First-order distortion estimation for efficient video streaming at moderate to high packet loss rates

Christian Schmidt and Kenneth Rose
Department of Electrical and Computer Engineering
University of California
Santa Barbara, California 93106–9560
Email: {schmidt,rose}@ece.ucsb.edu

Abstract— **This paper reports on advances in end-to-end distortion estimation for streaming of pre-compressed video. It builds on the "first-order distortion estimate" (FODE) which employs a Taylor expansion of the distortion about the operating point of zero packet loss rate (PLR), while accounting for packet loss, error propagation and error concealment at the decoder. Improved estimation can be achieved by similar expansion about operating points of non-zero PLR, but would normally involve impractical complexity to evaluate the impact of many possible packet loss patterns. Instead, we propose a new method where FODE is combined with the "recursive optimal per-pixel estimate" (ROPE) to provide accurate, low complexity estimates via expansion about an arbitrary reference PLR vector. The reference vector need not be uniform and can exhibit the expected profile of effective PLR versus packet position in the group of packets (GOP). The method further accounts for the commonly neglected error propagation across GOP boundaries due to error concealment at the decoder. The main experimental results show significant gains in estimation accuracy at medium to high packet loss rates. Some preliminary results on embedding the estimate within a simple delivery policy selection system are also provided.**

I. INTRODUCTION

Video streaming over packet networks has received wide attention in recent years. Despite many advances in network technology and infrastructure, the best-effort nature of current networks remains a fundamental problem and still poses a significant challenge to the design of high-quality video streaming systems. Since most networks are heterogeneous and do not offer guaranteed end-to-end quality of service (QoS), packet loss cannot be avoided. Hence, a good video streaming system needs to incorporate error resilience/robustness mechanisms to mitigate the impact of packet loss and enable graceful performance degradation.

In live streaming, source-channel coding algorithms can be used to optimize media delivery. The sender has access to the source signal, and is aware of current network conditions such as bandwidth, packet loss rate (PLR) etc. Encoding and transmission decisions can be adapted to optimize a tradeoff between source compression and robustness to packet loss. This problem is naturally formulated within a rate-distortion (RD) framework, wherein various coding and transmission options are mapped to points on the operational RD curve. The RD framework enables managing the tradeoff between some resource-cost measure (e.g., channel bit rate, transmission power) and the resulting video quality. The key challenge in this context is accurate estimation of the end-to-end distortion, which accounts for all relevant factors such as quantization, packet loss, error propagation, and error concealment at the decoder. Most relevant to the work herein is the accurate end-to-end distortion estimation technique, called the "recursive optimal per-pixel estimate" (ROPE) [1].

When streaming pre-compressed video, compression is performed off-line and ahead of time, without knowledge of the eventual network condition. At the time of delivery, the system only has access to the compressed video signal (and possibly some side information). Hence, error robustness can only be enhanced by optimizing transport/delivery decisions. Some possible scenarios include: (i) forward error correction (FEC) per packet (bit-error channels) or between packets (packet erasure channels), (ii) packet retransmission protocols, (iii) service class selection in QoS networks (e.g., DiffServ), (iv) adaptation of the transmission parameters such as transmission power in wireless networks, etc.

Robust streaming of pre-compressed video has been addressed extensively, see [2]–[5]. Here too, the core problem evolves around accurate distortion estimation, and various estimates have been proposed. The intractable complexity of distortion estimation has been addressed in various ways, including neglecting inter-frame error propagation [2], or ignoring the effects of error concealment [3], [4]. However, such simplification results in inaccuracy in the distortion estimate that potentially impairs the performance of adaptation strategies.

An alternative approach to reduce complexity was proposed by our group in [6], [7], which introduced the first-order distortion estimate (FODE). The approach performs a first-order Taylor expansion of the end-to-end distortion, viewed as function of the packet loss rate (PLR), about a given reference point (in fact the origin $p = 0$). It produces a linear estimate of the distortion at any given set of PLR values for the packets, in terms of contributions from individual packets. It provides good estimation accuracy at low to medium packet loss rates.

Chakareski et al. proposed distortion chains to predict the decoder distortion [5]. Their zeroth-order distortion chain bears some similarity to FODE, but they seem to rely on the assumption that errors fade over time due to refresh, rather than build up.

Motivated by the potential of FODE and the recognition of the need to expand its applicability, the main contribution of this paper is an estimation algorithm that builds on FODE to achieve: (i) improved estimation at medium to high packet loss rates and, more importantly, (ii) estimation at arbitrary reference points for better adaptation to the channel statistics. The main experimental results demonstrate the improved estimation accuracy over various video sequences and over a variety of network conditions. We also show preliminary results on exploiting the distortion estimate within a simple delivery policy optimization system.

This paper is organized as follows: Section II reviews the basic FODE algorithm for end-to-end distortion estimation in pre-compressed video streaming. In Section III, we identify the main limiting factors that compromise estimation accuracy at medium to high PLRs, and develop an extension of FODE which leverages ROPE to achieve better estimates at higher PLR while maintaining acceptable computational complexity. Results are provided to evaluate the estimation quality. In Section IV, we provide results for a straightforward integration of the estimate within an RD framework for delivery of pre-compressed video.

