

RATE-DISTORTION OPTIMIZED VIDEO STREAMING FOR NEW SCALABLE H.264

Sangho Yoon and Mark Mao

{holyyoon,markmao}@stanford.edu

ABSTRACT

We propose a new online scheduling algorithm based on SNR scalable H.264. We define a new importance measure for each packet based on the decoding dependency among packets across temporal and SNR layers. The new scheduling algorithm utilizes this new importance measure to minimize the expected reconstruction error at the decoder side under certain rate constraint. Experimental results show that the new online scheduling algorithm outperforms two simple heuristic scheduling algorithms.

1. BACKGROUND

The new H.264 standard promises higher quality video transmission for both high and low bandwidth networks. In order to improve the performance in case of varying link quality, a scalable version [?] of this standard has been recently proposed. The scalable H.264 partitions the compressed video data into layers so that different qualities of video can be transmitted according to the availability of network bandwidth. This scalability makes it ideal for video streaming over wireless network or internet where available bandwidth fluctuates over time.

The scalable H.264 coder provides three aspects of scalability, i.e. temporal, spatial and quality(SNR). The coding scheme for achieving spatial, temporal, and SNR scalability can be classified as a layered Motion-Compensated Temporal Filtering (MCTF) approach. The temporal scalability is mostly controlled by the temporal transform (or prediction). The spatial scalability is controlled by the pyramidal representation of the spatial scalability levels. And the quality scalability is controlled by the texture coding.

1.1. Temporal Scalability

The temporal decomposition framework of MCTF inherently provides temporal scalability. By using n decomposition stages, up to n levels of temporal scalability can be provided. In Figure 1, an example for the temporal decomposition of a group of 16 pictures using 4 decomposition stages is illustrated. If only the I and the B1 packets are transmitted, the picture sequence can be reconstructed at the decoder side has $\frac{1}{8}$ of the temporal resolution of the input

sequence. By additionally transmitting the B2 packets, the decoder can reconstruct an approximation of the picture sequence that has $\frac{1}{4}$ of the temporal resolution of the input sequence. And finally, if all the remaining B packets are transmitted, a reconstructed version of the original input sequence with the full temporal resolution is obtained

In general, by using n decomposition stages, the decomposition structure can be designed in a way that n levels of temporal scalability are provided with temporal resolution conversion factors of $\frac{1}{m_0}, \frac{1}{(m_0m_1)}, \dots, \frac{1}{(m_0m_1m_{n-1})}$, where m_i represents any integer number greater than 1. Therefore, a picture sequence has to be coded in groups of $N0 = (jm_0m_1\dots m_{n-1})$ pictures with j being an integer number greater than 0. The GOP size does not need to be constant within the picture sequence.

In our project, 4 decomposition stages are used with fixed GOP size 16, and temporal resolution conversion factor of $\frac{1}{2}$ between each stage.

1.2. Spatial Scalability

In the block transform based approach of MCTF, spatial scalability is provided by concepts used in the video coding standards H.262/MPEG-2 Visual, H.263, or MPEG-4 Visual. Conceptually a pyramid of spatial resolutions is provided. The base layer represents the lowest spatial resolution that can be decoded from the bitstream.

In our project, to simplify the problem, the spatial scalability provided by the coder is not used. There is only one spatial resolution(Qcif 176x144).

1.3. SNR (Quality) Scalability

As indicated above, the texture information is encoded in an base layer that provides a minimum quality at a given quantization level. The texture base layer is encoded using AVC entropy coding, including the block transformation, quantization and CABAC as specified in AVC. Within each spatial resolution SNR scalability is achieved by encoding successive refinements of the transform coefficients, starting with the minimum quality provided by AVC compatible texture encoding. This is done by repeatedly decreasing the quantization step size and applying a modified CABAC entropy

coding process akin to sub-bitplane coding. This coding mode is referred to as progressive refinement.

In our project, 2 enhancement layers are used on top of the base layer.

