Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

An optimal quality adaptation mechanism for end-to-end FGS video transmission*

FENG Shun†, ER Gui-hua, DAI Qiong-hai, LIU Ye-bin

(*Broadband Networks & Digital Media Laboratory, Department of Automation, Tsinghua University, Beijing 100084, China*)

†E-mail: fengs03@mails.tsinghua.edu.cn

Received Dec. 5, 2005; revision accepted Feb. 25, 2006

Abstract: In this paper, we propose a novel optimal quality adaptation algorithm for MPEG-4 fine granular scalability (FGS) stream over wired network. Our algorithm can maximize perceptual video quality by minimizing video quality variation and increasing available bandwidth usage rate. Under the condition that the whole bandwidth evolution is known, we design an optimal algorithm to select layer. When the knowledge of future bandwidth is not available, we also develop an online algorithm based on the optimal algorithm. Simulation showed that both optimal algorithm and online algorithm can offer smoothed video quality evolution.

Key words: MPEG-4 fine granular scalability (FGS), Video streaming, Transmission, Quality adaptation

doi:10.1631/jzus.2006.AS0119 **Document code:** A **CLC number:** TN919.8

INTRODUCTION

With the increasing multimedia application, internet video streaming has become an important research area. Delay sensitivity, high bandwidth fluctuation at multiple time scales, and buffer size limitation pose a great challenge for video transmission. All of these limitations make it difficult to deliver video streaming with full quality. Small time scale bandwidth variation can be accommodated by client buffer. Fine granular scalability (FGS) (Li, 2001) video coding algorithm is a good way to accommodate large time scale bandwidth variation. In FGS video coding, frames are encoded in several layers. Base layer is encoded in the least bit rate and decoded to provide the minimum video quality. Enhancement layer has higher bit rate and can be decoded cumulatively to improve video quality. When transmitting FGS stream, if bandwidth is in bad condition, we just deliver base layer; if bandwidth is in good condition, enhancement layer can be transmitted. Available bandwidth is always at variable bit rate, while consistent video quality leads to significantly variable bit rate. So the problem is to accommodate the mismatch between available bandwidth variability and encoded video variability.

In this paper, we briefly review past works in (Kim and Ammar, 2003; Nelakuditi *et al*., 2000) and then develop an algorithm for FGS streaming that maximizes perceptual video quality through minimizing quality variation while at the same time increasing available bandwidth usage rate. The main differences from previous algorithms (Kim and Ammar, 2003; Nelakuditi *et al*., 2000) are as follows: we give a novel equation to define maximum available buffer resource, and use it to select layer; our algorithm can provide consistent video quality for VBR FGS streaming, and previous algorithm (Kim and Ammar, 2003) can be seen as its special case; previous algorithm (Kim and Ammar, 2003) does not consider frames buffered in client buffer and so sometimes buffer overflow happens, but our algo-

* Project supported by the National Natural Science Foundation of China (No. 60432030) and the National Science Fund for Distinguished Young Scholars (No. 60525111), China

rithm dynamically considers the effect of buffer consumption on buffer constraint, such that no buffer overflow happens.

This paper is organized as follows. In Section 2, we formulate the problem. In Section 3, we review past works and present our offline algorithm and online algorithm. We give simulation results and compare our algorithm with previous algorithm in Section 4. In Section 5, we conclude this paper.

PROBLEM FORMULATION

The objective of smoothing quality is to make the variation of received video quality as small as possible. When formulating this problem, we consider the video sequence as a discrete-time model at frame level. Each time slot is the time needed to decode one frame. At every time slot, what we need do is to decide which layer to transmit, while not violating bandwidth constraints and receiver buffer constraints. Before the playback at receiver, there is some delay and some frames are stored in receive buffer. We also assume that there is no packet loss and network delay. The notations used in this paper are listed as follows:

C_i^j: Available transmit capacity for layer j of frame i;

B_i^j: Buffer status for layer j at time slot i;

$Size_i^j$: Buffer occupancy in the buffer for layer j at time slot i;

MO_i^j: Maximum possible buffer occupancy in the buffer for layer j at time slot i;

F_i^j: The size of layer j in frame i;

Bf_j: Buffer size allocated for layer j;

c_i^j: Available bandwidth for layer j of frame i;

c_i: Available bandwidth at time slot i;

Se_i^j: Transmit decision made for layer j of frame i. When layer is selected, $Se_i^j = F_i^j$ and when layer is discarded, $Se_i^j = 0$;

λ: The maximum possible number of frames in receive buffer.

