

where M_i and $\tilde{M}_i^k$ are the ith macroblocks respectively in the frames F_n and $\tilde{F}_n$, n representing the decoding reference frame index. The frame F_n is the current frame, decoded from a lossless bitstream, and the frame $\tilde{F}_n$ is the substitute of F_n in case of loss and concealment in a previous reference frame F_{n-k}. The indexes k and n are reported to the reference frame encoding order [§]. It is assumed here that no loss occurs in the reference frames between frames F_{n-k} and the current frame F_n, besides the loss in F_{n-k}.

The AMIS mechanism is based on a probability weighted distortion measure. In other words, it computes not only the distortion due to an hypothetical loss in the bitstream, but considers also the probability for this loss to occur. In the next sections we compute these probabilities, for both spatial and temporal error propagation. The probability to loose the macroblock M_i will be called $L(i)$ and $\mathbf{E_n^k}$ will represent the probability that pixels in the frame F_n suffer from data loss in F_{n-k}.

4.2. Pixel Loss Probability

The macroblock loss probability is defined as the probability for the macroblock information in the bitstream to be lost, entirely or even partly. A macroblock is considered as lost either when encapsulated in a lost transmission packet or when previous macroblocks of the same slice are lost, due to spatial error propagation (see Sec. 2.2) . The macroblock loss probability is then strongly dependent on both the packet loss process and the highly hierarchical structure of MPEG-2.

[§]Only the I- and P-frame are considered, since the B-frames do not propagate errors in subsequent frames

Let the transmission packet loss process be modelized by a two-state Markovian model (Gilbert model[8]). It is assumed that a uniform loss distribution represents the worst loss pattern in a MPEG-2 video transmission in terms of video quality. Therefore, the probability for a TS packets to be lost is set to PLR (Packet Loss Ratio), with a loss burst length of one.

At this stage, without any other information about the loss process, each packet has the same average probability to be lost. Therefore all macroblocks have the same probability to be part of lost packets, unless they are encapsulated into several transport packets. Indeed, if no assumption is made about the video stream transport, a macroblock can be encapsulated into several loss entities, especially at high rates. Consequently, losses obviously have a higher probability to occur in the largest encoding size areas since these regions need more fixed-size packets to be encapsulated in.

Within a frame, the probability for a macroblock M_i to be lost is given by:

$$\lambda(i) = PLR \times P(i), \tag{3}$$

where $P(i)$ is the number of packets the macroblock M_i belongs to. The larger the number of packets P, the higher the probability for the macroblock to be lost. Generally, the loss entities (e.g. MPEG-2 TS packets) are larger than the macroblock size, even at high encoding rates, and the macroblocks belong to at most two packets.

As stated above, another process must be considered now, the spatial loss propagation. In case of loss, an unsupervised MPEG-2 decoder skips all video information up to the next encountered slice header, which acts as a spatial resynchronization point. Consequently, when a macroblock is lost within a slice, all subsequent macroblocks of the same slice are also considered as being lost even if they do not belong to the lost packet.

Finally, the macroblock loss probability, which should be understood as the probability that a macroblock could not be normally decoded, can be expressed as follows:

$$L(i) = \lambda(i) + PLR \times S(i) = [P(i) + S(i)] \times PLR \tag{4}$$

where $S(i)$ represents the number of transmission packets that incorporates information from macroblocks located in the same slice, before M_i. Obviously the packets that contain information about M_i are not part of S (see Fig. 3) since this situation is taken into account by Eq. 3.
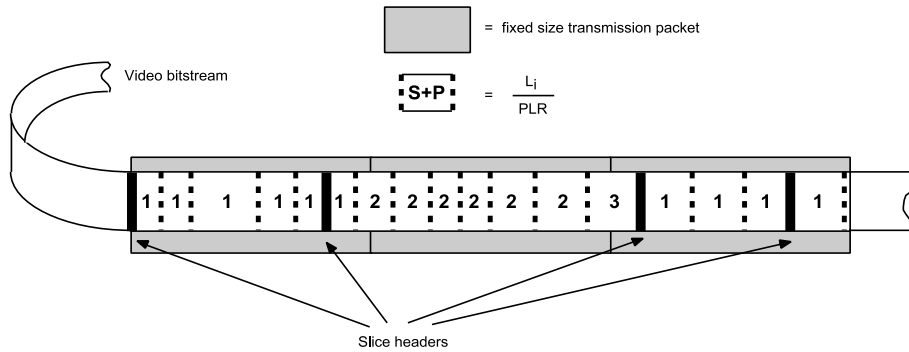

