MULTIPLE DISTORTION MEASURES FOR SCALABLE STREAMING WITH JPEG2000

Carri Chan — Stanford University, Stanford, CA

Susie Wee, John Apostolopoulos — Mobile & Media Systems Lab, Hewlett-Packard Labs, Palo Alto, CA

ABSTRACT

The increasing diversity of end-user devices and networks allow different users to receive and view images and video at different resolutions and rates. Scalable coding methods allow streaming media systems to easily adapt media by selecting packets according to schedules that reflect their importance. Traditional scheduling algorithms are based on a single distortion measure such as mean-squared error relative to the original high resolution image. In this paper, we present a new approach of using *Multiple Distortion Measures* to schedule packets in a manner that explicitly accounts for a range of rates and resolutions. We show the effectiveness of this approach and examine a common scenario where multiple distortion measures are helpful. We present an algorithm that generates embedded JPEG2000 schedules and achieves 1 to 4 dB improvement over conventional approaches.

***Index Terms*—** Multiple distortion measures, scalable streaming, embedded packet schedules, scalable video coding, JPEG2000

1. INTRODUCTION

Users are viewing media content on an increasing variety of devices such as cell phones, PDAs, TVs, laptops, and desktop computers. Since these devices have different display resolutions and network bandwidths, it is desirable for streaming media systems to be able to quickly adapt the transmission of the coded media stream in a manner that maximizes the quality for each user. Scalable coding methods such as JPEG2000 allow quick adaptation by simply selecting media packets according to schedules that reflect their importance. Resolution can be reduced by selecting low resolution packets and rate can be adapted by scheduling packets based on rate-distortion performance.

Streaming media systems typically use scheduling algorithms that are based on minimizing a single distortion measure, such as mean-squared error with respect to the original high resolution image. The problem with this approach is that it neglects the needs of many users, such as low resolution users. For example, a scheduling algorithm optimized for high resolution users may send high frequency edge information at low rates, but these edges may not be visible to low resolution users. Since low resolution users are more likely to be accessing the media at the lower rates, it is desirable to have a scheduling algorithm that reflects these needs.

In this paper, we present a new approach of using *Multiple Distortion Measures* to consider the needs of a variety of users. With this approach, we can consider the relative importance of media packets for each user and develop scheduling algorithms that explicitly account for a variety of rate and resolution requirements. Specifically, we develop an algorithm that generates an embedded packet schedule using multiple distortion measures. We show that our algorithm outperforms the standard approach of using a single distortion measure.

This paper proceeds as follows. In Section 2, we present our problem formulation and review an embedded scheduling algorithm for a single distortion measure [1]. Section 3 formally introduces the idea of multiple distortion measures and demonstrates its importance. In Section 4, we study the properties of the schedules optimized for different distortion measures. In Section 5, we present an algorithm that generates an embedded schedule based on multiple distortion measures. We develop a framework to evaluate the performance of these scheduling algorithms and show the performance of our algorithm in this context.

2. PROBLEM FORMULATION AND BACKGROUND

We consider a streaming media system that encodes content once and adaptively streams it to a number of users. One such system is shown in Fig. 1, where a coded media stream is sent to a relay node that adaptively transcodes the stream for each receiver according to its display capabilities and network conditions. When a scalable coder such as JPEG2000 is used, the relay's transcoding operation is simply packet selection, and this operation can be further simplified to a single stream truncation operation by using embedded schedules. Embedded schedules are schedules that incrementally add packets–all packets for rate R_1 will also be included for rate $R_2 > R_1$. If all users have the same viewing capabilities, an optimal high resolution embedded schedule can be generated as in [1, 2].

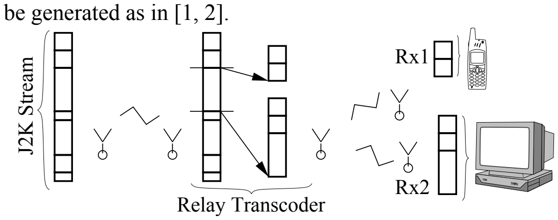

Fig. 1. Embedded schedules allow a mid-network transcoder to adapt media for different users with a stream truncation operation.