II. End-to-end Distortion Estimation

This section reviews the basic FODE algorithm for end-to-end distortion estimation in streaming of pre-compressed video. FODE is also compared to Chou and Miao's "incremental additive distortion estimate" (IADE) [3], [4], to illustrate the performance potential of FODE.

A. Distortion Analysis

Without loss of generality, we assume that the compressed video is packetized into independent groups of packets (GOPs). The expected distortion of each GOP can be calculated separately since there is no coding dependency across GOPs (we will revisit and refine this statement later). However, packets within a GOP may depend on each other due to prediction or error concealment. Thus, the distortion for all packets in one GOP must be calculated *jointly*.

Let there be N source packets per GOP. Let p_i denote the effective PLR of packet i. Note that p_i is a function of both network condition and resilience strategy used for this packet. The PLR vector for the entire GOP is given by:

$$\mathcal{P} = \{p_0, ..., p_i, ..., p_{N-1}\}. \tag{1}$$

Packet i can either be received correctly, or considered lost, and we denote the random outcome by the binary variable b_i. The packet is received correctly if $b_i = 0$, and is lost if $b_i = 1$. The delivery status of the entire GOP is denoted by the binary random vector $\mathcal{B}$. There are a total of 2^N possible delivery events for each GOP. A particular event vector of the entire GOP is the realization of the delivery status vector and is represented by:

$$\mathcal{B}^{(k)} = \{b_0^{(k)}, b_1^{(k)}, ..., b_i^{(k)}, ..., b_{N-1}^{(k)}\}, \tag{2}$$

where $k = 0, 1, ..., 2^N - 1$ denotes the index of the event. The probability of the kth event vector $\mathcal{B}^{(k)}$ is given by:

$$p^{(k)} = \prod_{i=0}^{N-1} (1 - p_i)^{(1-b_i^{(k)})} p_i^{b_i^{(k)}}. \tag{3}$$

Let f, $\tilde{f}$ denote the value of some pixel in the original video and its corresponding reconstruction at the receiver, respectively. For the transmitter, $\tilde{f}$ is a *random* variable, since it depends on the actual delivery event, which is unknown to the transmitter. However, the decoder reconstruction is completely determined if the event vector of the entire GOP is given. Thus, the decoder reconstruction for the pixel under the kth event $\tilde{f}^{(k)}$ can be computed *exactly*. The end-to-end distortion of this pixel under the kth event is given by:

$$d^{(k)} = (f - \tilde{f}^{(k)})^2. \tag{4}$$

The overall GOP distortion under the kth event is:

$$D^{(k)} = \sum_{f \in GOP} d^{(k)}. \tag{5}$$

During the compression phase, the encoder can (in principle) compute $D^{(k)}$ for $k = 0, 1, ..., 2^N - 1$ and store these quantities as side-information at the server.

We can calculate the *expected* GOP distortion given the PLR vector $\mathcal{P}$ (which depends on the current channel status and delivery strategy):

$$\begin{aligned} E_{\mathcal{P}}\{D\} &= \sum_{k=0}^{2^N-1} p^{(k)} D^{(k)} \\ &= \sum_{k=0}^{2^N-1} \Big(\prod_{i=0}^{N-1} (1 - p_i)^{(1-b_i^{(k)})} p_i^{b_i^{(k)}}\Big) D^{(k)} \end{aligned} \tag{6}$$

Note that this expectation is *exact*: it considers all possible events and accounts for all sources of distortion.

In practical applications, however, there are two major and obvious drawbacks. First, 2^N values for D^k need to be computed and stored as side information for each GOP. Second, the expected distortion is a complicated function of the individual PLRs as seen in (6). Therefore, the exact analysis requires impractical complexity.

B. First-Order Approximation

Note that the exact expression for the expected end-to-end distortion in (6) is a polynomial function of the p_i in the PLR vector. Hence, we proposed to approximate (6) at any PLR vector $\mathcal{P}$ by using a first-order Taylor expansion around a reference PLR vector $\bar{\mathcal{P}}$:

$$E_{\mathcal{P}}\{D\} \approx E_{\bar{\mathcal{P}}}\{D\} + \sum_{i=0}^{N-1} \frac{\partial E_{\mathcal{P}}\{D\}}{\partial p_i}|_{\mathcal{P}=\bar{\mathcal{P}}}(p_i - \bar{p}_i)$$
$$= E_{\bar{\mathcal{P}}}\{D\} + \sum_{i=0}^{N-1} \gamma_i \Delta p_i, \quad (7)$$

where p_i, $\bar{p}_i$ denote the effective PLR and the reference PLR of packet i, respectively, and

$$\gamma_i = \frac{\partial E_{\mathcal{P}}\{D\}}{\partial p_i}|_{\mathcal{P}=\bar{\mathcal{P}}}, \quad (8)$$
$$\Delta p_i = p_i - \bar{p}_i. \quad (9)$$

