RATE-DISTORTION OPTIMIZED STREAMING OF VIDEO WITH MULTIPLE INDEPENDENT ENCODINGS

Mark Kalman and Bernd Girod

Information Systems Laboratory
Stanford University
{mkalman, bgirod}@stanford.edu

ABSTRACT

We extend recent work on rate-distortion optimized streaming of video to include the case when multiple, independently encoded and packetized versions of the video are available for transmission. While there has been much recent work on the rate-distortion optimized streaming of packetized video, previous work has focussed on layered video encodings and encodings that are comprised of a single set of interdependent packets. In this paper we extend the rate-distortion optimizing packet transmission algorithms, a major benefit of which is that they select the optimal source rate from a rate-scalable encoding on a per-transmission basis, to handle the case of multiple independent encodings, the most ubiquitous form of rate-scalable video in use. In our experimental results, we show up to a 3 dB performance improvement for our multiple-encoding optimizing scheduler over a non-optimizing scheduler.

1. INTRODUCTION

While there has been much recent work on the rate-distortion optimized streaming of packetized video, previous work has focussed on layered video encodings and encodings that are comprised of a single set of interdependent packets. The case of rate-distortion optimized streaming of video encoded independently at multiple bit-rates and display qualities has not been previously addressed.

Much of the video content on the Internet, however, is encoded independently at multiple bit-rates and display qualities. In the absence of a widely accepted layered-scalable codec, multiple encodings has become the preferred means of providing rate-scalability. With multiple encodings, content providers can accommodate users connecting to the Internet at varying link speeds, and in addition, systems like Real Networks' Surestream technology (to name one) use multiple independent encodings to respond to Internet packet loss and delay in a TCP-friendly way [1]. Systems using multiple independent encodings respond to changing TCP-friendly rate constraints by dynamically switching to the encoding with the appropriate rate.

With the ubiquity of video encoded in multiple independent streams as motivation, we have extended recent work on R-D optimized video streaming to include the case of multiple independently encoded streams. In this paper we are building upon work that we presented in [2] which in turn was based on the framework for R-D optimized streaming in [3]. We begin in Sec. 2 by reviewing the framework for R-D optimized streaming presented in [3]. In Sec. 3, we show how we extend the framework in order to accommodate multiple independent streams. In Sec. 4, we present performance results for our optimized scheduling formulation.

This work was funded by a generous grant from ST Microelectronics Inc.

2. BACKGROUND

In this section we briefly review the framework for rate-distortion optimized streaming in [3] which serves as our point of departure.

The framework assumes that a compressed media representation has been assembled into packets or data units. Each data unit has a size in bytes and a time deadline by which it must arrive in order to be useful for decoding. Each data unit is also associated with a value for the amount of distortion that will be removed from the decoded video if the unit is available when needed, and each data unit has a set of decoding interdependencies with other data units that are expressed with a single directed acyclic graph.

In the framework, the goal of the optimizing packet transmission scheduler is to choose the best set of these data units (packets) to transmit at successive discrete-in-time transmission opportunities. Because of decoding dependencies among data units, the importance of transmitting a packet at a given transmission opportunity often depends on which packets will be transmitted in the near future. The scheduler therefore bases its transmission decisions on an entire plan governing all the transmissions that will occur in the near future

The plan governing packet transmissions that will occur during a time horizon of discrete transmission opportunities is referred to as a *transmission policy*, $\boldsymbol{\pi}$. Assuming a time horizon of length N, $\boldsymbol{\pi}$ can be represented as a collection of length-N binary vectors π_l, with one such vector for each packetized data unit l under consideration for transmission. In this representation, the N binary elements of a policy vector π_l indicate whether, under the policy, the data unit l will be transmitted or not at each of the next N transmission opportunities, unless an acknowledgement indicates that the packet has been successfully received.

