CONTENT-BASED ADAPTIVE MEDIA PLAYER FOR NETWORK VIDEO

Chih-Hao Liang and Chung-Lin Huang
Electrical Engineering Department
National Tsing-Hua Univrsity
Hsin-Chu, Taiwan

ABSTRACT

This paper proposes an Adaptive Media Play-out (AMP) method based on Perceived Motion Energy (PME). Most of current researches of AMP are focusing only on buffer control and packet scheduling. Here, we develop a PME-based threshold adjustment scheme for determining the play-out rate. Because human eyes are more sensitive to high motion videos, we avoid over-slowing the play-out rate in high motion video clips. We adjust the play-out rate based on the PME variation to improve the visual perceptibility.

1. INTRODUCTION

Adaptive media play-out (AMP) is a dynamic delay-based method for receiver-driven play-out rate control. It employs reduced play-out rates if the current number of packets in the play-out buffer falls below a given threshold. For video, the client simply adjusts the duration that each frame is displayed. For audio, the client performs signal processing in conjunction with time scaling to preserve the pitch of the signal. Formal subjective tests show that slowing the play-out rate of video up to 25% is often unnoticeable, and that time-scale modification is preferable subjective to play-out halting or errors due to missing data [1~3].

Let the frame space be t_F and $T_{playout}=t_F$, the AMP play-out time is defined as $T^{AMP}=sT_{playout}$, or $T^{AMP}=fT_{playout}$, with slowdown factor $s\geqq 1$ or speed-up factor $f\leqq 1$. During the burst loss no further packets are received and the buffer fullness decreases, and the AMP decreases the playout speed. After the burst loss is over, the buffer level will begin to increase, after the buffer reaches the target level, the play-out speed return to normal ($s=f=1$). If the buffer level continues to increase, then the play-out speed can be further increased (with $f>1$) to compensate for the additional delay (latency) introduced by the previous slow play-out period.

Liu *et al.*[4] have proposed a perceptual frame dropping algorithm for adaptive video streaming. Dropping frames is simple and efficient, however, it causes motion judder since the dropped frames are replaced by replaying previous frames. Different from the frame dropping, AMP lowers the frame rate and thus introduces latency time. Frame dropping tries to keep up with the schedule of the regular video player with minimal delay.

Yuang [5] proposed a Video Smoother (VS), which consists of an arrival video stream, a finite play-out buffer, an output video stream, and a play-out rate controller. The play-out rate depends on the current number of video frames in the frame buffer and the threshold (*TH*). When the number of frames in the buffer exceeds *TH*, the VS employs a maximum play-out rate; otherwise, it uses proportionally reduced rates to eliminate play-out pauses resulting from buffer emptiness.

Laoutaris *et al.*[6] present another AMP strategy based on M/G/1 queuing system with finite buffer capacity. They proposed a new metric named as Variance of Distortion of Playout (VDOP), which accounts for the overall display disruptions (or gaps) during slowdown periods and data loss from overflows. Kalman *et al.*[7] have proposed a different method based on the rate-distortion optimization concept. They proposed a scheme that combines AMP with the rate-distortion optimized packet transmission scheme. They minimized the function of the subjective cost of play-out rate modifications for the best display schedule.

Because human eyes are more sensitive to high motion objects, we propose a better AMP in consideration of perceptual effect. Based on perceived motion energy (PME)[8] and the number of frames in the buffer, we developed a content-based AMP. The play-out rate is either decreased or increased when the number of frames stored in the buffer is smaller than or greater than a threshold value. The threshold of buffer is also dynamically adjusted based on the buffer quantity and PME value.

2. PERCEIVED MOTION ENERGY

The perceived motion energy (PME)[8] was first applied in frame dropping, which will cause motion judder. The dropped frame is usually replaced by the previous frame. AMP is similar to frame dropping in which it prolongs the playtime of each frame that may cause no annoyance when no motion is presented. However, it fails in fast moving video.