1.4. The GOP structure and decoding dependency

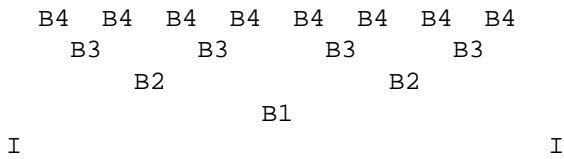
In our project, an open GOP structure as following is used. For each SNR layer

I N × [B4 B3 B4 B2 B4 B3 B4 B1 B4 B3 B4 B2 B4 B3 B4 I]

The first picture is always coded as single I (or more accurately IDR [Instantaneous Decoder Refresh] picture). The remaining of the stream is coded in groups of 16 pictures, with the anchor pictures at the end and 15 (hierarchically coded) B pictures between each pair of anchor pictures.

The following figure shows the structure of one GOP for all SNR layers(one base layer and 2 enhancement layers)

SNR L0 / 1 / 2



At the decoder side, for the base layer, B1 packet can only be decoded when the I packet of the current GOP and the I packet of the previous GOP are decoded. The B2 packet can only be decoded only when the closest I packet and B1 packet are decoded. For B3 and B4 packets, the dependency is the same, each packet depends on the closest lower level packet that precedes/follows it.

For the enhancement layers, the dependency is somewhat different. For any packet X in the first enhancement layer, it can only be decoded when the corresponding packet in the base layer is decoded. For any packet Y in the second enhancement layer, it can only be decoded when the corresponding packets in the base layer and the first enhancement layer are both decoded.

For example, if a B3 packet in the second enhancement layer is lost, the adjacent B4 packets can still be decoded, as long as the corresponding B4 packets in the base layer and enhancement layer 1 and the corresponding B3 packet in the base layer are decoded. But the distortion removed as a result of second enhancement layer B4 packet being decoded will be less compared to when the B3 packet in the second enhancement is available.

Apparently, the I packets in the base layer are the most important packets, since many packets depend on it. The B1 packets in the base layer and the I packets in the first enhancement layer are next important.

1.5. Related work

In rate-distortion optimized streaming, we want to minimize the expected distortion given a rate constraint. This problem can be formulated by the following Lagrangian cost function:

$$J(\pi) = D(\pi) + \lambda R(\pi) \quad (1)$$

where, π is the policy to transmit L data units, and $D(\pi)$ and $R(\pi)$ are the expected distortion and expected transmission rate for the transmission policy π .

Finding the optimal policy minimizing $J(\pi)$ above becomes intractable when we have multiple data units and each of them has multiple transmission opportunities. In the case of multiple data units, there is a transmission policy for each data unit ($\pi_l, 1 \leq l \leq L$) and π is a vector $(\pi_1, \pi_2, \dots, \pi_L)$. Therefore the complexity of (1) grows exponentially in the number of data units as well as the number of transmission opportunities [?]. Chou et al. [?] solved this problem by using an iterative descent algorithm where they optimize the Lagrangian of each π_l shown below separately until (1) converges.

$$J_l(\pi_l) = d(\pi_l) + \lambda_l r(\pi_l) \quad (2)$$

Even though this effectively decouples the packet dependencies, they could reduce the complexity of (1) to roughly proportional to the number of data units. However, the complexity is still exponential in the number of transmission opportunities. The complexity comes from the fact that the transmission policy is optimized for the current and future transmissions of multiple data units. They also showed how they could transmit only one data unit at a time, but they needed to find appropriate λ_l for each data unit. Even further, to perform rate control, their algorithm needs to adjust λ_l by iterations to meet the rate constraint. This should be done off-line and their algorithm is not feasible for on-line application.

We can think of transmitting only one data unit at a time, but finding the best order of transmission sequence is still exponential in the number of data units. Since we are considering to transmit only one data unit at a time, we actually need to choose only one data unit to transmit. Therefore rather than searching the entire policy space exhaustively to transmit one data unit at the current time, we first want to define an importance measure for each data unit and find the one which has the highest importance measure.