In the framework, at any transmission opportunity the optimal $\boldsymbol{\pi}$ is the one that minimizes the Lagrangian cost function

$$D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi}), \tag{1}$$

where, for a given transmission policy, $D(\boldsymbol{\pi})$ is the expected reconstruction distortion and $R(\boldsymbol{\pi})$ is the expected transmission rate. λ controls the trade-off between rate and distortion.

The framework's expression for $D(\boldsymbol{\pi})$ is written in terms of $\epsilon(\pi_l)$, the data units' error probabilities under transmission policy $\boldsymbol{\pi}$. 'Error' is the event that the data unit does not arrive by its time deadline for decoding. The framework's expression for

0-7803-8554-3/04/$20.00 ©2004 IEEE.

$R(\boldsymbol{\pi})$ is written in terms of the data unit sizes B_l in bytes, and $\rho(\pi_l)$, the expected number of times each unit will be transmitted under a given transmission policy. Delays and losses experienced by packets transmitted over the network are assumed to be random and independent from transmission to transmission. Packet loss is modeled as Bernoulli with some probability, and packets not lost are assumed to be delayed according to a shifted-Γ distribution. Expressions for $\epsilon(\pi_l)$ and $\rho(\pi_l)$ are given in terms of the probability distribution functions, transmission policy and transmission history, and the data units' arrival deadlines.

The scheduler re-optimizes the entire policy $\boldsymbol{\pi}$ at each transmission opportunity to take into account information learned since the previous transmission opportunity. As a method of actually finding the optimal transmission policy, exhaustive search is not generally tractable as noted in [4]. Avoiding an exhaustive search of the entire space of transmission policies is the main contribution of [3]. The authors introduce an iterative descent algorithm that simplifies the search for an optimal $\boldsymbol{\pi}$. The iterative descent algorithm begins with an initial set of transmission policies, and then proceeds to minimize (1) iteratively. At each iteration (1) is minimized with respect to the transmission policy π_l of one data unit while the transmission policies of other data units are held fixed. Data units' policies are optimized in round-robin order until the Lagrangian cost converges to at least a local minimum. Rewritten in terms of the transmission policy of one data unit, (1) becomes

$$J_l(\pi_l) = \epsilon(\pi_l) + \lambda' \rho(\pi_l). \tag{2}$$

where $\lambda' = \frac{\lambda B_l}{S_l}$ incorporates the rate-distortion trade-off operator λ from (1), the data unit size B_l, and S_l, a term that expresses the sensitivity of the overall expected distortion to the error probability $\epsilon(\pi_l)$ of data unit l. The sensitivity term represents the relative importance of a particular data unit.

Another contribution of [3] is its method for modeling the reconstruction distortion of a video depending on what packets are available at the decoder. The distortion model assumes that packet interdependencies can be expressed with a single, directed acyclic graph (DAG) in which the packets form the nodes of the graph and the decoding dependencies are expressed as the directed edges. The algorithm assumes that as each successive node (representing a packetized data unit) on the graph becomes decodable, there is an incremental reduction in the distortion of the decoded media.

3. PROPOSED FORMULATION

In the following subsections we describe how we modify the rate-distortion optimized streaming framework of [3] to take into consideration multiple independent encodings and the effects of error concealment that substitutes missing frames with the most recent, highest quality frame that is decodable. In addition, as in [2] we extend the formulation to consider the distortion value of packets at multiple deadlines.

3.1. Modeling Distortion with Multiple Streams

While elegant, the model of distortion and decoding dependency in [3] does not hold in the case when the scheduler has multiple independent packetized encodings of the same video available. Suppose, for example, that there are two independent encodings of a video and both are available at the decoder. During decoding, the decoder would simply decode the higher quality stream and ignore the other. In this case, each of the independent encodings is worth a certain amount in terms of distortion reduction, but the value when both streams are available is not the sum of the two individual values. In the DAG model, on the other hand, the effect on distortion when a packet becomes available is always additive.