Our goal is to generate embedded schedules that minimize the distortion at various rates and resolutions. In this work, we measure distortion using mean-squared error (MSE). Due to coding dependencies, distortions across multiple packets are not strictly additive, as some packets can only be decoded with the inclusion of other packets. We map this dependence to precedence constraints, and note that JPEG2000 packets have precedence constraints across layers. Therefore, a JPEG2000 packet i is defined by its size, s_i, its multiple distortion values, p_{ij} for multiple j, and its precedence constraints of which packets must be included before it. We calculate distortion values, and subsequent profit values p_{ij}, by incrementally dropping the JPEG2000 packets and calculating the resulting distortion via our multiple distortion metrics. We use distortion and profit interchangeably–the distortion value is the distortion incurred with packet exclusion and the profit value is the profit gained with packet inclusion. These are equivalent.

The idea of viewing scalable media with different types of displays has been examined in the past. Temporal and spatial adaptation of scalable video is examined in [3]. In this work, performance metrics were user/classification-based upon subjective tests, whereas our work uses the standard measure of MSE. Spatial resolution reduction with video transcoding is studied in [4]. They focus on downsampling methodologies and use the original, high resolution image as the benchmark.

1-4244-0728-1/07/$20.00 ©2007 IEEE

In our work, we use multiple benchmarks and distortion measures.

2.1. Fused-Greedy Algorithm

We briefly review the fused-greedy algorithm presented in [1]. This algorithm gives the optimal embedded schedule for a single distortion measure. It leverages the idea that a packet's importance can be measured by its profit-to-size ratio. Because of the dependencies among JPEG2000 packets, some packets must be *fused* together to ensure that the precedence constraints are not violated. A violation occurs when the profit-to-size ratio of packet i is larger than that of packet j which must precede it. This causes a packet ordering of i then j, which violates the precedence constraints. Fusing the packets together prevents this violation from occurring.

The fused-greedy algorithm first computes the profit-to-size ratio of each packet, then fuses together the packets that violate a precedence constraint. Finally, the packet schedule is obtained by arranging the packets in decreasing order of the fused profit-to-size ratios. Let p_i, s_i, and r_i be the profit, size, and profit-to-size ratio of packet i.

FUSED-GREEDY(p, s)

$r_i = \frac{p_i}{s_i}$
Check precedence constraints
if Violation between packet i and $i+1$
  then Fuse packets:
    $r_i = \frac{p_{i,i+1}}{s_{i,i+1}} = \frac{p_i + p_{i+1}}{s_i + s_{i+1}}$
    $r_{i+1} = 0$
Sort packets by r_i

3. MULTIPLE DISTORTION MEASURES

In current media systems, distortion is typically measured against the original high resolution image. However, a low resolution viewer is most concerned with distortion compared to the low resolution image. For this reason, we propose using *Multiple Distortion Measures* to measure performance relative to each target device.

To calculate the distortion measures, we evaluate packet importance compared to a benchmark image, which may be the original image, as done conventionally, as well as the original image downsampled to different resolutions. To do this, we incrementally drop packets, decode and possibly downsample the image, and calculate the resulting MSE relative to the appropriate benchmark image. Thus, each packet has multiple distortion values associated with it: one for each resolution. By applying the fused-greedy algorithm to the high resolution metric, we get an *embedded high-optimized schedule*, as in the traditional case; by applying it to the low resolution metric, we get an *embedded low-optimized schedule*.

Many different downsampling methods exist, and it is important to note that our methodology applies to any downsampling method. Specifically, once the distortion values are computed, the rest of the analysis follows identically. In our experiments that follow, we examine the following two linear methods for 2×2 downsampling: (1) basic 2×2-pixel averaging, and (2) the 13-tap downsampling filter developed in the Scalable Video Coding (SVC) effort [5] applied separably, denoted here as "SVC".

3.1. Packet scheduling for low resolution viewing

Scalable coders allow low resolution images to be constructed by decoding only the low resolution packets or equivalently by downsampling a full size decoded image with the low resolution wavelet filter. However, we cannot guarantee that all devices will downsample with the wavelet filter. Thus, in this section we examine the packet selections and embedded schedules that result when using the fused-greedy algorithm for the two downsampling methods described above.

