

Modeling and Analysis of Distortion Caused by Markov-Model Burst Packet Losses in Video Transmission

Zhicheng Li, Jacob Chakareski, Xiaodun Niu, Yongjun Zhang, and Wanyi Gu

Abstract—This paper addresses the problem of distortion modeling for video transmission over burst-loss channels characterized by a finite-state Markov chain. Based on a detailed analysis of the error propagation and the bursty losses, a distortion trellis model is proposed, enabling us to estimate at the both the frame level and sequence level the expected mean-square error (MSE) distortion caused by Markov-model burst packet losses. The model takes into account the temporal dependencies induced by both the motion-compensated coding scheme and the Markov-model channel losses. The model is applicable to most block-based motion-compensated encoders, and most Markov-model lossy channels as long as the loss pattern probabilities for that channel is computable. Based on the study of the decaying behavior of the error propagation, a sliding window algorithm is developed to perform the MSE estimation with low complexity. Simulation results show that the proposed models are accurate for all tested average loss rates and average burst lengths. Based on the experimental results, the proposed techniques are used to analyze the impact of factors such as average burst length on the average decoded video quality. The proposed model is further extended to a more general form, and the modeled distortion is compared with the data produced from realistic networks loss traces. The experiment results demonstrate that the proposed model is also accurate in estimating the expected distortion for video transmission in real networks.

Index Terms—Burst-loss channel, distortion modeling, error propagation, Markov-model loss process, video transmission.

I. INTRODUCTION

PACKET loss is a key factor degrading the reconstructed video quality in multimedia streaming services such as video conferencing and internet protocol television [1]. Modeling the effect of packet loss on the end-to-end video quality

Manuscript received August 26, 2008; revised January 5, 2009. First version published May 12, 2009; current version published July 22, 2009. This work was supported in part by Hi-Tech Research and Development Program of China, Nos. 2007AA01Z2a4 and 2009AA01Z218, the National Natural Science Foundation, Nos. 60877052 and 60702005, PCSIRT, No. IRT0609 and the Programme of Introducing Talents of Discipline to Universities, No. b07005. This paper was recommended by Associate Editor D. S. Turaga.

Z. Li, Y. Zhang, and W. Gu are with the Beijing University of Posts and Telecommunications (BUPT), Beijing 100876 China (e-mail: lizhicheng@bupt.cn; zc.li@sub.siat.ac.cn; yjzhang@bupt.edu.cn; wyg@bupt.edu.cn).

J. Chakareski is with the Swiss Federal Institute of Technology (EPFL), 1015 Lausanne, Switzerland (e-mail: jakov.cakareski@epfl.ch).

X. Niu is with the Nanyang Technological University, 639789 Singapore (e-mail: niux0001@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2009.2022806

is often critical for constructing joint source-channel rate-distortion optimized schemes [2] and channel error control techniques such as inter/intra mode switching [3] and forward error correction (FEC) [4].

When modeling the packet-loss-induced distortion, the network loss characteristics should be known *a priori*. Many researchers have studied the actual network loss behavior, and most of these studies agree that internet packet loss often exhibits finite temporal dependency [5]. For example, if current packet is lost, the next packet is also likely to be lost. This leads to bursty packet losses [6], [7]. Still, to the best of our knowledge, a complete mathematical model relating the channel-induced distortion and the bursty packet loss process has not been proposed yet. In particular, very little analytical work has been done on 1) estimating the expected distortion at the encoder given a burst-loss channel model and 2) for such a channel and a given average loss rate, how average burst length affects the average video quality at the receiver. Answering these questions can help improving existing error resilient techniques for video transmission over burst-loss channels. The above problems highlight the need for modeling the distortion caused by bursty packet losses.

So far, many distortion models for video transmission over lossy channels have been proposed. However, in these existing works, the packet loss events are modeled as independent and identically distributed (i.i.d.) random variables with an average loss rate that can be characterized by a Bernoulli-model loss process. Examples are the well-known recursive optimal per-pixel estimation method in [3] and its extensions in [8] and [9], the macroblock (MB) level distortion model in [10], the frame level channel-induced distortion models in [11] and [12] and the group of picture (GOP) level distortion model in [13]. These models mainly focus on characterizing the specific coding behavior but do not pay further attention to the loss behaviors of the underlying network. For example, the model in [12] takes into account almost all popular coding features, including intra prediction and deblocking filtering, in order to accurately estimate the expected channel distortion in terms of mean-square error (MSE). However, that work assumes that the underlying network packet losses are i.i.d. and therefore employs a Bernoulli model to this end. All the above distortion models only consider the average loss rate in the absence of another factor, i.e., the burst length, and therefore are less efficient for the case of video transmission over burst-loss channels.

The work in [14] shows that at the same average loss rate, different loss patterns lead to different decoding distortion values, which indicates that the burst length does matter. Similarly, the distortion chain model in [15] predicts the end-to-end distortion for arbitrary loss patterns based on a set of distortion values measured *a priori* at the encoder for a smaller set of loss pattern samples. However, both of these works do not consider explicitly in their analysis the channel correlation that exists between the individual packet losses when computing the expected distortion at the receiver.

In this paper, we develop a mathematical framework denoted as the distortion trellis model for estimating the expected MSE distortion caused by bursty packet losses. The model aims to answer satisfactorily the unresolved issues in video distortion modeling raised above.

Before constructing our distortion model, we analyze first the bursty loss process of the transmission channel. The work in [5] validates that the finite-state Markov chain is more accurate than the Bernoulli model to characterize the bursty packet losses in the Internet. Many other works such as [16]–[19] have shown that Markov-models are a good approximation of the actual packet loss processes for both wired and wireless channels. Therefore, in our subsequent analysis we assume that the channel loss behavior can be characterized by a finite state Markov-model. Without loss of generality and for simplicity, we derive our distortion model for a two-state Markov loss model, or the Gilbert model [20]. Note that the proposed techniques are applicable to more general cases, i.e., they can be easily extended to most finite state Markov loss models, for example, the Markov-models studied in [17] and [21]. We also extend the proposed distortion model to the case of an $(m+1)$ -state Markov loss model, which is studied in [17]. It is worth noting that our objective is not to demonstrate specific bursty loss models but rather to show how to model the video decoding distortion given such a channel model.

The contributions of this paper can be summarized as follows: 1) based on an in-depth analysis of both the error propagation and the loss burstiness, a distortion trellis model is established, enabling us to estimate at the encoder the expected MSE distortion at both frame and sequence level given Gilbert channel packet losses. The model is designed to be applicable to most block-based motion-compensated encoders. The model also allows for any temporal error concealment at the decoder; 2) based on the study of the decaying behavior of the error propagation, a sliding window algorithm is developed to compute the MSE distortion estimate with low complexity. Using the sliding window algorithm, in most cases more than 90% of the computational burden can be saved compared to the original distortion trellis model; 3) simulations are conducted to validate our proposed model and algorithm. Based on the simulation results, the proposed techniques are also used to study the impact of bursty losses and factors such as average burst length on the average decoded video quality; and 4) finally, we extend the aforementioned proposed model to estimate the expected distortion given $(m+1)$ -state Markov channel losses. Then, the modeled distortion is compared with the data produced from realistic network packet loss traces.

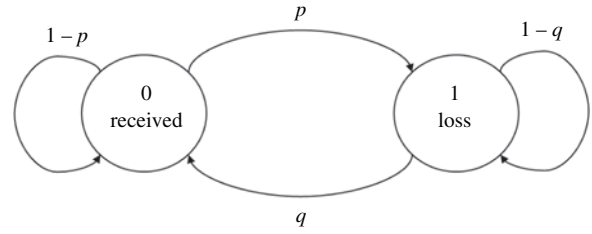


Fig. 1. Gilbert channel model.

The paper is organized as follows. Section II covers the related preliminaries and formulates the considered problem of distortion modeling. In Section III, we present the framework of the proposed distortion trellis model, while in Section IV, we describe the sliding window algorithm for calculating the MSE distortion with low complexity. In Section V, we study the performance of the proposed techniques through simulation experiments. In Section VI, we extend the proposed model to a more general form and compare the modeled distortion with actual measured data in realistic networks. Section VII concludes the paper and discusses prospective future work.

II. PRELIMINARIES

In this section, we provide the necessary preliminaries and then formulate the problem under consideration. First, let us state some general assumptions in this paper.

A. General Assumptions

We assume that at the encoder, a raw video sequence is separated into GOPs and each GOP starts with an I-frame followed by P-frames. Because the loss of a B-frame generally does not interfere with other frames, we do not consider B-frames in this paper. However, the proposed distortion model can be easily extended to the case with B-frames. In a P-frame, MB intra-refreshing can be used for either coding efficiency or error resilience. Subsequently, we assume that all the MBs in a frame are grouped into one slice and each slice is coded into one network packet. Note that the results in this paper can also be easily extended to the case when one frame is separated into multiple slices.¹ We assume that the channel losses can be characterized via a Gilbert model, and the channel drops or delivers the packet according to the current channel state. At the decoder, we assume that certain temporal error concealment strategy is applied when a P-frame is lost. We assume that the correct reception and decoding of the I-frame can always be guaranteed. Hence, error propagation is only due to the loss of P-frames.

B. Channel Model

The presence of temporal memory and correlation in packet losses in wired/wireless internet suggests the use of a Markov-model to describe the long-term network packet loss. Here

¹In the case when the content of one video frame is spread over multiple network packets, the issue of burst loss may actually be less relevant. More packets per frame will mean that burst losses will actually concentrate on affecting individual frames, rather than spread across multiple frames, as studied in [22].

we use a two-state Markov-model, or the Gilbert model, to emulate the loss process. Note again that our proposed techniques can be extended to most finite state Markov-model loss processes. Gilbert first proposed a two-state Markov-model in his studies to characterize the bursty losses [20]. In this model, the channel switches between the error state and the error-free state. When the channel is in the error state, the transmitted packet is always lost, while in the error-free state the packet is always correctly received. Let State 0 and State 1 respectively denote the error-free and the error states. As shown in Fig. 1, the parameter p is the transition probability from State 0 to State 1, and q denotes the probability of the opposite transition. Normally $p + q < 1$. If $p + q = 1$, the Gilbert model reduces to a Bernoulli model. From the definition, the stationary probability for State 0 and 1, denoted by π_0 and π_1 , can be computed as $\pi_0 = q/(p + q)$ and $\pi_1 = p/(p + q)$, respectively. Then, the mean packet loss ratio PLR equals π_1 , and the average burst length ABL is given by $1/q$. Thus, given PLR and ABL , the Gilbert model is determined.

C. Problem Formulation

Let x_n^i and y_n^i denote the reconstructed pixel values for frame n and pixel i at the encoder and at the decoder, respectively. Then, the average MSE distortion for frame n for channel realization c can be calculated as

$$d_n^c = E_i \left\{ \left(x_n^i - y_n^i \right)^2 \right\} = \frac{1}{XY} \sum_{i=1}^{XY} \left(x_n^i - y_n^i \right)^2 \quad (1)$$

where $E_i\{\cdot\}$ denotes the computation of the average MSE over all pixels in frame n , and X and Y respectively denote the frame width and height in pixels. Finally, the expected distortion of frame n can be defined as

$$d_n = E_c \{ d_n^c \} = E_c \left\{ E_i \left\{ \left(x_n^i - y_n^i \right)^2 \right\} \right\} \quad (2)$$

where $E_c\{\cdot\}$ denotes the expectation taken over all possible channel realizations. Note that the definition of d_n is generic and hence applies to most existing coding technologies and channel realizations.

