Energy-aware Video Streaming with QoS Control for Portable Computing Devices

Morihiko Tamai, Tao Sun, Keiichi Yasumoto, Naoki Shibata[†] and Minoru Ito

Grad. Sch. of Info. Sci., Nara Inst. Sci. and Tech.
Ikoma, Nara 630-0192, Japan
{morihi-t,song-t,yasumoto,ito}@is.naist.jp

[†] Dept. of Info. Proc. and Man., Shiga Univ.
Hikone, Shiga 522-8522, Japan
shibata@biwako.shiga-u.ac.jp

ABSTRACT

We propose an energy-aware video streaming system for portable computing devices, in which the video can be played back for the specified duration within the remaining battery amount. To save power, we introduce techniques (i) to reduce playback quality of a video at an intermediate proxy and (ii) to shorten working time of the network I/F card using periodic bulk transfer of the video data on the wireless LAN. To enable playback for the specified duration, we have developed a power consumption model for portable devices using parameters on playback quality, playback duration, battery amount, and so on. We have also developed an algorithm to assign different playback quality among multiple video segments based on the user's preference. Our experiments using PDAs and laptop PCs on 802.11b WLAN show that our system achieves less than 6 % prediction error in playback duration while adapting playback quality among video segments.

Categories and Subject Descriptors: C.3 [Computer Systems Organization]: Special-purpose and application-based systems

General Terms: Experimentation

Keywords: Energy-aware Systems, Video Streaming, QoS, Wireless LANs

1. INTRODUCTION

Owing to recent innovations in portable computing devices (such as PDAs and smart phones) and wireless communication infrastructure (such as WLAN and wideband CDMA), we can watch video contents using those devices every time and everywhere. In such an on-demand video streaming and playback, however, the portable computing device consumes much more battery amount than other business applications, due mainly to video decoding and wireless communication. Consequently, there will be a demand to control battery life depending on user requirements or situations so that the battery is not exhausted until the video playback finishes or the specified amount of battery (e.g., 50%) is left after playing back the video content.

In general, we can extend video playback duration by reducing video quality. However, it is difficult to estimate how long the battery lasts for each playback quality, since the battery consumption depends on not only video playback but also other factors such as OS, LCD, and so on. Besides it, these factors are device dependent. Moreover, when the remaining battery amount is small and the playback quality is reduced over the whole playback duration, users may get frustrated. In order to mitigate such a situation, 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 preferences such as balance of motion speed and vividness.

In this paper, we propose a method for energy-aware video streaming and playback which consists of (1) power-saving techniques for streaming video playback in portable computing devices on the wireless LAN, (2) a technique to estimate the suitable parameter values (such as picture size, frame rate and bitrate) which enable playback for the specified duration within the remaining battery amount, and (3) a QoS control mechanism which enables playback of multiple video segments with different playback quality based on the user's preference.

For (1), we introduce two techniques: (i) reduction of video quality with a transcoder; and (ii) periodic bulk transfer of the video data on the wireless LAN. For (i), we execute a proxy on an intermediate node (or on the content server) in the network so that the proxy receives the stream and forwards the transcoded stream to the user terminal. For (ii), the user terminal receives each fragment of the stream data at transmission rate as high as possible, stores the data in the local buffer, and stops supplying power to its WNIC (wireless LAN I/F card) until the buffer becomes empty.

For (2), we have developed a power consumption model for the user terminal, in which the parameter values can automatically be decided from the desired playback duration, the remaining battery amount, and device specific information.

For (3), we have developed an algorithm to distribute the battery amount among multiple video segments based on each user's preference. When a user specifies relative importance among video segments and preferred video property (proportion among picture size, frame rate and bitrate) for each segment, our algorithm determines the playback quality of each video segment so that the video playback can last for the specified duration within the remaining battery amount.

We have implemented the energy-aware video streaming system based on the proposed method. From experiments using a PDA and a laptop PC on IEEE 802.11b WLAN, we have confirmed that the proposed power saving method is effective, and that the estimation of playback quality based on the proposed model is accurate enough. Moreover, we have confirmed that the proposed QoS control method improves the playback quality of important video segments much better than flattening the quality over the playback duration.