Table 1 summarizes comparisons of PSNRs evaluated for low resolution viewing for a variety of standard test images. We compare the PSNRs when the image is reconstructed using all of the low resolution packets to when the image is reconstructed using packets selected from the embedded low-optimized schedules to match the same rate constraint. Note that the packets are selected from all the packets, not just the low resolution packets. It is particularly surprising to see the gain that can be achieved by considering the high resolution packets. By allowing the selection from all packets, one can get 1-2dB gains when using pixel-averaging. The gains are smaller when using the SVC downsampling filter. However, we will see that the SVC downsampling filter gives greater gains in Fig. 2.

Image	Rate (bytes)	PSNR of LowRes pkts		PSNR of opt pkts	
		Pix-Avg	SVC	Pix-Avg	SVC
Actor	20386	30.52	36.43	32.72	36.70
Aerial	16344	32.22	37.97	34.46	38.74
Barboo	16119	28.85	34.23	29.81	34.28
Bike	17583	28.10	34.24	31.36	35.27
Cafe	17270	25.38	31.09	26.76	31.24
Woman	16983	36.21	41.10	38.71	41.47

Table 1. Comparison of PSNR for 6 images when including all of the low resolution packets as compared to selecting the optimal packets at the same rate. Downsampling is done using pixel-averaging and the SVC downsampling filter.

3.2. Comparison of Low- and High-Optimized Schedules

Typically, embedded schedules are optimized for high resolution viewing even at low rates, where the viewer is likely to have a low resolution display. We can see in Fig. 2 the PSNR versus Rate curves for embedded schedules generated using the fused-greedy algorithm, optimized at low and high resolutions. The solid lines correspond to the schedules optimized and evaluated at the low and high resolutions. The dotted curves correspond to the schedules optimized at low and measured at high resolution and optimized at high, measured at low. When using pixel-averaging, there is up to a 2dB gain in the low resolution PSNR when the packet selection is optimized for low resolution viewing rather than high resolution viewing. When using the SVC filter, the gains that can be achieved are up to 4dB. One reason for the drop in PSNR is when the image is optimized for high resolution viewing, edges are important, and packets corresponding to edges will be transmitted. However, once the image is reduced in size, these edges are no longer important. Therefore, instead of transmitting information to improve the low resolution image quality, bytes have been wasted on packets that cannot be seen on a low resolution display. This is a key factor as to why multiple distortion measures are necessary. Blindly optimizing for high resolution viewing or simply selecting low resolution packets can lead to a drop in performance of multiple dB.

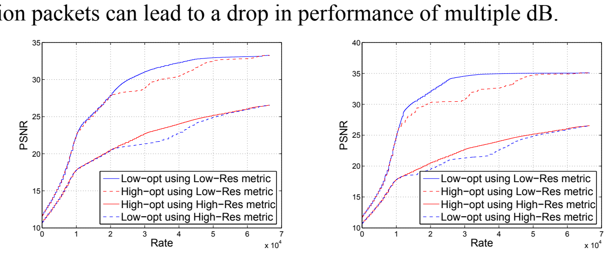

Fig. 2. PSNR vs Rate (in bytes) for schedules optimized for low and high resolutions measured at both low and high metrics. Downsampling by pixel-averaging (left) and SVC filter (right)

4. PROPERTIES OF SCHEDULES AT DIFFERENT DISTORTION MEASURES

In this section we examine some properties of schedules optimized for low resolution viewing compared to those optimized for high resolution viewing. For the sake of space, we present the results for a single image, Cafe; however, the trends are similar across multiple images.

Our goal is to find embedded schedules given multiple distortion measures. Given two schedules, S_1 and S_2, we define their correlation as the fraction of packets common in both schedules, i.e. $\frac{|S_1 \cap S_2|}{|S_1|}$. Therefore, embedded schedules have correlation equal to 1. Schedules that contains very different packets will have correlation close to 0. In Fig. 3, we see correlation is fairly varied between schedules optimized at different resolutions. At medium rates, the correlation is very low, which is why there are the PSNR gaps in Fig. 2 around the same rates. At high rates, most packets are included in the schedule, so correlation is close to 1. At low rates, the same few packets will create the foundation for the image, regardless of the viewing resolution, so correlation equals 1. When using the SVC filter, the correlation is much lower. These results reinforce the notion that packet selection at each distortion metric is very different, and optimizing for the high resolution measure does not equate to the selection of all low resolution packets.

