Selective Frame Prefetching for Reducing Disk Energy Consumption in Scalable Video Coding (SVC) Media Players

Seungsoon Lee and Minseok Song, *Member*, IEEE

Abstract - *We propose a new disk power management scheme for scalable video coding (SVC) video playback. Conventional disk power management techniques for video players usually prefetch video frames into buffer to allow disk to enter low-power mode, which may require significant buffer space. To address this, we exploit temporal scalability property in which multiple frame rates for the same video stream are provided, and allow selective frame prefetching with the aim of achieving specified quality levels while reducing disk energy consumption. Simulation results show that the disk energy consumption can be reduced by 8.4% on average without any noticeable quality degradation by users*[1].

Index Terms — Portable Media Player, Disk Power Management, Scalable Video Coding, Temporal Scalability

I. INTRODUCTION

Recent advances in multimedia and network technologies have made it possible for people to access video services using their portable consumer electronics such as netbooks, notebooks, personal multimedia players (PMPs), MP3 players and so on. Clients accessing such video services tend to control their quality of service (QoS) parameters, and scalable video coding (SVC) techniques provide easy QoS adatation by enabling the decoding of partial bit streams to provide video services with lower temporal, spatial resolutions or reduced fidelity [1], [2].

SVC divides a video into several layers, which can be decoded to extract low-quality video streams if all low-quality layers are available. This allows temporal, spatial and quality scalabilities, so a subset of a full-quality stream can represent a lower temporal, spatial, or quality video signal [1]-[4]. Among various SVC codecs, H.264/SVC, the SVC extension of the H.264/AVC standard, receives a lot of attention from research and industry communities because of its outstanding coding efficiency [1], [15].

To provide large storage requirements of multimedia contents, many portable devices use hard disk drives (HDD), which is known to be the most cost-effective storage medium [7], [9]. A disk, however, consumes a significant amount of energy, so it is important to reduce its energy consumption. For example, a microdrive of a personal digital assistant (PDA) is known to take 23% of the total power [10].

Modern disks have several power modes [5], [7], [8]: in active mode the head is reading or writing data; in seek mode the head is seeking; in idle mode the disk spins at full speed but is processing no requests for data; and in standby mode the disk stops spinning completely so no requests can be handled, but consumes much less energy than in any other mode.

To reduce disk energy, it is important to extend the length of time during which disk stays in standby mode [8], [9], [16]. To achieve this, existing low-power media players generally use prefetching techniques [7]-[9]; they prefetch many frames into memory to handle further requests from the memory without accessing the disk, allowing disk to enter standby mode. More data in the buffer increases the idle period extending the period during which disk stays in standby, but this may require a lot of buffer space, which may not be feasible for portable consumer devices that have small amount of memory space [10], [11].

A lot of previous works in network research communities made use of SVC adaptation characteristics to cope with unreliable network stability and shortage of wireless network bandwidth [12], [13], [15]. Although few works have been done to reduce system-level power consumption using the property [14], to the best of our knowledge, there was no previous work that handles disk power issues for the SVC playback.

We propose a new disk power management scheme for portable media players by making use of temporal scalability. We first present a QoS model that reflects users' subjective evaluation of video quality. Based on this, we propose a new prefetching scheme in which frames are selectively prefetched into buffer with the aim of reducing disk energy consumption without noticeable quality degradation by users, while providing real-time video service.

The rest of this paper is organized as follows: we explain the related work in Section II. We present our system model in Section III, and propose a new selective prefetching scheme in Section IV. We assess our scheme in Section V and conclude the paper in Section VI.

II. RELATED WORK

Disk power management techniques have received a lot of attention from research communities, and have been studied for both general-purpose and multimedia mobile systems. To reduce disk energy consumption, most techniques for the general-purpose systems put the disk into standby mode after

[1] This work was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology under grant 2010-0005743, in part by the IT R&D program of MKE/KEIT under grant 10035243, and in part by Inha University Research Grant.

Seungsoon Lee is with the School of Computer and Information Engineering, Inha University, Incheon, Korea (e-mail: veryslow@hotmail.com).

Minseok Song is with the School of Computer and Information Engineering, Inha University, Incheon, Korea (e-mail: mssong@inha.ac.kr) .

Contributed Paper
Manuscript received 12/20/11
Current version published 06/22/11
Electronic version published 06/22/11.

it has been idle for a while [5], [14], [18], [19]. When the next request arrives, the disk is returned to active mode as fast as possible to service that request. These approaches, however, cannot be used for video players, because video data must be read continuously and in a real-time fashion [7], [8], [16].