The PME of a frame is introduced to indicate the degree of perceived motion judder if this frame rate is dropped/adjusted. The PME is defined as the product of the average magnitude of motion vectors and the percentage of dominant motion direction. In this paper, we use MPEG video as our test data. The average magnitude of motion vectors is

$$Mag(t) = \sum_{i,j} \frac{MixEn_{i,j}(t)}{N} \qquad (1)$$

Where $MixEn_{i,j}$ represents the motion vector of block (i, j). Then, we define the percentage of dominant motion direction $\alpha(t)$ as

$$\alpha(t) = \frac{\max(AH(t,k), k \in [1,n])}{\sum_{k=1}^{n} AH(t,k)} \qquad (2)$$

The angle 2π is quantized into n ranges. Then the number of angles allocated in each range is accumulated over the whole motion vectors to form an angle histogram with n bins, denoted as $AH(t, k)$, $1 \le k \le n$. The $\max\{AH(t, k)\}$ is the dominant direction bin among all motion directions, and n=8.

Finally, we define $\boldsymbol{PME(t)= Mag(t)\alpha(t)}$. The first item on the right side is the average magnitude of motion vectors within a frame, which is used to indicate that the dropping/slowing-down frames of low motion activity is less perceptible than the frames of high motion activity. The second term represents the percentage of the dominant motion direction. For instance, $\alpha(t)$ will make the contribution of motion from a camera pan more significant to PME, because $\alpha(t)$ will be very large if a camera panning exists. Human's eyes tend to track dominant motion in the scene, and the PME feature is expected to be closely related to the characteristics of human perception.

3. CONTENT-BASED AMP

Here, we describe an analytical model for the study of a content-based AMP with finite buffer. Yuang *et al.*[5] proposed a AMP which employ a maximum play-out rate μ when the number of buffered frame $i \ge TH$, and reduce the play-out rate at $\mu(i)=\mu i/TH$ for $i<TH$. To avoid the maximum play-out rate larger than the arrival rate (*i.e.,* reduce latency time), Laoutaris *et al.* [6] assume a Poisson arrival process with mean arrival rate λ. They propose a similar play-out controller displaying frames at a rate μ=λ when the buffer frames $i \ge TH$. For other cases, it plays the frames at a linearly declining rate $\mu(i)$.

The determination of *TH* can profoundly affect the system performance. If *TH* is overestimated, the play-out rate tends to be reduced which results in serious degradation of play-out performance. On the other hand, if *TH* is underestimated, the probability of having an empty buffer increases which results in play-out discontinuity. The optimal *TH* can then be selected by trading off the rise of the probability of having an empty buffer against the increasing of the play-out rate.

TH is related to the play-out rate, the probability of an empty buffer, the buffer overflow, and the mean display rate. Therefore, *TH* is supposed not to be fixed during the display of an entire video sequence. We assume that $\{F_k\}$ are the frames received in the buffer and the PME of $\{F_k\}$ are provided by the encoder. With these frames in the buffer, we develop a play-out rate controller to determine how to display the stored frames based on the buffer quantity and PME.

The PME is calculated for video sequence with a window length 12 frames, and the windows are overlapped with 6 frames in order to get a more precise estimation. The play-out controller gathers the information of PME and the buffer quantity to calculate the play-out rate and dynamically adjust the threshold of the buffer. The play-out algorithm is illustrated as follows:

[Play-out Algorithm]

```
While (a sequence of frames is going to be play-out) do
 if (PME < K)
  if (buffer quantity < TH)
    play-out rate = linear adjust (TH, i);
  else
   if (TH<TH_upper_bound)
      TH_update;
if (PME > K)
  {if (buffer quantity < TH)
    if (TH > TH_lower_bound)
      TH_update;
    play-out rate = linear adjust (TH, i);
   if (buffer quantity > TH)
    play-out rate = maximum play-out rate;}
```

[End of Play-out Algorithm]

