

# Energy-aware QoS Adaptation for Streaming Video based on MPEG-7

Morihiko Tamai<sup>†</sup>, Tao Sun<sup>†</sup>, Keiichi Yasumoto<sup>†</sup>, Naoki Shibata<sup>‡</sup> and Minoru Ito<sup>†</sup>

<sup>†</sup> Graduate School of Information Science, Nara Institute of Science and Technology  
{morihiko-t,song-t,yasumoto,ito}@is.naist.jp

<sup>‡</sup> Department of Information Processing and Management, Shiga University  
shibata@biwako.shiga-u.ac.jp

## Abstract

*In this paper, we propose a QoS adaptation method for streaming video playback for portable computing devices where playback quality of each video fragment is automatically adjusted from the remaining battery amount, desirable playback duration and the user's preference to each fragment. In our method, we assume that video segments (or shots) are classified into some predefined categories. Each user specifies relative importance among categories and preferred video property such as proportion between motion speed and vividness for each category. From the information, playback quality and property of each category are determined so that the video playback can last for the specified duration within the battery amount. We have implemented a video streaming system consisting of a transcoder for PCs and a video player for PDAs.*

## 1. Introduction

Due to widespread of CATV, satellite broadcasting and digital terrestrial broadcasting in recent years, various video contents are becoming available. Moreover, video recorders with re-writable DVD and HDD are becoming popular. Some video recorders on the market can convert recorded videos to MPEG-4 files, and transmit them to users via the Internet or copy them to memory cards[5]. With such a product, we can now watch a recorded video at any place using a portable computing device such as PDAs and cellular phones via wireless LAN, PHS or wideband CDMA.

However, portable computing devices do not have sufficient battery for watching a video with a long duration. It is desirable that the battery lasts for the specified duration when watching cinemas, soccer games, etc which have fixed durations. Moreover, in video playback at portable devices, some fragments of a video important for a user should be played back with higher quality than others. Also, to each fragment, a user should be able to specify playback property such as balance of motion speed and vividness.

In this paper, we propose a QoS adaptation method for streaming video playback for portable computing devices where playback quality of each video fragment is automatically adjusted from the remaining battery amount, desirable playback duration and the user's preference to each

fragment. In our method, we assume that video segments (shots) of a video are classified into some predefined categories in advance. For example, video segments of a soccer game can be classified into categories: *shoot, normal-play, set-play, audience, other*, etc. These categories are described as meta information in MPEG-7 format. Classification can be done manually using annotation tools like [3], or done automatically using tools like [4]. Next, a user specifies priorities among categories. For each category, the user also specifies relative importance among playback parameters such as motion speed, vividness and sound.

From the information above, playback quality/property for each category are determined so that the video playback can last for the specified duration within the battery amount. In our previous work [6], we have proposed a method to determine fixed playback parameter values where a video can be played back for the specified duration with the fixed quality. In this paper, we enhance this algorithm so that the battery amount can be allocated to categories according to the specified priority and the playback property of each category is determined based on the specified preference.

We have implemented a video streaming system consisting of a transcoder which converts a video stream from a contents server to a new stream with any specified parameters, and a video player which can be executed on PDAs. From some experiments using our system, we have confirmed that the playback quality of important categories can be improved a few times better than flattening the playback quality over the playback duration.

### 1.1. Related Works

In previous transcoding techniques which simply reduce the picture size, objects in each picture frame becomes too small and difficult to identify. [2] copes with this problem by specifying the user's interesting area in the picture with the MPEG-21 DIA framework so that only the area is trimmed off and transcoded. In [1], a video in MPEG-4 format is divided into objects of several categories such as foreground objects and background objects. Here, playback qualities of important objects are kept high while qualities of other objects are lowered.

The objectives of these existing researches are to satisfy restrictions of portable devices w.r.t. picture size and available bandwidth. However, we believe that the restric-

tion w.r.t. the battery amount and the playback property of each fragment are also important. These points are new in our approach.

## 2. Describing Meta Data and Priorities

MPEG-7 has been standardized by ISO/IEC as a description method of meta information for audiovisual data contents in multimedia environments. In MPEG-7, meta data can be specified to any fragment of a video in order to facilitate users to search a specified fragment by its “feature data”.

### 2.1. Specifying feature data to each video segment

We use a keyword called *category* as feature data, and denote a set of categories by  $C = \{c_1, \dots, c_n\}$ . Here, a category  $c_i$  is specified by a string. For example, for the video of a soccer game, we may use a set of categories  $C = \{\text{shoot}, \text{play}, \text{audience}, \text{other}\}$ . A fragment in a video taken by the same camera work is called a *shot* or *segment*. In this paper, we suppose that a category  $c_i \in C$  is assigned to each segment.

