Low-Complexity Adaptive Streaming via Optimized A Priori Media Pruning

Jacob Chakareski and Pascal Frossard
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Institute - LTS4, CH-1015 Lausanne, Switzerland

Abstract— **Source pruning is performed whenever the data rate of the compressed source exceeds the available communication or storage resources. In this paper, we propose a framework for rate-distortion optimized pruning of a video source. The framework selects which packets, if any, from the compressed representation of the source should be discarded so that the data rate of the pruned source is adjusted accordingly, while the resulting reconstruction distortion is minimized. The framework relies on a rate-distortion preamble that is created at compression time for the video source and that comprises the video packets' sizes, interdependencies and distortion importances. As one application of the pruning framework, we design a low-complexity rate-distortion optimized ARQ scheme for video streaming. In the experiments, we examine the performance of the pruning framework depending on the employed distortion model that describes the effect of packet interdependencies on the reconstruction quality. In addition, our experimental results show that the enhanced ARQ technique provides significant performance gains over a conventional system for video streaming that does not take into account the different importance of the individual video packets. These gains are achieved without an increase in packet scheduling complexity, which makes the proposed technique suitable for online R-D optimized streaming.**

I. Introduction

Pruning of compressed and packetized sources is quite common in the media world today. This operation, also known as packet dropping, is performed by discarding packets from the compressed representation of the media source, either for communication or storage applications. Source pruning is necessary whenever the data rate of the source exceeds the available capacity of the storage system or the communication channel.

Deciding which packets to discard from a compressed media source can be very involved. This is due to the interdependencies between the media packets created at compression and their influence on the reconstruction quality of the source. Specifically, media is typically compressed using predictive schemes where the successful decoding of a packet is dependent on successful decoding of previous (and even future) packets. Then, pruning a packet that appears early in the prediction chain may trigger a significant amount of quality degradation along the successive packets in the prediction chain.

Scalable coding [1] has been proposed to deal with this problem, since the scalable (or layered) representations provide an intuitive way to select which parts of the compressed media to retain/discard so that the data rate constraint is met. However, scalable coding techniques have not gained a wide acceptance in practice, due to a few shortcomings, e.g., their coding inefficiency. Furthermore, the presence of different frame types, namely I, P and B, in conventional MPEG coding also provides a natural way of prioritizing segments of the media content when source pruning is performed. However, pruning non-scalable or non-prioritized packetized media presents a more challenging problem as the compressed data does not suggest a straightforward way of placing priorities on the media packets. In this paper, we focus on the problem of source pruning for non-scalably coded video streams. It should be noted that an alternative solution to source pruning for data rate adaptation is to simply re-encode the compressed media presentation at the available data rate. However, this approach, known as transcoding [2], may not be always feasible due to the higher complexity that is involved.

In this paper, we propose a framework for rate-distortion optimized pruning of compressed video sources. The framework selects which packets, if any, from the compressed representation of the source should be discarded so that the data rate of the pruned source is adjusted accordingly, while the resulting reconstruction distortion is minimized. The framework relies on a rate-distortion preamble that comprises the video packets' sizes, interdependencies and distortion importances. The framework can be exploited for efficient rate adaptation at a streaming server or at an intermediate proxy (for both unicast and broadcast applications), as it provides for fine packet classification based on pruning the compressed media stream at different target data rates. In particular, each packet can be tagged with a single rate threshold value above which the packet should be selected for streaming and below which the packet should be discarded. To this end, in conjunction with the pruning framework, we design an enhanced ARQ scheme for video streaming that achieves significant improvement in quality relative to conventional ARQ streaming, however without an increase in online complexity.

Most closely related contemporaneous works to the present paper are those on packet dropping in media networking and communication, such as [5–8], that consider various approaches for making intelligent packet dropping decisions, with or without R-D optimization. Another body of related works is that on low-complexity and R-D optimized streaming, such as those in [9, 10], where strategies for R-D optimized streaming with reduced complexity are proposed.

The paper proceeds as follows. In the next section, we present the rate-distortion preamble that succinctly describes the compressed video packets. How this information is employed to perform rate-distortion optimized pruning of a packetized video source is then described in Section III. The design of the enhanced ARQ scheme that relies on the pruning framework is the subject of Section IV. Next, in Section V we examine the performance of the pruning framework as a function of the employed distortion model, as described in Section III. In addition, we study in Section V the loss in performance of source pruning relative to re-encoding the original video signal at various data rates. We conclude this section with experimental results that explore the performance of the enhanced ARQ technique for streaming packetized video content and compare it to that of conventional ARQ video streaming. Finally, concluding remarks are provided in Section VI.