Figure 3. Example of $L(i)$ values

However, there is an exception to the rule defined hereabove. Indeed, according to the MPEG-2 syntax each picture is preceded by a header. When the packet containing the picture header is lost, the entire frame is generally skipped, making the previous computation totally useless. This case, of very small probability anyway, could nevertheless be neglected in this development.

Finally, for the following sections, there is a need to increase the probability calculation granularity to the pixels, instead of the macroblocks. From Eq. 4, a pixel probability map can easily be drawn. The map $\mathbf{L_n}$ is the loss probability matrix of the frame F_n. In this map, each pixel of the macroblock M_i has the same value $L(i)$ (see Fig. 4).Within a macroblock, each pixel has the same probability to be lost, and this probability is non-decreasing with the macroblock relative position within the slice.
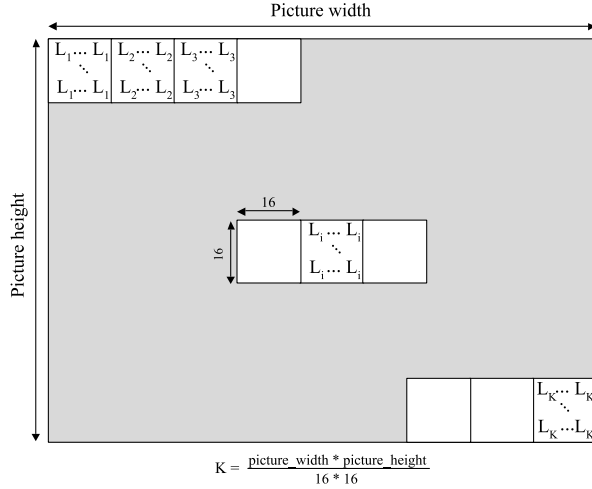
Figure 4. Representation of $\mathbf{L_n}$, the map of pixel loss probability values

4.3. Erroneous Pixel Probability

In the previous section, the probability for a pixel to be lost has been analyzed. Let's now define the erroneous pixel probability as the probability for a pixel to be erroneously decoded due to temporal error propagation. In a simplification purpose, the erroneous pixel probability matrix will generally be referred to a particular previous reference frame.

This probability is more difficult to compute than the loss probability. Two phenomenons have to be taken into account: the temporal and the spatial propagation of errors. The second phenomenon is due to spatial error propagation in previous reference frames which influence the corresponding erroneous pixel probability. Now, as the only way to reset the temporal propagation of error is intra coding, a pixel will be not only damaged by loss within its direct reference frame (the previous I- or P- frame), but also by losses within other previously decoded reference frames, unless the referred areas are intra-coded.

Moreover, to analyze the effect of temporal error propagation, the way the motion compensation is performed should be considered. Video areas each pixel refers to could be found by recursively following the successive motion vectors within the video sequence. However, motion vectors generally do not refer to the macroblocks as entities, but rather as 16×16 pixels areas, without macroblock boundaries considerations. This means that, within a macroblock, even though each pixel has the same probability to be lost, it does not have the same probability to be decoded into an erroneous value.

This study will be conducted in two steps. First, the error pixel probability matrix will be computed for losses occurring in the reference frame right before F_n (i.e. F_{n-1}). Then, the influence of losses in any of the previous reference frame, called F_{n-k}, with $k \leq n$, will be computed. Finally, the general erroneous pixel probability matrix $\mathbf{E_n^k}$ will be computed.