In the following Sect. 2, we briefly survey related work. Sect. 3 presents the power saving techniques in playback of streaming video and the power consumption model to derive parameter values to satisfy the playback duration and the battery amount. In Sect. 4, we introduce our energy-aware QoS control technique based on relative importance among video segments. In Sect. 5, we show evaluation of the proposed method through experiments and analysis. Finally, we conclude the paper in Sect. 6.

2. RELATED WORK

Power consumption when playing back streaming video on portable computing devices via WLAN consists mainly of (i) decoding and drawing video frames and (ii) packet transmission on WLAN.

For (i), we can use a transcoding technique. For example, [1] proposes a power saving technique which reduces the total data size of the video by selectively dropping I, P and B frames when streaming MPEG videos. For (ii), there are several approaches. [9] proposes a power saving technique by regulating power for radio wave output in IEEE 802.11b WLAN. [8] proposes another power saving technique which reduces power consumption by shortening time for communication using data compression.

These existing researches focus mainly on power saving techniques, but do not propose battery control depending on user requirements such as the playback duration within the battery amount.

In recent years, semantic transcoding techniques are paid much attention. Here, the user's satisfaction is improved by transcoding a video according to the contents and semantics of each fragment of a video[2, 4]. In previous transcoding techniques which simply reduce the picture size, objects in each picture frames becomes too small and difficult to identify. [4] 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 [2], 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 maintained while qualities of other objects are degraded.

The objectives of these existing researches are to satisfy restrictions of portable devices w.r.t. picture size and available bandwidth. However, they do not treat restrictions on the battery amount.

3. METHOD FOR ENERGY-AWARE VIDEO STREAMING AND PLAYBACK

In this section, we present two power saving techniques for playback of streaming videos on portable computing devices, and then introduce the power consumption model to determine the parameter values which enable playback for the specified duration within the battery amount.

3.1 Power Saving Techniques

When playing back a streaming video on a portable computing device (called the *user terminal*, hereafter), the consumed power consists of (i) power consumed by wireless transmission of packets, (ii) power consumed by decoding and drawing video frames, and (iii) power consumed by other factors such as operating systems, LCD back-light and so on

When playing back an MPEG1 stream (288×216 pixels, 24 frames per second, 327 Kbps bitrate) on a PDA via IEEE 802.11b WLAN (the device names are shown in Table 2), the power consumed by (i) and (ii) was more than 78 % of the whole power. So, it will be effective to save the power for (i) and (ii).

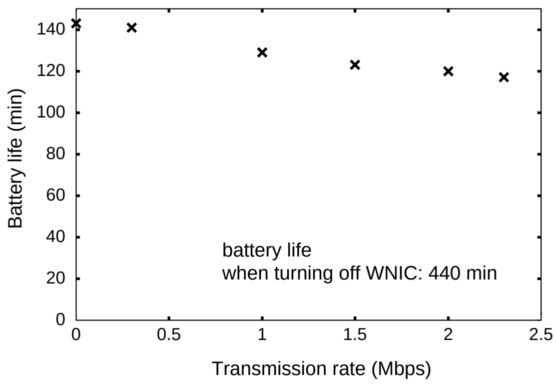

Figure 1: Battery Life vs. Transmission Rate on WLAN

Power saving for decoding/drawing video frames

Since power consumed by video playback depends on the picture size, the frame rate and the bitrate, it is possible to save power by reducing the values of those parameters. In the proposed system, we adopt to use a transcoder called the *transcoding proxy* which receives the video stream from the content server, converts the stream to the stream with the lower parameter values in real time and forwards the new stream to the user terminal (see Fig. 3). The transcoding proxy can be executed on any node (including the content server) reachable from the user terminal in the network.

Power saving for wireless communication

We have investigated how the battery life changes depending on the transmission rate on the IEEE 802.11b WLAN. In the experiment, we used the PDA and the WNIC in Table 2. The result is shown in Fig. 1. Fig. 1 suggests us that power consumed by wireless communication increases in proportional to the transmission rate, but the minimum power consumed by the WNIC is much larger than the difference by transmission rate. So, if we can shorten the time to supply power to the WNIC, we can save the power.