When calculating d_n for a Bernoulli channel, an important problem is to model the error propagation due to decoding dependencies between temporally adjacent frames. In the case of a Gilbert channel, the channel states, or the packet losses also exhibit temporal dependencies. Hence, when calculating d_n for a Gilbert channel, the decoding dependencies and the loss dependencies should both be considered. Therefore, it is more complex to model d_n for Gilbert losses than for Bernoulli losses.

This paper mainly focuses on modeling d_n for video transmission over a Gilbert channel. In the following sections, through analysis we present a mathematical model of the distortion caused by Gilbert losses. We also develop a sliding window algorithm allowing us to calculate the expected distortion with low complexity. At the end, we extend the proposed model to a more general form, for estimating the distortion caused by $(m + 1)$ -state Markov losses.

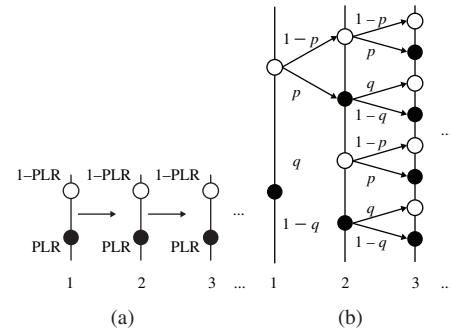


Fig. 2. (a) Packet loss in a Bernoulli channel and (b) Packet loss in a Gilbert channel. A white circle denotes a received packet while a black dot denotes a lost packet.

III. PROPOSED DISTORTION TRELLIS MODEL

A. Framework of the Distortion Trellis Model

The problem we will address in this subsection is how to perform the operation $E_c\{\cdot\}$ from (2) on frame n , in the case of a Gilbert-model loss process. Then, we will derive the general form of the proposed distortion trellis model.

In motion-compensated video coding, decoding error in a previous frame may propagate into the current frame. In such a case, the distortion of the current frame is affected not only by the transmission state (“Lost” and “Received”) of the current frame, but also by the transmission states of all previous frames in the same GOP. In other words, it is affected by the loss patterns of all transmitted frames in the same GOP (including the current frame). For a frame sequence of length n , the total number of all possible loss patterns is 2^n . Thus, theoretically, after decoding the n th frame in a GOP, the total number of all possible distortion values of the n th frame at the decoder is also 2^n .

In a Bernoulli channel, a packet is either lost with a probability PLR or received with a probability $1 - PLR$, independently of other loss events, as shown in Fig. 2(a). Thus, when calculating d_n , we do not need to calculate all 2^n possible distortions. Instead, most existing models define another two distortions d_n^L and d_n^R . The former is the expected distortion given that frame n is lost, while the latter denotes the expected distortion for the case when frame n is received. Often, d_n^L and d_n^R are calculated in a recursive approach to account for the error propagation. In such a case, we only need to calculate two distortions for each frame. Finally, d_n is calculated as

$$d_n = E_c \{ d_n^c \} = PLR \cdot d_n^L + (1 - PLR) \cdot d_n^R. \quad (3)$$

In a Gilbert channel, packet losses are no more i.i.d. but exhibit dependencies over time. Note that correct reception of the I-frame can always be guaranteed (as assumed in Section II-A), hence we only consider P-frame losses. Observed from the sender, the loss process of all P-frames in a GOP is a two-state Markov process, as shown in Fig. 2(b). In such a case, when calculating d_n , we need to consider all 2^n cases for frame n , which adumbrates a rather elaborate calculation process.

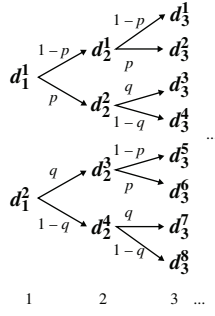


Fig. 3. Statistical dependencies $\{d_n^r, n = 1, 2, \dots\}$.

To numerically analyze the loss dependencies, we first study the statistic of the Gilbert-model loss process. Consider the impairments for a transmitted packet (frame) sequence of length n as an n -bit binary random variable $K_n = \{B_j\}_{j=1}^n$. The random variable B_j is over the binary alphabet $\{0, 1\}$. $B_j = 1$ indicates that the j th frame is lost. Then, the total number of all possible values of K_n is 2^n . Define moreover an ordered set $\mathbf{I}_n = \{k_n^r\}$, $r = 1, \dots, 2^n$, where k_n^r is an n -bit binary number and $k_n^1 = \underbrace{0 \dots 0}_{n \text{ bits}}$, $k_n^r = 1 + k_n^{r-1}$, $r = 2, \dots, 2^n$. Furthermore, we assume that the r th value of K_n is k_n^r , the r th element in \mathbf{I}_n . Note that in our analysis this is an important assumption, based on which we can recursively derive \mathbf{I}_n from \mathbf{I}_{n-1} in a simple way. Hereafter, we refer to k_n^r as the r th loss pattern of a frame sequence of length n .

Let $P(k_n^r)$ denote the probability that loss pattern k_n^r occurs, i.e., $P(k_n^r) = P_r(K_n = k_n^r)$. Note that different loss patterns lead to different distortion values. Let d_n^r be the decoder distortion of the n th frame in a frame sequence of length n under loss pattern k_n^r . Then, d_n^r can be defined as

$$d_n^r = E_i \left\{ \left(x_n^i - y_{n,r}^i \right)^2 \right\} \quad (4)$$

where $y_{n,r}^i$ denotes the decoder reconstructed value of pixel i in the n th frame for an n -length frame sequence under loss pattern k_n^r . Thus, from the definition of d_n^r , we can obtain an important probability relation as follows: $\Pr(\text{at the decoder the distortion of frame } n \text{ is } d_n^r) = P(k_n^r)$.²

In essence, the definitions of k_n^r and d_n^r lay a foundation for the proposed model. First, they establish the relation between various loss patterns and their corresponding decoding distortions. Second, they enable us to recursively analyze the loss dependencies and decoding dependencies. Third, they supply the knowledge and means to develop a low-complexity algorithm for distortion estimation. The second and third items will be discussed later. Fig. 3 shows the statistical dependencies between the elements of the set $\{d_n^r, n = 1, 2, \dots\}$. Furthermore, since the packet loss dependencies of the channel loss process and the distortion/decoding dependencies of the

video frames can both be depicted by a trellis graph, as illustrated in Figs. 2(b) and 3, respectively, we refer to the proposed distortion estimation method as the distortion trellis model.

Then, we can calculate the expected distortion of frame n by taking an expectation over all possible decoder distortion values for frame n

$$\begin{aligned} d_n &= E_c \{d_n^c\} \\ &= \sum_{r=1}^{2^n} d_n^r \cdot \Pr(\text{distortion of frame } n \text{ is } d_n^r) \\ &= \sum_{r=1}^{2^n} d_n^r \cdot P(k_n^r), \quad n = 1, 2, \dots \end{aligned} \quad (5)$$

The formula in (5) is the general form of the proposed distortion trellis model. From (5), it is clear that the computation of d_n necessitates knowledge of both d_n^r and $P(k_n^r)$, $r = 1, \dots, 2^n$. We must emphasize that (5) is applicable to most channel models and hence is general. For different channel models, the only difference in using the distortion trellis model is the computation of $P(k_n^r)$, because generally the same loss pattern occurs with different probabilities in different channels. On the other hand, d_n^r is uncorrelated with a specific channel model but only depends on the video sequence. That is why the distortion trellis model can be easily extended to arbitrary finite-state Markov loss model. In a Bernoulli channel, (5) can be reduced to (3), which will be discussed in the following sections. In a Gilbert channel, $P(k_n^r)$ can be derived recursively, as follows. From the definition of $\{k_n^r\}$, it is clear that given $P(k_{n-1}^t)$, $t = 1, \dots, 2^{n-1}$, the loss pattern probabilities can be written as

$$\begin{cases} P(k_n^{4r-3}) = (1-p) \cdot P(k_{n-1}^{2r-1}) \\ P(k_n^{4r-2}) = p \cdot P(k_{n-1}^{2r-1}) \\ P(k_n^{4r-1}) = q \cdot P(k_{n-1}^{2r}) \\ P(k_n^{4r}) = (1-q) \cdot P(k_{n-1}^{2r}), \quad r = 1, \dots, 2^{n-2}. \end{cases} \quad (6)$$

The computation of the loss pattern probabilities is illustrated in Fig. 4, which reveals the loss dependencies for the Gilbert channel and is also in a trellis shape. The remaining task in this section is how to calculate d_n^r .

B. Recursive Computation of d_n^r

In this section, we aim to establish a generic distortion model for calculating d_n^r that is able to capture the effect of error propagation and is applicable to most block-based motion-compensated coding schemes. Related models are proposed in [3], [11], [12], where some recursive distortion models are developed. However, they are methods for calculating d_n^L and d_n^R (as mentioned in Section III-A) rather than d_n^r . On the other hand, most of them are overly complex for our objective. Hence, using the recursion-based analytic methodology, we aim to obtain a more explicit and more generic model. Recall that our purpose is to calculate d_n by (5) and then to analyze the impact of bursty losses on average video quality. Therefore,

²Note that for different r and r' , maybe the distortion value d_n^r could be nearly equal to $d_n^{r'}$. However, from the definition of d_n^r , the probabilities $\Pr(\text{distortion} = d_n^r)$ and $\Pr(\text{distortion} = d_n^{r'})$ cannot be the same, because we are dealing with distinct loss patterns. In this sense, we believe the probability equation $\Pr(\text{at the decoder the distortion of frame } n \text{ is } d_n^r) = P(k_n^r)$ is true.

we attempt to establish a distortion model that can well account for bursty losses and is compatible with our distortion trellis framework. Additionally, we would like the model to be convenient for deriving a low-complexity algorithm for distortion estimation.

From the definition of k_n^r , it can be observed that for loss pattern k_n^{2r-1} , the n th packet is received, while for loss pattern k_n^{2r} , the n th packet is lost, where $r = 1, \dots, 2^{n-1}$. Thus, given that the loss pattern of the previous $n-1$ frames is k_{n-1}^r, d_n^{2r-1} is the frame-average distortion if the n th frame is received while d_n^{2r} denotes the same quantity for the case when the n th frame is lost. Next, we separately consider computing d_n^{2r-1} and d_n^{2r} .