Similar work was done by Miao et al. [?][?]. They proposed an algorithm for scalable media streaming. Their algorithm tries to maximize the quality of reconstructed media at client side by on-line packet scheduling policy. To achieve that, they compute the expected distortion for each packet based on the transmission history and packet dependencies. Their distortion measure is simple and fast. However, in their distortion model, they did not consider the future schedules, but considered only the history of the transmissions. In addition, they simply modelled channel with a fixed value of packet loss probability without consideration of channel delay. More importantly, rate constraint was not incorporated into their distortion measure.

Therefore we want to propose a new distortion measure, which can be applied to on-line scheduling of scalable media with rate constraint. Also we want our algorithm to consider the future scheduling in a simple manner. If we have rate constraint, in selecting a packet to transmit, we not only consider the distortion that will be decreased by the current packet considered, but also the cost (e.g. in bytes) to pay for transmitting the current packet considered. To take into account of these two facts, we define a cost normalized distortion measure in the following section. Our algorithm chooses the most important packet at each time and discard less important ones to meet the rate constraint. By doing that, our algorithm effectively prunes the original data units in temporal and SNR domains and does not interact with the encoder. The benefit of non-interaction with the encoder is that we do not need to have multiple encodings of the same media for various rate constraints and this saves cost in encoding time and storage.

For simplicity, we will assume that each data unit can be placed on one packet. Thus data unit and packet can be used interchangeably without ambiguity throughout this paper.

2. SCHEDULING ALGORITHM BASED ON IMPORTANCE MEASURE

2.1. Problem formulation

We want to minimize the expected reconstruction error at client side under certain rate constraint. If the data unit l is decodable by the receiver on time, then the reconstruction error is reduced by Δd_l . To be decodable, all packets, which data unit l is dependent on, should have been arrived on time. Otherwise, we can not decode the received data unit l . Thus, the expected reconstruction error resulted by transmitting data units based on a policy π becomes:

$$D(\pi) = D_0 - \sum_l \Delta d_l \prod_{l' \in M(l)} (1 - P_e(l', \pi)) \quad (3)$$

$$= D_0 - D_c(\pi) \quad (4)$$

where, $M(l)$ is a set of packets which packet l depends on, $P_e(l', \pi)$ is the loss probability of packet l' under policy π , and π transmits one packet at a time.

We also have a measure of cost for π (=total number of bytes transmitted by π):

$$R(\pi) = \sum_l B_l \rho(l, \pi) \quad (5)$$

where B_l is the packet size in bytes, and $\rho(l, \pi)$ is the number of transmissions of packet l under policy π .

With the rate constraint R , our optimal policy π^* is:

$$\pi^* = \operatorname{argmin}_{\pi, R(\pi) \leq R} D(\pi) \quad (6)$$

$$= \operatorname{argmax}_{\pi, R(\pi) \leq R} D_c(\pi) \quad (7)$$

As discussed before, the complexity of finding π^* by exhaustive searching is exponential in the number of packets. In addition to that, if we have some feedback from channel or receiver, we need to incorporate that into the policy and re-search the optimum policy π^* accordingly. Therefore, full-searching algorithm is not suitable for on-line application.

Finally, the quantity Δd_l is not unique in highly complicated GOP structure. For example, there can be two kinds of dependencies between packets: direct and indirect. If there is a direct dependency between packets, then child packet can only be decoded when parent packets are received. However, in indirect relationship, child packets can still be decoded without parent packets. Indirect parent packets affects only on the amount of distortion reduction of child packets. This necessitates considering past and future transmissions of parent and child packets to find Δd_l of each packet, and leads to quite high computation. Thus, we propose a simple distortion measure, which not only reduces searching complexity, but also improves the quality of media played at client side.

2.2. Packet Importance Measure

We define the importance measure of each packet and transmit the packet with the highest importance measure. Especially we are interested in the case where there are indirect dependencies between packets.