In general, when playing back a video stream whose bitrate is b, the user terminal should receive the stream at transmission rate b and its WNIC should be always turned on during the playback. From the result in Fig. 1, we adopt the following periodic bulk transfer to save the power: the user terminal (1) receives each fragment of the stream data at the maximum available bandwidth more than b, (2) stores it in the local buffer and (3) turns off the WNIC while playing back the data in the buffer until the buffer becomes empty. We call this scheme the *buffered playback*.

For push type video sources which transmit video streams at their bitrates, say, b, the buffered playback cannot be applied as is even when the available bandwidth is much larger than b. So, we let the transcoding proxy to store the transcoded data in its buffer until the buffer is filled, and transmit the data at the available transmission rate (denoted by b_{max}) to the user terminal. In this case, playback of the video is delayed for $(M/b_1) - (M/b_{max})$ which means the time until the buffer with M bit is filled. If the user specifies the maximum tolerable delay, buffer size M can be determined.

3.2 Power Consumption Model

We should be able to estimate the parameter values satisfying the specified playback duration and the remaining battery amount, for any combination of portable computing devices and WNICs. So, we construct an equation which represents the relation among the above parameters and device specific constants for user terminals.

Let us denote the powers consumed by playing back video and receiving video data by P_v and P_N, respectively. Let S denote the power consumed by other factors (operating system, back-light, memory and so on). Here, S is the device specific constant. Thus, power P consumed by the user terminal is represented by the following equation.

$$P = S + P_v + P_N \tag{1}$$

3.2.1 *Power consumed by video processing*

Playback of MPEG-1 videos needs two operations: decoding and drawing picture frames. The power consumed by drawing each frame is proportional to the number of pixels. The decoding operation consists of Huffman decoding, dequantization and inverse discrete cosine transformation. The power consumed by Huffman decoding is proportional to the bitrate of the video, and the power consumed by other processes is proportional to the number of pixels. Accordingly, the total power P_v to play back a video is represented by the following equation.

$$P_v(r, f, b) = \alpha r f + \beta b$$

where r, f and b denote the picture size (number of pixels), the frame rate and the bitrate, respectively, and α and β are device specific constants. Finally, we obtain a power consumption model for playback of a video

$$E_0 = (S + \alpha r f + \beta b)T \tag{3}$$

where T and E_0 are the expected battery life and the remaining battery amount.

Here, the actual values of S, α and β can be calculated by measuring the battery lives when playing back n videos with different parameters $\{(r_1, f_1, b_1), \ldots, (r_n, f_n, b_n)\}$, and using multiple regression. Since we have 3 variables, we should use $n \geq 3$.

3.2.2 *Power consumed by communication*

From the result in Fig. 1, we assume that power P_N consumed by communication is represented by

$$P_N(b) = N + \gamma b$$

where γ and N are device specific constants. From equation (1), we can derive

$$E_0 = (S + N + \gamma b)T \tag{5}$$

representing a relation among battery life T, battery amount E_0 and transmission rate b when the user terminal receives the video stream at b (without decoding and drawing video frames). The values of N and γ can be calculated by measuring battery lives when data are received at different transmission rates $b_1, ..., b_m$, and using multiple regression.

3.2.3 *Power consumed by buffered playback*

Let b_{max} denote the maximum available bandwidth at the user terminal. Let T_{on} denote the time interval to fill up the buffer since the buffer is empty. During this time interval, the WNIC is turned on. Let T_{off} denote the time interval to consume all of the data in the buffer since the buffer is full. During this interval, the WNIC is turned off. Let τ_{on} and τ_{off} denote the time intervals to resume the WNIC and to suspend it respectively. Let τ denote $\tau_{on} + \tau_{off}$. τ is a device specific constant and can be easily obtained (the actual value of τ in our environment was about 3sec).

Fig. 2 depicts the time chart of how the stream data is filled and consumed in the buffer when using the buffered playback, where M and b denote the buffer size and the bitrate of the video, respectively.