Conventional low-power media players prefetch frames into buffer to allow disk to enter standby mode. Ridenour et al. [6] have presented how prefetching can be used for media players to reduce disk energy consumption. Pettis et al. [8] analyzed the effect of buffer size on disk energy consumption and provided the analytical solution for the optimal buffer size. Won et al. [9] introduced several memory management techniques for prefetching. Go et al. [7] have implemented this kind of buffered video player on Linux.

Several works employ flash memory to reduce disk energy consumption [10], [11], [17], [18], [20]. Khatib et al. [10] have used flash memory as a buffer and addressed the tradeoff between memory size and disk energy. Kim et al. [11] analyzed the energy consumption of dynamic random access memory (DRAM), flash memory and disk when frames are prefetched into flash memory. Ryu et al. [20] modified an Mplayer on Linux to use flash memory as a buffer to reduce disk energy consumption. All of these low-power media players, however, are developed for non-scalable video files.

Multi-speed disks may be used to reduce disk energy consumption for media players. Liu et al. [21] examined several I/O speed setting strategies for multimedia applications. Rao et al. [22] investigated the problem of choosing disk speeds for video playback with the aim of minimizing disk energy consumption. Even though multi-speed disks are now in a market, single-speed disks are still widely used for mobile devices.

III. System Model

SVC supports temporal scalability, so a subset of a stream represents the source content with a reduced frame rate. We assume that there are N_r temporal resolution levels where the j^{th} temporal resolution level has r_j frames per second (fps) $(j = 1,...,N_r)$. We suppose that the $N_r{}^{\text{th}}$ resolution level represents the full resolution requiring the highest bit-rate; thus, $r_{N_r} > ... > r_1$. Let D_j (fps) be the decoding period of the video with the j^{th} temporal resolution level, so D_j is calculated as $\frac{1}{r_j}$. Obviously, $D_1 > ... > D_{N_r}$. Let S_i be the size of the i^{th} frame ($i = 1,...,N_f$), where N_f represents the total number of frames for a given video. We assume that a given video has a length of T_t seconds.

Decreasing temporal resolution level reduces the frame rate, which requires more frames to be skipped. To represent this, we introduce a binary variable for each frame i when the j^{th} resolution level is selected as follows: if frame i is skipped when the j^{th} resolution level is selected, $x_{i,j} = 0$; otherwise. $x_{i,j} = 1$.

IV. Selective Frame Prefetching

A. Main Idea

To allow disk to enter standby mode, video frames are prefetched into a buffer in memory, and a number of consecutive requests for the frames can then be handled by the buffer without accessing the disk. To make full use of available memory, the disk enters standby mode when the player has prefetched as many frames as possible into memory, and returns to the active mode before the buffer becomes empty. Therefore, disk alternates between active and standby modes during the entire playback; based on this, we divide our disk activity cycle into prefetching and standby periods as shown in Fig. 1.

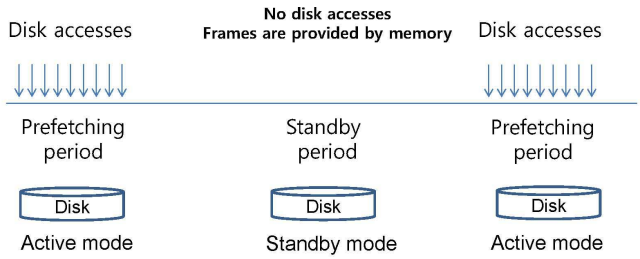


Fig. 1. Our disk activity cycle model.

Increasing the temporal resolution level decreases the decoding period, which empties the buffer quickly, reducing the standby period; conversely, decreasing the temporal level makes a longer standby period. To illustrate this possibility, we examined how the lengths of the standby period depend on four different temporal resolution levels of 30, 15, 7.5 and 3.75 fps as shown in Table I. We here suppose that every frame has 8KB and the prefetching buffer size is 2MB. The results in Table I clearly demonstrate that decreasing resolution level increases the length of standby period, which gives disk more opportunities to reduce its energy consumption.