In [2] we presented an alternative paradigm for distortion modeling that *can* be used in the cases of error concealment and multiple independent streams. In this paradigm, a video frame (or portion thereof) is assumed to be dependent on some set $\mathcal{L}$ of packets in order to be decoded and displayed, and that the distortion for that decoded frame may be dependent on exactly which of the packets in $\mathcal{L}$ are available and which are not. By assuming copy error concealment, the complexity of this paradigm can be brought down to $\mathcal{O}(|\mathcal{L}|)$. Below, we apply our distortion modeling paradigm to the case of multiple independent encodings and previous-frame copy error concealment. We derive the expressions for the expected distortion $D(\boldsymbol{\pi})$ and for the sensitivity S_l of the expected distortion to the loss probability of a particular packet l. With these in hand we can then apply the iterative descent algorithm of [3] to optimize packet transmission policies.

In our derivation, we assume that there are Q individually encoded video streams indexed by $q \in \{0, 1, \ldots, Q-1\}$, with $q = 0$ being the lowest quality and and $q = Q-1$ being the highest. Each encoding has a prediction structure I-P-...-P-P with GOPs of length G frames. For simplicity, we assume that each frame of each encoding is placed into one packet. When a frame is due for decoding, the highest-quality frame that is decodable is decoded and displayed. If no encoding of the current frame is decodable (because packets have not arrived in time) the nearest previous frame that can be decoded is shown. If more than one quality of the nearest previous frame is available, the highest quality version is shown.

Let $D_n(\boldsymbol{\pi})$ be the expected distortion of frame n under transmission policy $\boldsymbol{\pi}$. The overall distortion is then $D(\boldsymbol{\pi}) = \sum_{n \in \mathcal{W}} D_n(\boldsymbol{\pi})$, where $\mathcal{W}$ is the set of data units (packets) belonging to frames in the window considered for transmission. Assuming 'copy' error concealment and assuming that the video is independently encoded at Q different quality levels, when it comes time to display frame n, one of $nQ+1$ different images may be shown. For each of these possible display outcomes, there is a particular mean-squared error (MSE) between the display outcome and the pre-encoded image of the frame.

Let $d^n_{q,g,i}$ be the MSE value when frame n is the intended frame for display, but the q-th quality of the i-th frame of the g-th GOP must be shown instead. Let $\epsilon^n_{q,g,i}$ be the probability that the packet containing the i-th frame of the g-th GOP at quality q is not available for decoding at frame n's decoding time (under policy $\boldsymbol{\pi}$, implicitly). As in Sec. 2 we use the idea of a data unit 'error' probability $\epsilon(\pi_l)$, but now instead of absolute indexes l, we index by quality q, GOP g, and position in GOP i, and we associate the error probability with a time deadline that can be specific to frame n.

In order to find the expected distortion of a frame $D_n(\boldsymbol{\pi})$, we need to find the relative probabilities of each of the distortion outcomes $d^n_{q,g,i}$. To help us write an expression for these $\Pr\{d^n_{q,g,i}\}$, we first write some intermediate expressions. Let $A^n_{q,g,i}$ be the probability that for encoding q, all the packets in GOP g leading up to and including i are available for decoding at frame n's display time.

$$A^n_{q,g,i} = \prod_{l=0}^{i} \left(1 - \epsilon^n_{q,g,l}\right)$$

Let $B^n_{q,g,i}$ be the probability that for encodings with higher quality than q, not all of the packets needed to display the i-th frame of the g-th GOP are available at frame n's display time.

$$B^n_{q,g,i} = \prod_{j=q+1}^{Q-1} \left[1 - \prod_{l=0}^{i} \left(1 - \epsilon^n_{j,g,l}\right) \right]$$

In the case when the i-th frame of the g-th GOP precedes frame n, then the image (q, g, i) will not be shown if any lower-quality stream is decodable through $(i+1)$-th frame. Let $C^n_{q,g,i}$ be the probability that no lower-quality stream is decodable beyond position i at frame n's decoding deadline.

$$C^n_{q,g,i} = \prod_{j=0}^{q-1} \left[1 - \prod_{l=0}^{i+1} \left(1 - \epsilon^n_{j,g,l}\right) \right]$$