Let's consider the frame F_{n-1} as the direct reference frame of the current frame F_n (i.e $k = 1$). Given the motion vectors of F_n, the pixel probability to be damaged by loss occurring in F_{n-1} can be computed by mapping the loss probability matrix of F_{n-1}, following the same process as in the pixel motion compensation. In other words, the motion estimation is performed with the motion vectors given for the frame F_n, but with reference to $\mathbf{L_{n-1}}$. The $\mathbf{L_{n-1}}$ matrix acts as any luminance frame and replaces the reference frame F_{n-1}. Such a probability matrix mapping could be represented by the function $\mathcal{M}_n$, where the index n is referred to the use of the motion vectors of the frame F_n (see Fig. 5).

Following the proposed notation, the probabilities that pixels of F_n are erroneously decoded due to loss in F_{n-1} are given by the matrix $\mathbf{E_n^1}$. It is obtained by referencing $\mathbf{L_{n-1}}$ according to the motion vectors of the frame F_n, through the mapping function $\mathcal{M}_n$. However, the loss pixel probability in the current frame F_n has also to be considered. Indeed, there is no need to compute the probability for a pixel to be damaged in a previous frame if it is lost in the current frame. Finally, the matrix $\mathbf{E_n^1}$ could be written as follows:
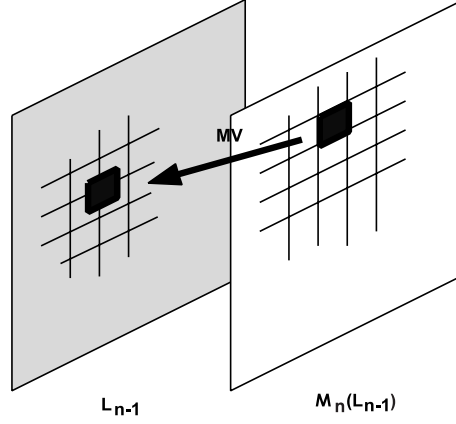
7

Figure 5. Mapping function $\mathcal{M}_n$ representation.

$$\mathbf{E_n^1}(p) = \mathcal{M}_n(\mathbf{L_{n-1}})(p) \times \bar{\mathbf{L}}_{\mathbf{n}}(p), \tag{5}$$

where $\mathbf{L_n}$ and $\mathbf{L_{n-1}}$ are the lost pixel probability matrices defined in Fig. (4) and

$$\bar{\mathbf{L}}_{\mathbf{n}}(p) = 1 - \mathbf{L_n}(p) \tag{6}$$

Then, according to the previous relation, a pixel is erroneous when it has not been lost but refers to a lost pixel in the reference frame. Obviously, if a pixel q does not have any correspondence in the reference frame F_{n-1}, or belongs to an intra-coded macroblock,

$$\mathbf{E_n^1}(q) = 0. \tag{7}$$

The next step is the generalization of the Eq. (5) to consider not only losses in the direct reference frame F_{n-1}, but in any of the kth previous reference frame, with $k \leq n$. The generic erroneous pixel probability matrix $\mathbf{E_n^k}$, which captures the influence of losses in the frame F_{n-k} can be obtained by recursivity.

Indeed, similarly to Eq. (5), the erroneous pixel probability matrix of the frame F_{n-k+1} due to losses in the frame F_{n-k} could be computed by

$$\mathbf{E_{n-k+1}^k}(p) = \mathcal{M}_{n-k+1}(\mathbf{E_{n-k}^k})(p) \times \bar{\mathbf{L}}_{\mathbf{n-k+1}}(p), \tag{8}$$

where initially,

$$\mathbf{E_{n-k}^k}(p) = \mathbf{L_{n-k}}(p). \tag{9}$$

By recursivity, the process could then be generalized from the initial condition above (Eq. (9)) and written as

$$\mathbf{E_{n-k+j}^k}(p) = \mathcal{M}_{n-k+j}(\mathbf{E_{n-k+j-1}^k})(p) \times \bar{\mathbf{L}}_{\mathbf{n-k+j}}(p), \quad j = 1, 2, 3, ..., n. \tag{10}$$

Following the notation introduced before, $\mathcal{M}_{n-k+j}$ refers to the motion vectors of the frame F_{n-k+j}. Moreover, as in the relation (5), when a pixel q in one of the reference frames F_{n-k+j} belongs to an intra-coded macroblock or has no correspondence in its direct reference frame (according to $\mathcal{M}_{n-k+j}$),

$$\mathbf{E_{n-k+j}^k}(q) = 0. \tag{11}$$

Finally, the generic erroneous pixel probability matrix $\mathbf{E_n^k}$ is given when $j = k$ in the equation (10). Losses in-between F_{n-k} and F_n are not taken into account. The influence of each of the reference frames will thus be considered separately.