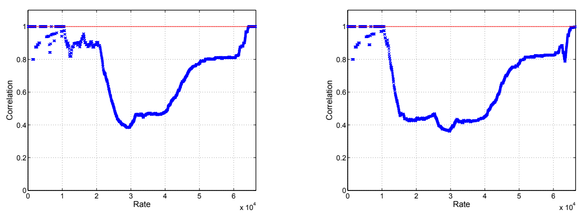

Fig. 3. Correlation between low and high resolution schedules. Downsampling by pixel-averaging (left) and SVC filter (right)

In Fig. 4, we compare the packet selection between schedules. We sort packets, so packet 1 is the first packet included for the low resolution schedule. The red line corresponds to the rate at which each packet enters the low resolution schedule. In gray, we plot the entry of the same packets into the high resolution schedule. In general, the more important a packet, the lower the entry rate; therefore, we can compare the varied packet importance of the two distortion metrics. The disparity between the two schedules is very large. In fact, a packet which is not important for the low resolution image and is included only at very high rates can be quite important for the high resolution image. Likewise, packets that are important in the low resolution image could be relatively unimportant at high resolution. This disparity between profit values causes the low correlations in Fig. 3 and the drop in PSNR in Fig. 2. There seems to be a larger discrepancy between schedules when downsampling is done using the SVC filter, which can justify the larger gaps in the PSNR curves.

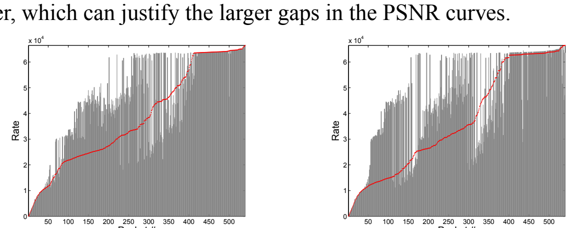

Fig. 4. Packet entry of the embedded schedule optimized at low resolution (red) and at high resolution (gray). Downsampling by pixel-averaging (left) and SVC filter (right).

5. SCHEDULING GIVEN A SWITCHING RATE

In this section we look at a special case of users with multiple distortion metrics. We present a framework to evaluate schedules for this scenario and present an algorithm to generate high-performance embedded schedules.

At a given rate, there may be multiple users who wish to view the content at low, medium, and high resolution. At other rates, for instance low rates, users may only view the content on low resolution displays. We focus on the specific problem of a fixed switching rate, R_s, where the distortion metric switches from low to high resolution. For all schedules with $R \leq R_s$, MSE is measured against the low resolution image whereas for schedules with $R > R_s$, MSE is measured against the high resolution image. Because packets have different importance at low and high resolutions, optimizing the schedule at low resolution for rates less than R_s and then switching to high resolution will give suboptimal performance and multiple distortion measures must be accounted for. We also assume that the probability of a user's channel having rate-constraint R is uniformly distributed from 0 to R_{max}, where R_{max} is the total size of the coded image.

If $R_s = R_{max}$, all distortions are measured by the low resolution distortion metric and the schedule is given by the fused-greedy algorithm on the low resolution profit values. Likewise, if $R_s = 0$, the schedule is determined by the fused-greedy algorithm on the high resolution profit values. Given these optimal schedules and R_s, we want to find an algorithm to generate schedules which switch between them.