If we let $\bar{\mathcal{P}} = \bar{\mathcal{P}}_0 = \{0, \ldots, 0\}$, i.e., use the origin as reference PLR, then the value of $E_{\bar{\mathcal{P}}}\{D\}$ can easily be pre-calculated. For this choice, $E_{\bar{\mathcal{P}}}\{D\}$ becomes the source coding (quantization) distortion per GOP and is simply computed during encoding. The only remaining task is to derive an algorithm to calculate the partial derivatives γ_i. The expected distortion (6) can be rewritten as

$$E_{\mathcal{P}}\{D\} = (1 - p_i)E_{\mathcal{P}}\{D|b_i = 0\} + p_i E_{\mathcal{P}}\{D|b_i = 1\}.$$

The first-order partial derivative is hence given by

$$\gamma_i = \frac{\partial}{\partial p_i} E_{\mathcal{P}}\{D\}|_{\mathcal{P}=\bar{\mathcal{P}}}$$
$$= E_{\bar{\mathcal{P}}}\{D|b_i = 1\} - E_{\bar{\mathcal{P}}}\{D|b_i = 0\}. \quad (10)$$

Note that, in general, computation of γ_i involves two distortion calculations (i.e. decoding runs), one with packet i received and one with packet i lost. This needs to be performed for each packet. However, the choice of $\bar{\mathcal{P}} = \bar{\mathcal{P}}_0$ implies that $E_{\bar{\mathcal{P}}}\{D|b_i = 0\} = E_{\bar{\mathcal{P}}}\{D\}$, which is already known. Hence the calculation of γ_i only requires one decoding run per packet, and the γ_i values for all packets in a GOP are easily computed by simulating all single packet loss patterns.

C. Estimation Accuracy

Since the actual PLR in the network is usually not far from zero, it is indeed convenient to use an "all-zero" reference PLR vector $\bar{\mathcal{P}}_0 = \{0, \ldots, 0\}$ for the Taylor expansion. To illustrate the estimation accuracy, we encoded the qcif sequence coastguard as H.264 [8] bitstream (JM 12.2 [9], 75kbps, 10fps, GOP=30 frames, 512 bytes/packet). We implemented basic FODE to calculate the partial derivatives during compression and use them for distortion estimation at different PLR vectors. For comparison, we implemented IADE [3], [4], which neglects error concealment (yielding a pessimistic bias to its estimate).

Figure 1(a) plots the FODE and IADE distortion estimates and the effective average distortion (averaged over 1000 patterns) in dB as a function of the PLR p. FODE tracks the distortion accurately. It provides a good estimate at low to medium PLRs, and the estimation error rises slowly towards the high PLRs as the distance from the reference PLR vector $\bar{\mathcal{P}}_0$ increases. The IADE algorithm overestimates the distortion due to the underlying assumption that a packet can only be decoded (and hence reduce the distortion), if all packets it depends on have been received correctly.

D. Delivery Optimization

Following [6], [7], we integrated both distortion estimates into an RD framework to optimize delivery policies for unequal error protection (UEP) by protecting the data packets of each frame with a variable number of FEC redundancy packets (for details refer to Section IV). We evaluated the delivery performance for the salesman sequence (qcif, 75kbps, 512 bytes/packet, 10fps, GOP=30 frames) achieved with the delivery policies selected under basic FODE and IADE. Performance was averaged over 1000 simulated loss patterns at each PLR.

Note that while IADE neglects error concealment during distortion estimation, during delivery FODE and IADE employ the same concealment algorithm. Hence any difference in delivery performance can be attributed to improved estimation and subsequent delivery policy selection. The FODE algorithm outperforms IADE (Figure 1(b)).

III. Reference PLR Vector

We start this section with some observations about FODE's estimation accuracy at medium to high PLRs. We identify issues that impair the estimate, which relate to the fact that the original FODE algorithm expands its estimate about the no-loss reference PLR vector $\bar{\mathcal{P}}_0$. Hence, a different, non-zero reference PLR vector should improve estimation, and we investigate how this can be achieved while avoiding an excessive increase in complexity.