Now, the problem can be formulated as follows. For a video sequence which has K frames and L layers, find an optimal transmit decision which can minimize received video quality variation under bandwidth constraints and buffer constraints. We use Eq.(1) to describe it:

$$\{Se_i^j\} = \arg\min_{\text{Select}} F(Se_i^j, c_i, Buffer_j, F_i^j), \tag{1}$$

where $\{Se\}$ is the set of Se, $1 \le i \le K$, $1 \le j \le L$, $i \in \mathbb{Z}, j \in \mathbb{Z}$.

We use AQT (average quality transition) (Kim and Ammar, 2003) and ARL (average run length) (Nelakuditi *et al.*, 2000) to measure quality variation:

$$AQT = \frac{1}{L}\sum_{j=1}^{L}\sum_{i=1}^{K} I_j(i), \tag{2}$$

$$ARL = \frac{1}{L}\sum_{j=1}^{L}\frac{1}{T_j}\sum_{p=1}^{T_j} n_j(p), \tag{3}$$

where $I_j(i)=1$, if there is quality transition, i.e., layer j of both frame i and frame $i+1$ is not selected or selected at the same time; $I_j(i)=0$, otherwise. T_j is quality transition number of layer j; $n_j(p)$ is frame number between two consecutive quality transitions at layer j.

We quantize the objective of minimizing quality transition, to minimize which, we have to minimize AQT and maximize ARL.

SMOOTHING ALGORITHM

Some algorithms (Kim and Ammar, 2003; Nelakuditi *et al.*, 2000) have been proposed to smooth quality for FGS streaming. Algorithm in (Nelakuditi *et al.*, 2000) uses bi-directional optimal layer selection to maximize perceptual quality. The forward scan can be seen as a step that identifies the end of each run and the minimal number of runs. The backward scan extends each run towards the front of the run while maximizing residual buffer made available to higher layer. But the algorithm is proposed for CBR video, it has no effect on VBR video. Algorithm in (Kim and Ammar, 2003) is more sophisticated and proposed to adapt VBR video quality. It uses cumulative bandwidth resource and buffer to select layer to deliver. According to this algorithm, cumulative bandwidth capability is calculated by the following equation:

$$Cu_j[i] = \min(S_j[i-1] + Bf_j,\ C_j[i-1] + c_i^j), \tag{4}$$

where $Cu_j[i]$ is the cumulative bandwidth capability for layer j at time slot i, $S_j[i]$ is the cumulative selected data defined by $\sum_{k=1}^{i} Se_k^j$. In fact, the equation assumes that there is only one frame in the buffer. When there are several frames in the buffer and bandwidth is in good condition, the available cumulative bandwidth capability should be

$$Cu_j[i] = \min(S_j[i-1] + Bf_j - B_j,\ C_j[i-1] + c_i^j), \quad (5)$$

where B_j is the size of frames in buffer j. In such case, the result from Eq.(4) will lead to buffer overflow.

In order to overcome such drawback, we propose a novel quality smoothing algorithm for MPEG-4 FGS stream. In our algorithm, the video stream used for analysis is a scalable VBR stream with three layers: base layer, FGS layer, and FGST layer. The receiver has three buffers to store the data from base layer, FGS layer, and FGST layer respectively. Before the start of playback, the receiver stores some frames in the buffer.

Preliminaries

In order to simplify the problem, we assume that both send order and receive order of the packets are the same as the decode order at the client. Layer selection of frame i is denoted by transmit decision $\{Se_i^j, 1\le j\le L, 1\le i\le K\}$. At the end of every time slot i, we will record receive buffer status $\{B_i^j, 1\le j\le L, 1\le i\le K\}$ of buffers allocated for base layer, FGS layer, and FGST layer. It is an array, where stored data order is the same as the decode order, i.e., $B_i^j(1)$ is the data which will be decoded in buffer j at time slot i+1. At the end of every time slot i, element $B_i^j(m)$ in buffer j is the size for layer j of frame i+m, if layer j is selected, $B_i^j(m) = F_{i+m}^j$; if layer j is discarded, $B_i^j(m) = 0$. So, we can get $B_i^j(m) = Se_{i+m}^j$. Given $\{B_i^j(m)\}$, we can calculate the whole occupancy of buffer j at the end of time slot i as $Size_i^j = \sum_m B_i^j(m)$. And we can derive B_i^j from B_{i-1}^j as follows:

$$B_i^j(k) = B_{i-1}^j(k+1),\ 1\le k\le \lambda-1,\ k\in\mathbb{Z},\ B_i^j(\lambda) = Se_i^j. \quad (6)$$