1) *Computation of d_n^{2r}* : We first develop the recursion model for computing d_n^{2r} . In our assumption, if a frame is lost, all MBs in this frame are recovered using some temporal error concealment strategy, regardless whether they are coded in inter or intra mode. Let $f_l(i)$ denote the index of the l th pixel in frame $n-1$ that is used to estimate pixel i in frame n . Then the final concealed value of $y_{n,2r}^i$ can be expressed as $\Phi_l(y_{n-1,r}^{f_l(i)})$, where Φ_l represents the pixel operation on $y_{n-1,r}^{f_l(i)}$ for all l used in obtaining the final concealed value of $y_{n,2r}^i$. For example, in video coders using sub-pixel motion estimation, Φ_l denotes the interpolation operation. For another example, in video coders using deblocking filters, Φ_l denotes the deblocking operation. For previous frame copy concealment, $\Phi_l(y_{n-1,r}^{f_l(i)}) = y_{n-1,r}^i$. Φ_l could also denote weighted prediction and so on. It is a reasonable assumption that Φ_l is a linear pixel filtering operation and can be considered the same for different frames. Then, d_n^{2r} can be derived as follows:

$$\begin{aligned}
 d_n^{2r} &= E_i \left\{ \left(x_n^i - \Phi_l \left(y_{n-1,r}^{f_l(i)} \right) \right)^2 \right\} \\
 &= E_i \left\{ \left(x_n^i - \Phi_l \left(x_{n-1}^{f_l(i)} \right) + \Phi_l \left(x_{n-1}^{f_l(i)} \right) - \Phi_l \left(y_{n-1,r}^{f_l(i)} \right) \right)^2 \right\} \\
 &= E_i \left\{ \left(x_n^i - \Phi_l \left(x_{n-1}^{f_l(i)} \right) \right)^2 \right\} \\
 &\quad + E_i \left\{ \left(\Phi_l \left(x_{n-1}^{f_l(i)} \right) - \Phi_l \left(y_{n-1,r}^{f_l(i)} \right) \right)^2 \right\} \\
 &= ECD_n \\
 &\quad + E_i \left\{ \left(\Phi_l \left(x_{n-1}^{f_l(i)} - y_{n-1,r}^{f_l(i)} \right) \right)^2 \right\}, \quad r = 1, \dots, 2^{n-1}
 \end{aligned} \tag{7}$$

where $ECD_n = E_i \left\{ \left(x_n^i - \Phi_l \left(x_{n-1}^{f_l(i)} \right) \right)^2 \right\}$. Note that ECD_n is the average error concealment distortion of frame n . Given specific coding scheme and error concealment strategy, Φ_l and $f_l(i)$ are determined and then ECD_n is determined. It is worth noting that ECD_n is the new added distortion if frame n is lost. $E_i \left\{ \left(\Phi_l \left(x_{n-1}^{f_l(i)} - y_{n-1,r}^{f_l(i)} \right) \right)^2 \right\}$ is the temporal propagation distortion from frame $n-1$. Note that the third identity in (7) is based on the assumption that the concealment error $x_n^i - \Phi_l \left(x_{n-1}^{f_l(i)} \right)$ and the propagation error $\Phi_l \left(x_{n-1}^{f_l(i)} \right) - \Phi_l \left(y_{n-1,r}^{f_l(i)} \right)$ are uncorrelated [11], [12]. The fourth identity is

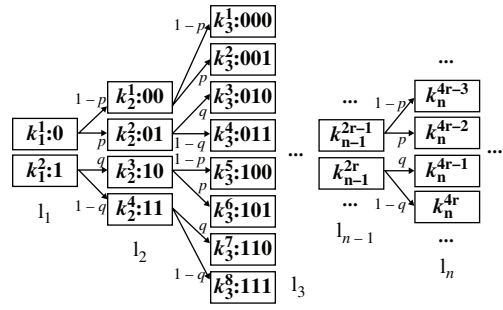


Fig. 4. Computation of the loss pattern probabilities $\{P(k_n^r), n = 1, 2, \dots\}$.

based on the assumption that Φ_l is linear, and we believe that this is reasonable in most cases.

Furthermore, as is well known in the video coding community, that when the error in a previous frame propagates into the current frame, it is typically attenuated by the adoption of some coding schemes such as deblocking filtering and sub-pixel motion estimation [23], whose effect can be regarded as a spatial filter or more precisely as an error attenuator. In other words, the temporal propagation distortion in the current frame can be considered as the filtered output of the distortion in a previous frame. Following this reasoning, the term $E_i \left\{ \left(\Phi_l \left(x_{n-1}^{f_l(i)} - y_{n-1,r}^{f_l(i)} \right) \right)^2 \right\}$ in (7) can be approximated as

$$\begin{aligned}
 E_i \left\{ \left(\Phi_l \left(x_{n-1}^{f_l(i)} - y_{n-1,r}^{f_l(i)} \right) \right)^2 \right\} &= u \cdot E_i \left\{ \left(x_{n-1}^i - y_{n-1,r}^i \right)^2 \right\} \\
 &= u \cdot d_{n-1}^r.
 \end{aligned} \tag{8}$$

Then (7) can be rewritten as

$$d_n^{2r} = ECD_n + u \cdot d_{n-1}^r, \quad r = 1, \dots, 2^{n-1} \tag{9}$$

where u is the error attenuation factor for a lost frame.

Then, d_n^{2r} can be estimated as a sum of two separate parts. One part is the average concealment distortion ECD_n , which can be directly calculated at the encoder just after encoding frame n . A related problem here is that some error-resilient video communication techniques, such as joint source-channel rate-distortion optimized schemes, operate before encoding the current frame. Hence, at that moment, x_n^i is not available. Several strategies have been proposed to solve this problem. For example, in [11], ECD_n is estimated by the product of the MSE between the original frames n and $n-1$ by a factor α and then the estimated ECD_n are used in an adaptive mode selection and a rate control scheme. Nevertheless, in this paper we mainly aim to propose and validate a distortion model. Therefore, when we evaluate the proposed model via simulations in Section V, we measure the ECD_n by running the decoder error concealment algorithm on x_n^i . In other words, ECD_n is regarded as precalculated data. Still, using the proposed model in some error-resilient techniques may require estimating ECD_n which in turn may affect the accuracy of the proposed model.

The second term in (9) denotes the temporal propagation distortion and indicates the relation between d_n^{2r} and d_{n-1}^r . In particular, this term reveals the numerical relationship between the distortions of frame $n-1$ and n when frame n is lost. For

a practical application, the parameter u has to be estimated for the specific video coder and content that are employed. The estimation method is important for the performance of the proposed model. In this paper, we use simulation data to estimate u , which will be described in detail in Section V.

2) *Computation of d_n^{2r-1}* : We now turn to computing d_n^{2r-1} . As is well known, a received frame may still contain distortion due to error propagation from an impaired previous frame. In such a case, the coding modes should be considered because the distortions in received inter-coded MBs and intra-coded MBs are different. We first consider the case when all MBs are coded in inter mode, and then we will extend our result to the more general case of having mixed MB coding modes in a frame. Let $g_l(i)$ denote the index of the l th pixel in frame $n-1$ that is used to estimate pixel i in frame n . Note that $g_l(i)$ may differ from $f_l(i)$. Then, at the encoder, the predicted value of x_n^i can be expressed as $\Psi_l(x_{n-1}^{g_l(i)})$, where Ψ_l represents the pixel operation on all $x_{n-1}^{g_l(i)}$ used for obtaining the predicted value of x_n^i , such as when performing interpolation or deblocking filtering. We also assume that Ψ_l is linear and has the same form for different frames. Similarly, at the decoder, the predicted value of $y_{n,2r-1}^i$ is $\Psi_l(y_{n-1,r}^{g_l(i)})$. Then, d_n^{2r-1} can be derived as follows:

$$\begin{aligned} d_n^{2r-1} &= E_i \left\{ \left(\Psi_l(x_{n-1}^{g_l(i)}) - \Psi_l(y_{n-1,r}^{g_l(i)}) \right)^2 \right\} \\ &= E_i \left\{ \left(\Psi_l(x_{n-1}^{g_l(i)} - y_{n-1,r}^{g_l(i)}) \right)^2 \right\}, \quad r = 1, \dots, 2^{n-1}. \end{aligned} \quad (10)$$

As in the case of d_n^{2r} , the operator Ψ_l can be regarded as a spatial filter that will attenuate the error propagation. Hence, we similarly employ $v_0 \cdot d_{n-1}^r$ to approximate $E_i \left\{ \left(\Psi_l(x_{n-1}^{g_l(i)} - y_{n-1,r}^{g_l(i)}) \right)^2 \right\}$ and therefore we can rewrite (10) as

$$d_n^{2r-1} = v_0 \cdot d_{n-1}^r, \quad r = 1, \dots, 2^{n-1} \quad (11)$$

where v_0 is the error attenuation factor for a received frame, in which all MBs are coded in inter mode.

The development of (11) assumes that all MBs in a P-frame are coded in an inter mode. However, a P-frame often contains intra-coded MBs, which will effectively restrain the error propagation [3], [12]. For example, if constrained intra prediction is used, the distortion in received intra-coded MBs equals to zero. Regardless whether constrained intra prediction is employed or not, the effect of macroblock intra refreshing can also be considered as an attenuator that attenuates the error signal from an impaired previous frame [23]. Therefore, to take this into account we introduce a new constant λ and rewrite (11) as

$$d_n^{2r-1} = v \cdot d_{n-1}^r, \quad r = 1, \dots, 2^{n-1} \quad (12)$$

where $v = \lambda \cdot v_0$.

Note that (12) is also applicable to the case when unconstrained intra prediction is employed, such as in an H.264 codec [24], mainly because the error in a received intra-coded MB, if any, also comes from an impaired previous frame, and is also attenuated by the intra refreshing. In essence, (12)

reveals the numerical relation between the distortions of frame $n-1$ and n when frame n is received. Finally, as in the case of u from the previous section, the parameter v needs to be estimated, which is discussed in Section V.

So far, an explicit and generic distortion model is established by the two recursion formulae in (9) and (12). This model is suitable for most motion-compensated coding schemes. Note that although we have assumed in Section II-A that one transmission packet contains exactly one frame, the distortion model in (9) and (12) is also applicable to the case when one frame is packetized into multiple packets. Moreover, the model enables us to calculate d_n^r for all 2^n lost patterns, and therefore can be used in (5) to calculate the expected distortion d_n .

C. Recursive Computation of d_n and Further Analysis

Based on (9) and (12), we can recursively obtain the distortion d_n^r for $r = 1, \dots, 2^n$. The loss pattern probability $P(k_n^r)$ can be recursively calculated with (6). Then, using (5), the expected distortion d_n for Gilbert channel packet losses can be estimated as

$$\begin{aligned} d_n &= \sum_{t=1}^{2^n} d_n^t \cdot P(k_n^t) \\ &= \sum_{r=1}^{2^{n-2}} \left[P(k_n^{4r-3}) d_n^{4r-3} + P(k_n^{4r-2}) d_n^{4r-2} \right. \\ &\quad \left. + P(k_n^{4r-1}) d_n^{4r-1} + P(k_n^{4r}) d_n^{4r} \right] \end{aligned} \quad (13)$$

where

$$\begin{cases} P(k_n^{4r-3}) = (1-p) P(k_{n-1}^{2r-1}), & d_n^{4r-3} = v \cdot d_{n-1}^{2r-1} \\ P(k_n^{4r-2}) = p \cdot P(k_{n-1}^{2r-1}), & d_n^{4r-2} = ECD_n + u \cdot d_{n-1}^{2r-1} \\ P(k_n^{4r-1}) = q \cdot P(k_{n-1}^{2r-1}), & d_n^{4r-1} = v \cdot d_{n-1}^{2r-1} \\ P(k_n^{4r}) = (1-q) P(k_{n-1}^{2r-1}), & d_n^{4r} = ECD_n + u \cdot d_{n-1}^{2r-1} \end{cases} \quad r = 1, \dots, 2^{n-2}. \quad (14)$$

It can be seen that d_n depends on $u, v, ECD_n, p,$ and q . The former three parameters depend on the video sequence. The parameter pair p and q is used to describe the Gilbert channel and is equivalent to another parameter pair PLR and ABL , which are more commonly used. Then, for video transmission over a Gilbert channel, given the average packet loss ratio PLR , the average burst length ABL , the initial probability distribution $P(k_1^1)$ and $P(k_1^2)$, and the initial distortion distribution d_1^1 and d_1^2 , the expected distortion of each frame in a GOP can be estimated in a frame recursion approach using (13) and (14).