In general, MPEG files do not contain the boundary information of each segment. The tool named VideoAnnEx (IBM MPEG-7 Annotation Tool) [3] can read a MPEG1 file, identify each video segment automatically, assign a string to each segment, and output an MPEG-7 file as shown below.

```
<VideoSegment>
  <TextAnnotation>
    <FreeTextAnnotation> shoot
  </FreeTextAnnotation>
</TextAnnotation>
<MediaTime>
  <MediaTimePoint> T00:00:00:0F25
</MediaTimePoint>
  <MediaIncrDuration mediaTimeUnit
    = "PT1N25F"> 78
</MediaIncrDuration>
</MediaTime>
</VideoSegment>
```

In the above file, string *shoot* is specified to a video segment as a category using tag `<TextAnnotation>`. Tag `<MediaTime>` describes the starting time and the duration of this segment.

### 2.2. Specifying importance among categories

It is desirable for users to be able to specify what part of a video will be played back with higher quality. So, we allow users to specify relative importance among categories as priority values. Let  $p_i$  denote the priority specified to category  $c_i$  where  $p_i$  is an integer number such that  $p_i \geq 1$ .

The *playback property* of a video is decided by the balance of its picture size, frame rate and bitrate. In general, users have different preferences for the playback property of each category. Also, there may be various properties which consume the same electric power. So, we allow users to specify a preference to the property of each category by the proportion of relative importance among three factors: motion speed, vividness and sound. We denote these factors by  $spd_i$ ,  $vid_i$  and  $snd_i$  for category  $c_i$ .

For example, in a video of a soccer game, suppose that sound is not very important in all categories, that both the

motion speed and the vividness are very important in category *shoot*, that only the motion speed is somewhat important in category *play*, and that only the vividness is somewhat important in categories *audience* and *other*. In such a case, users give the following preference.

category	spd	vid	snd
<i>shoot</i>	3	3	1
<i>play</i>	2	1	1
<i>audience</i>	1	2	1
<i>other</i>	1	2	1

## 3. Algorithm for Deciding Playback Quality

### 3.1. Battery distribution among categories

Let us denote battery amount of a portable computing device by  $E_0$ , and the desirable playback duration of a video by  $T$ . We denote by  $w_0$  the power consumed while no video is played back (i.e., the power consumed by the operating system, the back-light for LCD, and so on). Thus, the battery amount which can be used for playback of a video with duration  $T$  is denoted by  $E = E_0 - w_0T$ . Here, we can easily measure the actual value of  $w_0$  for any device.

For each category  $c_i \in C$ , the product of its importance and playback duration is called the *virtual playback time* of  $c_i$ . We denote it by  $T'_i (= p_i T_i)$ . Also, the total sum of the virtual time of all categories is denoted by  $T' (= \sum_{c_i \in C} T'_i)$ .

In our algorithm, we distribute the remaining battery amount  $E$  among categories according to the proportion of the virtual time  $T'_i/T'$  of each category. That is,  $E_i (= ET'_i/T')$  is allocated for playback of each category  $c_i$ .

The property of each video is represented by picture size  $r$ , frame rate  $f$  and bitrate  $b$ . We denote it by  $(r, f, b)$ . We denote the properties of videos with the maximum quality and with the minimum quality by  $(r_{max}, f_{max}, b_{max})$  and  $(r_{min}, f_{min}, b_{min})$ , respectively. Here, the video with the maximum quality might be the one with satisfactory quality or the maximum one which the device can play back without changing its property. The video with the minimum quality can similarly be defined.

In [6], we have confirmed that the battery amount  $E$  consumed by video playback on PDAs is approximately proportional to the product of picture size  $r$ , frame rate  $f$ , bitrate  $b$  and playback duration  $T$ . That is,  $E = \alpha rfbT$ . Here,  $\alpha$  is a device specific constant and can be measured for any device using our technique in [6].

Due to this fact, if  $E_i > \alpha r_{max} f_{max} b_{max} T_i$ ,  $E_i$  is too much for playback of video segments in  $c_i$ . Similarly, if  $E_i < \alpha r_{min} f_{min} b_{min} T_i$ ,  $E_i$  is too small for playing back segments in  $c_i$ . In either case, we fix  $E_i = \alpha r_{max} f_{max} b_{max} T_i$  or  $E_i = \alpha r_{min} f_{min} b_{min} T_i$ , and distribute the remaining battery amount  $E' (= E - E_i)$  among remaining categories  $C - \{c_i\}$ . Consequently, we can obtain battery amount  $E_i$  for playback of category  $c_i$  as a constant value.