5.1. Integral Distortion Performance Metric

Given a switching rate, R_s, we would like to find the schedule that minimizes distortion over all rates and users. In order to evaluate and compare the performance of a schedule, we propose the *Integral Distortion*, $D_I(S)$, performance metric. We define this metric as the sum of the distortion of the schedule at each rate.

$$D_I(S) = \sum_{R=0}^{R_s} d_L(S,R) + \sum_{R=R_s}^{R_{max}} d_H(S,R) \tag{1}$$

where S is the schedule, R_{max} is the total number of bytes in the image, $d_L(S,R)$ is the distortion measured at low resolution for schedule S evaluated at rate R, and $d_H(S,R)$ is the distortion measured at high resolution for schedule S evaluated at rate R. Because distortion at each rate is weighted equally, $\frac{D_I(S)}{R_{max}}$ is the expected distortion for uniformly distributed access rates. Therefore, minimizing D_I is equivalent to minimizing the expected distortion. Because we wish to minimize distortion, high-performing schedules will have very low Integral Distortion. This measure gives us a systematic way to compare different schedules. If $D_I(S_1) < D_I(S_2)$, then we can say that S_1 is a better schedule than S_2. This approach can be straightforwardly extended to more resolutions and non-uniform probability of viewing.

5.2. Changing Rate

Now that we have a performance metric to evaluate schedules, we introduce a scheduling algorithm to generate embedded schedules. This algorithm utilizes the fused-greedy algorithm and a changing rate, R_c.

One approach is to generate schedules by starting with the low resolution schedule and then when the distortion metric changes at R_s, the remaining packets should be added into the schedule according to the high resolution schedule. In this case, the changing rate would be equal to the switching rate, $R_c = R_s$. We examine this hypothesis in this section. We define R_c as the rate at which we change from low resolution to high resolution prioritization for packet scheduling, i.e., for $R < R_c$ we use the solution from the low resolution fused-greedy solution, and for $R > R_c$, we fill using the high resolution fused-greedy schedule. The scheduling algorithm is:

CHANGING-RATE(p_L, p_H, s)

$S_L = $ FUSED-GREEDY(p_L, s)
$S_H = $ FUSED-GREEDY(p_H, s)
Fill S according to S_L until S has R_c bytes
Fill remaining packets into S according to S_H

In Fig. 5, we can see the PSNR versus Rate curves as we sweep over various values of R_c for a fixed switching rate, R_s. As we decrease R_c and switch to the high resolution schedule at earlier rates, the performance of the low resolution schedules takes a hit. However, the performance of the high resolution schedules improves.

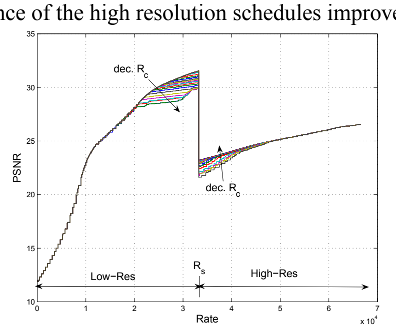

Fig. 5. PSNR vs. Rate for varying values of the changing rate, R_c. $R_s = 33000$. Downsampling using pixel-averaging.

Sweeping over different values of R_c vastly affects the integral distortion, as seen in Fig. 6. Empirically, we have seen for various switching rates, there is a unique R_c that corresponds to the minimum integral distortion. Certainly, the optimal changing rate, R_c^*, will depend on the switching rate, R_s, as well as the image/packet properties. However, given packet profits and R_s, we can search over R_c to find the optimal changing rate which minimizes the integral distortion.

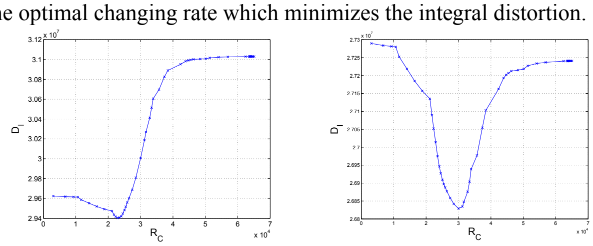

Fig. 6. Integral Distortion vs. changing rate with $R_s = 30000$ (left) and $R_s = 40000$ (right). Downsampling by pixel-averaging.

5.3. Performance Results

In this section we present results for the performance of the algorithm presented in Section 5.2. We compare the performance to the standard policy which generates schedules using the fused-greedy algorithm on the high resolution profit values. We also compare to the fused-greedy algorithm on the low resolution profit values. As a more multiple-distortion-aware policy, we also examine the case of fixing $R_c = R_s$ where no optimization is done to find the optimal changing rate R_c^*. All results shown are for actual decoded images; however, the optimal R_c^* is found by assuming an additive model, with precedence constraints, for distortions and minimizing integral distortion values.