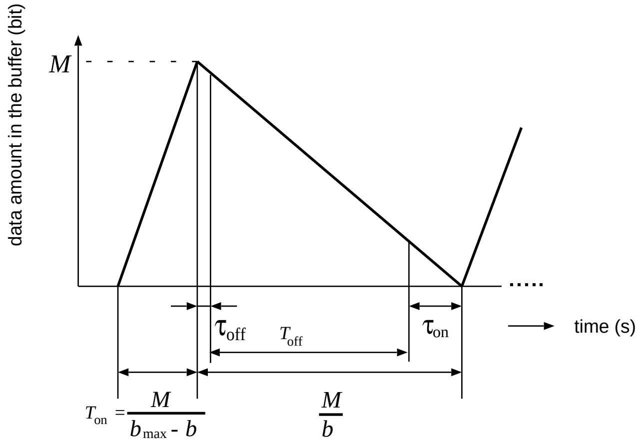

Figure 2: Time Chart of Buffered Playback

From Fig. 2, we obtain $T_{on} = \frac{M}{b_{max} - b}$, $\quad T_{off} = \frac{M}{b} - \tau$.

The power consumed by receiving the stream data is equal to $P_N(b_{max})$ while the WNIC is turned on, and 0 while turned off. So, the average power P_{buf} consumed by receiving the stream data (whose bitrate is b) is represented by

$$P_{buf}(b) = \frac{T_{on}}{T_{on} + T_{off} + \tau} P_N(b_{max}) = \frac{b}{b_{max}}(N + \gamma b_{max})$$

During time interval τ, the WNIC does not receive any data but consumes power to negotiate with the WLAN access point for authentication and address assignment by DHCP if necessary.

The average power P_{oh} consumed during time interval τ is represented by

$$P_{oh}(b) = \frac{\tau}{T_{on} + T_{off} + \tau} P_N(0) = \frac{b(b_{max} - b)\tau}{M b_{max}} N$$

Finally, we can derive a power consumption model when using the buffered playback.

$$E_0 = (S + \alpha r f + \beta b + P_{buf}(b) + P_{oh}(b))T \tag{6}$$

3.3 Algorithm for Deciding Parameter Values

Let (r_0, f_0, b_0) denote parameter values of the original video transmitted from the content server.

We assume that the desirable proportion among picture size r, frame rate f and bitrate b is given as follows.

$$r/r_0 : f/f_0 : b/b_0 = x : y : z \tag{7}$$

Equation (7) means that for example, if (x, y, z) are set to be $(1, 2, 1)$ and the picture size is reduced to $1/3$ by the transcoding proxy, frame rate and bitrate are reduced to $2/3$ and $1/3$, respectively.

From equation (7), b and f can be represented by expression consisting of r and constants.

Consequently, when desirable playback duration T, remaining battery amount E_0 and the above proportion among r, f and b are given, we can calculate the values of r, f and b by solving equation (6). In our implementation, we have used Newton's Method to solve the equation.

Calculation of appropriate bitrate

When we specify the ratio among r, f and b by hand, the value of b may not be adequate due to quantization noise. We assume that the bitrate is linear to the picture size when the frame rate is fixed, and that the bitrate is linear to the frame rate when the picture size is fixed. From these assumptions, we can derive the following equation to obtain the appropriate bitrate from r and f.

$$b = c_0 rf + c_1 r + c_2 f + c_3, \tag{8}$$

where c_0, c_1, c_2 and c_3 are constants, and can be calculated using multiple regression.

We have prepared 64 videos with different parameter values which have similar balances of playback quality, and obtained the above constants using multiple regression. As a result, we got $c_0 = 7.9$e-5, $c_1 = 4.2$e-4, $c_2 = 13$, $c_3 = -16$.

4. ENERGY-AWARE QOS CONTROL

In this section, we propose an algorithm to assign different playback quality among multiple video segments based on the user's preference, satisfying the specified playback duration within the remaining battery amount. Here, we assume that a video consists of multiple segments and those segments are classified into some predefined categories $C = \{c_1, ..., c_n\}$, e.g., by using MPEG-7 labeling tools [5].

4.1 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 *importance degrees*. Let p_i denote the importance degree 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 may have different preferences for the playback property among categories. So, we allow users to specify a preference to the playback property of each category by the proportion among the picture size, the frame rate and the bitrate $r/r_0 : f/f_0 : b/b_0 = x : y : z$ as explained in Sect. 3.3. An integer number 1 or more than 1 can be specified to x, y and z. When we use equation (8), we need not specify b/b_0.