Next, the cumulative expected distortion over the entire GOP D_N can be defined as $D_N = \sum_{n=1}^N d_n$. Note that the sequence level expected distortion D_N can be used as an objective metric to assess the average video quality. Using the proposed method, D_N can be directly derived as explained in the following. Recall first that a GOP consists of one I-frame and N P-frames and that we assume that I-frames are never lost. Then, we define $D(k_n^r)$ as the total distortion of a sequence from the first P-frame to the n th P-frame, for a

given loss pattern k_n^r . Hence, D_N can be estimated by taking an expectation over all possible loss patterns as follows:

$$D_N = \sum_{r=1}^{2^N} D(k_N^r) \cdot P(K_N^r). \quad (15)$$

With the help of the distortion model in (9) and (12), $D(k_n^r)$ can be calculated as follows:

$$\begin{cases} D(k_n^{2^r}) = D(k_{n-1}^r) + d_n^{2^r} \\ D(k_n^{2^r-1}) = D(k_{n-1}^r) + d_n^{2^r-1}, \quad \text{for } r = 1, 2, \dots, 2^{N-1} \end{cases} \quad (16)$$

where $P(k_N^r)$ in (5) and $d_n^{2^r}, d_n^{2^r-1}$ in (16) can be calculated using (14). Then, using (16), the total distortion for an arbitrary loss pattern can be calculated. The formula in (5) provides a way to estimate and analyze the impact of the bursty loss behavior on the average video quality.

Based on the distortion trellis model, let us go back to discuss the case of a Bernoulli channel. We know that the Bernoulli channel is a special case of a Gilbert channel when $p + q = 1$. Substituting $p = PLR$ and $q = 1 - PLR$ into (14), we have $P(k_n^{2^r-1}) = (1 - PLR) \cdot P(k_{n-1}^r)$ and $P(k_n^{2^r}) = PLR \cdot P(k_{n-1}^r)$. Then with the help of (9) and (12), (13) can be rewritten as

$$\begin{aligned} d_{n,\text{Bernoulli}} &= \sum_{r=1}^{2^{n-1}} \left[P(k_n^{2^r-1}) d_n^{2^r-1} + P(k_n^{2^r}) d_n^{2^r} \right] \\ &= (1 - PLR) d_n^R + PLR \cdot d_n^L \end{aligned} \quad (17)$$

where

$$\begin{aligned} d_n^R &= \sum_{r=1}^{2^{n-1}} P(k_{n-1}^r) d_n^{2^r-1} = \sum_{r=1}^{2^{n-1}} P(k_{n-1}^r) v \cdot d_{n-1}^r \\ &= v \cdot d_{n-1,\text{Bernoulli}}, \end{aligned} \quad (18)$$

$$\begin{aligned} d_n^L &= \sum_{r=1}^{2^{n-1}} P(k_{n-1}^r) d_n^{2^r} = \sum_{r=1}^{2^{n-1}} P(k_{n-1}^r) (ECD_n + u \cdot d_{n-1}^r) \\ &= ECD_n + u \cdot d_{n-1,\text{Bernoulli}}. \end{aligned} \quad (19)$$

From (17), we can see that when $p + q = 1$, the Gilbert channel reduces to a Bernoulli channel and the proposed distortion trellis model in (5) reduces to the traditional model in (3), as mentioned in Section III-A. In general, the distortion trellis is applicable to an arbitrary channel model, as long as the loss pattern probability $P(k_n^r)$ is computable for that channel.

Using the distortion trellis model, one can estimate the expected distortion d_n caused by Markov-model bursty losses, at the encoder/sender. However, the model often fails to compute d_n within acceptable time. In particular, when calculating d_n , one needs to compute the terms d_n^r and $P(k_n^r)$ associated with $r = 1, \dots, 2^n$. Consequently, the complexity for calculating d_n is $O(2^n)$ while that for calculating D_N is $O(N2^n)$. Thus, it is desirable to develop a low-complexity algorithm for distortion estimation.

IV. SLIDING WINDOW ALGORITHM

The distortion trellis model assumes that the loss pattern of all previous frames in the same GOP could affect the distortion of the current frame, mainly because of the error propagation effect. However, the propagation of error typically decays in magnitude over the subsequent frames due to the intra refreshing and the spatial filtering [22], [24]. In fact, when the distance between a previous frame and the current frame is big enough, the influence of an error in that previous frame on the current frame can be ignored. Therefore, it is a reasonable assumption that the distortion of frame n is independent of the transmission state of frame m , when $|m - n| > W$, where W is an integer constant. Based on the assumption, we now propose a sliding window (SW) algorithm to calculate d_n for $n > W$ with low complexity. When $n \leq W$, we employ the same approach described in Section III to calculate d_n .

The basic methodology of the SW algorithm is that when calculating d_n for $n > W$, we only consider the loss patterns of the previous W frames rather than all previous frames. Assume that frame n , for $n > W$, has a corresponding sequence segment, or a window W , comprising frames $n - W + 1$ to n , which loss patterns only are considered for calculating the corresponding distortion. In particular, we assume that the first frame in the segment $n - W + 1$ is either received with probability $P(k_1^1)$ or lost with probability $P(k_1^2)$, independently of any frame prior to it, and the corresponding distortion is d_1^1 and d_1^2 , respectively. The loss process of the frames within W is also considered to be a two-state Markov process, or a Gilbert process. In such a case, there are in total 2^W loss patterns that should be considered for each frame $n > W$. That means when calculating d_n using (5), we only need to calculate 2^W corresponding decoder distortion values rather than 2^n . The window slides ahead one frame at a time, and the expected distortion d_n for all $n > W$ can be obtained in this manner. It can be seen that, instead of considering the loss process of all P-frames in a GOP as Markovian, the SW algorithm limits the Markov loss process within each window W and ignores the frames outside the window. As a result, the SW algorithm always underestimates the expected distortion. The overall SW algorithm is summarized in Algorithm 1.

The window length W is an important parameter of the SW algorithm. Generally, a big W leads to more accurate prediction but increases the algorithm's complexity. An appropriate W implies that the tradeoff between the estimation accuracy and the computation complexity is achieved. From (9) and (12), we can see that parameters u and v determine the distortion fading speed over the subsequent frames, which can be considered when selecting the appropriate W . Generally, a small to middle u and v indicate a quick fading, in which case a relatively small W may be acceptable.

The SW algorithm provides an efficient way to calculate d_n . To obtain d_n for $n > W$ based on the SW algorithm, we need only to compute the quantities d_n^r and $P(k_n^r)$ associated with $r = 1, \dots, 2^W$. Together with (15) and (16), the GOP level expected distortion can also be calculated. In this paper, we find that a window size $W \leq 16$ is sufficient to achieve acceptable prediction accuracy for most examined cases. Hence, the computation cost is reduced significantly compared to the

Algorithm 1 SW for calculating d_n for $n = 1, \dots, N$

```

1: Input:  $PLR, ABL, u, v, W, N, \{ECD_n, n = 1, \dots, N\}$ ;
2: Output: the expected distortion  $d_n$  for  $n = 1, \dots, N$ ;
3: Procedure:
4: Initialization:  $d_1^1 = 0, d_1^2 = ECD_1, P(k_1^1) = 1 - PLR,$ 
    $P(k_1^2) = PLR, p = PLR/(ABL(1 - PLR)), q = 1/ABL$ ;
5: for  $n = 1$  to  $N$  do
6:   if  $n \leq W$  then
7:     Compute  $d_n$  using (13) and (14), note that when
        $n = W, P(k_W^j), j = 1, \dots, 2^W$  are obtained;
8:   else
9:     Reset the initial distortion values:  $d_1^1 = 0,$ 
        $d_1^2 = ECD_{n-W+1}$ ;
10:    for  $i = 2$  to  $W$  do
11:      for  $j = 1$  to  $2^{i-1}$  do
12:         $d_i^{2^{j-1}} = v d_{i-1}^j, d_i^{2^j} = ECD_{n-W+i} + u d_{i-1}^j$ ;
13:      end for
14:    end for
15:    After the two for loops,  $d_W^j, j = 1, \dots, 2^W$ 
       are obtained, then the output is
        $d_n = \sum_{j=1}^{2^W} P(k_W^j) d_W^j$ ;
16:   end if
17: end for

```

original distortion trellis model. For example, when we use the SW algorithm with $W = 15$ to calculate $d_n, n = 1, \dots, N$ for a GOP with $N = 36$, the number of iteration cycles that need to be run reduces from $\sum_{n=1}^{36} 2^n$ to $\sum_{n=1}^{15} 2^n + (36 - 15) \cdot 2^{15}$. This translates to a more than 90% reduction in computational complexity.

V. SIMULATION RESULTS

A. Simulation Setup

The H.264 reference software encoder JM12.2 [25] with the baseline profile is used to encode the test sequences used in our experiments. Four QCIF sequences are used, the low-motion sequence *News*, the moderate motion sequence *Foreman*, and the high motion sequences *Stefan* and *Football*. The former three are coded at 15 frames/s, while the sequence *Football* is coded at 30 frames/s. All sequences are coded with a constant QP = 28. The first frame is coded as an I-frame, while the remaining frames are coded as P-frames with a forced intra refresh rate of 9/99 (every nine frames a row of MBs is intra refreshed in a round-robin fashion). Intra prediction and 1/4-pel motion estimation are enabled. Inter pixels are not used for intra prediction. The search range is set to 16. The reference frame is set to be the previous frame. We use one slice per frame and one frame per packet. At the decoder, the simple frame-copy scheme is used for concealment, so that ECD_n for each sequence can be easily premeasured using $ECD_n = (1/XY) \sum_{i=1}^{XY} (x_n^i - x_{n-1}^i)$, where X and Y once again respectively denote the width and the height of frame n in pixels. The concealment frame is displayed instead of the missing frame, and is also stored in the reference frame buffer for decoding subsequent frames.

To validate the accuracy of the proposed estimation methods and to study the impact of the Gilbert losses on the average video quality, extensive simulations are conducted. The average loss rates in our simulations range from 3 to 10%, and the average burst length is set to 1, 1.5, 2, \dots , 5. Each pair of PLR and ABL values is translated into the corresponding p and q values for the Gilbert channel. Then, with each pair of p and q we simulate a Gilbert packet loss process and generate 50 000 to 90 000 loss traces with random loss patterns. For each loss trace, we decode the video and calculate the MSE distortion between each transmitted and received P-frame. The expected distortion for each frame is then obtained by averaging the distortion of that frame over all traces. The GOP size for each sequence used is 390 for *Foreman*, 240 for *Football*, and 200 for both *News* and *Stefan*.