### 3.2. Decision of each category's playback property

We would like to decide the playback property of each category  $c_i$  as picture size (i.e., number of pixels)  $r_i$ , frame

rate  $f_i$  and bitrate  $b_i$  from battery  $E_i$  assigned for  $c_i$ , playback duration  $T_i$  and the user preference ( $spd_i, vid_i, snd_i$ ) for playback property of  $c_i$ .

Here, it is considered that motion speed  $spd_i$  and vividness  $vid_i$  influence the picture size and the frame rate, respectively. On the other hand, bitrate  $b_i$  is influenced from all of  $spd_i, vid_i$  and  $snd_i$ . Here, we assume that the proportion of  $b_i$  will be  $(vid_i + spd_i + snd_i)/3$ . If we do not use sound (i.e.,  $snd_i = 0$ ), the proportion will be  $(vid_i + spd_i)/2$ . For the sake of simplicity, we suppose  $snd_i = 0$ , hereafter.

**When playing back from storage** From user preference ( $spd_i, vid_i$ ), we would like to decide playback property ( $r_i, f_i, b_i$ ) such that  $E_i = \alpha r_i f_i b_i T_i$ . Since we cannot directly compare the ratio between the picture size, the frame rate and the bitrate, we use the ratio of each video parameter to the corresponding one of an original video ( $r_0, f_0, b_0$ ) as follows.

$$\frac{r_i}{r_0} : \frac{f_i}{f_0} : \frac{b_i}{b_0} = vid_i : spd_i : \frac{vid_i + spd_i}{2}$$

From the above equation, we can derive  $f_i = \frac{spd_i f_0}{vid_i r_0} r_i$  and  $b_i = \frac{(spd_i + vid_i) b_0}{2 vid_i r_0} r_i$ . When we assign these equations to formula  $E_i = \alpha r_i f_i b_i T_i$ ,

$$E_i = \alpha \frac{spd_i (spd_i + vid_i) f_0 b_0}{2 vid_i^2 r_0^2} T_i r_i^3$$

is derived. Then, we can calculate the value of  $r_i$  as follows.

$$r_i = \sqrt[3]{\frac{2 E_i vid_i^2 r_0^2}{\alpha T_i spd_i (spd_i + vid_i) f_0 b_0}}$$

Similarly, the values of  $f_i$  and  $b_i$  can be obtained as follows.

$$f_i = \sqrt[3]{\frac{2 E_i spd_i^2 r_0^2}{\alpha T_i vid_i (spd_i + vid_i) r_0 b_0}} \quad b_i = \sqrt[3]{\frac{E_i (vid_i + spd_i)^2 b_0^2}{4 \alpha T_i spd_i vid_i f_0 r_0}}$$

#### When playing back streaming video via wireless LAN

When playing back a video with bitrate  $b$  bps using IEEE 802.11b, the power consumed for communication by a portable computing device and a WNIC (wireless network interface card) can be approximated by the linear expression of bitrate  $b$  [6]. That is,  $\beta + \gamma b$ . Here,  $\beta$  and  $\gamma$  are device specific constants and can be measured for any portable devices and WNIC.

Our preliminary experiments using IEEE 802.11b have shown that  $\beta$  is much larger than  $\gamma b$ . Owing to this fact, when available bandwidth is larger than  $b$  bps, we can vastly reduce battery consumption [6] by dividing a video (whose bitrate is  $b$  bps) to fragments with  $M$  bit, transmitting each fragment at  $B$  bps ( $B > b$ ) every  $M/b$  seconds so that the portable device stores each received fragment in a local buffer, turns off its WNIC until the next transmission period comes, and plays back the fragment from the buffer. We call this scheme *buffered playback*. In the buffered playback, when transmitting each fragment at  $k (= B/b)$  times of original bitrate  $b$ , the portable device can receive it in

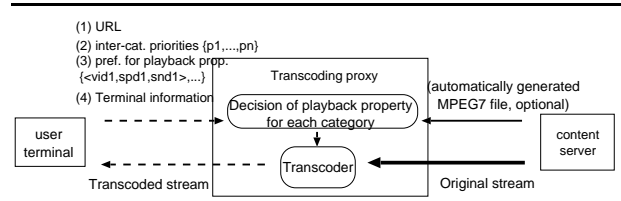


Figure 1. Energy-aware streaming system

$1/k$  of the originally required time. So, the power supply to WNIC can be stopped during most of the playback time. However, actually, it takes a few seconds (denoted by  $t_{on/off}$ ) to stop/resume WNIC during which some power (denoted by  $\tau$ ) is consumed.