For example, suppose that a video of a soccer game consists of video segments classified into three categories $\{$*shoot, play, other*$\}$. A user may want to play back the video segments of category *shoot* at as high quality as possible, and the segments of category *play* at the medium quality, while reducing the playback quality of the segments of category *other*. The user may have the preference for playback property such that both the motion speed and the vividness are similarly important in category *shoot*, that the motion speed is more important in category *play*, and that the vividness is more important in category *other*. In such a case, the user gives the following preference.

category	importance degree	r/r_0	f/f_0	b/b_0
$shoot$	4	1	1	–
$play$	2	1	2	–
$other$	1	2	1	–

4.2 Algorithm for Determining Playback Quality among Categories

For each category $c_i \in C$, the product of its importance degree p_i and playback duration T_i 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 battery amount $E = E_0 - S \sum_{i=1}^{|C|} T_i$ among categories according to the proportion of each category's virtual time T'_i/T'. That is, $E_i(= ET'_i/T')$ is allocated for playback of each category c_i.

However, the above algorithm may result in over/under assignment of battery amount to some categories. 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 maximum quality's video might be the video with satisfactory quality or the maximum video which the device can play back. The minimum quality's video can similarly be defined. So, we let our algorithm to adjust battery amounts in some categories so that restrictions of the maximum/minimum battery amount can be kept.

As explained in Sect. 3.3, battery amount E consumed by video playback is represented by equation (6) including parameters r, f, b and T. Consequently, if $E_i > (\alpha r_{max} f_{max} + \beta b_{max} + P_{buf}(b_{max}) + P_{oh}(b_{max}))T_i (\stackrel{def}{=} E_{max})$, E_i is too much for playback of c_i. Similarly, if $E_i < (\alpha r_{min} f_{min} + \beta b_{min} + P_{buf}(b_{min}) + P_{oh}(b_{min}))T_i (\stackrel{def}{=} E_{min})$, E_i is too small for playing back c_i. In those cases, we fix $E_i = E_{max}$ or $E_i = E_{min}$ and distribute remaining battery $E'(= E - E_i)$ among remaining categories $C - \{c_i\}$. Consequently, we can obtain battery amount E_i for playback of category c_i as a constant value.

Using battery amount E_i, playback duration T_i, and the proportion $(r_i/r_0 : f_i/f_0 : b_i/b_0)$ and algorithm in Sect. 3.3, we can calculate parameter values r_i, f_i and b_i for each category c_i.

4.3 Bitrate Extension Algorithm for Low Bandwidth Environment

When available network bandwidth b_{max} at the user terminal is low (e.g., 128Kbps), playback quality cannot be improved better than this value even if its remaining battery amount is sufficient. To cope with this problem, we propose an algorithm to enable playback of some video segments with higher bitrates than b_{max}.

The proposed algorithm delays the start time of the video playback where the user terminal buffers enough amount of data before starting video playback.

We assume that we play back video segments $v_1, ...,$ and v_m in this order. Let p_i and T_i denote the importance degree and the playback duration of segment v_i. Let P_{min} denote the power consumed by the playback of video segments with the minimum importance degree. Let D denote the delay time to start the playback. Then the following equation holds.

$$E_0 - S \cdot (D + \sum_{i=1}^{m} T_i) - D \cdot P_N(b_{max}) = \sum_{i=1}^{m} P_{min} \cdot p_i \cdot T_i \tag{9}$$

When battery amount assigned to segment v_i (represented by $P_{min} \cdot p_i \cdot T_i$) is determined, then we can obtain the corresponding r_i, f_i and b_i using the algorithm in Sect. 3.3. On the other hand, D can be calculated by

$$D = Max(\sum_{j=1}^{i-1} b_j \cdot T_j / b_{max} - \sum_{j=1}^{i-1} T_j), i = 2, ..., n+1 \tag{10}$$

Consequently, we can represent D as the function of P_{min}. From equation (9), we can calculate the value of P_{min} by using the Newton's Method.