II. Rate-Distortion Preamble

When a media presentation is compressed, the encoded data are packetized into *data units* and stored in a file on a media server. All of the data units in the presentation have interdependencies, which can be expressed by a directed acyclic graph (DAG). Each node of the graph corresponds to a data unit, and each edge of the graph directed from data unit l' to data unit l implies that data unit l can be decoded only if data unit l' is first decoded. Associated with each data unit l is its size R_l in bits.

In addition, from the DAG we extract for every data unit l a set of data units $\mathcal{A}_l$ that can be potentially used to reconstruct l at decoding. Specifically, $\mathcal{A}_l$ includes all of the ancestors of l in the DAG that are necessary for decoding l, but also it contains in addition other data units that may be used to reconstruct l in case of an error event. Then, for every subset $\mathcal{C} \subset \mathcal{A}_l$ we can calculate the resulting reconstruction distortion $\Delta d_l(\mathcal{C})$ associated with data unit l for that particular event. This can be obtained as an auxiliary information when the media presentation is compressed.

Finally, we define the *rate-distortion preamble* for the media presentation to be the collection of packet sizes and reconstruction distortion information for every data unit in the presentation. To compute optimal packet selection decisions when adjusting the data rate of a media presentation, a pruning algorithm only needs to consider this compact description of the media presentation rather than the actual compressed content. How this is performed is the subject of the next section. It should be noted that the concept of a rate-distortion preamble has been considered earlier in the context of proxy-driven streaming [11].

III. R-D Optimized Source Pruning

Let there be L packetized data units in the media presentation. We are interested in finding the vector of packet selection actions $\boldsymbol{a} = (a_1, \ldots, a_L)$ for the presentation, where $a_i = 1$ denotes the action of keeping data unit i in the presentation, while $a_i = 0$ signifies the converse. The incurred reconstruction error (or distortion) for the media presentation associated with a particular vector $\boldsymbol{a}$ is denoted $D(\boldsymbol{a})$ and can be computed as

$$D(\boldsymbol{a}) = \sum_{i=1}^{L} \Delta d_i(\mathcal{A}_i(\boldsymbol{a})) \qquad (1)$$

where the notation $\mathcal{A}_i(\boldsymbol{a})$ simply signifies the fact that the choice of a subset from $\mathcal{A}_i$ that will be used to reconstruct data unit i depends on the selection vector $\boldsymbol{a}$.

Similarly, the associated data rate of the source $R(\boldsymbol{a})$ as a function of the selection vector can be computed as $R(\boldsymbol{a}) = \sum_{i=1}^{L} R_i a_i$. Finally, as described earlier we are interested in finding the optimal selection vector $\boldsymbol{a}^*$ that minimizes the resulting reconstruction error and for which the data rate of the source does not exceed the available resource as given by R^*, i.e.,

$$\boldsymbol{a}^* = \arg\min D(\boldsymbol{a}), \text{ s.t. } R(\boldsymbol{a}) \leq R^* \qquad (2)$$

Using the method of Lagrange multipliers the solution to the constrained optimization problem from (2) can be replaced with an equivalent convex hull approximation that is obtained as a solution of the unconstrained optimization problem given as

$$\boldsymbol{a}^* = \arg\min D(\boldsymbol{a}) + \lambda R(\boldsymbol{a}), \qquad (3)$$

where $\lambda > 0$ is a Lagrange multiplier. Adjusting λ according to the rate constraint R^* is usually done in an iterative fashion using fast convex search techniques such as the bisection search technique.

Now, solving for the optimal vector $\boldsymbol{a}^*$ as given in (3) can be very difficult, due to the interdependencies between the data units and their influence on the reconstruction distortion, especially for media presentations that contain a large number of data units. Therefore, we employ an iterative gradient descent procedure that minimizes the Lagrangian $J(\boldsymbol{a}) = D(\boldsymbol{a}) + \lambda R(\boldsymbol{a})$ one component at a time, until convergence. Specifically, let $\boldsymbol{a}^{(0)}$ be any initial selection vector and let $\boldsymbol{a}^{(n)} = (a_1^{(n)}, \ldots, a_L^{(n)})$ be determined for $n = 1, 2, \ldots$, as follows. We select one component $l_n \in \{1, \ldots, L\}$ to optimize at step n in a round-robin fashion, i.e., $l_n = (n \bmod L)$. Then for $l \neq l_n$, we let $a_l^{(n)} = a_l^{(n-1)}$, while for $l = l_n$, we compute