At every time slot i, the channel provides available bandwidth c_i, if all of the layers below layer j are selected to transmit, the available bandwidth c_i^j for layer j is $c_j - \sum_{n=1}^{j-1} F_i^n$. In order to avoid buffer overflow, we define available transmit capacity C_i^j for layer j of frame i as follows:

$$C_i^j = \min\{c_i^j, Bf_j + B_{i-1}^j(1) - Size_{i-1}^j\}. \quad (7)$$

The second item in the above equation can guarantee that buffer overflow will not happen.

While not violating bandwidth constraint and buffer constraint, we define maximum possible buffer occupancy MO_i^j of buffer j at the end of time slot i. MO_i^j can be seen as buffer resource of buffer j at the end of time slot i+1. If MO_i^j increases, it means that there are more buffer resources, and we can transmit more data to receiver; If MO_i^j decreases, it means that there are less buffer resources, and we should transmit less data to receiver.

$$MO_i^j = \min\{Bf_j, MO_{i-1}^j - B_{i-1}^j(1) + C_i^j\}. \quad (8)$$

Algorithm

Algorithm 1 gives the details of offline quality smoothing algorithm. Before the start of playback, we let receiver store λ frames $\{B_0^j, 1\le j\le 3\}$ in the buffer. If layer j of previous frame is selected, layer j of current frame is selected only when $MO_i^j \ge Temp_i^j$ (at line 13). $Temp_i^j$ is a temporary variable, which is the buffer occupancy when layer j of frame i is selected. This can be explained that when maximum possible buffer occupancy is more than current buffer occupancy, we can utilize more buffer space and so we can select layer to transmit. If layer j of previous frame is not selected, layer j of current frame will not be selected until $MO_i^j = Bf_j$ and $Size_{i-1}^j = 0$ (at line 19). This can be explained that in order to minimize quality transition, we accumulate buffer resource. When $MO_i^j = Bf_j$ and $Size_{i-1}^j = 0$, the available buffer resource is the maximum, the expected run will be the maximum. Now, we will prove this point.

Proof We assume that a discard phase starts at t_0 and ends at t_0+m-1. Then a select phase starts at t_0+m and ends at $t_0+m+n-1$. We now prove that when $MO^j_{t_0+m}=Bf_j$ and $Size^j_{t_0+m-1}=0$, the run for discard phase will be the maximum, that is, n is the maximum.

Consider selecting run n' when $MO^j_{t_0+m}=a$ and $Size^j_{t_0+m-1}=b$. At time slot t_0+m+n', maximum buffer occupancy is equal to current buffer occupancy:

$$MO^j_{t_0+m+n'}=Size^j_{t_0+m+n'}. \tag{9}$$

From Eq.(7), we can get

$$MO^j_{t_0+m+n'}=MO^j_{t_0+m}-\sum_{k=t_0+m}^{t_0+m+n'-1}B^j_k(1)+\sum_{k=t_0+m+1}^{t_0+m+n'}C^j_k. \tag{10}$$

And buffer occupancy at time slot t_0+m+n' can be

$$Size^j_{t_0+m+n'}=Size^j_{t_0+m}-\sum_{k=t_0+m}^{t_0+m+n'-1}B^j_k(1)+\sum_{k=t_0+m}^{t_0+m+n'-1}F^j_k. \tag{11}$$

From Eqs.(7)~(9), we can have

$$MO^j_{t_0+m+n'}-Size^j_{t_0+m+n'}=\sum_{k=t_0+m}^{t_0+m+n'-1}F^j_k-\sum_{k=t_0+m+1}^{t_0+m+n'}C^j_k, \tag{12}$$

if we write it in continuous function, we can get

$$a-b=\int_{t_0+m}^{t_0+m+n'-1}F^j_t\,\mathrm{d}t-\int_{t_0+m+1}^{t_0+m+n'}C^j_t\,\mathrm{d}t=n'(\bar{F}^j-\bar{C}^j), \tag{13}$$

where $\bar{F}^j$ and $\bar{C}^j$ is the average size of F^j_k and C^j_k.