We have applied the proposed algorithm to a video (640×480, 1150Kbps, 30fps, 2300sec) using E_0 as 50% of fully charged battery amount in the laptop PC environment in Table 2, where we set b_{max} to be 384Kbps. The result is shown in Table 1. We see that bitrates much higher than b_{max} are assigned to video segments 3 and 8 whose importance degrees are 4, while the delay time is reasonable (59sec).

Table 1: Quality Improvement When Using Bitrate Extension Algorithm

video			With bitrate extension (delay time=59sec)			Without bitrate extension		
segment (No.)	duration (sec)	priority	picture size (pixel)	frame rate (fps)	bit rate (Kbps)	picture size (pixel)	frame rate (fps)	bit rate (Kbps)
1	217	1	354×265	9.2	210	419×314	12.8	338.4
2	136	2	432×324	13.6	369.8	437×328	14	384
3	231	4	524×393	20.1	655.3	437×328	14	384
4	132	2	432×324	13.6	369.8	437×328	14	384
5	530	1	354×265	9.2	210	419×314	12.8	338.4
6	276	1	354×265	9.2	210	419×314	12.8	338.4
7	232	2	432×324	13.6	369.8	437×328	14	384
8	138	4	524×393	20.1	655.3	437×328	14	384
9	408	2	432×324	13.6	369.8	437×328	14	384

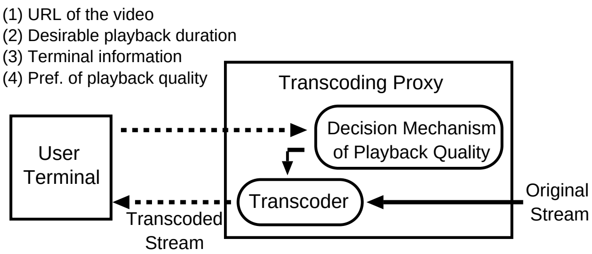

Figure 3: Energy aware video streaming system

Table 2: Experimental Environments

device type/ name/ setting	WNIC type/ name/ setting	CPU/OS
PDA/ Sharp Zaurus SL-C700/ brightness small	CF/ WN-B11/CF (I/O Data)/ low power mode off	XScale PXA250/Linux (Embedix)
laptop PC/ Toshiba SS S4/275PNHW brightness small	PCMCIA/ GW-NS11S (Planex)/ low power mode off	PentiumIII 750MHz/Linux Kernel 2.4.24

Table 3: Actual and Predicted Battery Life

picture size (pixel)	frame rate (fps)	bit rate (kbps)	actual (min)	predicted (min) (error %)
PDA				
streaming playback				
288x216	24	327	120	122 (1.6 %)
buffered playback				
288x216	24	327	189	185 (2.3 %)
288x216	8	110	289	285 (1.3 %)
166x124	24	321	242	239 (1.1 %)
166x124	8	109	340	323 (5.0 %)
laptop PC				
streaming playback				
400x300	24	499	6.67	6.72 (0.8 %)
buffered playback				
400x300	24	499	6.90	7.23 (4.8 %)
400x300	4	78	10.92	11.02 (0.9 %)
164x120	24	498	10.32	9.85 (4.6 %)
164x120	4	83	12.03	11.78 (2.1 %)

5. EXPERIMENTAL RESULTS

We have implemented an energy aware video streaming system as shown in Fig. 3. In the system, the user specifies the location of the video, the playback duration and the preference including importance degrees to categories and the proportion $r/r_0 : f/f_0 : b/b_0$ to each category. The information is sent to the transcoding proxy with the terminal information including the remaining battery amount and so on. The transcoding proxy calculates the appropriate playback quality from the received information, receives the video stream from the content server, transcodes it to the stream with the playback quality obtained with algorithms in Sect. 3.3 and Sect. 4 and forwards the new stream to the user terminal.

In order to evaluate the effectiveness of the proposed method, we have carried out several experiments using two different environments in Table 2. For both environments, we have used IEEE 802.11b WLAN. To restrain the power consumption due to rotation of HDD, we used `noflushd` for the laptop PC environment.