$$\begin{aligned} a_l^{(n)} &= \arg\min_{a_l} J(a_1^{(n)}, \ldots, a_{l-1}^{(n)}, a_l, a_{l+1}^{(n)}, \ldots, a_L^{(n)}) \\ &= \arg\min_{a_l} S_l^{(n)}(1 - a_l) + \lambda R_l a_l, \end{aligned} \qquad (4)$$

where the second equality follows by grouping terms that do not depend on a_l and where $S_l^{(n)}$ can be regarded as the *sensitivity* to losing data unit l, i.e., the amount by which the reconstruction distortion will increase at decoding if data unit l is discarded, given the current selection choices for the other data units. Note that the algorithm is guaranteed to converge because the Lagrangian $J(\boldsymbol{a}^{(n)})$ is non-increasing with n and is bounded from below with zero, since it is non-negative.

The minimization (4) is now simple, since each data unit l can be considered in isolation. Indeed, the optimal selection decision $a_l^* \in \{0, 1\}$ for data unit l minimizes $\lambda_l(1-a_l)+\lambda a_l$, where $\lambda_l = S_l^{(n)}/R_l$ can be considered as the distortion per bit utility associated with data unit l. Finding a_l^* is then done by comparing λ_l and λ: for $\lambda_l > \lambda$, $a_l = 1$, while for $\lambda_l \leq \lambda$, a_l should be set to zero.

Computing the sensitivity $S_l^{(n)}$ can exhibit various degrees of complexity depending on the employed distortion model. This in turn is determined by the assumptions that are made about the set $\mathcal{A}_l$ and the reconstruction distortion function Δd_l for a data unit l that were introduced in Section II. In the section with the experiments, we examine the performance of the pruning algorithm based on two different distortion models that have very different implementation complexities. The first model is additive, i.e., it assumes additivity of the distortions associated with the events of discarding individual data units [12, 13]. This implies that $\mathcal{A}_l$ contains only the data unit l itself and Δd_l accounts for the total increase in reconstruction error for the media presentation associated exclusively with l. Computing $S_l^{(n)}$ based on this model is quite simple due to its low complexity. The second distortion model that we will consider is more complex as it accounts for the influence of the packet interdependencies on the reconstruction distortion associated with a data unit [4]. Computing the sensitivity $S_l^{(n)}$ in this case can be much more involved. Finally, it should be noted that iterative descent algorithms analogous to (3)-(4) have been considered in [3, 4] in the context of packet scheduling.

IV. Rate-Distortion Optimized ARQ

In this section, we explain the design of a rate-distortion optimized ARQ scheme based on the pruning algorithm from the previous section. Let r_i, for $i = 1, \ldots, N$ be a series of monotonically decreasing transmission rates at which we would like to be able to stream a media presentation. Then, we employ the optimization algorithm from Section III to adjust, i.e., to match the data rate of the presentation to each of the rates r_i. In particular, we set $R^* = r_i$ in (2) and perform the optimization from Section III in order to find the set of data units $\mathcal{DU}_i$ from the presentation that we need to keep so that the resulting data rate of $\mathcal{DU}_i$ does not exceed r_i, while its associated reconstruction error is minimized.

The sets $\mathcal{DU}_i$ obtained in this manner are typically embedded, i.e., $\mathcal{DU}_i \subset \mathcal{DU}_{i-1}$, for $i = 2, \ldots, N$. This provides an additional benefit to the ARQ technique as it can be used for dynamic rate adaptation while streaming. Specifically, consider that during streaming the available transmission rate has suddenly changed from r_i to r_j, for $r_i > r_j$. A sender based on the ARQ scheme has a window $\mathcal{W}$ of data units to transmit at present. These data units are from $\mathcal{DU}_i$ and the sender cannot transmit them all due to the rate reduction. Rather than dropping all of them or making a random selection from $\mathcal{W}$ to account for the reduced rate, the sender can simply find which data units from $\mathcal{W}$ belong to $\mathcal{DU}_j$, i.e., it can determine $\mathcal{W} \cap \mathcal{DU}_j$. Then, only these data units are sent and the rest of them from $\mathcal{W}$ are omitted. Finally, the sender continues to stream from $\mathcal{DU}_j$ having achieved the smoothest possible transition from r_i to r_j.