We first measure the distortion reduction Δd_l as the decrease in distortion by decoding packet l . To take into account of indirect dependencies between packets, we weight Δd_l by w_l :

$$\Delta d_{l,weighted} = w_l \Delta d_l \quad (8)$$

Basically, the distortion would decrease by Δd_l when we decode packet l assuming all of its direct and indirect parents are available. However, when some of indirect parents are not available at decoder, then the distortion decrease

would be less than Δd_l . By using some reasonable pre-determined weight w_l based on the transmission history, we can avoid computing the actual distortion decrease expected by decoding packet l .

We also try to incorporate the transmission history of packet l into its importance measure. Intuitively, we do not want to re-transmit those packets transmitted short time before. Thus, we compute the probability of future loss conditioned on the knowledge of feedback and the deadline:

$$\begin{aligned} P_{e, fu}(l) &= \prod_{i=1}^{n(l)} P(FTT > t_{d,l} - t_x(i) | RTT > t_c - t_x(i)) \\ &= \prod_{i=1}^{n(l)} \frac{P(FTT > t_{d,l} - t_x(i))}{P(RTT > t_c - t_x(i))} \end{aligned}$$

where, $n(l)$ is the total number of previous transmission trials of packet l including the current trial, t_c is the current time, $t_{d,l}$ is the dead line of packet l , $t_x(i)$ is the transmission time stamp of i^{th} trial of packet l , FTT is the forward travel time and RTT is the round travel time.

We also define another probability of loss only for the past transmission trials of packet l' as follows:

$$P_{e, pa}(l') =$$

$$\left\{ \begin{array}{l} 0 \rightarrow \text{packet } l' \text{ ACKed} \\ 1 \rightarrow \text{packet } l' \text{ Not yet sent} \\ \prod_{i=1}^{n(l')} P(FTT > t_{d,l'} - t_x(i) | RTT > t_c - t_x(i)) \\ = \prod_{i=1}^{n(l')} \frac{P(FTT > t_{d,l'} - t_x(i))}{P(RTT > t_c - t_x(i))} \\ \rightarrow \text{packet } l', \text{ sent } n(l') \text{ times before} \end{array} \right.$$

The importance measure of packet l considers the above probabilities, weighted distortion, and the cost in transmitting packet l by normalizing with the packet size B_l :

$$I(l) = \frac{P_{e, fu}(l) \Delta d_{l, weighted} \prod_{l' \in M(l)} (1 - P_{e, pa}(l'))}{B_l}$$

2.3. Scheduler with rate constraint

In [?], they achieved rate control by iteratively finding λ . However, this is not suitable for on-line application. In [?][?], they did not consider rate control. Simple algorithm to control rate is to pre-select the set of packets to transmit. However, we need to update the set of packets to transmit based on the feedback. This can be easily achieved by our algorithm. Since we are selecting the most important packet at each time, we achieve rate control by stopping transmission when the total number of bytes transmitted is greater than or equal to the maximum allowable transmission bytes ($=R\Delta T$, ΔT is the total transmission period). Therefore, our algorithm based on the importance measure $I(l)$ not

only chooses a packet to transmit fast, but also meets the rate constraint at the same time. In our experiments, we perform the rate control GOP by GOP.