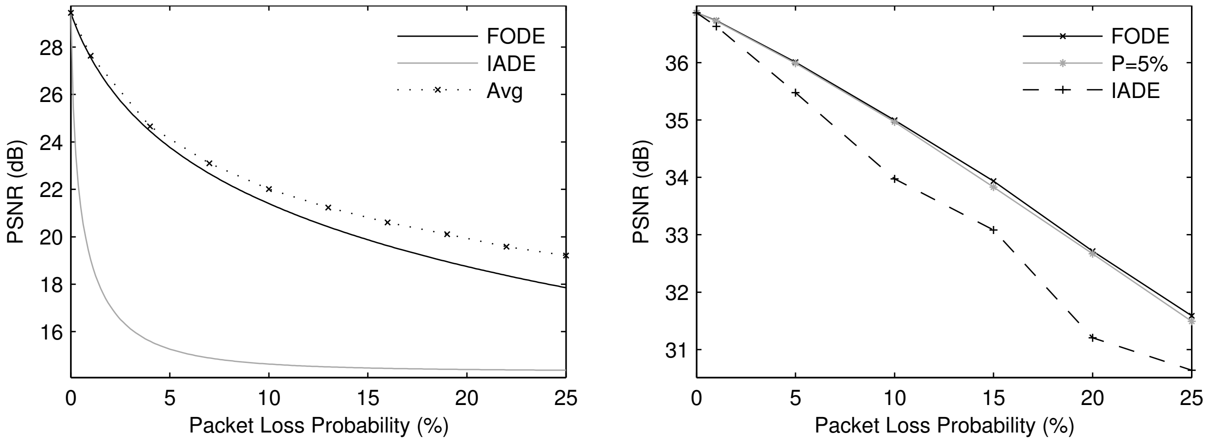
A. Estimation Mismatch

Referring back to Figure 1(a), we notice that FODE provides good estimation accuracy at low to medium PLRs, but the estimation error grows with the PLR. Recall that FODE is based on the first-order Taylor approximation (7). The quality of the approximation can be improved at the cost of increased complexity, e.g., via a second-order expansion [6], [7]. However, the fundamental problem remains that the approximation accuracy deteriorates with the distance from the reference point $\bar{\mathcal{P}}_0$. Note further that, during delivery (e.g., using UEP), the effective PLR for each packet p_i is adjusted individually, thus further increasing the distance from the uniform reference PLR vector $\bar{\mathcal{P}}_0$.

Moreover, the assumption that GOPs are independent is not truly valid. When data from the first frame in a GOP is lost, the accumulated error from the previous GOP spills over into the current one due to error concealment at the decoder — this effect is more pronounced at higher PLRs.

B. Non-zero Reference PLR Vector

For improved estimation accuracy, it would be beneficial to use a reference PLR vector closer to the actual channel PLR p. Hence, it would be beneficial to expand the FODE estimate about a non-zero PLR reference vector. Estimating the expected distortion and



(a) Estimation accuracy: FODE, IADE vs. actual distortion (PSNR vs. PLR p); coastguard, 75kbps

(b) Delivery (PSNR vs. PLR p): salesman, 75kbps

Fig. 1. Algorithm comparison FODE vs. IADE: (a) estimation accuracy, (b) delivery performance

partial derivative thereof at non-zero reference vectors may, however, be costly in computational complexity.

One approach would be to use the "many decoders in the encoder" (MDE) estimate [10], [11] during the computation of FODE's partial derivatives. A typical MDE implementation (see, e.g., the JM reference sofware [9]) requires the encoder to track M different reconstructions for the current frame from M simulated loss patterns. In this case, the complexity is increased by a factor of M. Recall that, in general, two decoding runs are needed per packet (10). Since these distortion terms must be computed over the *entire* GOP, this requires the simulation of $2MN$ loss patterns compared to only N patterns for the basic FODE. Moreover, to achieve good accuracy, M will be considerably large as we will see in simulations. Hence, the MDE estimate involves prohibitive complexity in terms of computation and storage.

Our objective is, therefore, to find an alternative approach to achieve FODE estimation about a non-zero reference PLR $\bar{\mathcal{P}}$ without incurring a substantial increase in complexity. In the context of live video networking, our group has previously proposed the recursive optimal per-pixel estimate (ROPE) [1]. ROPE is precisely the needed tool as it offers a means to efficiently and optimally track error propagation at any given PLR. Hence, we propose to complement FODE with a ROPE module for computation of the partial derivatives at low complexity, yet high accuracy. This combination provides a viable solution and offers FODE functionality at any appropriate reference PLR, thereby enabling considerably higher accuracy.

C. ROPE

This subsection provides a brief review of ROPE in its original setting of live video coding where the encoder has access to the source video signal, and some statistical information about the network. For simplicity (but without loss of generality) assume independent, uniformly distributed PLR p, and let the decoder employ standard temporal concealment of lost data. From the perspective of the encoder, the decoder reconstruction is a random process. It is not simply given by some additive uncorrelated noise as is often assumed in basic communication problems, but the ultimate effect of channel loss depends on error propagation through the prediction loop, error concealment efforts at the decoder and the impact of defensive measures at the encoder, etc.

1) Distortion Analysis: Let f_n^i denote the original value of pixel i in frame n, and let $\hat{f}_n^i$ denote its *encoder* reconstruction. The reconstructed value at the *decoder*, possibly after error concealment, is denoted by $\tilde{f}_n^i$. Note that, for the encoder, $\tilde{f}_n^i$ is a random variable. Using mean-squared error (MSE) as the distortion metric, the overall expected distortion for this pixel is

$$\begin{aligned} d_n^i &= E\{(f_n^i - \tilde{f}_n^i)^2\} \\ &= (f_n^i)^2 - 2f_n^i E\{\tilde{f}_n^i\} + E\{(\tilde{f}_n^i)^2\}. \end{aligned} \quad (11)$$

The computation of d_n^i requires the first and second moments of each random variable in the sequence $\tilde{f}_n^i$. These can be computed recursively, based on the moments from the previous frame, $\tilde{f}_{n-1}^i$. For the recursion step, there are two cases depending on whether motion-compensated prediction is disabled or enabled, called intra-coding or inter-coding, respectively.