In a video, there are some video segments (the number of segments is denoted by  $seg_i$ ) which belong to category  $c_i$ . Total size of video segments in  $c_i$  is  $b_i T_i$ . When we divide it to  $M$  bit fragments, the total number of transmissions can be denoted by  $\frac{b_i T_i}{M} + seg_i$  in the worst case. Practically, we can omit “ $+seg_i$ ” from the expression.

Consequently, battery consumption for playing back  $c_i$  when using buffered playback, is represented by

$$E_i = \alpha r_i f_i b_i T_i + (\beta + \gamma B) \frac{b_i T_i}{B} + \frac{b_i T_i}{M} \tau t_{on/off}$$

By assigning equations  $f_i = \frac{spd_i f_0}{vid_i r_0} r_i$  and  $b_i = \frac{(spd_i + vid_i) b_0}{2 vid_i r_0} r_i$ , we can get the following equation of  $r_i$ .

$$E_i = \alpha \frac{spd_i (spd_i + vid_i) f_0 b_0}{vid_i^2 r_0^2} T_i r_i^3 + \left( \frac{\beta + \gamma B}{B} + \frac{\tau t_{on/off}}{M} \right) \frac{(spd_i + vid_i) b_0}{2 vid_i r_0} T_i r_i$$

We can obtain the value of  $r_i$  from the above equation, for example, using Newton's method.

If either of the calculated values of  $r_i, f_i$  and  $b_i$  is larger/smaller than the maximum/minimum threshold, we can fix the parameter value, and re-calculate the values of the other parameters. For example, the value of  $r_i$  is larger than portable device's screen size  $r_{max}$ , we fix  $r_i$  to  $r_{max}$  and re-calculate  $f_i$  and  $b_i$  using the algorithm recursively.

## 4. Streaming System

We have implemented a video streaming system consisting of a movie player and a transcoding proxy as shown in Fig.1. The transcoding proxy is supposed to be executed at a contents server or at an intermediate node on the network. Each user sends (1) a video's URL with the desirable playback duration, (2) priorities among categories, (3) a preference to the playback property for each category and (4) the device specific information (values of  $E, \alpha, \beta, \gamma$ , etc) to the transcoding proxy. The transcoding proxy transcodes a video stream transmitted from a contents server to a new stream with the playback quality and property calculated by the algorithm described in Sect. 3, and relays the stream to the portable computing device.

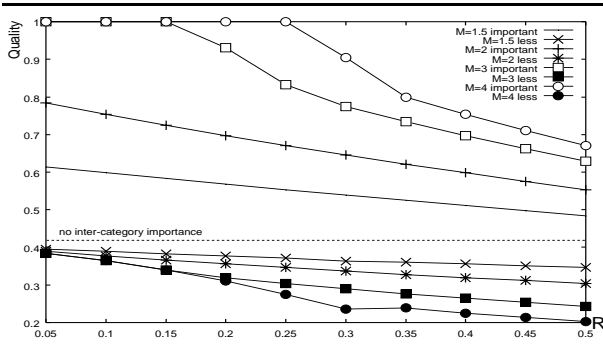


Figure 2. Quality improvement

## 5. Experimental Results and Evaluation

**Playback quality in important categories** Using the algorithm in Sect. 3, we have investigated to what extent the playback quality of important categories is improved and the quality of the other categories is degraded.

In the experiment, we assume that video segments in a video are classified into two categories: important category  $c_1$  and less-important category  $c_2$ . Let  $R$  denote the ratio of playback duration  $T_1$  of  $c_1$  to total playback duration  $T_1 + T_2$  (i.e.,  $R \stackrel{\text{def}}{=} T_1 / (T_1 + T_2)$ ). Let  $p_1$  and  $p_2$  denote the priorities for  $c_1$  and  $c_2$ , respectively. Let  $M$  denote the ratio of  $p_1$  to  $p_2$  (i.e.,  $M \stackrel{\text{def}}{=} p_1 / p_2$ ). We have observed variation of the playback qualities of video segments in  $c_1$  and  $c_2$  by changing  $R$  from 0.05 to 0.5 by 0.05 step and  $M$  in 1.5, 2, 3 and 4. The resulting graphs are depicted in Fig. 2, where the horizontal axis and the vertical axis represent  $R$  and playback quality  $Q$ , respectively. Since quality  $Q$  is defined as  $\sqrt[3]{r_i f_i b_i / r_0 f_0 b_0}$ ,  $Q$  varies between 0 and 1, where  $(r_0, f_0, b_0)$  is the property of an original video before transcoding. Since we mainly focus on the use of PDAs, we set  $(r_0, f_0, b_0)$  to  $(320 \times 240, 30 \text{ fps}, 700 \text{ Kbps})$  in this experiment.  $Q$  becomes 0.41, if all categories have the same priorities, that is,  $p_1 = p_2$ .