To estimate the model parameters u and v , we use the least square fitting method based on training data. In this estimation, we apply least square fitting to the proposed model for Bernoulli losses in (17) rather than that for Gilbert losses in (13), because (17) is much simpler. Remember that the parameters u and v only depend on the sequence/content, and are not correlated with the transmission channel. Therefore, although we apply least square fitting to (17) to determine u and v , the so-derived parameters are also accurate for (13) and will fit the simulation data well. For instance, when we determine the values of u and v for QCIF *Foreman*, we first simulate a Bernoulli channel and generate 50 000 loss traces, so that the measured distortion $d_{n,\text{measured}}$ can be obtained by averaging the MSE distortion of frame n over all traces. Then, the parameters u and v are determined by minimizing $\sum_n (d_{n,\text{measured}} - d_{n,\text{Bernoulli}})^2$, where $d_{n,\text{Bernoulli}}$ is the estimated distortion of frame n and can be replaced by (13). Note that in the estimation, the used sequence contains 390 frames for *Foreman*, 240 frames for *Football*, and 200 frames for both *News* and *Stefan*. In such a case, although in the following experiments we will use different GOP sizes sometimes, for each tested sequence we still use one pair of parameters u and v , which will be validated rather accurate.

B. Simulation Results and Discussion

In the first set of experiments, the measured expected distortion, the estimate using the original distortion trellis model, and the estimate based on the SW algorithm are all compared. Fig. 5(a) plots the average expected distortion for PLR values from 3 to 10% at $ABL = 2$. Due to the high complexity of the original distortion trellis model, which is used as the performance benchmark here, we test the model over short sequence segments in this simulation. Particularly, we encode 20-frame segment starting at different positions in the original sequence. For each tested PLR and ABL pair, we generate 50 000 loss traces for each segment. The average expected distortion is then obtained by averaging all segments and all loss traces.

It can be seen that the original distortion trellis model provides better prediction of the expected distortion than the SW algorithm along the whole tested PLR range. Although the SW algorithm is less accurate, it still matches the measured

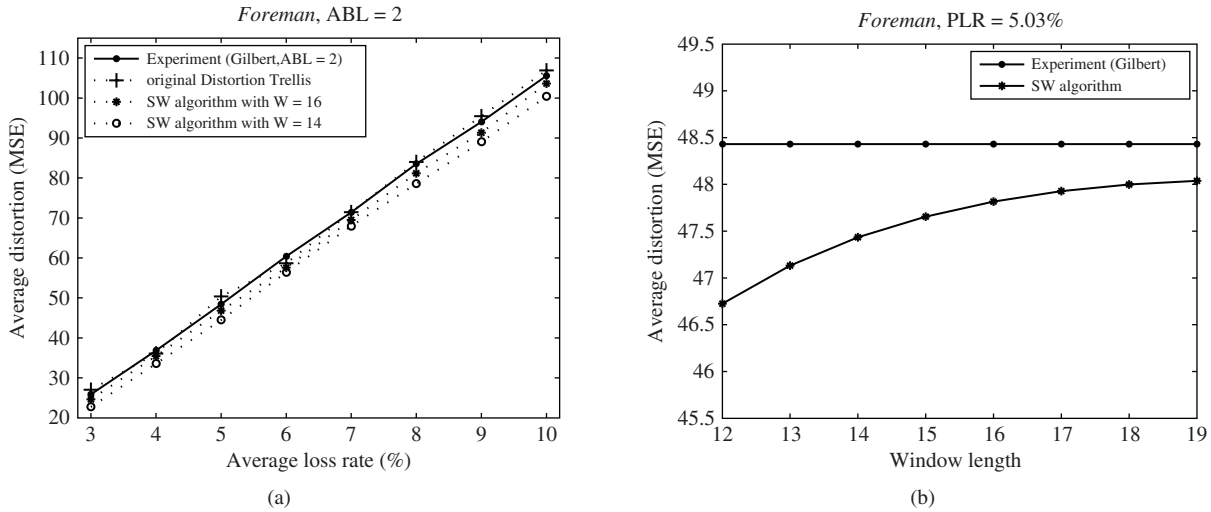


Fig. 5. (a) Average distortion comparison and (b) average expected distortion over all P-frames of *Foreman* versus window length.

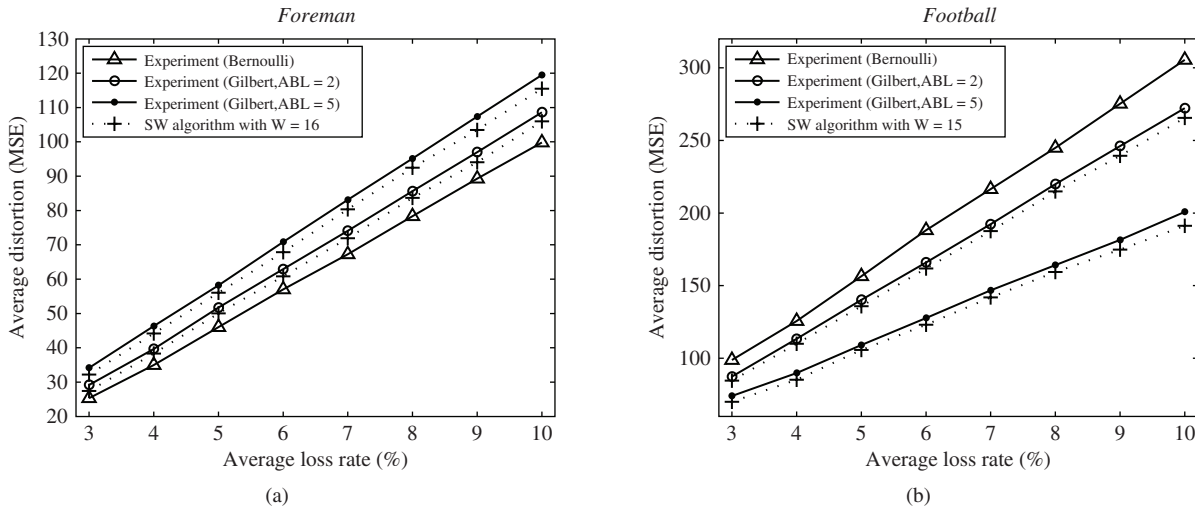


Fig. 6. Average distortion versus PLR for both ABL = 2 and 5; The corresponding distortions for a Bernoulli channel are also shown.

expected distortion quite well. As expected, we see that the estimation curve using the SW algorithm is always under the experimental curve and that its performance improves as the window length increases. These plots indicate that the SW algorithm could be used as a very good approximation of the original model, especially at larger window lengths.

To examine the influence of the window length on the accuracy of the SW algorithm, Fig. 5(b) plots the average expected distortion over all P-frames of *Foreman* versus the window lengths from 12 to 19 at $PLR = 5\%$ and $ABL = 2$, based on the same simulation data set as used in Fig. 5(a). We clearly see that the SW algorithm generally underestimates the expected distortion. This is because when calculating the expected distortion of the current frame, the distortion from frames outside the sliding window is ignored by the SW algorithm. In particular, we observe that smaller window lengths lead to larger estimation error values, because a smaller window ignores more distortion components from the past. We also observe that with the increase of the window length W from 12 to 16, the performance of the SW algorithm increases gradually, as discussed in Section IV. However,

increasing the window length further does not bring as much performance gain. We believe that this is because the fading behavior of the impulse channel distortion often follows an exponential decay curve [26], or at least follows a similar degrading trend. That is why the SW algorithm performance does not increase linearly with the window length. In our experiments, the SW algorithm with $W \leq 16$ is good for most examined cases, whereas for some fast decaying sequences even smaller W also provides acceptable results. Hereafter, we will use only the SW algorithm to estimate the expected distortion for much longer sequences.

Next, the average expected distortion over all P-frames (i.e., the quantity D_N/N) versus from 3 to 10%, for both $ABL = 2$ and 5, is plotted in Fig. 6. The tested sequences include *Foreman* and *Football*. For *Foreman*, the first 390 frames are coded, while for *Football*, the first 240 frames are coded. For each tested PLR and ABL pair, we simulate a Gilbert loss process and generate 90 000 random loss patterns. For each loss pattern, the distortion model in (16) is used to predict the decoder distortion. The window length used in the SW algorithm is 16 for *Foreman* and 15 for *Football*. The average expected distortion

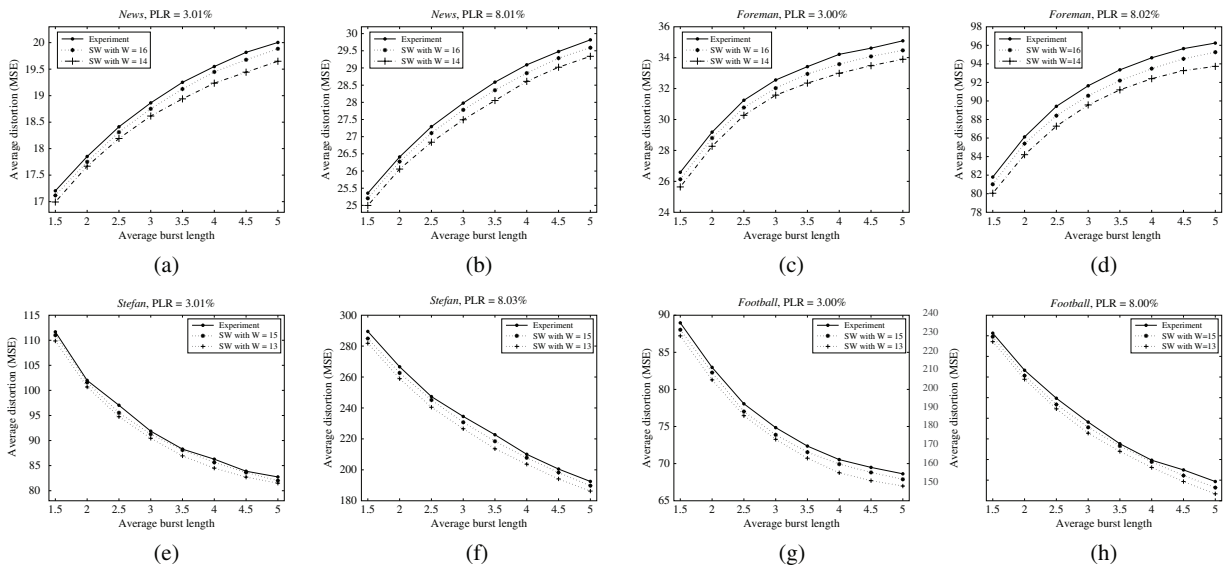


Fig. 7. Average distortion versus burst length of 1, 1.5, ..., 5 for a given packet loss rate (3% or 8%).

tion for the case of a Bernoulli channel at the same loss rate is also plotted in the same figure for comparison, where 1000 loss traces are generated at each loss rate for this channel model.

We can see that the SW algorithm accurately estimates the average expected distortion over most of the range of the average loss rate. At high loss rate, the SW is less accurate, but still matches well with the actual distortion curve. The good match between the theoretical data and the measured data tells us that the proposed model can be used to estimate and analyze the impact of bursty losses on the average video quality. Moreover, we see that though the window length used for *Football* is smaller than that for *Foreman*, the accuracy of the SW algorithm is similar. We will discuss this later. From both figures, we also see that at the same *ABL*, the average distortion increases linearly with the *PLR*.