2) Intra-coding: The packet containing pixel i is received correctly with probability $1-p$, producing $\tilde{f}_n^i = \hat{f}_n^i$. If the packet is lost, the lost block is concealed as $\tilde{f}_n^i = \tilde{f}_{n-1}^i$, with probability p. Therefore, the first and second moments of $\tilde{f}_n^i$ for an intra-coded pixel are computed as:

$$E\{\tilde{f}_n^i\} = (1-p)\hat{f}_n^i + pE\{\tilde{f}_{n-1}^i\} \quad (12)$$

$$E\{(\tilde{f}_n^i)^2\} = (1-p)(\hat{f}_n^i)^2 + pE\{(\tilde{f}_{n-1}^i)^2\} \quad (13)$$

3) Inter-coding: Let pixel i be predicted from pixel j in the previous frame, i.e. the encoder prediction is $\hat{f}_{n-1}^j$. The prediction error, e_n^i, is quantized to the value $\hat{e}_n^i$, which is transmitted together with the motion vector. Even if the current packet is correctly received, the decoder uses the *decoder* reconstruction of pixel j in the previous frame, $\tilde{f}_{n-1}^j$, for prediction, which is potentially different from the value used by the encoder: $\tilde{f}_n^i = \tilde{f}_{n-1}^j + \hat{e}_n^i$. The first and second moments of $\tilde{f}_n^i$ for an inter-coded pixel are:

$$\begin{aligned} & E\{\tilde{f}_n^i\} \\ = \;& (1-p)\left(\hat{e}_n^i + E\{\tilde{f}_{n-1}^j\}\right) + pE\{\tilde{f}_{n-1}^i\}, \qquad (14) \\ & E\left\{(\tilde{f}_n^i)^2\right\} \\ = \;& (1-p)E\left\{(\hat{e}_n^i + \tilde{f}_{n-1}^j)^2\right\} + pE\left\{(\tilde{f}_{n-1}^i)^2\right\} \\ = \;& (1-p)\Big((\hat{e}_n^i)^2 + 2\hat{e}_n^i E\{\tilde{f}_{n-1}^j\} \\ & +E\{(\tilde{f}_{n-1}^j)^2\}\Big) + pE\left\{(\tilde{f}_{n-1}^i)^2\right\} \qquad (15) \end{aligned}$$

Due to the recursive nature of the above formulae, we call this distortion estimate the "recursive optimal per-pixel estimate" (ROPE). It is optimal in the mean squared error sense, and uses all information available at the encoder to estimate the moments of the decoder reconstruction. Despite their imposing appearance, the recursion formulae are clearly benign in actual update computation of the first and second moments in terms of those of the previous frame.

4) Estimation Accuracy: In order to evaluate ROPE's estimation accuracy, we compare it against the MDE algorithm [10], [11] with $M = 60$ packet loss patterns. Figure 2 plots the absolute estimation error (in dB) of the ROPE and MDE estimates for the qcif sequence news. ROPE achieves a low estimation error across the whole PLR range. MDE60 provides a reasonable estimate, but it fluctuates. Most of the time, its estimation error is significantly bigger, but sometimes it is closer to the actual distortion than ROPE. This effect is caused by the discrete nature and insufficient statistics of the MDE60 estimate (this effect is even more pronounced on a per-frame basis).

In terms of complexity, the amount of computation required for ROPE is similar to MDE for $M = 2 \sim 3$. Even with twenty-fold complexity (MDE with $M = 60$), MDE has worse estimation accuracy than ROPE. MDE's estimation precision increases with M, but so does the complexity. Comparing the memory overhead, ROPE needs two floating point values per pixel ($8 = 2 \times 4$ bytes at single precision), while MDE requires M bytes/pixel ($2M$ if pixels have more than 8 bits). Overall, ROPE's complexity is significantly lower, while providing a better estimate. Note that a reduced complexity scheme has been proposed for MDE, called "few decoders in the encoder" (FDE) [12]. It models the underlying PLR distribution in order to simulate fewer patterns, but it assumes a limited window length for error propagation.

5) Subpixel Estimation: The original ROPE [1] was developed in the context of full-pixel motion compensation. Subsequently, an extension to estimate cross-correlation terms (e.g. due to pixel filtering operations such as subpixel interpolation) was proposed [13]. However, this method incurred substantial computational and

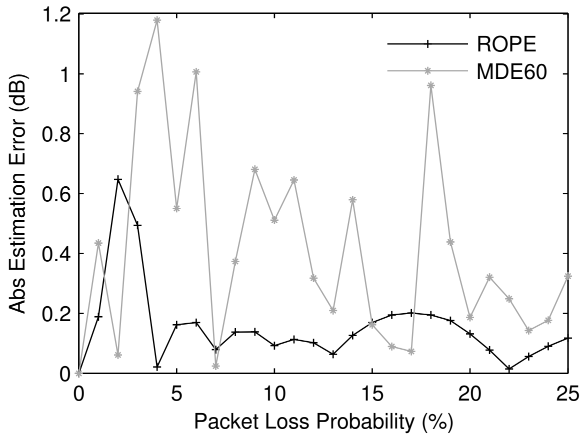


Fig. 2. Estimation error (absolute difference in dB) vs. PLR p (news, qcif, 10fps, 60kbps, 512 bytes/packet, GOP=30 frames)

storage complexity. Recently, a low complexity and high accuracy method to enable ROPE to accurately deal with subpixel interpolation and subsequent rounding [14], [15].