TABLE I
AN EXAMPLE OF HOW THE LENGTHS OF STANDBY PERIOD DEPENDS ON THE RESOLUTION LEVELS

Fps	30	15	7.5	3.75
Standby period (s)	8.5	17.1	34.1	68.3

Most people watching a video tend not to notice brief interruptions in playback continuity [1], [23]. In particular, they do not perceive the decrease in frame rates for a clip with static scene changes but feel quality degradation greatly for a video clip with a lot of scene changes [23]. We exploit these

users' perception to the video quality, and decrease the temporal resolution level with the aim of reducing disk energy consumption, if this does not degrade quality perceived by users.

To balance video quality with disk energy consumption, we divide a given video into N_s segments and evaluate how the quality of each video segment depends on the temporal resolution level. For the low-resolution segments, users may feel no, small, or great degradation depending on the scene characteristics. To reflect this, we calculate the minimum acceptable resolution level L_k for segment k, and select the temporal resolution level of each segment differently to guarantee that users watch each video segment k with L_k resolution level .

B. Design

Prefetching can be implemented by retrieval and decoder threads [7]: the retrieval thread reads frames from the disk and fills the buffer, and the decoder thread reads frames from the buffer and decodes them at playback rate.

Power mode transitions take time and require energy, so it may increase disk energy consumption to allow disk to enter standby mode. We refer to the shortest length of the standby period that justifies the energy cost of spinning up again as breakeven time [7], [8], [9]. Based on this, to determine whether to allow disk to enter standby mode, we consider two constraints as follows:

R1: To fully utilize available buffer space, disk can enter standby mode when the player has prefetched as many frames as possible into memory.
R2: The standby period must exceed the breakeven time.

When the player reads frame i of segment k, it decides whether to read it or not depending on current L_k values: if $x_{i,L_k}=1$, then it reads frame i ; otherwise, it skips reading.

Let B be the buffer size. Let G_i denote the segment number to which frame i corresponds. Let F_i be the frame index of the first frame in the buffer when frame i is the last frame in the buffer. When frame i has been read from disk, the player checks the following conditions to meet the requirement **R1**:

$$\sum_{m=F_i}^{i} S_m x_{m,L_{G_m}} \le B , \qquad (1)$$

$$\sum_{m=F_i}^{i+1} S_m x_{m,L_{G_m}} > B . \qquad (2)$$

If these two conditions are met, then the player has prefetched as many frames as possible into buffer and checks the requirement **R2**.

Let E_s be the spinup energy and E_f the spindown energy. Let T_u and T_d be the disk spinup and spindown times, respectively. Let P_a be the power in active mode, P_i the power in idle mode, and P_l the power in standby mode. Then, the breakeven time T_b can be calculated as follows [7]:

$$T_b = \frac{E_s + E_f - P_l \times (T_u + T_d)}{P_i - P_l}.$$

We now calculate the total remaining time R_i that can be played using the frames between F_i and i in the current buffer, which depends on the temporal resolution level of each frame as follows:

$$R_i = \sum_{m=F_i}^{i} D_{L_{G_m}} .$$

To meet the requirement **R2**, the following condition must be satisfied:

$$R_i > T_b . \qquad (3)$$

If conditions in (1), (2) and (3) are satisfied when frame i has been read, then disk goes into standby mode; otherwise, it stays in idle mode to read the next frame $i+1$.

Before emptying the buffer, disk needs to be spun up to fill the buffer again. Delayed spinup causes the playback to be stopped or become distorted. To guarantee continuity of video playback, we must spin up the disk at least T_u seconds before all the frames in the memory are exhausted. We also assume that several frames are prefetched into memory before the buffer becomes empty, so the disk is spun up $T_u + \alpha$ seconds before all the frames in the buffer are exhausted, where α is an additional time reserved. When disk is in standby mode and frame F_i is going to be decoded, we check that $R_i \le T_u + \alpha$, and spin up the disk at a point in time in which R_i becomes less than or equal to $T_u + \alpha$.

C. Energy consumption calculation

To estimate the disk energy consumption, we calculate the length of periods in which disk stays in standby, active, idle and seek modes. We assume that videos are stored sequentially on disk, which requires only one seek operation for each prefetching.