Interestingly, we observe from Fig. 6 that for the same average loss rate the expected distortion for the Gilbert channel can be smaller or larger than that for the Bernoulli channel depending on the video content. Specifically, for the *Foreman* sequence the former is larger, while for the *Football* sequence the opposite is true. Even more interestingly, we observe that increasing the average burst length does not always contribute to a larger expected distortion, for a given average loss rate. For example, in the case of *Foreman*, a larger *ABL* leads to a larger expected distortion, at the same *PLR*, as seen from Fig. 6(a). However, the opposite holds in the case of *Football*, as seen in Fig. 6(b). To the best of our knowledge, the aforementioned experimental result is reported for the first time here. This proves again that the burst length does matter, i.e., it does affect the video quality.

To study further the impact of the average burst length on the average video quality, in Fig. 7 we show the average expected distortion over all P-frames (D_N/N) versus burst length of 1, 1.5, ..., 5 for a given packet loss rate (3% or 8%). The tested sequences include *News*, *Foreman*, *Stefan*, and *Football*. For each sequence, the first 200 frames are coded. The examined loss rates include 3 and 8%. For each tested *PLR* and *ABL* pair, 60000 loss traces are generated.

From Fig. 7, we see that the estimated distortion matches the measured data well along the whole tested burst length range at both tested loss rates.

From Fig. 7, we clearly observe that the average expected distortion D_N/N does NOT always increase as the average burst length increases at the same average loss rate. For the sequences *Stefan* and *Football*, increasing the average burst length will reduce D_N/N . This confirms the earlier findings from Fig. 6 that at the same average loss rate, a larger average burst length does not always lead to a larger distortion in the case of a Gilbert channel.

It is worth noting that the above observations seem different from those in [14], where it can be found that “longer burst length always causes larger MSE distortion.” We believe that this is due to a difference in the experimental setup. In particular, [14] aims to test whether burst length matters, and therefore it measures the total distortion at the decoder versus varying burst lengths, implying that the average loss rates that are used are proportional to each burst length; while in our experiment we consider how the average burst length affects the expected distortion if the average loss rate remains constant. In particular, from the Gilbert channel model shown in Fig. 1, we can see that increasing *ABL* while keeping *PLR* constant implies at the same time reducing the parameters q and p proportionally (recall that $ABL = 1/q$ and $PLR = p/(p+q)$).

In the following, we analyze our experimental results using the proposed distortion trellis model. From (15), it is clear that D_N is the sum of 2^N components, i.e., $D_N = \sum_{r=1}^{2^N} D(k_N^r) \cdot P(h_N^r)$. Define $\mathcal{D}_r = P(k_N^r) \cdot D(k_N^r)$ as the r th component. We observe that several components are much larger than almost all the other components. For the D_N versus *ABL* curve, these large components will determine its trend. Many other components are relatively small and thus make less contribution to D_N , though they may still affect the shape of the D_N versus *ABL* curve.

We first try to explain why D_N/N is an increasing function of *ABL* for *Foreman* and *News*. Specifically, we employ multiple 10-frame segments taken at different positions in the

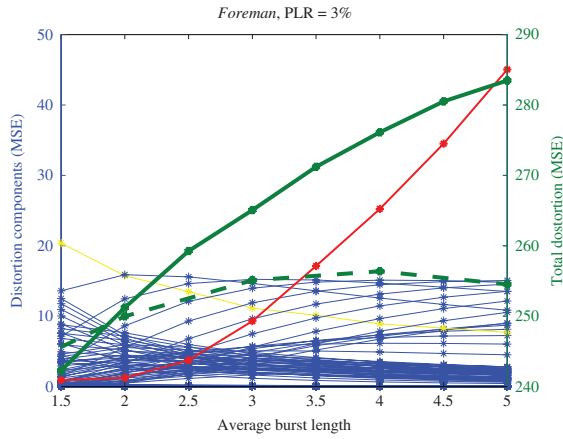


Fig. 8. Distortion components \mathcal{D}_r , $r = 1, \dots, 2^{10}$ of D_{10} versus ABL . The red curve is \mathcal{D}_{210} . The green solid curve is D_{10} . The green dash curve is $D_{10} - \mathcal{D}_{210}$. The yellow curve is \mathcal{D}_{1+29} .

Foreman sequence to calculate an “average D_{10} ” and plot all components of the average D_{10} versus ABL in Fig. 8. We discover that some relatively large components monotonically increase with ABL , which makes D_{10} an increasing function of ABL . Although many other components decrease with ABL , they are relatively small and thus cannot influence the overall trend of D_N as a function of ABL . Note that the component \mathcal{D}_{210} (the red curve in Fig. 8) increases quite quickly and becomes much larger than all the others starting from $ABL = 3.5$. The quantity $D_{10} - \mathcal{D}_{210}$ is also plotted with a dashed line. We see that without \mathcal{D}_{210} , the total distortion becomes a decreasing function of ABL for $ABL \geq 4$. This implies that the single component \mathcal{D}_{2N} contributes the most to make D_N increase with ABL at high average burst lengths. Additionally, note that component \mathcal{D}_{1+29} (plotted in yellow) is the main decreasing component. Although the results are obtained by analyzing short segments of *Foreman*, we believe it also holds in more general cases. Applying the same analysis based on distortion components can help to explain the upward trend of the D_N versus ABL curve for the *News* sequence. Note that this figure is not included here in order to conserve space.

Moreover, it is an interesting observation from Fig. 7(a)–(d) that the rate of increase of the average distortion D_N/N gradually decreases as ABL increases. We believe this is because there are still many components of D_N decreasing with ABL , as shown in Fig. 8. Though these components are too small to make the trend of D_N versus ABL curve change from upward to downward, they still slow down the increasing rate of D_N as a function of ABL . In other words, they gradually decrease the slope of the D_N versus ABL curve.

Next, we apply the same analysis to explain why D_N/N is a decreasing function of ABL for *Stefan* and *Football*. Using the same approach as used for creating Fig. 8, we compute the 10-frame average D_{10} for the *Football* sequence and plot in Fig. 9 all components of D_{10} for ABL from 1.5 to 5 at $PLR = 3\%$. Compared to the corresponding graphs from Fig. 8, it can be seen that many distortion components in Fig. 9 have a similar shape, however, exhibiting different relative magnitudes. For example, the component \mathcal{D}_{210} (the red curve in Fig. 9) is not that large now, while \mathcal{D}_{1+29} becomes a large and important component. Finally, from Fig. 7(g, h) we can see that the

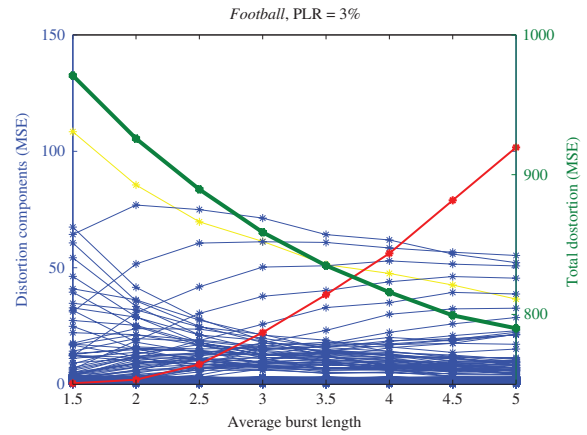


Fig. 9. Distortion components \mathcal{D}_r , $r = 1, \dots, 2^{10}$ of D_{10} versus ABL . The red curve is \mathcal{D}_{210} . The green solid curve is D_{10} . The yellow curve is \mathcal{D}_{1+29} .

rate of decrease of the average distortion D_N/N reduces as ABL increases. This is due to the fact that many small but increasing components slow down the decreasing rate of D_N as a function of ABL , as shown in Fig. 9. The results for *Stefan* can be explained in a similar fashion and are not included here for space considerations.

Although we know that the trend of the D_N versus ABL curve is affected by the sequence characteristics, we should point out that the above results do not answer specifically how. A plausible explanation in our opinion can be the following observation: *Stefan* and *Football* both have small values for the parameter v compared to *News* and *Foreman*. Heuristically, it is a reasonable conjecture that in a Gilbert channel at the same average loss rate, for sequences with small parameter v , especially smaller than u , increasing the average burst length may decrease the average distortion and vice versa. We leave the detailed investigation of this conjecture as future work.

From Fig. 7 we can see that the SW algorithm better matches the measured data in the case of *Stefan* and *Football* relative to *Foreman* and *News*, for the same window length. That is because the parameter v for *Stefan* and *Football* is smaller. In particular, in the case of *Foreman* and *News* the distortion in a lost frame decays over the subsequent frames more slowly, which implies that a larger window length is needed to achieve the same prediction accuracy.

Fig. 10 shows the expected distortion versus frame number for *Foreman* and *Football* at $PLR = 8\%$ and $ABL = 3$. For each sequence the first 100 frames are coded. We see that the estimated distortion using the SW algorithm with $W = 16$ fits the measured distortion values well. This tells us that the proposed model predicts well the frame level expected distortion and therefore can be employed to improve the performance of some frame-based error resilient techniques for video transmission over burst loss channels.

In order to understand how the average loss rate and the average burst length jointly affect the video quality, Fig. 11 plots the average expected distortion versus both parameters PLR and ABL for the sequences *Foreman* and *Football*. The predicted data is obtained by the SW algorithm. The tested window length is 15 for *Foreman* and 14 for *Football*. In order

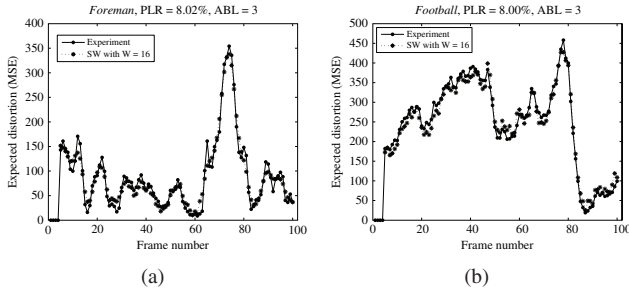


Fig. 10. Expected distortion versus frame number: (a) *Foreman* and (b) *Football*. PLR = 8% and ABL = 3.

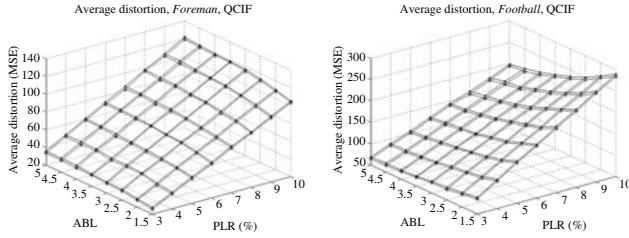


Fig. 11. Average expected distortion versus both ABL and PLR. The black mesh denotes the measured data and the red mesh denotes the estimated data by the SW algorithm.

to more easily differentiate the measured and the theoretical data in the 3-D graphs, we use smaller window lengths than those used in Fig. 7. The tested *PLRs* are from 3 to 10%, and the tested *ABLs* are 1.5, 2, ..., 5. First, we see from Fig. 11 that at all tested *PLR* and *ABL* values the SW algorithm provides a rather accurate prediction of the expected channel distortion. Next, we observe that at the same *ABL*, the expected distortion increases linearly with the *PLR*. Finally, we also observe that for the same *PLR*, as the *ABL* increases, the expected distortion increases with a decreasing rate for *Foreman*, while for *Football* the expected distortion decreases with a decreasing rate. This last observation matches what we saw earlier in Fig. 7.