In this paper, we focus on full-pixel estimation only, but these ROPE extensions can easily be integrated into the FODE-ROPE algorithm.

D. FODE-ROPE Estimation

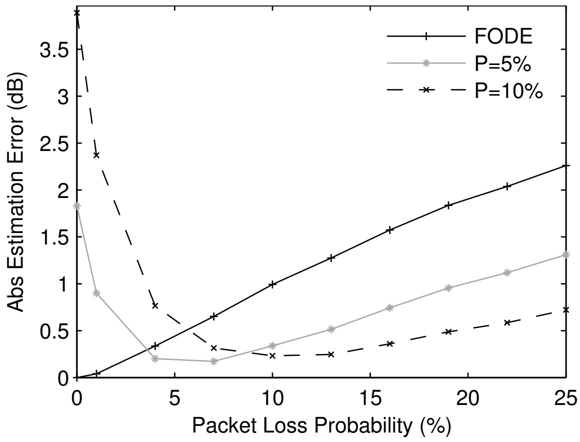
As proposed in Section III-B, we complemented FODE with a ROPE module to extend the applicability to non-zero reference PLR vectors.

Estimation of the expected distortion $E_{\mathcal{P}}\{D\}$ in equation (7) involves the distortion at the reference PLR vector $E_{\bar{\mathcal{P}}}\{D\}$ and the partial derivatives γ_i. Recall from Section II-B that, in the basic FODE algorithm, these were computed by conventional decoding, accumulating the distortion over all frames in a GOP. The calculation of $E_{\bar{\mathcal{P}},}\{D\}$ requires decoding the GOP once without losses, while the γ_i for the N GOP source packets are calculated by simulating all N single packet loss patterns.

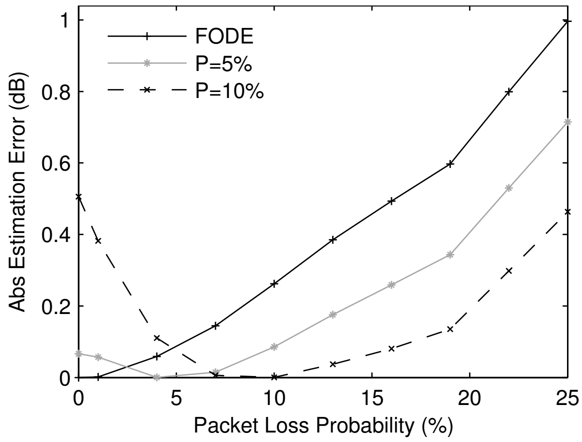
For FODE-ROPE estimation, the same basic procedure is performed. However, instead of conventional decoding, we track the (first and second) ROPE moments and employ those to calculate the (expected) distortion terms. The PLR p in the ROPE update equations (12)–(15) is replaced with the $\bar{p}_i$ corresponding to the source packet i from the reference PLR vector $\bar{\mathcal{P}}$. Note that the error propagation across GOP boundaries is elegantly accounted for as part of $E_{\bar{\mathcal{P}},}\{D\}$. Further, calculation of the γ_i in equation (10)) now requires two decoding runs, one with $p_i = 1$ and another with $p_i = 0$ (the other $p_j = \bar{p}_j$ from $\bar{\mathcal{P}}$ for $i \neq j$).

To evaluate the impact of the reference vector, we plot the resulting estimate for $\bar{\mathcal{P}} = 5\%$ and $\bar{\mathcal{P}} = 10\%$ (uniform) in Figure 3 (qcif sequences, 10 fps, 512 bytes/packet, GOP = 30 frames). Note that there is no requirement that $\bar{\mathcal{P}}$ be uniform, and these vectors were chosen to provide a simple evaluation of the new capabilities.

Figure 3(a) plots the estimation accuracy for a conventional loss-less encoding of the sequence stefan (150kbps,



(a) stefan, 150kbps (encode for error-free channel)



(b) carphone, 60 kbps (error-resilient encode for p=5%)

Fig. 3. Absolute estimation error (dB) vs. PLR p (qcif, 10 fps, 512 bytes/packet, GOP=30 frames)

10fps). Clearly, the $\bar{\mathcal{P}} = 10\%$ estimate tracks the distortion better at higher PLRs than the conventional FODE algorithm. However, there is an inherent trade-off between accuracy at low and high PLRs. As expected, at low PLRs the FODE estimate outperforms the $\bar{\mathcal{P}} = 10\%$ reference vector. The estimate based on the $\bar{\mathcal{P}} = 5\%$ reference PLR vector is closer to FODE at low PLRs (but still worse), performs better for medium PLRs, but worse than $\bar{\mathcal{P}} = 10\%$ at higher PLRs.

Figure 3(b) shows the distortion accuracy for an error-resilient encoding of the sequence carphone at 60kbps. Once again, the new estimates outperform the FODE $\bar{\mathcal{P}}_0$ estimate, with the $\bar{\mathcal{P}} = 10\%$ maintaining good accuracy until $p = 20\%$, and then slowly diverging.

IV. Delivery of Pre-Compressed Video

In this section, we integrate the different distortion estimates into an RD framework to optimize the delivery decisions for a pre-compressed video stream. As an example, we provide results for a non-scalable encoder using unequal error protection (UEP) based on FEC protection. We demonstrate that improved estimation translates to better delivery performance.