4.4. AMIS: Adaptive MPEG-2 Information Structuring

As seen before, the temporal and spatial error propagations can be considered as distinct processes. The AMIS mechanism could then be divided into a spatial and a temporal mechanism, limiting respectively the spatial and temporal error propagation visibility. Indeed, adding slice headers has no effect on temporal error propagation and, adding intra-coded macroblock does not help in limiting the spatial error propagation. Therefore, these two mechanisms could be considered independently, besides the fact that the slice structure of previous reference frames influences the decision of intra-coded macroblocks insertion (see Fig. 3).

4.4.1. Spatial AMIS

The spatial part of AMIS aims at limiting the spatial error propagation, or at least its visible degradations. It introduces an extra slice header as soon as the expected distortion reaches a given threshold. Clearly a new slice is inserted as soon as:

$$\sum_{M_i \in S} \delta_s(i) \times L(i) \geq T_S, \tag{12}$$

where M_i is the current macroblock belonging to slice S and $\delta_s(i)$ is a distortion measure like the one defined in Eq. (1). L(i) defined in Eq. (4) represents the probability for the macroblock to be lost. The weighting factor acts in an adaptive manner. There is indeed no need to add protection for an area which is not likely to be lost, even if the involved degradation would be high.

The spatial threshold T_S regulates the acceptable level of impairments. This parameter could be adapted to the transmission conditions (i.e. the expected loss rate) or to other QoS parameters. It defines a kind of bitstream vulnerability degree, since the smaller the threshold, the smaller the visual impact of data loss.

Furthermore, this mechanism takes the packetization process into account. Indeed, there is no need to put more than one slice header in the same network loss entity.[12] Then, before inserting a new slice header, the encoder compares the size of the current slice to the expected size of transmission packets taking into account the packetization process.

4.4.2. Temporal AMIS

The temporal AMIS mechanism is much more complex, due to the fact that the temporal error propagation and its effect are much more difficult to compute. Let's assume here first that losses in different reference frames could be considered independently in regards to their effect on the current frame. This assumption, though not completely correct, places anyway the encoding process in the worst case from the degradations point of view. It will tend to add more protection than effectively needed, but simplifies greatly the encoding mechanism.

The decision on intra-coding is analyzed for each macroblock in the encoding process. It is based on the degradation in the current macroblock, due to losses in reference frames, weighted by the probability for the impairment to appear.
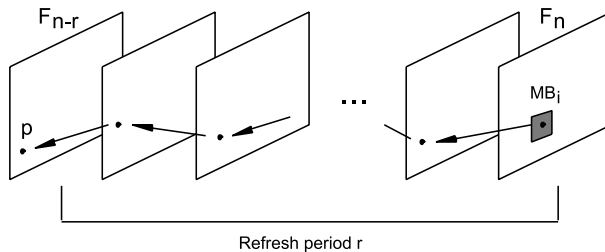

Figure 6. Maximum refresh period r of pixel p. The macroblock M_i in frame F_n is intra-coded, because the pixel p has reached the maximum refresh period (the grey areas intra-coded).

Finally, the selection of intra-coded macroblocks is defined as follows:

- A maximum refresh period can be imposed. This period corresponds to the maximum number of frames for a pixel without any intra reference. When one of the pixels of a macroblock has no intra reference for the maximum refresh period, the macroblock shall be Intra coded (see Fig. 6). This condition is important to avoid the need of a regular intra frame coding.