[TH updating Algorithm]

```
while (buffer quantity < TH)
 if (TH_lower_bound < TH < TH_upper_bound)
  {TH_low = TH- 1;
  if ( distortion_low < distortion_original )
   TH = TH_low;}
while (buffer_quantity > TH)
 if ( TH < TH_upper_bound)
  {TH_up = TH+1;
  if (distortion_up < distortion_original)
  TH = TH_up;}
```

[End of TH updating Algorithm]

K is a video content-based variable, which is defined as $K=Mean(PME)–VAR(PME)$. It is important that the threshold *TH* can be dynamically changed to minimize the Distortion function. Each time when the frame rate is being adjusted, we determine the threshold *TH* that leads to the minimum perceptual distortion. The perceptual distortion function is defined as

$$Distortion = w_1 \cdot dis\_PME - w_2 \cdot log(1/\pi_0) \tag{3}$$

where *dis_PME* is the variation of PME value due to the frame rate adjusting of the video player. π_0 is the underflow probability, w_1 and w_2 are the two different weightings for two different costs. Since the play-out rate is linearly adjusted, the frame discontinuity $d(i)$ is the difference between the normal frame duration and extended frame duration which is defined as $d(i)$=max{$(TH-i)/\mu i$, 0} for $1 \leqq i \leqq N$.

We combine the above frame discontinuity equation with the PME measurement by replacing frame rate 1/μ in $d(i)$ with *PME*, and define the *PME discontinuity* measure as follows

$$dis\_PME = \sum_{m=1}^{M} \frac{TH-i}{i} \cdot PME\_value(m) \tag{4}$$

where $1 \leq i \leq TH$, the index m is the video section, and *PME_value(m)* is the perceived motion energy of the *m-th* video section. The *dis_PME* represents the total motion energy that is disturbed by play-out rate adjustment, which is a reliable measurement. In certain cases, we may decrease the threshold to maintain the play-out rate without increasing the buffer underflow probability. It may create the gain of the value of *dis_PME*. Because we have to find the optimal threshold adaptively to determine the play-out rate, the overall underflow probability π_0 cannot be obtained using the formula from[5,6]. Here, we adjust the threshold adaptively, and different selected thresholds may influence the overall statistics of the state transition probability p_{ij} and the underflow probability π_0. The "*current underflow-probability* " is defined as

$$current\_\pi_0 \equiv max(p_{i,j}) \cdot (1-\beta)^j, \quad \forall j < i \tag{5}$$

where $p_{i,j}$ can be defined as

$$p_{i,j} = \begin{cases} \sum_{r \in \{ON,OFF\}} \Pi_r \cdot \sum_{s \in \{ON,OFF\}} a_{r,s}^{m_i}(j-i+1) & if \quad i>0 \; and \; j<N; \\ \sum_{k=1}^{\infty} \sum_{r \in \{ON,OFF\}} \Pi_r \cdot \sum_{s \in \{ON,OFF\}} a_{r,s}^{m_i}(j-i+k) & if \quad i>0 \; and \; j=N; \end{cases} \tag{6}$$

Our definition of the distortion function is similar to [5,6].

When the PME is large and buffer quantity is smaller than *TH*, we need to maintain the play-out rate by decreasing *TH* to a lower value that makes it be closer to buffer quantity. Once the buffer quantity is larger than *TH*, we need to update *TH* to a higher value to avoid the increasing underflow probability (due to lower *TH*). We assume that the incoming frame-rate is not faster than the maximum play-out rate so that the buffer overflow probability is neglect. Our scheme updates *TH* based on current PME value.

4. EXPERIMENTAL RESULTS

The Internet is often simulated as a packet-based transmission channel. When packets of data are delivered over the Internet, usually a packet is either received correctly or lost. These losses are mainly caused by congestion and sever queuing delay. Thus, the packet loss probability can be modeled by using a conventional two-state Markov model[9].