3. RESULTS AND FUTURE WORK

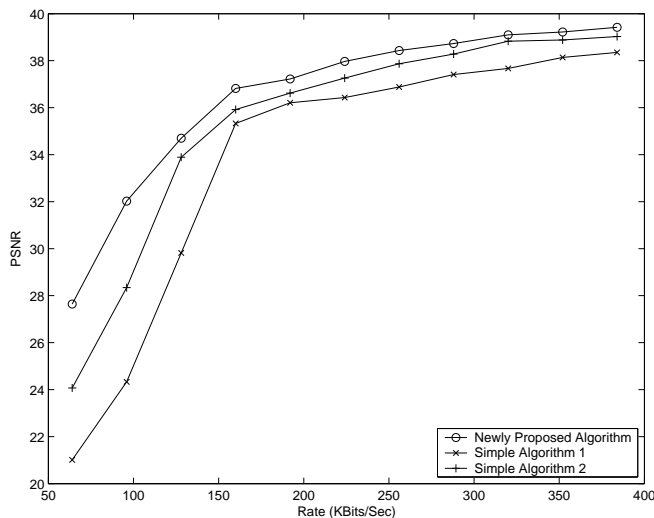
We simulate our algorithm for *foreman* video sequences encoded with three layers by new H.264 scalable coder [?] at frame rate of 30 Hz. Each layer of frame is put into one packet. We use 16 frames per GOP. We model the network as in [?], with independent delay assumptions. Delay is modelled by a shifted Gamma distribution with shift $\kappa = 10$ ms, mean 40 ms, and the standard deviation 10 ms. Packets are delayed both in forward and backward directions. The packet loss probability in both forward and backward directions are 0.1. We use start-up delay of 1000 ms.

Our algorithm is compared with two simple heuristic algorithms. First simple algorithm transmits every packet twice starting from the SNR base layer. Second one transmits in the same way, but checks ACK after 120 ms (=mean RTT time + 2σ). If ACK not received by that time, we re-transmit the previous one.

As we can see from the rate-PSNR curve, our algorithm outperforms the first simple algorithm by about 4 db on average, and outperforms the second simple algorithm by about 2 db on average.

Table 1. Comparison between proposed algorithm and two simple algorithms in PSNR

Rate(kbps)	Proposed	Simple 1	Simple 2
64	27.6400	21.0100	24.0700
96	32.0200	24.3300	28.3400
128	34.7000	29.8200	33.8900
160	36.8200	35.3300	35.9200
192	37.2200	36.2100	36.6200
224	37.9700	36.4300	37.2600
256	38.4300	36.8800	37.8700
288	38.7300	37.4100	38.2800
320	39.1000	37.6700	38.8300
352	39.2200	38.1400	38.8800
384	39.4200	38.3500	39.0300



4. CONCLUSIONS

Fast scheduling algorithm for on-line application is considered. Compared to previous heuristic algorithms [?][?], we use more realistic channel delay model. In addition to that, we define a new importance measure suitable for complex GOP structure, where there are both direct and indirect dependencies between packets. Finally, rather than pre-determine the set of packets to meet to rate constraint, we update the set at each transmission time by using the feedback. This can be simply achieved by our algorithm without extra cost. We choose the most important packet each time and stop transmission when we meet the rate constraint. Simulation results show that our algorithm is superior to simple algorithms.

5. ACKNOWLEDGEMENT

We want to thank Prof. Girod for his excellent lectures and TA David for his assistance in the logistics. We want to give special thanks to Mark Kalman for his guidance and help throughout the project.

6. REFERENCES

- [1] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," Microsoft Research, Tech. Rep. MSR-TR-2001-35, February 2001, (also submitted to IEEE Transactions on Multimedia).
- [2] M. Kalman, and B. Girod, "Rate-Distortion Optimized Streaming of Video With Multiple Independent Encodings," Proc. IEEE International Conference on Image Processing, ICIP-2004, Singapore, October, 2004.

- [3] Z. Miao and A. Ortega, "Optimal Scheduling for the Streaming of Scalable Media", Proc. of Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, CA, Oct. 2000
- [4] Z. Miao and A. Ortega, "Expected Run-Time Distortion Based Scheduling for Delivery of Scalable Media", Packet Video Workshop 2002, Pittsburgh, PA, Jan. 2002.
- [5] M. Podolsky, S. McCanne, and M. Vetterli, "Soft ARQ for layered streaming media," Tech. Rep. UCB/CSD-98-1024, University of California, Computer Science Division, Berkeley, CA, Nov. 1998.
- [6] Heiko, "Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)14th Meeting: Hong Kong, CN, 17-21 January, 2005"