If conditions in (1), (2) and (3) are satisfied after frame i has been read, then disk goes into standby mode. We maintain a set of frame indices i at this point, I_f as follows: $I_f = \{M_1, ..., M_{N_p}\}$, where M_n is the n^{th} frame index in I_f and N_p is the number of elements in set I_f . We can now determine our energy properties as follows:

(1) We can calculate the length of time in which disk stays in active mode by dividing the total size of frames by disk

transfer rate, r_t, which is $\frac{\sum_{m=1}^{N_f} S_m x_{m,L_{G_m}}}{r_t}$. We can thus calculate energy consumption during the active phase, E_a as follows:

$$E_a = P_a \frac{\sum_{m=1}^{N_f} S_m x_{m,L_{G_m}}}{r_t}.$$

(2) Disk enters standby mode N_p times during the entire playback. Let P_s be the power in seek mode and T_s the seek time. Seek operation is required whenever it starts prefetching, so the total seek time is $N_p T_s$. Thus, the seek energy, E_s can be calculated as $P_s N_p T_s$.

(3) When disk enters standby mode after frame M_n in I_f has been read, disk stays in standby mode for $R_{M_n} - T_u - T_d - \alpha$, so the total length of standby time during the entire playback is $\sum_{n=1}^{N_p}(R_{M_n} - T_u - T_d - \alpha)$. Because disk consumes standby power of P_l, the standby energy consumption, E_l can be calculated as follows:

$$E_l = P_l \sum_{n=1}^{N_p}(R_{M_n} - T_u - T_d - \alpha).$$

(4) Disk stays in idle mode, when no disk activity is taking place, so we can calculate the total idle time during the entire playback by subtracting the seeking and standby and active time from the total playback time, T_t. We therefore calculate the energy consumption during the idle mode, E_i as follows:

$$E_i = P_i(T_t - \sum_{n=1}^{N_p}(R_{M_n} - T_u - T_d - \alpha) - N_p T_s - \frac{\sum_{m=1}^{N_f} S_m x_{m,L_{G_m}}}{r_t})$$

Based on the energy calculation above, we now determine the total energy consumption during the entire playback, E_t as follows:

$$E_t = E_i + E_l + E_s + E_a$$

V. Experimental Results

A. Simulation setup

We evaluated the effectiveness of our scheme by simulations. We consider three disks with different form factors as shown in Table II [10]. We set α to 1 seconds, and consider three different types of H.264/SVC video clips: animation, sports and TV shows, each of which has the characteristics in Table III. We consider five temporal resolution levels between 5 and 1 with 30, 15, 7.5 and 3.75, 1.875 fps. The length of each video is 8 minutes, and we divide each video into 48 segments each of which contains frames of 10 seconds. We consider four buffer sizes: 1MB, 2MB, 4MB and 6MB.

We requested five users to grade videos at each temporal resolution level. Based on this, we calculate L_k values at each segment k differently depending on the users' subjective satisfaction about video quality as follows:

(1) **M1**: This method selects the lowest resolution level L_k at which users watch each segment with small but acceptable quality degradation as shown in Table IV.

(2) **M2**: This method selects the lowest resolution level L_k at which users watch each segment with no noticeable quality degradation as shown in Table V.

TABLE II
DISK CHARACTERISTICS

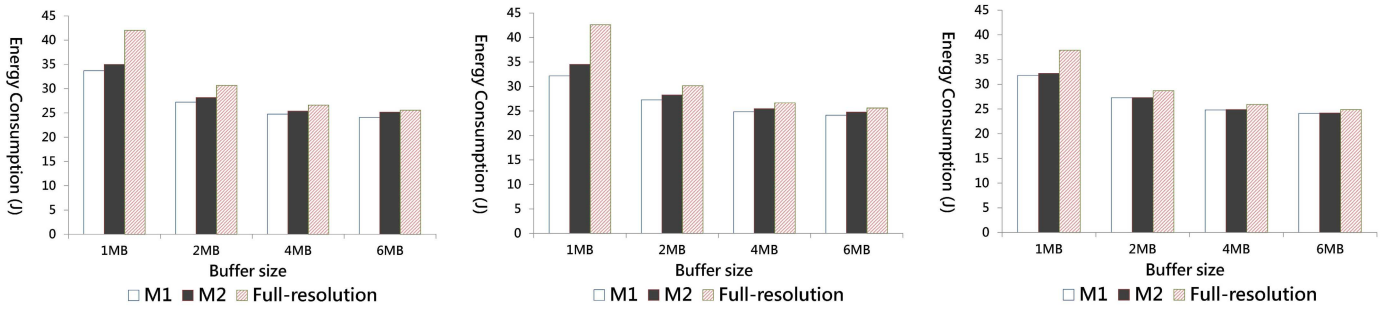
Parameters	Microdrive (1.0 inch)	Disk drive (2.5 inch)	Disk drive (3.5 inch)
Spinup time	0.5s	4.0s	15.0s
Spindown time	0.5s	1.0s	5.0s
Seek time	0.012s	0.016s	0.020s
Disk transfer rate	96Mbps	318.5Mbps	383.2Mbps
Active power	0.99W	2.0W	11.0W
Seek power	0.66W	2.3W	10.0W
Idle power	0.215W	0.85W	8.0W
Standby power	0.043W	0.2W	1.0W
Spinup energy	1.023W	5.5W	29.5W
Spindown energy	0.215W	1.8W	8.0W