We obtained device specific constants S, N, α, β and γ using four videos with different quality and two different available bandwidth for each environment as explained in Sect. 3.2. In the PDA environment, we have used the fully charged battery amount as E_0. In the laptop PC environment, we have used the 10 % of the fully charged battery amount as E_0 to shorten measurement time.

5.1 Effectiveness of Power Saving Techniques and Power Consumption Model

In order to show effectiveness of our power saving techniques and validity of our power consumption model, we have measured the actual playback durations when using the buffered playback, and calculated errors from predicted durations. We used 2.3 Mbps and 1.6 Mbps as the average transmission rates for the PDA and the laptop PC, respectively, and used a buffer with 5 Mbyte. The experimental results are shown in Table 3. In table 3, prediction error defined as $|t_e - t|/t$ is a difference between the predicted battery life t_e and the actual battery life t.

Table 3 shows that the buffered playback could extend the battery life up to about 1.5 times in the PDA environment. On the other hand, in the laptop PC environment, the effect of the buffered playback is not so much. This is because the ratio of the power consumed by the WNIC to the whole consumed power is small in the laptop PC environment. When using the transcoding with the buffered playback, the battery lives could be extended up to about 2.8 times (PDA) and 1.7 times (laptop PC), respectively. The prediction errors were within 5% in both environments.

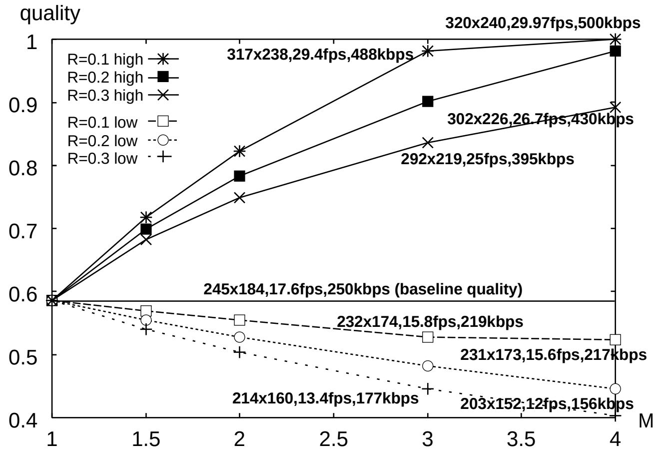

Figure 4: Quality Improvement

5.2 Effectiveness of QoS Control

In this section, we assume that the laptop PC environment is used and that 30% of the fully charged battery amount is used as E_0. An MPEG1 video with 320×240, 29.97fps, 500Kbps and 1800 sec is used as the original video denoted by $v_0 = (r_0, f_0, b_0, T_0)$.

Playback quality in important categories

Using the algorithm in Sect. 4.2, we have investigated to what extent the playback quality of important categories is improved and the quality of the other categories is degraded.

We assume that video segments in v_0 are classified into two categories: important category c_1 and less-important category c_2. We assume that the preference to each category is $r/r_0 : f/f_0 = 1 : 1$ (b is automatically calculated using equation (8)).

Let R denote the ratio of playback duration T_1 of c_1 to total playback duration $T_1 + T_2$ (i.e., $R \stackrel{def}{=} T_1/(T_1 + T_2)$). Let p_1 and p_2 denote the importance degrees for c_1 and c_2, respectively. Let M denote the ratio of p_1 to p_2 (i.e., $M \stackrel{def}{=} p_1/p_2$). Using our algorithm in Sect. 4.2, we have calculated the playback quality in c_1 and c_2 by changing R from 0.1 to 0.3 by step 0.1 and M from 1 to 4 by step 0.5.

The resulting graphs are depicted in Fig. 4, where the horizontal axis and the vertical axis represent M and playback quality r/r_0, respectively. Here, r/r_0 becomes 0.58, if all categories have the same priorities, that is, $p_1 = p_2$ (baseline quality).

Fig. 4 shows that when R is small (e.g., 0.1) and we specify $M \geq 3$, the playback quality in important categories can be improved significantly by slightly reducing the playback quality of less-important categories.