V. Experimental Results

This section examines the performances of the optimization framework for source pruning from Section III and the enhanced ARQ technique for video streaming from Section IV. The packetized video content used in the experiments are the standard test video sequences Foreman and Mother & Daughter in QCIF format encoded at 10 fps using JM 2.1 of the JVT/H.264 video compression standard. Each sequence is coded with a constant quantization level at an average luminance (Y) PSNR of about 36 dB and a Group of Pictures (GOP) size of 20 frames, where each GOP consists of an I frame followed by 19 consecutive P frames.

A. Pruning Experiments

Here, we examine the performance of the pruning framework as a function of the employed distortion model, as explained in Section III. Performance is measured in terms of the average Y-PSNR (dB) of the reconstructed video sequence as a function of the data rate at which the encoded video is pruned. The two distortion models that were described in Section III are the additive model from [12, 13] denoted henceforth *Model 2*, and the model from [4], denoted *Model 1*.

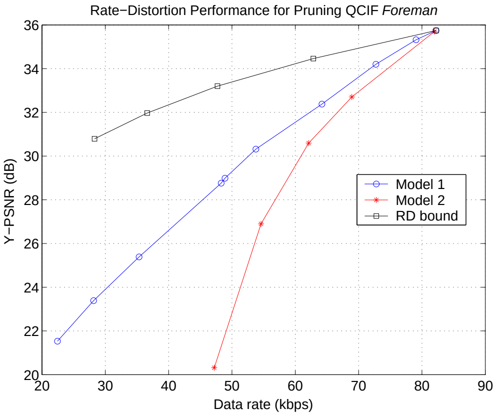

Fig. 1. Pruning performance for *Foreman*.

In Figure 1, we show the performance of the optimization framework for pruning Foreman based on these two distortion models. It can be seen that *Model 1* outperforms *Model 2* for pruning Foreman which is expected since *Model 1* is far more sophisticated and therefore more accurate. Furthermore, the performance difference between the two models increases as the available data rate that is used to prune the source decreases. This is also expected since the number of pruned data units increases as the data rate is increased which in turn affects inversely the accuracy of *Model 2*. Specifically, the underlying assumption of this model is that the effects of omitting individual data units are independent relative to the reconstruction distortion for the media presentation as explained in Section III. This certainly holds less true when more data units needs to be discarded since their locations cannot be placed sufficiently far apart in order to preserve the independence assumption, as recognized, e.g., in [14].

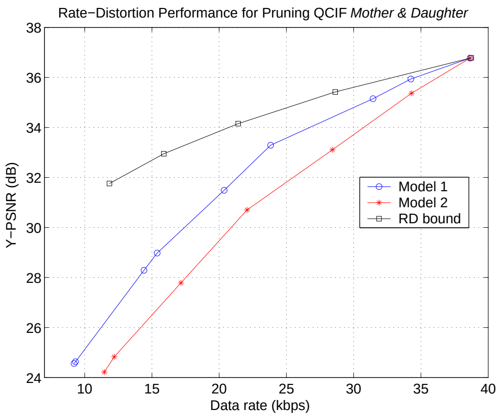

Fig. 2. Pruning performance for *Mother and Daughter*.

Next, in Figure 2 we examine the pruning performance of the two models for the sequence Mother & Daughter. It can be seen that again *Model 1* outperforms *Model 2* over the whole range of available data rates under consideration. However, it should be noted that the performance difference between the two models is not so significant here as in the case of Foreman. This is due to nature of the Mother & Daughter sequence which exhibits less motion and scene complexity relative to Foreman. Hence, error concealment can be performed more successfully on missing data units of Mother & Daughter, which makes the selection of data units to be discarded not so critical in terms of resulting reconstruction quality of the video sequence. In the experiments in the next section, we employ *Model 1* for the design of the proposed ARQ scheme from Section IV.

For comparison purposes, in Figure 1 and 2, we also show the R-D performance for encoding the original video content at different rates, denoted as *RDbound*. It can be seen that the loss in performance of pruning relative to re-encoding increases as the available data rate decreases. Furthermore, we can also see that this loss is content dependent and is due to the differences in terms of content complexity, as explained above. In summary, the comparison with *RDbound* suggests that re-encoding should always be preferred relative to pruning if such an option is feasible.