In the two-state Markov model, the packet loss process is modeled as a discrete-time Markov chain with two states. The states are either good or bad, and a packet is assumed correctly received in good state, and is lost or received an error packet in bad state. The current state S_i of the stochastic process depends only on the previous value S_{i-1}. P_{01} and P_{10} are the state transition probabilities from good to bad and from bad to good, respectively. The packet-error statistics will be different according to the values of the transition probability. Average burst length (L_b) is defined as $1/P_{10}$. The average packet error rate is defined as PER=$P_{01}/(P_{01}+P_{10})$. For instance, P_{01}=0.0111 and P_{10}=0.1 correspond to the average burst length of 10 packets and the average packet-error-rate of 0.1.

We use two timers to simulate the input (timer 1) and output (timer 2) of the buffer. The execute frequency of timer 1 is set as λ_{ON}, where λ_{ON} is the mean arrival rate in good state. Then the mean throughput is $\lambda=\lambda_{ON}P_{10}/(P_{01}+P_{10})$. The status of timer 1 is either good or bad, which depends on the channel states. When timer 1 is good, it reads data into the buffer. Similarly, the executed frequency of timer 2 determines the play-out rate.

Here, we compare the performance of our content-based AMP scheme with Steinbach's method[1] which has fixed the threshold *TH*. With the same total rate, we run the simulations for these two different methods on different channel conditions. The test sequences are coded at 128k bits/s, with QCIF format using MPEG-4 FGS codec released by Microsoft, v6.0. To prevent error propagation, I frames are inserted every 100 frames. We run several video sequences including Foreman and apply the two-state Markov model for our simulations.

Once the buffer storage is less than the threshold, the play-out rate is lowered a little bit to maintain the buffer storage. When the buffer storage is larger than *TH*, the play-out rate is back to normal rate. As shown in Figure 1, our play-out rate is higher than the play-out rate of the linear method. Figure 2 shows the VOD (variance of discontinuity) comparison of these two methods. The definition of VOD is as $VOD=E\{(d_i-E[d_i])^2\}$ where d_i is the *i-th* discontinuity duration during the play-out. The VOD function is used as a measurement of smoothness in [5, 6]. The VOD comparison of these two methods is to share on a fifty-fifty basis. Our VOD is smaller than the linear method.

Figure 3 shows the comparison of the latency time. It is the sum of delayed video playing time due to AMP. The result shows that linear adjust method has a latency time of about 2.75 sec, whereas our method only has a latency time of 0.75 sec. It means that our algorithm has a higher mean playout rate (MPR). It is a good result because AMP increases the frame durations and thus increases the length of video play-out time. Lower latency time means better timing control. The distortions (*i.e.,* defined in equation (3)) of these two methods are also compared in Figure 4. We can see clearly that our method also has lower distortion and thus has better visual quality.

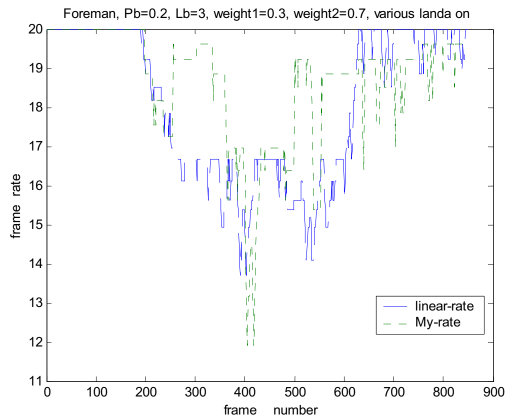

Figure 1. Play-out rates of playing Foreman sequence.

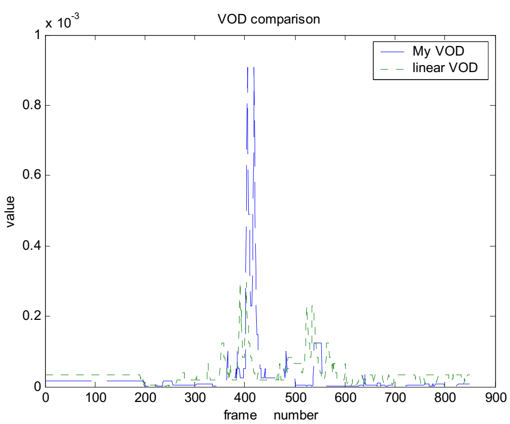

Figure 2. VOD (TH-initial= 30) of playing Foreman.