Fig. 2 shows that while  $R$  is less than 0.2, the playback quality in important categories can be improved significantly by a small reduction of the playback quality of less-important categories. Even when  $R$  is high (around 0.4), we can improve the quality of important categories much with about 20 % quality degradation in less-important categories, by controlling  $M$  under 2.

**Ratio of prediction error** We have measured actual playback durations of a video within the remaining battery using preferences in Table 1. In the experiment, a PDA (SHARP, ZAURUS SL-C700) with an IEEE 802.11b WLAN card (WN-B11/CF, I-O DATA Device, inc.) has been used.

For *pref1* and *pref2*, video segments are played back with the playback qualities of their categories shown in Table 1. In general, the battery life (time until the battery is exhausted during video playback) may be differ from the specified playback duration due to inaccurate information of the remaining battery amount, available bandwidth, and so on. For *pref1* and *pref2*, the specified playback duration is 180

	cat.	$(T_i, p_i, spd_i, vid_i)$	$(r, f, b)$
pref1	$c_1$	(59, 1, 1, 1)	$(178 \times 132, 9.2, 214K)$
	$c_2$	(69, 2, 1, 2)	$(260 \times 196, 9.9, 346K)$
	$c_3$	(52, 4, 2, 3)	$(320 \times 240, 20.4, 400K)$
pref2	$c_1$	(59, 1, 1, 1)	$(178 \times 132, 9.2, 214K)$
	$c_2$	(69, 2, 2, 1)	$(184 \times 136, 19.8, 347K)$
	$c_3$	(52, 4, 3, 2)	$(294 \times 224, 24.0, 400K)$

Table 1. Preferences and playback qualities

minutes, and the battery lives were 175 minutes and 171 minutes, respectively. In this case, the prediction errors are less than 5%. This result is close to our previous result when playing back videos with the fixed quality [6].

**Evaluation** We have evaluated the impact of the proposed method by means of questionnaire. In the evaluation, we used a soccer video with 180 minutes and let four testers watch the video with dynamic QoS adaptation using *pref1* in Table 1 and that with the fixed quality (picture size of  $230 \times 172$ , 15.51 *fps*, 362Kbps) obtained by using the same importance among categories.

As a result, all of testers preferred the playback quality in important categories using the proposed method to the fixed playback quality. Some of testers preferred larger picture size to larger frame rate. There are different opinions on the playback quality in less-important categories. Some said that the picture size is too small and the motion speed is too clumsy, others said no problem. Also, there is a comment that the sudden picture size change is a bit unnatural.

## 6. Conclusions

In this paper, we proposed an energy-aware QoS adaptation method for streaming video playback for portable computing devices, based on MPEG-7 meta information and priorities among segments in a video. We confirmed that on portable devices with limited battery amount, the user's feeling of satisfaction can be improved to some extent compared with flattening playback quality over the playback duration.

## References

- [1] Cavallaro, A., Steiger, O. and Ebrahimi, T.: Semantic Segmentation and Description for Video Transcoding, *Proc. of the 2003 IEEE Int'l. Conf. on Multimedia and Expo. (ICME2003)*, Vol. 3, pp. 597–600 (2003).
- [2] Lim, J., Kim, M., Kim, J and Kim, K.: Semantic Transcoding of Video based on Regions of Interest, *Proc. of Visual Communications and Image Processing 2003 (VCIP2003)* (2003).
- [3] Lin, C.-Y., Tseng, B.L. and Smith, J. R.: VideoAnnEx: IBM MPEG-7 Annotation Tool, <http://www.alphaworks.ibm.com/tech/videoannex>
- [4] Lin, C.-Y., Tseng, B. L., Naphade, M., Natsev, A. and Smith, J. R.: MPEG-7 Video Automatic Labeling System, *Proc. of the 11th ACM Int'l. Conf. on Multimedia*, pp. 98–99 (2003).
- [5] Sharp Corp: Personal Server HG-01S, <http://sharp-world.com/corporate/news/030117.html>
- [6] Tamai, M., Yasumoto, K., Shibata, N. and Ito, M.: Low Power Video Streaming for PDAs, *Proc. of the 8th IEEE Int'l. Workshop on Mobile Multimedia Communications (MoMuC2003)*, pp. 31–36 (2003).