At the end of this section, we would like to study the complexity of the proposed SW algorithm with simulations. We consider the distortion estimation complexity for all P-frames in a GOP, using SW with window length W . According to Algorithm 1, for estimating d_n , $n \leq W$ one needs to calculate both d_n^r and $P(k_n^r)$, for $r = 1, \dots, 2^n$, i.e., to run the loops from 6 to 7 in Algorithm 1, which approximately include $3 \cdot 2^n$ multiplication operations and $3 \cdot 2^{n-1}$ addition operations, while for estimating d_n , $n > W$, one only needs to calculate the distortion within the sliding window, i.e., to run the procedures from 8 to 12 in Algorithm 1. These procedures approximately include $2 \cdot 2^W$ multiplication operations and $3 \cdot 2^{W-1}$ addition operations. In summary, to calculate d_n , $n = 1, \dots, N$, the SW algorithm with window length W requires totally $(N - W) \cdot 2^{W+1} + \sum_{n=1}^W 3 \cdot 2^n$ multiplication operations and $(N - W) \cdot 3 \cdot 2^{W-1} + \sum_{n=1}^W 3 \cdot 2^{n-1}$ addition operations. To show some specific examples, we use the SW algorithm to estimate d_n for the first 100 P-frames of the QCIF *Foreman*, and the used window lengths are 13, 14, 15, and 16. Simulations were run on Intel Core2 Duo T8100 2.10 GHz

with 2-GB RAM. Table I lists the complexity results. The second column indicates the total cost of the multiplication operations when calculating d_n for all 100 P-frames, while the third column indicates the total cost of the addition operations. Table I also gives the average elapsed time per frame in the fourth column. From Table I, we can see generally that the computational cost increases linearly with the window length W . It also shows that the computation can be completed within acceptable time.

VI. DISTORTION TRELLIS FOR EXTENDED GILBERT CHANNEL

The distortion trellis model described before focuses on the case of the two-state Markov loss process or the Gilbert loss process. However, researchers have proposed many other models for the internet loss process. For example, the work in [17] proposes a more general finite-state Markov-model (so-called extended Gilbert model) to characterize the internet packet loss. We now show given the extended Gilbert loss model, how the distortion trellis works.

Fig. 12 shows the Markov chain of the extended Gilbert loss model with $m + 1$ states. In this model, State 0 represents “packet received” and State k , $0 < k < m$ represents “following a received packet, k or more consecutive packets lost” and State m represents “ m consecutive packets lost.” In the view of [17], State m can be considered as a bursty loss event over a window of size m . The detailed model description can be found in [17]. Define $p_{(k-1)k}$ as the transition probability from State $k - 1$ to State k , $k = 1, 2, \dots, m$. Then, the transition probability matrix for the extended Gilbert model with $(m + 1)$ states can be set up as

$$\mathbf{P} = \begin{bmatrix} 1 - p_{01} & 1 - p_{12} & 1 - p_{23} & \dots & 1 - p_{(m-1)m} & 1 - p_{mm} \\ p_{01} & 0 & 0 & \dots & 0 & 0 \\ 0 & p_{12} & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & p_{(m-1)m} & p_{mm} \end{bmatrix}.$$

Then, the stationary probability $\boldsymbol{\pi} = (\pi_0, \pi_1, \dots, \pi_m)$ can be computed as follows:

$$\boldsymbol{\pi} \cdot \mathbf{P} = \boldsymbol{\pi}, \quad \sum_{i=0}^m \pi_i = 1 \quad (20)$$

where $1 - \pi_0$ equals the mean packet loss ratio. It can be seen that in an extended Gilbert model with $(m + 1)$ states, the transmission state (“Lost” or “Received”) of current packet is affected by the past (up to) m consecutive loss events. Note that the two-state Gilbert model is a special case of the extended Gilbert model when $m = 1$. Recall that the distortion trellis is applicable to any channel model as long as the loss pattern probability $P(k_n^r)$ can be obtained. In an extended Gilbert channel, $P(k_n^r)$ can also be derived recursively, as follows. Given m and $P(k_{n-1}^t)$, $t = 1, \dots, 2^{n-1}$, the loss

TABLE I
COMPUTATION COMPLEXITY FOR SW ALGORITHM

Window length	Total Mul	Total Add	Average time per frame
13	1474554	1093626	14 ms
14	2916346	2162682	52 ms
15	5767162	4276218	116 ms
16	11403258	8454138	261 ms

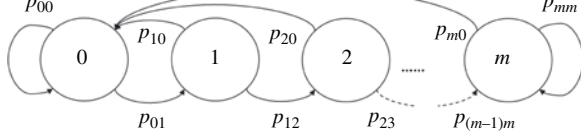


Fig. 12. Extended Gilbert channel model with $(m + 1)$ states.

pattern probabilities can be calculated as

$$\left\{ \begin{array}{l} P(k_n^{2^{m+1} \cdot r - 2^{m+1} + 1}) = p_{00} \cdot P(k_{n-1}^{2^m \cdot r - 2^m + 1}) \\ P(k_n^{2^{m+1} \cdot r - 2^{m+1} + 2}) = p_{01} \cdot P(k_{n-1}^{2^m \cdot r - 2^m + 1}) \\ \dots\dots\dots \\ P(k_n^{2^{m+1} \cdot r - 1}) = p_{m0} \cdot P(k_{n-1}^{2^m \cdot r}) \\ P(k_n^{2^{m+1} \cdot r}) = p_{mm} \cdot P(k_{n-1}^{2^m \cdot r}), \\ r = 1, \dots, 2^{n-m+1}. \end{array} \right. \quad (21)$$

Using (21), the loss pattern probabilities $P(k_n^r)$, $r = 1, \dots, 2^n$ can be obtained. Recall that the frame distortion d_n^r is uncorrelated with a specific channel model and can be frame-recursively calculated using (9) and (12). Then, based on (5), the expected distortion d_n for $(m + 1)$ -state extended Gilbert channel packet loss can be estimated as

$$\begin{aligned} d_n &= \sum_{t=1}^{2^n} d_n^t \cdot P(k_n^t) \\ &= \sum_{r=1}^{2^{n-m-1}} \sum_{i=1}^{2^{m+1}} d_n^{2^{m+1} \cdot r - i + 1} \cdot P(k_n^{2^{m+1} \cdot r - i + 1}). \end{aligned} \quad (22)$$

Note that when $m = 1$, the more general model in (22) reduces to the Gilbert loss distortion model in (13). Equation (22) provides an explicit way to estimate the expected distortion d_n caused by $(m + 1)$ -state Markov-model burst losses at the encoder/sender. The cumulative expected distortion D_N then can be estimated directly using (15) and (16). Still, the complexity of this model grows exponentially with n , as discussed at the end of Section III-C. Here we also employ the SW algorithm to calculate d_n with low complexity. Because the SW algorithm is uncorrelated with a specific channel model, here it has the same form as described in Algorithm 1. Note that although the extended Gilbert channel model is somewhat more complex than the Gilbert one, the proposed distortion trellis and the SW algorithm for both channel models have almost the same complexity.

To validate the accuracy of the proposed distortion trellis for extended Gilbert losses, we conduct a set of simulations

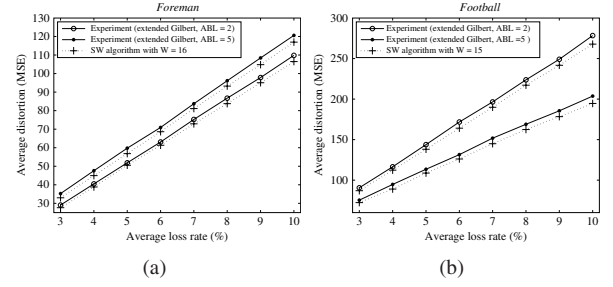


Fig. 13. Average distortion versus PLR for an extended Gilbert channel.

with two QCIF sequences, *Foreman* (15 frames/s) and *Football* (30 frames/s). For *Foreman*, the first 390 frames are coded, while for *Football* the first 240 frames are coded, both in the IPP... manner. The encoding and decoding setup follows the line described in Section V-A, so the parameters u and v are the same. We again assume that the I-frame can always be decoded correctly. Without losing generality, we use the six-state extended Gilbert lossy channel to model the end-to-end network, i.e., m is set to 5. We simulate different extended Gilbert lossy channels with different transition probability matrices, with the corresponding average loss rate *PLR* from 3 to 10%. For each simulation channel, we generate 90 000 loss traces with different loss patterns. For each loss pattern, we only use the SW algorithm to predict the decoder distortion in this simulation. Based on the observation results in Section V, we believe it is a reasonable deduction that, if the SW algorithm performs well, the original model would be more accurate.

Fig. 13 plots the average expected distortion (D_N/N) over all P-frames versus *PLR* from 3 to 10% for two different average burst lengths. Note that the average burst lengths here are calculated according to the burst length distribution for the simulation loss traces. The window length used in the SW algorithm is 16 for *Foreman* and 15 for *Football*. From Fig. 13, it is clear that the proposed model performs well for the extended Gilbert losses, as shown by its accuracy in estimating the measured distortion along all tested *PLRs*. As expected, we also see that the SW algorithm always underestimates the measured distortion. Moreover, we also observe that at the same *PLR*, for *Foreman*, increasing the *ABL* leads to a larger distortion, while for *Football*, the opposite is true. This observation matches with what we found earlier for Gilbert bursty losses in Section V.

So far, we examined the proposed distortion trellis model based on Markov-model loss simulations. Next, we would like to compare the modeled data with the results produced from realistic network packet traces. With this comparison, we can assess whether the proposed model is useful in estimating the expected distortion in reality. For this purpose, we performed the following experiment in December 2008.

First, the first 381 frames of the QCIF sequence *Foreman* are encoded in the IPP... manner, using the H.264 encoder JM12.2 [25] with the baseline profile at 15 frames/s and $QP = 28$. To create the packet traces, we set two Unix workstations as the sender and receiver, respectively located at Nanyang Technical University, Singapore and Beijing University of Posts and Telecommunications, Beijing. The

TABLE II
PACKET LOSS STATISTICS: DISTRIBUTION OF BURST LENGTH

Burst length	0	1	2	3	4	5
Occurrence	263792	993	662	309	176	68

software running on the workstations are not real video communication applications. However, since our purpose is to obtain loss traces for packet video transmission over realistic networks, we could achieve this by appropriate and reasonable experiment setup. A sample sequence containing 380 user datagram protocol (UDP) packets is used in this experiment, where the transmission spacing is set to 30 ms, i.e., the UDP packet is transmitted periodically per 30 ms. The packet size is set to 1500 bytes, so that each frame of the coded *Foreman* is smaller than the payload size and the assumption that one packet contains one frame could hold true. Then, We transmit the sample sequence between the two workstations, and record the index numbers of missing packets into loss trace file. In this case, the trace can be used as if it were created by real internet video communication applications. For accounting for the channel statistical characteristics well, we transmit the sequence for 700 times, and for each transmission session we generate one loss trace. Then, each loss trace can be regarded as a loss pattern for all P-frames of the *Foreman* sequence. Here the assumption of the correct decoding for every I-frame is implicit. For each loss trace, we decode the *Foreman* and record the MSE distortion for each P-frame. Note that the previous frame copy is used as the concealment strategy. Then, the actual average distortion D_{actual} could be measured by averaging the distortion of that frame over all traces, and $D_{\text{actual}} = 8.59$. Note that the measurement method is similar to what we have used in Section V, and the only different is that here the loss trace is obtained from realistic networks, rather than Markov loss simulations.