Because $0\le b\le a\le Bf_j$, we get

$$n'=\frac{a-b}{\bar{F}^j-\bar{C}^j}\le\frac{Bf_j-0}{\bar{F}^j-\bar{C}^j}=\frac{Bf_j}{\bar{F}^j-\bar{C}^j}=n. \tag{14}$$

Algorithm 1 Offline quality smoothing algorithm

1. **Procedure** offline algorithm $(c_i, Bf_j, B^j_0, F^j_i, \lambda)$;
2. Initialization $c^1_i=c_i$, $Size^j_0=\sum_{m=1}^{\lambda}B^j_0(m)$, $MO^j_0=Size^j_0$, where $1\le i\le K$, $1\le j\le L$;
3. for j=1 to L
4.   for i=1 to K
5. $C^j_i=\min\{c^j_i, Bf_j+B^j_{i-1}(1)-Size^j_{i-1}\}$
6. $MO^j_i=\min\{Bf_j, MO^j_{i-1}-B^j_{i-1}(1)+C^j_i\}$
7. $Temp^j_i=Size^j_{i-1}-B^j_{i-1}(1)+F^j_i$
8.   if $Se^j_{i-1}=F^j_{i-1}$
9.   if $MO^j_i\ge Temp^j_i$, $Se^j_i=F^j_i$
10.   else, $Se^j_i=0$
11.   end if
12.   else
13.   if $MO^j_i=Bf_j$ and $Size^j_{i-1}=0$, $Se^j_i=F^j_i$
14.   else, $Se^j_i=0$
15.   end if
16.   end if
17. $c^{j+1}_i=c^j_i-Se^j_i$
18. $B^j_i(k)=B^j_{i-1}(k+1)$, $1\le k\le\lambda-1$, $B^j_i(\lambda)=Se^j_i$
19. $Size^j_i=\sum_{m=1}^{\lambda}B^j_i(m)$
20.   end for
21. end for
22. end procedure

Offline algorithm is based on the complete knowledge of bandwidth evolution. However, when we implement online algorithm, the future bandwidth is unavailable. So we need to predict bandwidth, then the predicted bandwidth is used to guide the online algorithm. Although the performance of online algorithm depends on predicted bandwidth, this paper cares about the efficiency of our quality smoothing algorithm. Therefore, we assume that predicted bandwidth is reliable. The framework of online algorithm is similar to the counterpart of offline algorithm. Detailed algorithm is depicted in Algorithm 2. The main differences between online algorithm and offline algorithm are that (1) online algorithm uses predicted bandwidth to select layer (line 4); (2) offline algorithm receives buffer update (lines 20~21 in Algorithm 2).

Algorithm 2 Online quality smoothing algorithm

1. **Procedure** online algorithm $(c_i, Bf_j, B^j_0, F^j_i, \lambda)$;
2. Initialization $c^1_i=c_i$, $Size^j_0=\sum_{m=1}^{\lambda}B^j_0(m)$, $MO^j_0=Size^j_0$, where $1\le i\le K$, $1\le j\le L$;
3. for i=1 to K
4.   *predict_bandwidth*(c^j_i)

for j=1 to L
$C_i^j = \min\{c_i^j, Bf_j + B_{i-1}^j(1) - Size_{i-1}^j\}$
$MO_i^j = \min\{Bf_j, MO_{i-1}^j - B_{i-1}^j(1) + C_i^j\}$
$Temp_i^j = Size_{i-1}^j - B_{i-1}^j(1) + F_i^j$
if $Se_{i-1}^j = F_{i-1}^j$
if $MO_i^j \geq Temp_i^j$, $Se_i^j = F_i^j$
else, $Se_i^j = 0$
end if
else
if $MO_i^j = Bf_j$ and $Size_{i-1}^j = 0$, $Se_i^j = F_i^j$
else, $Se_i^j = 0$
end if
end if
$c_i^{j+1} = c_i^j - Se_i^j$
end for
$B_i^j(k) = B_{i-1}^j(k+1),\ 1 \leq k \leq \lambda - 1,\ B_i^j(\lambda) = Se_i^j,\ 1 \leq j \leq L$
$Size_i^j = \sum_{m=1}^{\lambda} B_i^j(m),\ 1 \leq j \leq L$
end for
end procedure

SIMULATION RESULTS

In this section, we show results of experiment which evaluate our quality smoothing algorithm. We use ns-2 (Http://www.isi.edu/nsnam/ns/) to set up simulation experiment. Fig.1 gives the topology of the simulation experiment. In order to simulate the realistic network bandwidth, we set a flow between Node 4 and Node 7 as a compound flow with self-similar characteristics. Flow between Node 5 and Node 8 is TFRC stream. Flow between Node 3 and Node 6 is TCP stream, whose throughput will be recorded to do simulation.