B. Streaming Experiments

This section investigates the end-to-end distortion-rate performance for streaming packetized video content using different algorithms. Three closed-loop streaming systems are employed in the experiments. *Conv. RaDiO* is a streaming system that employs a conventional RaDiO technique for packet scheduling such as the one from [4]. *Enh. ARQ* is the enhanced ARQ technique proposed in this paper in Section IV. Finally, the streaming system labelled *Conv. ARQ* is a conventional streaming system which does not take into account the importance of individual packets in terms of reconstruction distortion. In particular, when making transmission decisions, *Conv. ARQ* does not distinguish between two packets that contain two different P frames, except for the size of the packets. Therefore, *Conv. ARQ* randomly chooses between two P-frame packets of the same size, for example, when it needs to reduce the number of transmitted packets.

The forward and the backward channel on the network path between the server and the client are modeled as follows. Packets transmitted on these channels are dropped at random, with a drop rate $\epsilon_F = \epsilon_B = \epsilon = 10$ %. Those packets that are not dropped receive a random delay, where for the forward and backward delay densities p_F

and p_B we use identical shifted Gamma distributions with parameters (n, α) and right shift κ, where $n = 2$ nodes, $1/\alpha = 25$ ms, and $\kappa = 50$ ms for a mean delay of $\kappa + n/\alpha = 100$ ms and standard deviation $\sqrt{n}/\alpha \approx 35$ ms.

The retransmission time-out (RTO) interval for the two ARQ systems is set to $\mu_R + 3\,\sigma_R$. For the *Enh. ARQ* system, a sufficiently large series of rates $r_i, i = 1, .., N$, is chosen to prune the compressed video sequences, as described in Section IV, so that the pruned encodings $\mathcal{DU}_i$ cover a wide range of transmission rates on the forward channel. Finally, streaming performance is measured in terms of the average Y-PSNR (dB) of a reconstructed video sequence at the client as a function of the average transmission rate (kbps) on the forward channel. The play-out delay for the videos is set to 600 ms.

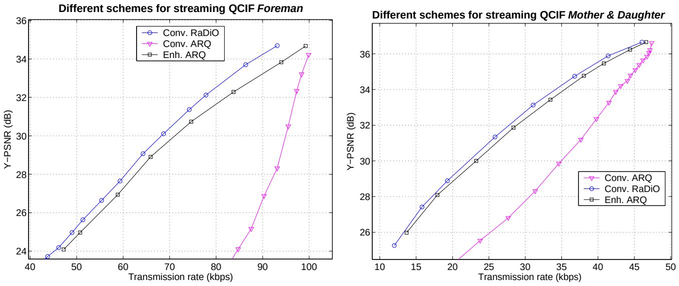

Fig. 3. Streaming performance for Foreman (left) and Moth. & Daug. (right).

In Figure 3 (left), we show the performances of the three systems for streaming Foreman. It can be seen that *Conv. RaDiO* outperforms the other two systems over the whole range of transmission rates. This is expected, as this system optimizes its current and future transmission decisions jointly over a certain time horizon, while no such optimization is present in the two ARQ systems. Furthermore, the performance of *Conv. ARQ* is the worst of the three, which is also expected. As no preferential treatment is given to packets by *Conv. ARQ*, its performance degrades quickly as the available transmission rate is decreased. Finally, what is most important is that *Enh. ARQ* provides a substantial gain over *Conv. ARQ*. For example at 90 kbps, the gain is around 6 dB which is quite significant. At the same time, the performance loss of *Enh. ARQ* relative to *Conv. RaDiO* does not exceed 1 dB over the whole range of rates.

Similar outcome is observed for streaming Mother & Daughter, as shown in Figure 3 (right). *Conv. ARQ* outperforms the other two systems, while *Enh. ARQ* provides an improved performance over *Conv. ARQ*. For example, the gain of *Enh. ARQ* relative to *Conv. ARQ* is 4 dB at 35 kbps. Note that in this case the relative performance differences between the three systems are not so large as those for Foreman. This is due to the low complexity nature of the Mother & Daughter sequence, which makes error concealment perform well on missing packets at the client, as explained in Section V-A. Note also that now, the performances of *Conv. ARQ* and *Enh. ARQ* are quite similar, with their difference not exceeding 0.2-0.3 dB.