Prediction Errors When using QoS Control

To investigate the prediction errors of the actual playback duration within battery amount E_0 from its original playback duration 1800sec, we played back video v_0 at different playback quality calculated from four different preferences as shown in Table 4.

For *pref1*, *pref2*, *pref3* and *pref4*, actual playback durations until E_0 was exhausted were 1710sec, 1696sec, 1692sec, and 1727sec, respectively. The prediction errors are less than 6%.

6. CONCLUSION

In this paper, we proposed an energy-aware video streaming system for streaming video playback on portable computing devices, based on the user requirement including the battery amount, the desirable playback duration and the relative importance among video segments. We have implemented the system and carried out some experiments using a PDA and a laptop PC on IEEE 802.11b WLAN.

Through experiments, it is confirmed that (1) our system can control the playback duration within 6% error even when controlling QoS dynamically, and that (2) the proposed algorithm can be applied to various portable computing devices by obtaining several device specific constants. We believe that our model is accurate enough since some part of prediction errors is caused by the inaccuracy of the APM mechanism.

In this paper, we assumed that the available bandwidth at the user terminal is much larger than the bitrate of the video and that the fluctuation of the bandwidth is small. However, when applying the proposed method to the cellular phone environment, we must consider the fluctuation of the available bandwidth and the dynamic change of radio wave output depending on the location. In our future system, those problems will be treated.

Table 4: Preferences and Playback Qualities

	cat.	(T_i, p_i, f_i, r_i)	(r, f, b)
pref1	c_1	(678, 1, 1, 1)	$(245 \times 184, 17.60, 250K)$
	c_2	(662, 1, 1, 1)	$(245 \times 184, 17.60, 250K)$
	c_3	(460, 1, 1, 1)	$(245 \times 184, 17.60, 250K)$
pref2	c_1	(678, 1, 1, 1)	$(196 \times 147, 11.26, 143K)$
	c_2	(662, 2, 1, 1)	$(240 \times 180, 16.91, 238K)$
	c_3	(460, 4, 1, 1)	$(293 \times 219, 25.06, 395K)$
pref3	c_1	(678, 1, 1, 1)	$(196 \times 147, 11.26, 143K)$
	c_2	(662, 2, 1, 2)	$(292 \times 219, 12.47, 201K)$
	c_3	(460, 4, 2, 3)	$(320 \times 240, 21.61, 365K)$
pref4	c_1	(678, 1, 1, 1)	$(196 \times 147, 11.26, 143K)$
	c_2	(662, 2, 2, 1)	$(196 \times 147, 22.48, 288K)$
	c_3	(460, 4, 3, 2)	$(261 \times 196, 29.97, 439K)$

7. REFERENCES

[1] Agrawal, P., Chen, J-C, Kishore, S., Ramanathan, P. and Sivalingam, K.: Battery power sensitive video processing in wireless networks, *Proc. IEEE PIMRC'98*, Vol. 1, pp.116–120 (1998).

[2] 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).

[3] Lahiri, K., Raghunathan, A. and Dey, S.: Battery-Efficient Architecture for an 802.11 MAC Processor, *Proc. of the 2002 IEEE Int'l Conf. on Communications (ICC2002)*, pp.669–674 (2002).

[4] 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).

[5] 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).

[6] Pering, T., Burd, T. and Brodersen, R.: Dynamic Voltage Scaling and the Design of a Low-Power Microprocessor System, *Proc. of Driven Microarchitecture Workshop*, pp.107–112 (1998).

[7] Simunic, T., Benini, L., Acquaviva, A., Glynn, P. and Micheli, G. D.: Dynamic voltage scaling for portable systems, *Proc. of the 38th Design Automation Conf. (DAC2001)*, pp.524–529 (2001).

[8] Xu, R., Li, Z., Wang, C. and Ni, P.: Impact of Data Compression on Energy Consumption of Wireless-Networked Handheld Devices, *Proc. of the 23rd Int'l. Conf. on Distributed Computing Systems (ICDCS'03)*, pp.302–311 (2003).

[9] Yip, K.W. and Ng, T.S.: Fast Power Control, Transmit Power Reduction and Multimedia Communications over WLANs, *Proc. of First Int'l Conf. on Information Technology and Applications (ICITA2002)* (2002).