To obtain the modeled data, first the state transition probabilities of the channel should be calculated as follows. Based on the trace file, we can obtain the burst length distribution statistics as listed in Table II. We can see that the packet losses clearly show the expected burstiness, and the loss length distribution shows a heavy-tailed nature. From the statistics, one can easily calculate that the average loss rate is 0.83% and the average burst length equals 1.87. Let $o_l, l = 1, \dots, m$ denote the occurrence of a burst loss with length l , nr denote the number of the received packets, shown in Table II. Then, according to the conclusion in [17], the transition probabilities for a Gilbert loss model can be calculated as

$$p = \left(\sum_{l=k}^m o_l \right) / nr, \quad q = \left(\sum_{l=1}^m o_l - 1 \right) / \left(\sum_{l=1}^m l \cdot o_l - 1 \right). \quad (23)$$

If we consider the network loss as an $(m+1)$ -state extended Gilbert loss process, the corresponding transition probabilities can be calculated as

$$p_{01} = \left(\sum_{l=k}^m o_l \right) / nr, \quad p_{(k-1)k} = \left(\sum_{l=k}^m o_l \right) / \left(\sum_{l=k-1}^m o_l \right) \quad (24)$$

where $k = 2, \dots, m$.

Then, based on the proposed model, the expected distortion can be estimated using the SW algorithm. Let D_{gil} denote the modeled expected distortion for the Gilbert losses and D_{egil} denote that for the extended Gilbert losses. The modeled results are: $D_{gil} = 8.10$, $D_{egil} = 8.31$. Comparing them with the actual measure distortion $D_{\text{actual}} = 8.59$, it can be found that, both the proposed models, the Gilbert and the extend Gilbert, predict the measured distortion well, and the latter model is more accurate. The comparison demonstrates that the proposed distortion trellis model also performs well in estimating the expected distortion for actual network collected data and that therefore can be useful in real-life scenarios.

VII. CONCLUSION AND FUTURE WORK

Despite the fact that many researchers have modeled the current internet loss behavior using various Markov loss models [5], [16]–[19], there is a lack of work on modeling and analyzing the distortion caused by Markov-model bursty packet losses in decoded video. In this paper, we have proposed a mathematical model, which is denoted the distortion trellis model, for estimating the expected MSE distortion for video transmission over a Markov-model burst-loss channel. Without loss of generality and for simplicity, we established our distortion model based on a two-state Markov loss model or the Gilbert model [20]. We also extended the proposed model to a more general form, for $(m+1)$ -state Markov losses, or the extended Gilbert losses as in [17]. The proposed model can also be easily extended to most familiar Markov loss models [17], [21]. In general, the distortion trellis model can be used in all channels, only if in that channel the loss pattern probability is computable. The model takes into consideration both the decoding dependencies and the packet loss dependencies so that it accounts for the characterizations of the error propagation and the Markov loss process well.

Based on the fact that the error propagation often decays over the subsequent frames due to the effects of intra refreshing and spatial filtering [22], [24], a SW algorithm has been developed to compute the estimated expected distortion with low complexity. Using this algorithm, more than 90% of the computational burden can be saved compared to the original distortion trellis model. The comparison of the theoretical results and the measured data from our simulations shows that the proposed distortion trellis model and the SW algorithm are both accurate. Using the proposed models we also investigate the impact of the burst losses and the related average burst length on the resulting video quality. We establish that for the same average loss rate, the expected distortion does not always increase as the average burst length increases. Deeper understanding of this phenomenon may benefit from further investigations on this subject in the future.

Avenues for further exploration may also include employing our framework to study the impact of FEC on the received video quality for transmission over Markov-model lossy channels. Furthermore, it seems promising to combine the proposed models with some rate distortion optimization techniques in order to provide further improvements in end-to-end performance.

REFERENCES

- [1] N. Degrande, K. Laevens, and D. D. Vleeschauer, "Increasing the user perceived quality for IPTV services," *IEEE Commun. Mag.*, vol. 46, no. 2, pp. 94–100, Feb. 2008.
- [2] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.
- [3] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Selected Areas Commun.*, vol. 18, no. 6, pp. 966–976, Jun. 2000.
- [4] S.-R. Kang and D. Loguinov, "Modeling best-effort and FEC streaming of scalable video in lossy network channels," *IEEE/ACM Trans. Network*, vol. 15, no. 1, pp. 187–200, Feb. 2007.
- [5] M. Yajnik, S. B. Moon, J. Kurose, and D. Towsley, "Measurement and modeling of the temporal dependence in packet loss," in *Proc. IEEE INFOCOM '99*, vol. 1. New York, Mar. 1999, pp. 345–352.
- [6] D. Loguinov and H. Radha, "End-to-end internet video traffic dynamics: Statistical study and analysis," in *Proc. IEEE INFOCOM '02*, Jun. 2002, pp. 723–732.
- [7] J. C. Bolot, "End-to-end packet delay and loss behavior in the internet," in *Proc. ACM SIGCOMM*, Sep. 1993, pp. 289–298.
- [8] H. Yang and K. Rose, "Recursive end-to-end distortion estimation with model-based cross-correlation approximation," in *Proc. IEEE Conf. Image Process.*, vol. 3. 2003, pp. 469–472.
- [9] H. Yang and K. Rose, "Advances in recursive per-pixel end-to-end distortion estimation for robust video coding in h.264/avc," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 7, pp. 845–856, Jul. 2007.
- [10] S. Ekmekci and T. Sikora, "Recursive decoder distortion estimation based on ar(1) source modeling for video," in *Proc. IEEE Conf. Image Process.*, vol. 1. 2004, pp. 187–190.
- [11] Z. H. He, J. F. Cai, and C. W. Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 511–523, Jun. 2002.
- [12] Y. Wang, Z. Wu, and J. M. Boyce, "Modeling of transmission-loss-induced distortion in decoded video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 6, pp. 716–732, Jun. 2006.
- [13] C. Zhang, H. Y. S. Yu, and X. Yang, "Gop-level transmission distortion modeling for mobile streaming video," *Signal Process.: Image Commun.*, vol. 23, no. 2, pp. 116–126, 2008.
- [14] Y. J. Liang, J. Apostolopoulos, and B. Girod, "Analysis of packet loss for compressed video: Does burst-length matter," in *Proc. IEEE ICASSP*, Apr. 2003, pp. 684–687.
- [15] J. Chakareski, J. Apostolopoulos, S. Wee, W.-T. Tan, and B. Girod, "Rate-distortion hint tracks for adaptive video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, pp. 1257–1269, Oct. 2005.
- [16] J. Bolot and A. V. Garcia, "The case for fec-based error control for packet audio in the internet," in *Proc. ACM Int. Conf. Multimedia*, Sep. 1996.
- [17] H. Sanneck, G. Carle, and R. Koodli, "A framework model for packet loss metrics based on loss runlength," in *Proc. SPIE/ACM SIGMM Multimedia Comput. Network Conf.*, San Jose, CA, Jan. 2000, pp. 177–187.
- [18] B. Vucetic and J. Du, "Channel modeling and simulation in satellite mobile communication systems," *IEEE J. Selected Areas Commun.*, vol. 10, no. 8, pp. 1209–1218, Oct. 1992.
- [19] C. C. Tan and N. C. Beaulieu, "On first-order markov modeling for the rayleigh fading channel," *IEEE Trans. Commun.*, vol. 48, no. 12, pp. 2032–2040, Dec. 2000.
- [20] E. N. Gilbert, "Capacity of burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, no. 9, pp. 1253–1266, Sep. 1960.
- [21] L. N. Kanal and A. R. K. Sastry, "Models for channels with memory and their applications to error control," *Proc. IEEE*, vol. 66, no. 7, pp. 2032–2040, Jul. 1978.
- [22] T. Shu, J. Apostolopoulos, and R. Guerin, "Real-time monitoring of video quality in IP networks," in *Proc. ACM Int. Workshop Network Operating Syst. Support Digital Audio Video*, Jun. 2005.
- [23] B. Girod and N. Farber, "Feedback-based error control for mobile video transmission," *Proc. IEEE*, vol. 87, no. 10, pp. 1707–1723, Oct. 1999.
- [24] *Advanced Video Coding for Generic Audio-visual Services*, ITU-T Std. Recommendation H.264, May 2003.
- [25] *H.264/AVC Reference Software JM12.2* [Online]. Available: <http://iphome.hhi.de/suehring/tml/download/old-jm/jm12.2.zip>
- [26] Z. H. He and H. K. Xiong, "Transmission distortion analysis for real-time video encoding and streaming over wireless networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 9, pp. 1051–1062, Sep. 2006.



Zhicheng Li received the B.S. degree in electronic information engineering and the M.S. degree in signal processing from Shandong University, Jinan, China, in 2003 and 2006, respectively. He is pursuing the Ph.D. degree at Beijing University of Posts and Telecommunications, Beijing, China.

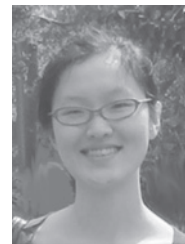
From 2007 to 2008, he was at the Nanyang Technological University, Singapore, as a government-sponsored Postgraduate. His research interests include image processing, multimedia communication, and traditional Chinese culture.



Jacob Chakareski received the M.S. degree in electrical and computer engineering from the Worcester Polytechnic Institute, Worcester, MA. He pursued the Ph.D. degree, also in electrical and computer engineering, at Rice University, Houston, TX, and at Stanford University, Palo Alto, CA.

From 2005 to 2006, he was a Postdoctoral Researcher with the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. From 2006 to 2007, he was a Senior Research Engineer with Vidyo Inc., Hackensack, NJ, where he worked on advanced methods for real-time video communication. Currently, he is a Senior Researcher with EPFL, where he investigates novel approaches to distributed computation and control in peer-to-peer systems.

Dr. Chakareski received the Best Student Paper Award at the IS&T/SPIE VCIP 2004 conference. He has held research positions with Microsoft and Hewlett-Packard. He has co-authored over 70 international publications and has several pending or approved patent applications. He actively participates in technical and organizing committees of several IEEE conferences and symposia every year.



Xiaodun Niu received the B.S. degree in power electronics from Nanyang Technological University, Singapore, in 2008.

Currently, she serves as the Team Leader of the IT Department of SNP Vite. Her fields of interests are signal processing and programming research.



Yongjun Zhang received the B.S. and M.S. degrees in telecommunication engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1991 and 1994, respectively, and the Dr. Ing degree in electrical and information engineering from the University Stuttgart, Germany, in 2003.

He is currently an Associate Professor with BUPT. His research interests include broadband network access and transport technologies. He has been in charge of and participated in three projects funded by the National High-Tech Research and Development Plan of China "863" Program and the National Natural Science Foundation of China. He has published one book in English and 30 technical papers.

Prof. Zhang was awarded with three national department-level Science and Technology Progress Prizes.



Wanyi Gu received the B.S. degree in physics from Peking University, Beijing, China, in 1970, and the M.S. degree from Beijing University of Posts and Telecommunications (BUPT), Beijing, in 1982.

She is currently a Professor with BUPT. Her research areas include optical communication networks, ultralong WDM optical transmission systems, and automatic switching optical networks, etc.