R-D HINT TRACKS FOR LOW-COMPLEXITY R-D OPTIMIZED VIDEO STREAMING

Jacob Chakareski[*†], *John Apostolopoulos*[†], *Susie Wee*[†], *Wai-tian Tan*[†] *and Bernd Girod*[*]

[†]Streaming Media Systems Group
Hewlett-Packard Labs, Palo Alto, CA 94304

[*]Information Systems Laboratory
Stanford University, Stanford, CA 94305

ABSTRACT

This paper presents the concept of Rate-Distortion Hint Track (RDHT), and evaluates two specific implementations of streaming systems that employ RDHT. Characteristics of a compressed media source that are often difficult to compute in realtime but crucial to general online optimized streaming algorithms are precomputed and stored in a RDHT. In such a way, low-complexity streaming can be realized for systems that adapt to variations in transport conditions such as bandwidth or packet loss. An RDHT-based streaming system has three components: (1) an R-D Hint Track, (2) an algorithm for using the RDHT to predict the distortion for different packet schedules, and (3) a method for determining the best packet schedule. Two RDHT-based systems are presented which perform R-D optimized scheduling with dramatically reduced complexity as compared to conventional on-line R-D optimized streaming algorithms. Experimental results demonstrate that for the difficult case of R-D optimized scheduling of non-scalably coded video (H.264, I-frame followed by all P-frames), the proposed systems provide 7-12 dB gain when adapting to a bandwidth constraint and 2-4 dB gain when adapting to random packet loss. Furthermore, the proposed RDHT-based systems achieve this R-D optimized performance with a complexity comparable to a conventional non-RD-optimized streaming system.

1. INTRODUCTION

Video streaming over bandwidth-constrained and lossy packet networks has been a practically important and challenging problem for a number of years. Video streaming typically involves pre-encoded and stored compressed media, and the pre-encoded content makes it harder to adapt to the available bandwidth and losses as compared to the case where real-time encoding is performed. Video transcoding can be performed in this situation, however this requires significant complexity and computation. Scalable coding techniques have been developed to solve these problems, where the compressed data is prioritized and the prioritization provides a natural method for selecting which portion of compressed data to deliver while meeting the transmission constraints. In addition, convention MPEG coding with I, P, and B frames also lends itself to a natural method of prioritizing the delivery. Recently Rate-Distortion Optimized (RaDiO) packet scheduling has been proposed and shown significant benefits[1–3], however this approach is also very compute intensive.

This paper proposes a method for designing and operating media streaming systems that can perform optimized streaming while being low complexity. Specifically, during encoding of a media object (e.g. video sequence), a Rate-Distortion Hint Track (RDHT) is generated that contains side information that are often difficult to compute on a realtime basis, but are useful to a general optimized streaming algorithm. The RDHT is a "track" because it is stored in the same file as the compressed media data but can be easily demultiplexed. It is a "hint track" in the sense that it provides "hints" for performing high quality streaming. Example information in the R-D hint track include the importance of each packet in an R-D sense. The computation of hints at encoding time relieves the burden of optimized streaming servers, which can simply read the hints from RDHT rather than estimating them on a realtime basis.

The term "hint track" has been used in the popular MPEG-4 File Format (MP4), and related streaming systems. An MP4 hint track contains information about media type, packet framing and timing information. With an MP4 hint track, media streaming is greatly simplified, both in terms of complexity and computation. This is because the streaming server no longer needs to (1) understand the compressed media syntax, and (2) analyze the media data in realtime for packet framing and timing information. In a similar spirit, RDHT enables low-complexity in optimized streaming by relieving the burden of analyzing R-D characteristics of media in realtime.

Related work on low-complexity and R-D optimized streaming is [4], where R-D information is placed in each packet header, thereby enabling efficient R-D optimized streaming and adaptation at the sender, or at a mid-network node or proxy, for scalable media content. In addition, [5] proposed a framework for scalable streaming media delivery, with similar attributes to the on-line optimization algorithms of, e.g. [2], but with a fast greedy search algorithm (and computationally simpler distortion metric) for determining the schedule with significantly lower complexity.

For scalably coded content, and MPEG I,P,B frames, the natural prioritization of the data simplifies the scheduling process. However, non-scalable or non-prioritized video does not suggest a natural scheduling method. In this paper we focus on this challenging problem of optimally streaming non-scalable video, specifically, when the coded video consists of all P-frames except for the initial I-frame.

This paper continues by providing an overview of RDHT-based video streaming in Section 2, and example RDHT systems are proposed in Section 3. Experimental results are presented in Section 4. Finally, concluding remarks are provided in Section 5.