In Fig. 7, we can see that the high resolution schedule performs very well if the switching rate is very low because there is little difference between the high and low schedules at low rates and at most rates, high resolution viewing is desired. However, as R_s increases, the performance drops because the high schedule ignores the low resolution profits. Likewise, the low resolution schedule performs well for high R_s, but very poorly for low R_s. Setting $R_c = R_s$ can outperform

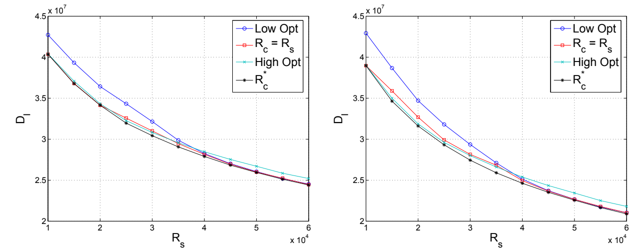

Fig. 7. Integral Distortion vs. switching rate for various algorithms. Downsampling by pixel-averaging (left) and SVC filter (right).

the low and high resolution schedules because it does switch between the low and high distortion metrics. However, we can see that if we optimize R_c, we can achieve even higher performance.

In Fig. 8, we compare the different policies for a fixed switching rate, $R_s = 35000$. The $R_c = R_s$ policy performs quite similar to the low optimized schedule because it switches much too late, though its performance improves at high rates. Our policy clearly outperforms the others. A very slight drop in PSNR for the high optimal schedule occurs just after the switching rate. However, at rates $R < R_s$, the R_c^* policy performs nearly 2dB better than the standard high optimized schedule and nearly as well as the low optimized schedule. When downsampling using the SVC filter, we can achieve over 3dB gains.

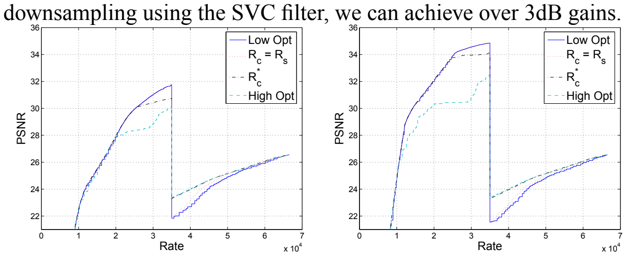

Fig. 8. PSNR versus rate with $R_s = 35000$ for various algorithms. Downsampling by pixel-averaging (left) and SVC filter (right).

6. CONCLUSION

In this paper we have proposed the use of multiple distortion measures to capture the need of users with varying display capabilities. By capturing the viewing needs of all users, quality of service can vastly be improved. We examined a scenario of using low resolution distortion measures at low rates and switching to high resolution distortion measures at high rates. We presented a framework to evaluate scheduling algorithms for JPEG2000 images with multiple distortion measures. Using this framework, we show our new scheduling policy outperforms the standard approaches which use single distortion measures. When comparing PSNR performance to a conventional scheduling algorithm, we found that our algorithm can achieve gains over 3dB.

7. REFERENCES

[1] C. Chan, S. Wee, J. Apostolopoulos. "On Optimal Embedded Schedules of JPEG-2000 Packets", in *Proc. IEEE ICIP*, October 2006.

[2] S. Wu, S. Dumitrescu, N. Zhang. "On Multirate Optimality of JPEG2000 Code Stream." *IEEE Transactions on Image Processing,* December 2005.

[3] Y. Wang, M. van der Schaar, S-F. Chang, A. C. Loui. "Classification-Based Multidimensional Adaptation Prediction for Scalable Video Coding Using Subjective Quality Evaluation," *IEEE Transactions on Circuits and Systems for Video Technology*, October 2005.

[4] P. Yin, M. Wu, B. Liu, "Video transcoding by reducing spatial resolution," in *Proc. IEEE ICIP*, September 2000.

[5] Joint Video Team of ISO/IEC MPEG & ITU-T VCEG, "AHG Report on Spatial Scalability Resampling," Document: JVT-Q007, October 2005.