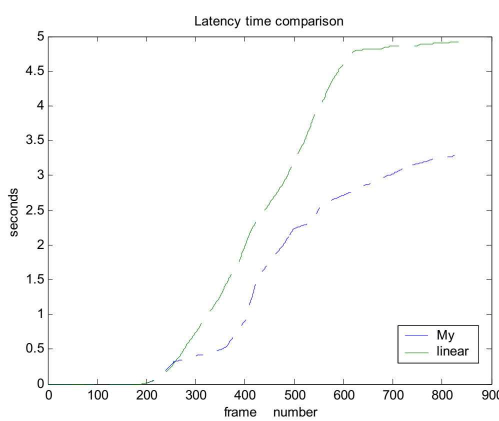

Figure 3. Latency time of playing Foreman.

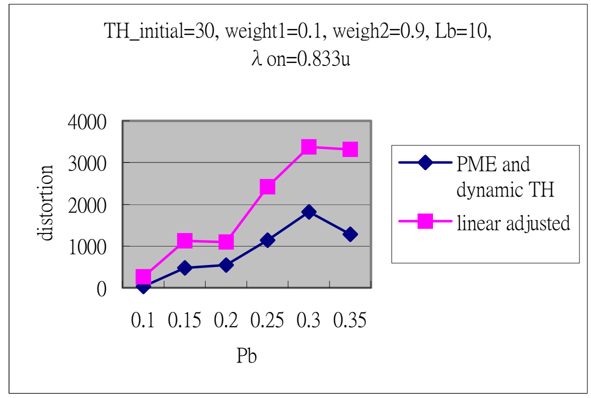

Figure 4. Distortion of playing Foreman sequence.

5. CONCLUSIONS

This paper has proposed a content-based AMP for network video. Performance comparison is made between our scheme and the conventional scheme for displaying the video received over network channel. Experimental results show that our scheme outperforms the conventional scheme in VOD with the same receiving frame-rate and channel conditions. Our scheme is a suitable receiver-driven AMP method for the packet video player.

REFERENCE

[1] E. Steinbach, N. Farber, and B. Girod, “Adaptive Play-out for Low Latency Video Streaming,” ICIP 2001, pp. 962-965, Thessaloniki, Greece, Oct. 2001.

[2] Y. J. Liang, N. Farber, and B. Girod, “Adaptive Playout Scheduling Using Timer-scale Modification in Packet Voice Communication,” Proc. ICASSP ’01, Salt Lake City, May 2001.

[3] M Kalman, B. Girod, and E. Steinbach, “Adaptive Playout for Real-time Media Streaming,” ISCAS 2002, Scottsdale Arizona, May 2002.

[4] T. Liu, H.-J. Zhang, F. Qi, “Perceptual Frame Dropping in Adaptive Video Streaming,” ISCAS2002, Scottsdale Arizona, May 2002.

[5] M. C. Yuang, S. T. Liang and Y. G. Chen, “Dynamic Video Playout Smoothing Method for Multimedia Applications,” Multimedia Tools and Applications, vol.6, 47-60, 1998.

[6] N. Laoutaris and I. Stavrakakis, “Adaptive Playout Strategies for Packet Video Receivers with Finite Buffer Capacity.” IEEE ICC2001,Helsinki, Finland, June 2001.

[7] M. Kalman, E. Steinback, and B. Girod, “Rate-Distortion Optimized video Streaming with Adaptive Playout,” IEEE ICIP 2002.

[8] Yu-Fei Ma and Hong-Jiang Zhang, “A New Perceived Motion Based Shot Content Representation.” ICIP 2001, Thessaloniki, Greece, October 7-10, 2001.

[9] M. Yajnik, S. Moon, and J. Kurose, “Measurement and Modeling of the Temporal Dependence in Packet Loss,” INFOCOM, pp. 78-98, 1999.