2. R-D HINT-TRACK BASED VIDEO STREAMING

The central issue of optimized streaming is to determine the best packet schedule to maximize the reconstructed quality at the receiver, subject to transmission contraints such as the available bandwidth or packet loss. A system that employs RDHT achieves this

This work was performed when Jacob Chakareski was a summer researcher at HP Labs, Palo Alto.

goal using the following three components:

1. Obtaining RDHT information
2. Method for using RDHT information to predict distortion for different packet schedules
3. Method for determining the best packet schedule

There are a number of tradeoffs in the design of RDHT. To provide high performance requires an "informative" RDHT, accurate distortion modeling of different feasible packet schedules, and comprehensive search for the best schedule. On the other hand, it is desirable to use relatively little storage for RDHT, and relatively little computation for predicting the distortion and searching for the best packet schedule.

3. R-D HINT TRACK DESIGN AND USE

In this Section, we propose two reasonable instantiations of RDHT which are used for evaluation purposes in Section 4. The storage cost of each RDHT, and the computation cost of associated distortion model and packet schedule search algorithms are discussed. We assume that each video frame corresponds to a transport packet for clarity purposes.

Two canonical problems are examined using RDHT-based streaming: (1) bandwidth adaptation, and (2) packet loss adaptation. Each one of these problems includes important subproblems. For example, in the context of adapting to available bandwidth, if the bandwidth constraint is measured in number of packets (ignoring packet size) then the problem is simpler than if the bandwidth constraint is measured in bits, in which case there may be many different subsets of different numbers of packets that must be examined. To adapt to the available bandwidth we must solve the problem of what is the best subset of packets to drop to meet the packet-rate or bit-rate constraint. To adapt to packet loss we must solve the problem of what is the best schedule for transmitting new packets and retransmiting previous lost packets to meet the bandwidth constraint.

3.1. A Linear Size RDHT Using DC^0

For the first RDHT, we simply store the distortion in MSE caused by isolated frame loss, assuming no other losses. For a video sequence with L frames, there are L possible isolated losses, resulting in a linear storage cost of L numbers. Figure 1 illustrates the distortion $D(k)$ caused by losing frame k,

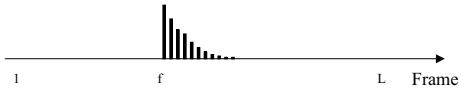

Fig. 1. Loss of single frame k induces distortion in later frames. $D(k)$ is the total distortion summed over all affected frames.

To model the distortion caused by an arbitrary loss pattern given only $D(1)$ to $D(L)$, we employ the zero-th order distortion chain model [6], DC^0, where zero-th order describes that we assume no memory and that the distortion that results for each loss packet is independent of any prior lost packets. This model is accurate when losses are spaced far apart, e.g., when loss rate is low. When N frames $\boldsymbol{k} = (k_1, k_2, \ldots k_N)$ are lost, the predicted total distortion is simply given by:

$$\widetilde{D}(\boldsymbol{k}) = \sum_{i=1}^{N} D(k_i) \tag{1}$$

As mentioned earlier, we need to find the best transmission schedule for the packets of a video stream subject to a transmission bandwidth constraint. This problem can be formalized as follows. Let $\mathcal{W}$ be a window of packets considered for transmission and let R^* be the bandwidth constraint, measured either in bits or number of packets. We need to decide on the subset of packets $\boldsymbol{k} \in \mathcal{W}$ that should not be transmitted in order to satisfy the bandwidth constraint. Let $R(\mathcal{W} \setminus \boldsymbol{k})$ be the rate associated with the packets from $\mathcal{W}$ that will be transmitted, where "\" denotes the operator "set difference". Thus, we are interested in finding the subset $\boldsymbol{k}$ such that the total distortion due to dropping $\boldsymbol{k}$ is minimized, while meeting the bandwidth constraint, i.e.,

$$\boldsymbol{k}^* = \underset{\boldsymbol{k} \in \mathcal{W} \,:\, R(\mathcal{W} \setminus \boldsymbol{k}) \le R^*}{\arg\min} \widetilde{D}(\boldsymbol{k}) \tag{2}$$

Now, consider first solving (2) in the case when the transmission bandwidth R^* is expressed in number of packets. Assume that $R^* = k$, i.e., we need to drop k packets from $\mathcal{W}$. Then $\boldsymbol{k}^*$ is easily found by sorting the distortions for each packet in increasing order, and selecting the first k packets (those with the k smallest associated distortions). In addition, if the problem changes to determine the best $k+1$ packets to drop, the solution then directly builds on the prior solution. Specifically, the selection of the best subset of k packets to drop is contained in the best subset of $k+1$ packets to drop. In contrast, an approach that does not provide this property would have to perform a completely new search for every k. The optimal schedule can therefore be obtained with very little computation.