- The distortion due to temporal error propagation is weighted by the erroneous pixel probabilities and compared to a threshold T_T, similar to the threshold used in the spatial AMIS mechanism. This distortion is obtained by summing effects of losses in each of the previous reference frames up to the last I-picture F_{n-I}. Finally, in the current frame F_n, the condition for a macroblock M_i to be intra-coded is given by

$$\sum_{k=1}^{I} \left(\frac{1}{256} \sum_{p \in M_i} \mathbf{E_n^k}(p) \times \delta_t^k \right) \geq T_T, \tag{13}$$

where $\mathbf{E_n^k}$ and δ_t^k are given in Eq. (10) and (2) respectively. The temporal threshold T_T can be adaptively modulated, like in the spatial mechanism, to vary the number of intra-coded macroblocks in the bitstream, or equivalently the stream robustness to temporal data loss propagation.

It should be noted that the B-frames have not been considered. Indeed, these frames offer the highest compression ratio, and adding Intra macroblocks would result in the highest relative overhead. Moreover these frames do not participate to the temporal error propagation. Therefore, the impact of data loss in B-frames is not visible (the human visual system temporal resolution is larger than a single frame duration[13]).

5. RESULTS

In this section, we compare the AMIS algorithm to the MPEG-2 TM5 encoding scheme. Figure 7 provides some experimental results on OL-VBR encoded streams with a constant MQUANT value of 20 and 32. It is shown that, in both cases, AMIS behaves much better than the traditional encoding scheme, especially for medium to high PLRs. AMIS also yields a better video quality at low PLRs due to the insertion of extra intra-coded macroblocks. Moreover, the video quality gain is even higher for higher MQUANT values as the proportional influence of intra-coded macroblocks is larger.
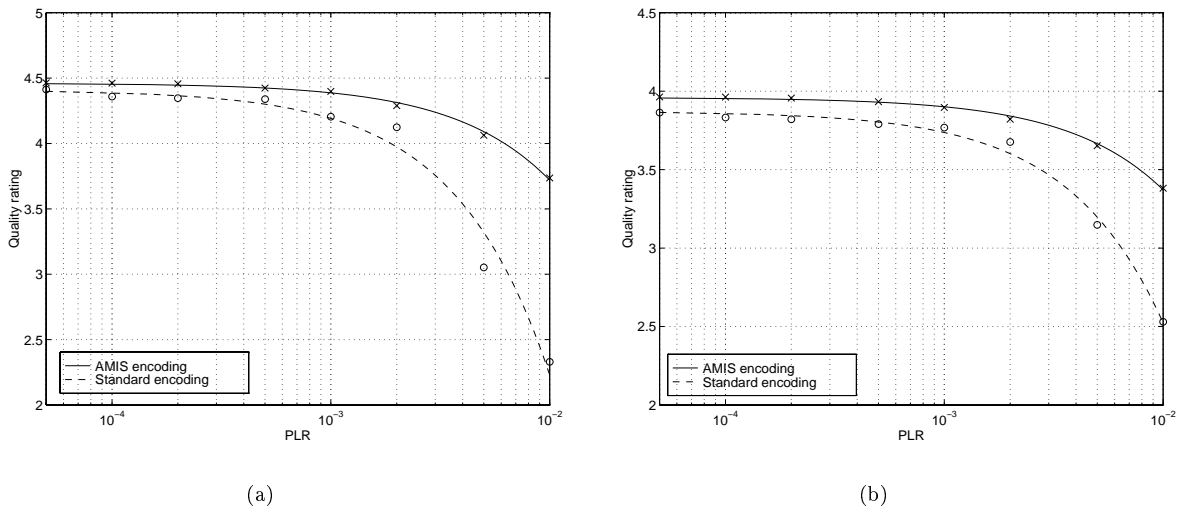

(a)   (b)

Figure 7. *AMIS algorithm and MPEG-2 TM5 encoding scheme: OL-VBR encoding quality versus the PLR experienced on the network. The AMIS thresholds have been fixed to $T_S = 10^3$ and $T_T = 2 \times 10^2$. The MQUANTs are set to 20 in (a) and 32 in (b).*

It should be noted that the perceived video quality evolves linearly with the PLR in both AMIS and standard MPEG-2 encoding schemes.[14] However, the slopes are very different. A FEC mechanism built on top of AMIS would further increase the slopes' difference.