A. RD Framework

An adaptive error-resilience scheme can be described as a set of policy choices $\pi \in \{\pi^{(0)}, \pi^{(1)}, \dots, \pi^{(S)}\}$ for each packet. Each choice π results in an effective loss rate p_i for packet i, and incurs a rate/cost $c(\pi)$. Depending on the employed resilience scheme, the policy choices could be the number of retransmissions, the strength of the applied FEC, or the transmission power or coding scheme for wireless transmission etc.

The policy vector for a group of source packets (GOP) is defined as $\Pi = \{\pi_0, \dots, \pi_{N-1}\}$. The corresponding PLR vector and the cost vector are denoted by $\mathcal{P}(\Pi)$ and $\mathcal{C}(\Pi)$, respectively. The RD costs (expected end-to-end distortion and total cost) for a GOP are

$$E\{D(\mathcal{P}(\Pi))\} \approx E\{D(\bar{\mathcal{P}})\} + \sum_{i=0}^{N-1} \gamma_i (p_i(\pi_i) - \bar{p}_i) \quad (16)$$

$$\mathcal{C}(\Pi) = \sum_{0}^{N-1} c_i(\pi_i) \quad (17)$$

.

The optimal delivery policy is the policy that minimizes the Lagrangian:

$$\begin{aligned} & E\{D(\mathcal{P}(\Pi))\} + \lambda \mathcal{C}(\Pi) \\ \approx \quad & E\{D(\bar{\mathcal{P}})\} + \sum_{i=0}^{N-1} \gamma_i (p_i(\pi_i) - \bar{p}_i) + \lambda c_i(\pi_i) \quad (18) \end{aligned}$$

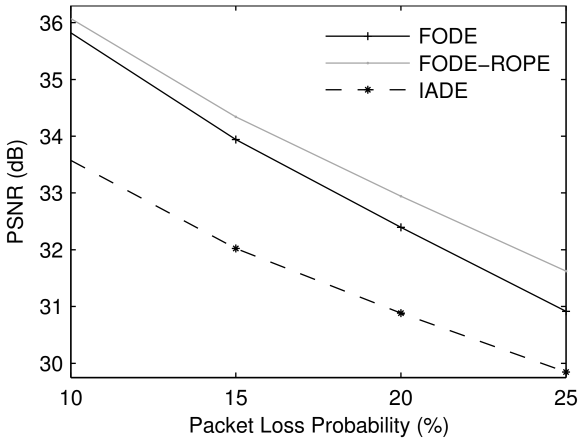
Note that the distortion estimate is additive in terms of contributions from the various p_i. Theoretically, the policies can be chosen independently for each packet, but practically the optimization may involve cluster decisions at the appropriate granularity level, e.g., grouping the packets in one frame for combined FEC policy selection.

B. Simulation Results

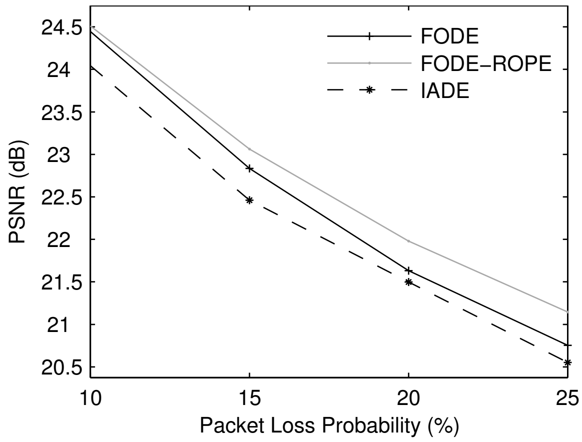
We simulate a non-scalable system with UEP using Reed-Solomon codes. The video is pre-compressed using the JM 12.2 reference software [9]. We use adaptive packetization: each packet is ≤ 512 bytes and contains one slice of the encoded frame (adaptive slicing).

Unlike the setting we employed for estimation accuracy evaluation (Section III-D), the current setting of delivery policy optimization calls for a nonuniform reference PLR vector closer to the effective loss rate experienced by each packet (after UEP).

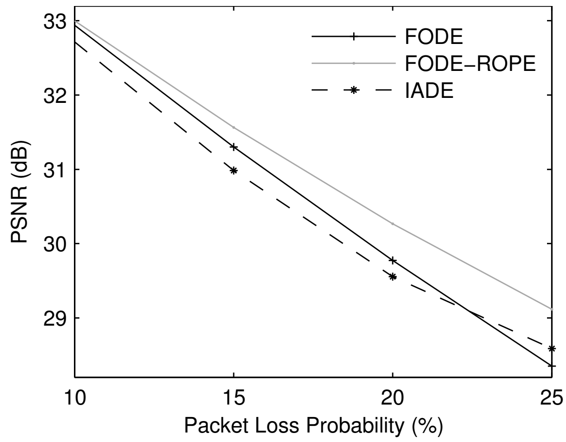
We compare the performance achieved under three algorithms: baseline FODE ($\bar{\mathcal{P}} = \bar{\mathcal{P}}_0$), FODE-ROPE and IADE. How do we generate the nonuniform reference PLR vector for FODE-ROPE? We observe that the *effective* PLR vector $\bar{\mathcal{P}}_{eff,FODE}$ from the delivery policy obtained under the baseline FODE algorithm provides a reasonable approximation. Hence, we choose $\bar{\mathcal{P}} = \bar{\mathcal{P}}_{eff,FODE}$ and compute the FODE-ROPE partial derivatives using that reference PLR vector. Finally, the FODE-ROPE delivery policy is computed using the new partial derivatives.