MPEG-4 FGS encoded stream of a river running through it is used in these experiments, and you can find its trace in (Http://www.cc.gatech.edu/computing/Telecomm/people/Phd/tkim/qa.html). The encoded stream has three layers: base layer, FGS layer, and FGST layer. Every layer is encoded in variable bit rate. A GOP includes 12 VOPs and there are 2 GOPs in 1 s. The whole size of receive buffer is set to be 600 kB. It is divided into three buffers for base layer, FGS layer, and FGST layer in the ratio of 0.04, 0.32, and 0.64 respectively. Bandwidth variability of TCP stream is shown in Fig.2. Small time scale variability is significant as much as about 2 Mbps in the steady state. In order to evaluate our algorithm, we first compare our algorithm with the algorithm in (Kim and Ammar, 2003). Fig.3 shows the performance of our offline algorithm (λ=1) and the offline algorithm in (Kim and Ammar, 2003).

From the figure, we can see that when the maximum possible number of frames in receiver

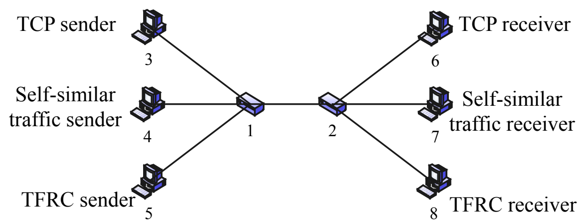

Fig.1 Network topology

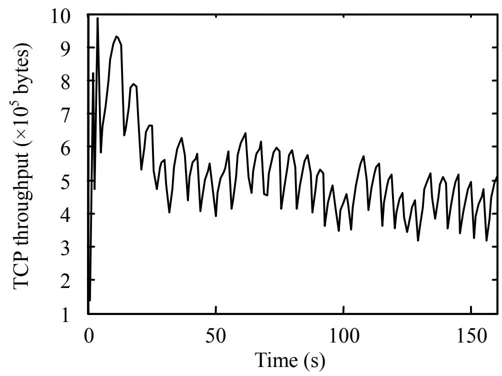

Fig.2 TCP throughput

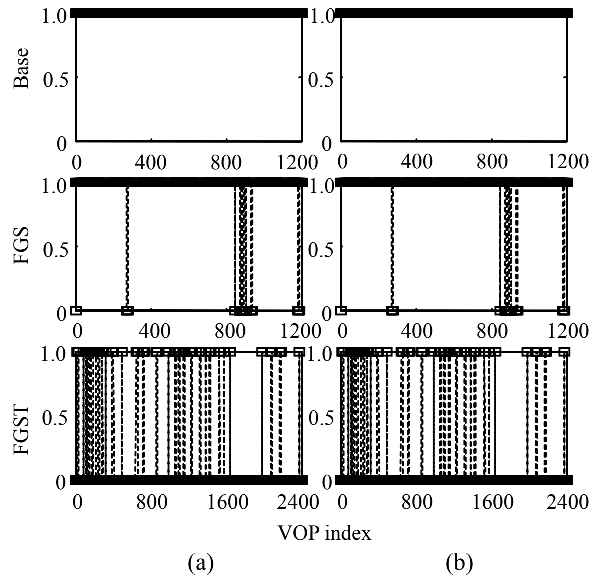

Fig.3 VOP selection of offline algorithm. (a) Offline algorithm in (Kim and Ammar, 2003); (b) Our offline algorithm

buffer is 1, both algorithms achieve the same result. AQT and ARL for both algorithms are 24.3 and 450.4 respectively. Previous algorithm in (Kim and Ammar, 2003) can be seen as a special case of our algorithm when λ=1. Fig.4 gives the buffer occupancy comparison between our algorithm and the algorithm in (Kim and Ammar, 2003) when λ=4. In Fig.4, straight line means allocated buffer size for some layer. From the graph, we can see that previous algorithm in (Kim and Ammar, 2003) will lead to buffer overflow while there is no buffer overflow in our algorithm. The reason for buffer overflow in the previous algorithm is that it does not consider dynamic buffer consumption process and the effect of present buffer occupancy on layer selection. Under the same network environment, we conduct online algorithm. Fig.5 gives the layer selection of three layers for online algorithm. In Fig.5, there is almost no quality transition in base layer and FGS layer. Because the residual bandwidth is no enough, a lot of VOPs of FGST layer are discarded. AQT and ARL for online algorithm are 37.7 and 800.8 respectively.