Nevertheless, this protection scheme introduces an overhead. However, AMIS keeps the protection overhead low in regards to the gain in video quality. The overhead never exceeds ten percent of the total bit rate. Obviously, this overhead, and thus the robustness, could vary when tuning the AMIS thresholds (see Sec. 4.4). For given thresholds, the overhead increases with the PLR and with the MQUANT value (adaptivity feature of AMIS, see Sec. 4.1).

Further studies are currently under investigation. These include the automatic regulation of the AMIS thresholds and the analysis of the algorithm under some given traffic constraints.

6. CONCLUSIONS

In this paper, we presented our adaptive MPEG-2 information structuring (AMIS) algorithm. AMIS adds extra resynchronization points (i.e. slice headers and intra-coded macroblocks) only where packet loss would lead to annoying video degradation according to the expected packet loss ratio and the error concealment implemented at the decoder. AMIS proved to behave much better in comparison to the MPEG-2 TM5 implementation under medium to high packet loss ratio experienced over an IP or an ATM network.

The addition of an application-level FEC technique is currently under study. A FEC packet is added when AMIS can not prevent a packet loss from introducing annoying degradation in the reconstructed video. The number of FEC packets should be minimal since the semantic information to be protected has been first well structured.

REFERENCES

1. I. J. 1, *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information - Part 1, 2 and 3*. ISO/IEC JTC 1, 1996.
2. B. G. Haskell, A. Puri and A. N. Netravali, *Digital Video: an Introduction to MPEG-2*, Digital Multimedia Standards Series, Chapman Hall, 1997.
3. O.A. Aho and J. Juhola, "Error resilience techniques for mpeg-2 compressed video signal," in *IEE International Broadcasting Convention (IBC)*, pp. 327–332, January 1994.
4. I. E. G. Richardson and M. J. Riley, "Varying slice size to improve error tolerance of mpeg video," in *SPIE - Electronic Imaging Conference*, vol. 2668, January 1996.
5. I. E. G. Richardson and M. J. Riley, "Controlling the rate of mpeg video by dynamic variation of sequence structure," in *International Picture Coding Symposium*, March 1996.
6. G. Karlsson, "Asynchronous Transfer of Video," *IEEE Communications Magazine* **34**, pp. 118–126, August 1996.
7. C. Fogg, "mpeg2encode/mpeg2decode," in *MPEG Software Simulation Group*, 1996.
8. J. W. Roberts, J. Guibert and A. Simonian, "Network Performance Considerations in the Design of a VBR Codec," in *Queuing Performance and Control in ATM*, pp. 77–82, J. W. Cohen and C. D. Pack, June 1991.
9. C. J. van den Branden Lambrecht and O. Verscheure, "Perceptual Quality Measure using a Spatio-Temporal Model of the Human Visual System," in *Proceedings of the SPIE*, vol. 2668, pp. 450–461, January 1996.
10. O. Verscheure and P. Frossard, "Perceptual MPEG-2 Syntactic Information Coding: A Performance Study based on Psychophysics," in *Picture Coding Symposium (PCS)*, September 1997.
11. M.-J. Chen, L.-G. Chen and R.-M. Weng, "Error Concealment of Lost Motion Vectors with Overlapped Motion Compensation," *IEEE Transactions on Circuits and Systems for Video Technology* **7**, pp. 560–563, June 1997.
12. J. Zhang, M.R. Frater, J.F. Arnold and T.M. Percival, "MPEG 2 Video Services for Wireless ATM Networks," *Journal on Selected Areas in Communications* **15**, pp. 119–128, January 1997.
13. C. J. van den Branden Lambrecht, *Perceptual Models and Architectures for Video Coding Applications*. PhD thesis, Swiss Federal Institute of Technology, Lausanne, Switzerland, 1996.
14. O. Verscheure, P. Frossard and M. Hamdi, "MPEG-2 Video Services over Packet Networks: Joint Effect of Encoding Rate and Data Loss on User-Oriented QoS," in *8th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, July 1998.