Next, consider the alternative case when R^* is measured in bits. The integer programming required to obtain the exact optimal solution is difficult to compute. A practical approximation that we employ is to drop packets that individually cause the least distortion per bit. Specifically, a packet $j \in \mathcal{W}$ is associated with a utility in terms of distortion per bit of $\lambda_j = D(j)/R(j)$. We obtain, $\boldsymbol{k}^*$ by sorting the packets in $\mathcal{W}$ in decreasing λ, and then transmit as many packets as possible starting from the highest utility (distortion-per-bit) packet. In this manner, once again we have an embedded search strategy with the associated low complexity benefits. While this search scheme is in general suboptimal, we believe it is reasonable under practical settings.

3.2. A (slightly larger) Linear Size RDHT Using DC^1

For the second RDHT, we store the distortion of all possible isolated losses, $D(k)$, as well as the distortion in MSE associated with all double losses, as illustrated in Figure 2. If $D(k_1, k_2)$

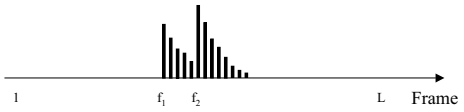

Fig. 2. $D(k_1, k_2)$ is total distortion summed over all frames caused by losing frames k_1 and k_2.

was stored for every possible pair $\{k_1, k_2\}$, then the total storage cost would be quadratic in L since there are L isolated losses and $L(L-1)/2$ distinct $D(k_1, k_2)$. However, since the distortion coupling between dropped packets decreases as the distance between the packets increases, one practical simplication is to assume $D(k_1, k_2) = D(k_1) + D(k_2)$ for $|k_1 - k_2| > M$, where M depends on the compression. For example, for a 15 frame GOP, M is at most the number of packets in the GOP. In the experiments in Section 4, M is 36 which is approximately the intra refresh period. This approximation reduces the required storage and computation for the RDHT from L^2 to LM.

The distortion for an arbitrary loss pattern is estimated using a first-order distortion chain, DC^1, where the distortion for a lost packet now depends on the last lost packet (memory of one). Specifically, for $k_1 < k_2 < \ldots < k_N$, where $N > 2$, we have,

$$\widetilde{D}(\boldsymbol{k}) = D(k_1, k_2) + \sum_{i=2}^{N-1} \{D(k_i, k_{i+1}) - D(k_i)\} \qquad (3)$$

Further results on the accuracy of first-order distortion chains are given in [6].

Searching for the optimal packet schedule (solving (2) exactly) using the above distortion estimate is expensive computationally due to the interdependencies between lost packets. Therefore, we employ an iterative descent algorithm in which we minimize the objective function $\widetilde{D}(\boldsymbol{k})$ one variable at a time while keeping the other variables constant, until convergence. In particular, consider first the case when R^* is expressed in number of packets and assume that $R^* = m$. Then, at iteration n, for $n = 1, 2, \ldots$, we compute the individual entries of the optimal drop pattern $\boldsymbol{k} = (k_1, \ldots, k_m)$ using

$$k_j^{(n)} = \arg\min_{k_j \in \mathcal{W}_j^{(n)}} \widetilde{D}(\boldsymbol{k}), \quad \text{for } j = 1, \ldots, m,$$

where the sets $\mathcal{W}_j^{(n)} = \{k_{j-1}^{(n)} + 1, \ldots, k_{j+1}^{(n-1)} - 1\}$. In other words, starting with a reasonable initial solution for $\boldsymbol{k}$, at each iteration we perturb the subset of selected packets $\boldsymbol{k}$ in order to find a subset that produces reduced distortion. At each iteration a subset with less or equal distortion is found, therefore the algorithm is guaranteed to converge, though not necessarily to the global optimum.

We solve the case when R^* is measured in bits using a similar gradient descent algorithm from above.

4. EXPERIMENTAL RESULTS

This section examines the end-to-end performance of the two RDHT approaches for streaming packetized video content. Performance is measured in terms of the average luminance peak signal-to-noise ratio (Y-PSNR) in dB of the decoded video frames at the receiver as a function of different channel parameters, namely, available transmission bandwidth and packet loss rate. Two scenarios are considered. In the first one, the network is lossless, but there is an insufficient transmission bandwidth to send all video packets across the channel. Therefore, the sender needs to decide which packets to send and which packets to drop. In the second scenario, there is enough channel bandwidth to transmit every packet of the video once, however the network is lossy and some of the transmitted packets are lost. Therefore, the sender needs to decide at each transmission opportunity whether to (1) retransmit a previous lost packet, or (2) transmit a new packet which has not been transmitted before.