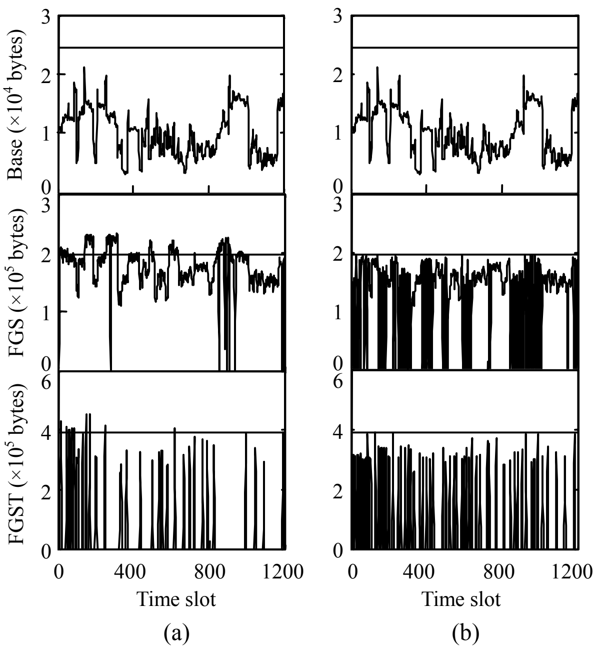

Fig.4 Buffer occupancy of offline algorithm. (a) Offline algorithm in (Kim and Ammar, 2003); (b) Our offline algorithm

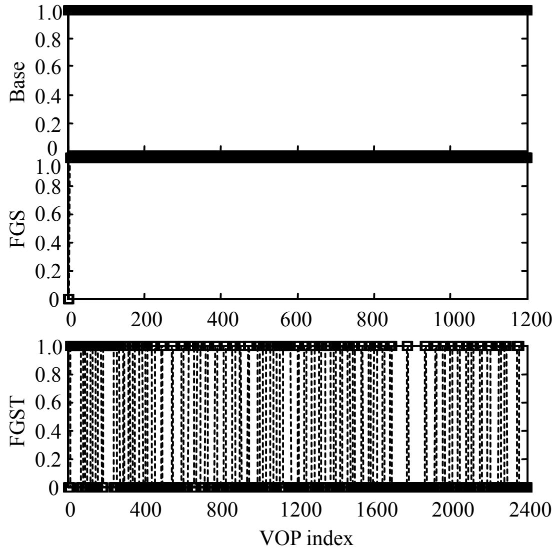

Fig.5 Online selection (Online algorithm)

CONCLUSION

We have investigated the adaptive layer selection of FGS streaming over TCP connection. In this paper, we design offline algorithm for full knowledge of bandwidth evolution and online algorithm without knowledge of bandwidth evolution. Through simulation, we can see that our algorithm can minimize AQT and maximize ARL, so as to achieve smooth video quality. Comparison of our algorithm with previous algorithm revealed that the previous algorithm can be seen as a special case of our algorithm when λ=1. Meanwhile, because we have dynamically considered the effect of buffer consumption on buffer constraint, our algorithm avoids buffer overflow which often happens in the previous algorithm when $\lambda \neq 1$. In the future, we plan to add some other feedback mechanism to improve our algorithm.

References

Kim, T., Ammar, M.H., 2003. Optimal Quality Adaption for MPEG-4 FGS Video. IEEE INFOCOM.

Li, W.P., 2001. MPEG-4 video standard. *IEEE Trans. on Circuits and Systems for Video Technology*, **11**(3):301-317. [doi:10.1109/76.911157]

Nelakuditi, S., Harinath, R., Kusmierek, E., Zhang, Z.L., 2000. Providing Smoother Quality Layered Video. Proceedings of NOSSDAV.