(a) akiyo, 60/75 kbps source/channel bit rate



(b) coastguard, 75/95 kbps source/channel bit rate



(c) container, 60/75 kbps source/channel bit rate

Fig. 4. Delivery performance, PSNR vs. PLR p (qcif, 10fps, 512 bytes/packet, GOP=30 frames)

Figure 4 shows the results for the three qcif sequences akiyo, coastguard and container for the medium to high PLR range (10 $\sim$ 25%). In Figure 4(a), the baseline FODE algorithm achieves a nice improvement over IADE, but the FODE-ROPE policy achieves an additional gain. In the case of the coastguard sequence (Figure 4(b)), baseline FODE only shows modest gains over IADE. Here, the FODE-ROPE method achieves significant gains over baseline FODE.

Finally, consider Figure 4(c). Baseline FODE barely improves upon IADE, and even performs worse than IADE at p=25% (far from its reference). Again, FODE-ROPE achieves a nice gain. We conclude that these gains are a direct result of improved estimation, since the underlying optimization algorithm is exactly the same for both FODE methods.

V. Conclusion

End-to-end distortion estimation is a fundamental and crucial problem in RD-optimized adaptive delivery of pre-compressed video over lossy networks. We build on the FODE algorithm, which offers accurate estimates of end-to-end distortion at low packet loss rates [6], [7]. Specifically, we complement FODE with the ROPE algorithm (originally proposed for distortion estimation in live video settings) to enable more flexible and accurate estimation about arbitrary reference PLR vectors. Simulation results demonstrate significant estimation gains at moderate to high packet loss rates.

The proposed estimation algorithm can be integrated with various delivery schemes, and provides robust transmission of pre-compressed video at low complexity. The basic approach is independent of the specific set of strategies or policies and is presented within a high-level rate-distortion optimized framework. It requires modest complexity due to the simplicity of the approximation / estimation model. A practical system with UEP through FEC is simulated and provides an example of potential gains for our approach.

In the future, we plan to investigate how estimates from different reference points can be combined for optimal estimation across the entire PLR range. Interesting applications include scalable coding with higher PLRs in the enhancement layers and packet scheduling.

References

[1] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience", *IEEE*

Journal on Selected Areas in Communications, vol. 18, no. 6, pp. 966–976, 2000.

[2] W. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 373–386, 2001.

[3] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra, "Error control for receiver-driven layered multicast of audio and video", *IEEE Transactions on Multimedia*, vol. 3, no. 1, pp. 108–122, 2001.

[4] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media", *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 390–404, 2006.

[5] J. Chakareski, J. G. Apostolopoulos, S. Wee, W.-T. Tan, and B. Girod, "Rate-Distortion Hint Tracks for Adaptive Video Streaming", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1257–1269, 2005.

[6] R. Zhang, S. L. Regunathan, and K. Rose, "End-to-end distortion estimation for RD-based robust delivery of pre-compressed video", *35th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 210–214, 2001.

[7] R. Zhang, S. L. Regunathan, and K. Rose, "Optimized video streaming over lossy networks with real-time estimation of end-to-end distortion", *IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, pp. 861–864, 2002.

[8] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Advanced video coding for generic audiovisual services", *ITU-T Recommendation H.264*, ISO/IEC 14496-10 AVC, March 2005.

[9] H.264/AVC Software Coordination: *http://iphome.hhi.de/suehring/tml/*

[10] T. Stockhammer, T. Wiegand, and S. Wenger, "Optimized transmission of H.26L/JVT coded video over packet-lossy networks",, *IEEE International Conference on Image Processing (ICIP)*, vol. 2, pp. 173–176, 2002.

[11] T. Stockhammer, D. Kontopodis, and T. Wiegand, "Rate-Distortion Optimization for H.26L Video Coding in Packet Loss Environment", *Packet Video Workshop*, 2002.

[12] G. Olmo, C. Cucco, M. Grangetto, and E. Magli, "Few decoders in the encoder: a low complexity encoding strategy for H.26L", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, pp. 641–644, 2003.

[13] H. Yang and K. Rose, "Recursive end-to-end distortion estimation with model-based cross-correlation approximation", *IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 469–472, 2003.

[14] H. Yang, and K. Rose, "Advances in recursive per-pixel estimation of end-to-end distortion for application in H.264", *IEEE International Conference on Image Processing (ICIP)*, vol. 2, pp. 906–909, 2005.

[15] H. Yang, and K. Rose, "Advances in Recursive Per-Pixel End-to-End Distortion Estimation for Robust Video Coding in H.264/AVC", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 7, pp. 845–856, 2007.