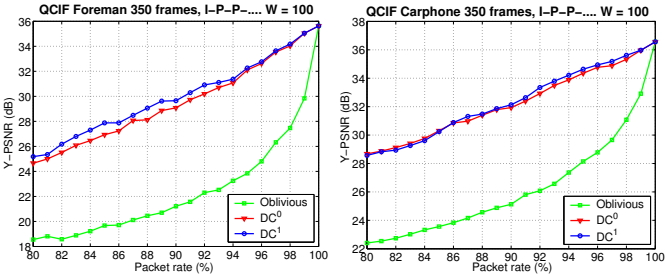

Fig. 3. Y-PSNR (dB) vs. Packet rate (%).

Two standard test video sequences in QCIF format, Foreman and Carphone, are employed in the experiments. The video sequences are coded using JM 2.1 of the JVT/H.264 video compression standard [7]. Each sequence has 300 frames at 30 fps, and is coded with a constant quantization level at an average Y-PSNR of about 36 dB. The first frame of each sequence is intra-coded, followed by P-frames. Every 4 frames a slice is intra updated to improve error-resilience by reducing error propagation (as recommended in JM 2.1), corresponding to an intra-frame update period of $4 \times 9 = 36$ frames. Three streaming systems are examined in the experiments: RD Hint Track systems using DC^0 and DC^1 as proposed in the prior section, while the system labeled *Oblivious* does not consider the distortion resulting from dropping a specific frame. In particular, when making transmission decisions, *Oblivious* does not distinguish between two packets that contain two different P frames, except for the size of the packets. Therefore, *Oblivious* randomly chooses between two P-frame packets of the same size, for example, when it needs to reduce the number of transmitted packets. Similarly, transmissions of new packets and retransmissions of old lost packets are also performed in a random order by this streaming system. In all three systems, packets are considered for transmission in non-overlapping windows of size $W = 100$. That is, at every transmission instance the sender considers 100 new packets for transmission, which correspond to 3.3 seconds of the video clip, given the selected frame rate. No retransmissions occur after the packets from the last transmission window are sent.

4.1. Adapting to Available Bandwidth

Figure 3 shows the performances of RDHT systems with DC^0 and DC^1, and *Oblivious* for streaming Foreman and Carphone as a function of the available packet rate measured in percent. For example, packet rate of 99% means that 99% of the packets in a transmission window can be transmitted. It can be seen that both DC^0 and DC^1 outperform *Oblivious* with quite a significant margin over the whole range of values considered for the available packet rate. This is due to the fact that DC^0 and DC^1 exploit the knowledge about the effect of loss of individual video packets on the reconstruction video quality, as discussed earlier. Therefore, both DC^0 and DC^1 RDHT approaches drop the video packets that will have the least impact on the quality of the reconstructed video. As seen in Figure 3 the performance gains reach up to 8 dB for Foreman and 7 dB for Carphone in the range of packet rates 86-96%. In

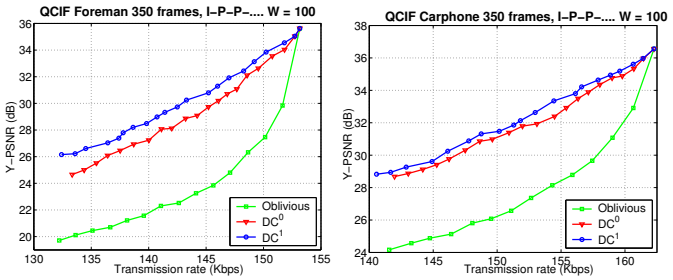

Fig. 4. Y-PSNR (dB) vs. Transmission rate (Kbps).

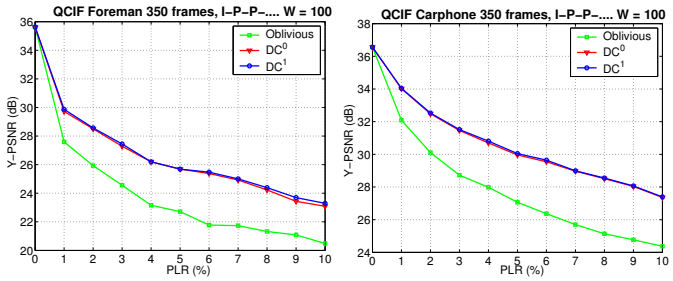

Fig. 5. Y-PSNR (dB) vs. PLR (%).

addition, even outside this range the gains in performance are still impressive and do not drop below 5 dB, except of course when we can send all the packets, i.e., for the case of packet rate of 100%. Finally, note that in this scenario the difference in performance between DC^0 and DC^1 RDHT systems is quite small.