This section concludes by briefly describing the computational requirements of the three streaming systems. As discussed in [9], the complexity of *Conv. RaDiO* is many times larger than that of conventional streaming systems such as *Conv. ARQ*. At the same time note that there is no difference in online complexity for each of the two ARQ systems used in our experiments. This is because the preprocessing for *Enh. ARQ* is done off-line, prior to streaming. Therefore, given the experimental settings that we used, it is encouraging to see the significant improvement in performance of *Enh. ARQ* over *Conv. ARQ* for the same online complexity, and the similar performance with *Conv. RaDiO* achieved by *Enh. ARQ* at a substantially smaller computational cost.

VI. Conclusions

We have presented a framework for rate-distortion optimized pruning of a packetized video source. The framework can be used to select which packets of the compressed source representation should be discarded so that the resulting data rate is adjusted accordingly while the resulting reconstruction distortion is minimized. In conjunction with the pruning framework, we design a low-complexity rate-distortion optimized ARQ scheme for video streaming. Our experimental results show that the performance of our framework can greatly vary depending on the accuracy of the distortion model that is employed to describe the effect of packet interdependencies on the reconstruction distortion. Furthermore, the experiments also show that the loss in performance of source pruning relative to re-encoding the original uncompressed source at different data rates increases significantly as the amount of media content that needs to be pruned increases. Finally, via another set of experiments we demonstrated that the enhanced ARQ scheme provides substantial improvement in performance with no increase in online complexity over conventional streaming systems, where no distortion information is taken into account for scheduling the packet transmissions. At the same time, the optimized ARQ technique achieves performance that is similar to that of conventional R-D optimized systems, with only a fraction of their computational complexity.

References

[1] B.-J. Kim, Z. Xiong, , and W. A. Pearlman, “Low bit-rate scalable video coding with 3D set partitioning in hierarchical trees (3-D SPIHT),” *IEEE Trans. Circuits and Systems for Video Technology*, Dec. 2000.

[2] J. Xin, C.-W. Lin, and M.-T. Sun, “Digital video transcoding,” *Proceedings of the IEEE*, Jan. 2005.

[3] P. A. Chou and Z. Miao, “Rate-distortion optimized sender-driven streaming over best-effort networks,” in *Proc. IEEE MMSP*, Cannes, France, Oct. 2001.

[4] J. Chakareski and B. Girod, “Rate-distortion optimized packet scheduling and routing for media streaming with path diversity,” in *Proc. IEEE DCC*, Snowbird, UT, Mar. 2003.

[5] R. Keller, S. Choi, M. Dasen, D. Decasper, G. Fankhauser, and B. Plattner, “An active router architecture for multicast video distribution,” in *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, Mar. 2000.

[6] I. Bouazizi, “Size-distortion optimized proxy caching for robust transmission of MPEG-4 video,” in *Proc. MIPS*, Napoli, Italy, Nov. 2003.

[7] Y. Bai and M. Ito, “Network-level loss control schemes for streaming video,” in *Proc. IEEE ICME*, Taipei, Taiwan, June 2004.

[8] W. Tu, W. Kellerer, and E. Steinbach, “Rate-distortion optimized video frame dropping on active network nodes,” in *Proc. Int’l Packet Video Workshop*, Irvine, CA, USA, Dec. 2004.

[9] J. Chakareski, J. Apostolopoulos, and B. Girod, “Low-complexity rate-distortion optimized video streaming,” in *Proc. IEEE ICIP*, Singapore, Singapore, Oct. 2004.

[10] A. Sehgal, A. Jagmohan, O. Verscheure, and P. Frossard, “Fast distortion-buffer optimized streaming of multimedia,” in *Proc. IEEE ICIP*, Genova, Italy, Sept. 2005, to appear.

[11] J. Chakareski, P. Chou, and B. Girod, “Rate-distortion optimized streaming from the edge of the network,” in *Proc. IEEE MMSP*, St. Thomas, US Virgin Islands, Dec. 2002.

[12] E. Masala and J. de Martin, “Analysis-by-synthesis distortion computation for rate-distortion optimized multimedia streaming,” in *Proc. IEEE ICME*, Baltimore, MD, July 2003.

[13] J. Chakareski, J. Apostolopoulos, W.-T. Tan, S. Wee, and B. Girod, “Distortion chains for predicting the video distortion for general packet loss patterns,” in *Proc. IEEE ICASSP*, Montreal, Canada, May 2004.

[14] J. Apostolopoulos, “Reliable video communication over lossy packet networks using multiple state encoding and path diversity,” in *Proc. SPIE VCIP*, San Jose, CA, Jan. 2001.