With A, B, and C we can write the probability that the display distortion for frame n is $d^n_{q,g,i}$ as

$$\Pr\{d^n_{q,g,i}\} = \begin{cases} A^n_{q,g,i} \cdot B^n_{q,g,i} \cdot C^n_{q,g,i} \cdot \epsilon^n_{q,g,i+1} & \text{if } g = \lfloor \frac{n}{G} \rfloor \\ & i \neq n \bmod G \\ A^n_{q,g,i} \cdot B^n_{q,g,i} & \text{if } g = \lfloor \frac{n}{G} \rfloor \\ & i = n \bmod G \\ \left(\prod_{m=g+1}^{\lfloor \frac{n}{G} \rfloor} \prod_{j=0}^{Q-1} \epsilon^n_{j,m,0} \right) \cdot & \\ \quad A^n_{q,g,i} \cdot B^n_{q,g,i} \cdot C^n_{q,g,i} \cdot \epsilon^n_{q,g,i+1} & \text{if } g < \lfloor \frac{n}{G} \rfloor \\ & i < G-1 \\ \left(\prod_{m=g+1}^{\lfloor \frac{n}{G} \rfloor} \prod_{j=0}^{Q-1} \epsilon^n_{j,m,0} \right) \cdot & \\ \quad A^n_{q,g,i} \cdot B^n_{q,g,i} & \text{if } g < \lfloor \frac{n}{G} \rfloor \\ & i = G-1 \end{cases} \tag{3}$$

In (3) the first case is the case where the frame (q, g, i) is in the same GOP as frame n (hence $g = \lfloor \frac{n}{G} \rfloor$), but is earlier in the GOP than n. The second case is when frame (q, g, i) is frame n itself at quality q. The third case is when the displayed frame (q, g, i) is in an earlier GOP than n and is not the last frame in that GOP. The product terms $\prod_{m=g+1}^{\lfloor \frac{n}{G} \rfloor} \prod_{j=0}^{Q-1}$ for this case are the probabilities that none of the I-frames for later GOPs are decodable. The final case is the case when (q, g, i) is in an earlier GOP than n but is the last frame in a GOP.

Using (3) we can write the expected distortion for frame n under policy $\boldsymbol{\pi}$ as

$$\begin{aligned} D_n = & \sum_{q=0}^{Q-1} \sum_{i=0}^{n \bmod G} \Pr\left\{ d^n_{q,\lfloor \frac{n}{G} \rfloor,i} \right\} \cdot d^n_{q,\lfloor \frac{n}{G} \rfloor,i} \\ & + \sum_{q=0}^{Q-1} \sum_{g=0}^{\lfloor \frac{n}{G} \rfloor - 1} \sum_{i=0}^{G-1} \Pr\left\{ d^n_{q,g,i} \right\} d^n_{q,g,i} \end{aligned} \tag{4}$$

The sensitivity of the distortion of frame n to the availability of the packet indexed by (q, g, i) can be found by rewriting (4) in terms of $\epsilon^n_{q,g,i}$. For any packet which we index by (q, g, i), (4) can be rewritten as

$$D_n = c + \epsilon^n_{q,g,i} S^n_{q,g,i} \tag{5}$$

The terms that contain a factor of $\epsilon^n_{q,g,i}$ in (4) therefore form the sensitivity for that data unit. All other terms can be lumped into c, which does not affect the sensitivity. We don't include the exact expression for $S^n_{q,g,i}$ here because of space constraints, but the expression is simply found (if tediously) from (4).

3.2. Extension for multiple deadlines

In this paper as in [2] we consider the value of a packet at multiple arrival deadlines. We assume that the decoder uses Accelerated Retroactive Decoding (ARD) as presented in [2]. With this scheme, if packets arrive after they are first needed, the decoder can go back and quickly decode the dependency chain when the late packets do arrive in order to "catch-up" to the current playout position. Thus, late arriving packets still have value. The value of a packet depends on which frame's decoding deadline it meets.