Figure 4 examines the performances of DC^0, DC^1 RDHT systems and *Oblivious* for streaming Foreman and Carphone when there is a transmission constraint expressed in bits, rather than packets as in the prior experiment. Again, both DC^0 and DC^1 provide substantial performance gains over *Oblivious*. Improvement in performance is observed over the whole range of available transmission rates. The gains in performance remain steadily around 5-6 dB almost over the whole range of transmission rates under consideration, both for Foreman and Carphone. Note that in this case the performance difference between the RDHT-based systems and *Oblivious* is not as large as in the previous case. Having a transmission constraint expressed in bits makes predicting the resulting distortion at the receiver due to a packet drop pattern more difficult for the distortion chain based systems, as the number of dropped packets may need to vary over different transmission windows. Finally, note that the performance difference between the DC^0 and DC^1 RDHT systems is somewhat larger in this case.

4.2. Adapting to Packet Loss

The performance of the three streaming systems is now examined in the second scenario where we have packet loss. In contrast to the first scenario, here there is sufficient channel bandwidth to transmit once every packet of the video. However, there is random packet loss on the forward channel and the sender needs to decide whether it should retransmit previous lost packets or instead transmit new packets which have not been transmitted yet. In other words, in addition to the W packets from the current transmission window, the sender also considers for the present transmission past packets from previous transmission windows that have been lost during transmission. These experiments assume an ideal feedback channel, i.e., the sender is immediately notified of each lost packet, that the forward channel exhibits no packet delay, and that successive packet losses are independent and identically distributed.

Figure 5 shows that also in this scenario DC^0 and DC^1 RDHT systems outperform *Oblivious*. Performance improvement is observed over the whole range of packet loss rates (PLR) under consideration. Comparing Figures 3 and 5, we see that *Oblivious* attains similar PSNR at the same loss rate. In contrast, at the same loss rate, DC^0 and DC^1 achieves lower PSNR under the second scenario compared to that of the first scenario. One reason is the simple distortion model using Equations 1 and 3, does not take into account the possibility of packet losses in the channel. Instead, the systems under scenario two simply react to the losses, i.e. they decide on a packet drop pattern only after losses occur and they have a limited window over which to react. Nonetheless, even with the simple model, DC^0 and DC^1 still provide substantial performance gains. Specifically, for both sequences, a PSNR gain of 2-3 dB is maintained for packet loss rates greater than 3%.

5. CONCLUSIONS

This paper proposed Rate-Distortion Hint Track (RDHT) based systems for performing low-complexity rate-distortion optimized packet scheduling. An RDHT-based streaming system is composed of three components: (1) the R-D Hint Track information, (2) a simple algorithm for using the RDHT to predict the distortion for different packet schedules, and (3) a method for searching for the best packet schedule. Two RDHT-based streaming systems are examined, and experimental results demonstrate that for the difficult case of non-scalably coded video, the proposed R-D optimized scheduling systems provide 7-12 dB gain when adapting to a bandwidth constraint and 2-4 dB gain when adapting to random packet loss. Furthermore, the RDHT systems provide these gains with dramatically lower complexity than conventional R-D optimized scheduling approaches, and in particular the system using DC^0 provides these benefits with a complexity comparable to that of the non-RD-optimized oblivious system.

6. REFERENCES

[1] Z. Miao and A. Ortega, "Scalable proxy caching of video under storage constraints," *IEEE J. Selected Areas in Communications*, vol. 20, no. 7, pp. 1315–1327, September 2002, Special issue on Internet Proxy Services.

[2] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, 2001, submitted.

[3] J. Chakareski, P.A. Chou, and B. Girod, "Rate-distortion optimized streaming from the edge of the network," in *Proc. Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin Islands, Dec. 2002, IEEE, pp. 49–52.

[4] S.J. Wee and J.G. Apostolopoulos, "Secure scalable video streaming for wireless networks," in *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, USA, May 2001, IEEE, vol. 4, pp. 2049–2052.

[5] Z. Miao and A. Ortega, "Fast adaptive media scheduling based on expected run-time distortion," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2002, IEEE, vol. 2, pp. 1305–1309.

[6] 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. Int'l Conf. Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004, IEEE, to appear.

[7] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-C167 (T. Wiegand, ed.), "Committee draft number 1, revision 0 (CD-1)," *ITU-T Recommendation H.26L*, May 2002.