In our optimization, we use the same descent algorithm as in [3], but in addition to using new expressions for distortion and sensitivity, we also use a new cost function with respect to the transmission policy of a data unit,

$$J_l(\pi_l) = \rho(\pi_l) + \sum_{n \in \mathcal{W}} \nu_{t_n} \epsilon(\pi_l, t_n). \tag{6}$$

This expression takes into consideration that data unit l is needed by multiple frames n for decoding. In the expression, there is a distinct cost associated with the data unit's error probability at each of the frames' deadlines t_n. The quantity $\epsilon(\pi_l, t_n)$ is the probability that data unit l does not arrive by time deadline t_n. These are the same as the $\epsilon^n_{q,g,i}$ in Sec. 3.1 but now indexed in terms of l instead of (q, g, i). The quantity ν_{t_n} is given by $\nu_{t_n} = \frac{S_{l,t_n}}{\lambda' B_l}$, analogous to the reciprocal of λ' in (2). Thus for each data unit an array of sensitivities S_{l,t_n} are computed, one for frame (with associated deadline) that requires the data unit for decoding.

4. EXPERIMENTAL RESULTS

In this section we show simulation results that compare the rate-distortion performance of our multi-stream optimizing transmission scheduler formulation with the performance of a heuristic, non-optimizing scheduler. We also compare the performance of the multi-stream scheduler with an optimizing scheduler that has available to it only one of the pre-encoded streams.

4.1. Simulations

Our results are for 12 seconds of the *Foreman* video sequence encoded in five independent streams using an H.264 reference encoder. The frame-rate is 10 fps, the GOP length is 10 frames, and the prediction structure is I-P-. . .-P-P. The rates and distortions of the five encodings are (rate in kbps, PSNR in dB): (19.6, 27.3), (35.0, 30.5), (55.9, 33.0), (104.7, 36.3), (167.9, 39.0).

Each frame of each encoded sequence is placed into an individual packet. The packet network is simulated as in [3], with delay and loss events statistically independent from transmission to transmission. Packets are randomly delayed both in the forward and reverse directions according to a shifted-Γ distribution with shift $\kappa = 10$ ms, mean $\mu = 50$ ms, and standard deviation 23 ms. The packet loss probability in both directions is 0.20. The client sends an acknowledgment packets for each media packet it receives.

We assume that our transmission scheduler has perfect knowledge of the channel statistics. The scheduler's discrete transmission opportunities occur regularly every 50 ms. Playout begins at the client 400 ms after the first transmission. The transmission window is a fixed interval such that a frame's data units become

eligible for transmission 400 ms before the frame's playout deadline. As described in Sec. 3.2 and [2], the decoder uses Accelerated Retroactive Decoding (ARD), and the optimizing scheduler considers the value of a packet at multiple deadlines. The rate-versus-PSNR results that we show for the optimizing schedulers are for λ (as in (1)) swept over a range of values. 10 seeds were run for each λ. Each data point shown in our figures shows the averaged rate and PSNR outcomes for the 10 seeds run at a particular λ.

4.2. Heuristic Scheduler for Comparison

The non-optimizing heuristic scheme whose performance we include as a basis for comparison is an ARQ scheme that uses a prioritized transmission queue. Data units are appended to the queue when they enter the transmission window. When a transmitted frame is not ACKed by the 90% point of the round-trip time cdf, it is again appended to the transmission queue. Packets for retransmission are given priority over packets being transmitted for the first time. To achieve a fixed transmission rate R_h, each time a data unit l is transmitted another data unit is not transmitted for B_l/R_h seconds. Packets are no longer considered for transmission one mean forward-trip time before the last frame in the packet's GOP is due for playout. For each rate, the encoding (of the five available) that is observed to give the best performance at that rate, is chosen for transmission for the entire simulation.

Fig. 1 compares the performance of the optimizing and non-optimizing packet schedulers, for the case when multiple independent streams are available for transmission. We see that the optimizing scheduler outperforms the heuristic scheduler by up to 3 dB.

Fig. 2 compares the performance of our multiple independent encoding optimizing scheduler with the optimizing scheduler from [2] that has access to only one encoding at a time. There is a separate R-D curve for the case when the scheduler has available encoding 0, encoding 1, and so on. We see the advantage of a scheduler that can select appropriately from multiple independent encodings at different source rates: the multiple-encoding scheduler outperforms any one of the single-encoding schedulers over most of the range of rates. We do see though, that at the source rate best suited for an individual encoding, there is a penalty of up to 0.7 dB for using the multiple-encoding scheduler over a single-encoding scheduler. We attribute this to the fact that the descent algorithm is only guaranteed to find a local minimum. It is sensitive to the initial policies used and the order in which policies are optimized. For instance, we notice better performance from the multi-encoding scheduler when the algorithm updates policies for the packets in order of highest quality stream to lowest.

5. CONCLUSION

In this paper we have extended recent work on rate-distortion optimized streaming of video to include the case when multiple, independently encoded and packetized versions of the video are available for transmission. In our simulation results we have shown that our scheduling algorithm outperforms a non-optimizing packet transmission scheduler by up to 3 dB. We have also shown that our multiple-encoding optimizing scheduler outperforms an optimizing scheduler with only one encoding available at most bitrates, but at the source rate best suited for the single encoding, we have observed a penalty, not exceeding 0.7 dB, for the multiple-encoding scheduler compared to the single-encoding scheduler.

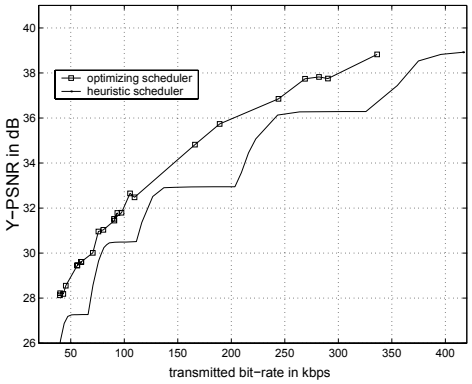

Fig. 1. Rate-Distortion performance of the optimizing and non-optimizing transmission schedulers for the case of multiple independent encodings of *Foreman*.

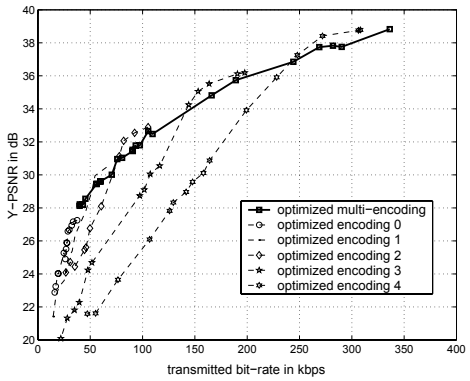

Fig. 2. Rate-Distortion performance of the multi-encoding optimizing scheduler compared to the performance of single-encoding optimizing schedulers each transmitting one of the multiple independent encodings.

We attribute the performance gap to local minima in the descent algorithm.

6. REFERENCES

[1] G.J. Conklin, G.S. Greenbaum, K.O. Lillevold, A.F. Lippman, and Y.A. Reznik, "Video coding for streaming media delivery on the internet," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 269–281, March 2001.

[2] Mark Kalman, Prashant Ramanathan, and Bernd Girod, "Rate distortion optimized streaming with multiple deadlines," in *IEEE International Conference on Image Processing*, Barcelona, Spain, September 2003.

[3] Philip A. Chou and Zhourong Miao, "Rate-distortion optimized streaming of packetized media," Tech. Rep. MSR-TR-2001-35, Microsoft Research, February 2001, (also submitted to IEEE Transactions on Multimedia).

[4] Matthew Podolsky, Steven McCanne, and Martin Vetterli, "Soft ARQ for layered streaming media," Tech. Rep. UCB/CSD-98-1024, University of California, Computer Science Department, Berkeley, CA, November 1998.