TABLE III
VIDEO CHARACTERISTICS

Movie name	fps	Average bit-rate (Kb/s)
Animation "Incredible"	30	964
	15	805
	7.5	594
	3.75	424
	1.875	301
TV show "I am a singer"	30	659
	15	557
	7.5	448
	3.75	351
	1.875	268
Sports Broadcasting: Soccer game	30	569
	15	485
	7.5	393
	3.75	308
	1.875	233

TABLE IV
A PAIR OF SEGMENT INDEX AND L_k, (k, L_k) WHEN SMALL BUT ACCEPTABLE QUALITY DEGRADATION IS PERCEIVED BY USERS

Animation							
(1,3)	(2,3)	(3,3)	(4,3)	(5,3)	(6,3)	(7,3)	(8,3)
(9,3)	(10,4)	(11,3)	(12,3)	(13,3)	(14,2)	(15,3)	(16,3)
(17,3)	(18,3)	(19,3)	(20,3)	(21,3)	(22,3)	(23,2)	(24,3)
(25,3)	(26,3)	(27,3)	(28,3)	(29,3)	(30,3)	(31,3)	(32,3)
(33,3)	(34,3)	(35,4)	(36,4)	(37,4)	(38,4)	(39,4)	(40,3)
(41,3)	(42,3)	(43,3)	(44,3)	(45,3)	(46,3)	(47,2)	(48,2)
TV Show							
(1,3)	(2,3)	(3,3)	(4,3)	(5,3)	(6,3)	(7,4)	(8,3)
(9,3)	(10,3)	(11,3)	(12,3)	(13,3)	(14,3)	(15,3)	(16,3)
(17,3)	(18,3)	(19,3)	(20,3)	(21,3)	(22,3)	(23,3)	(24,3)
(25,3)	(26,3)	(27,3)	(28,3)	(29,3)	(30,3)	(31,3)	(32,3)
(33,3)	(34,3)	(35,3)	(36,3)	(37,4)	(38,3)	(39,3)	(40,4)
(41,3)	(42,3)	(43,3)	(44,4)	(45,4)	(46,4)	(47,3)	(48,3)
Sports broadcasting							
(1,4)	(2,4)	(3,3)	(4,4)	(5,4)	(6,4)	(7,3)	(8,4)
(9,4)	(10,4)	(11,3)	(12,4)	(13,4)	(14,4)	(15,4)	(16,4)
(17,3)	(18,4)	(19,4)	(20,4)	(21,4)	(22,4)	(23,4)	(24,4)
(25,3)	(26,4)	(27,4)	(28,4)	(29,4)	(30,4)	(31,4)	(32,4)
(33,4)	(34,4)	(35,4)	(36,4)	(37,3)	(38,4)	(39,4)	(40,4)
(41,4)	(42,4)	(43,4)	(44,4)	(45,4)	(46,3)	(47,4)	(48,4)



(a) Animation **(b) TV Show** **(c) Sports broadcasting**

Fig. 2. Energy consumption against buffer size when a 1 inch disk drive is used.

TABLE V
A PAIR OF SEGMENT INDEX AND L_k , (k, L_k) WHEN NO QUALITY DEGRADATION IS PERCEIVED BY USERS

Animation							
(1,3)	(2,3)	(3,4)	(4,4)	(5,4)	(6,3)	(7,4)	(8,4)
(9,4)	(10,4)	(11,4)	(12,3)	(13,3)	(14,3)	(15,3)	(16,4)
(17,4)	(18,4)	(19,4)	(20,4)	(21,4)	(22,4)	(23,4)	(24,4)
(25,3)	(26,4)	(27,4)	(28,3)	(29,4)	(30,4)	(31,4)	(32,3)
(33,4)	(34,4)	(35,4)	(36,4)	(37,4)	(38,4)	(39,4)	(40,3)
(41,3)	(42,3)	(43,4)	(44,4)	(45,4)	(46,4)	(47,3)	(48,3)
TV Show							
(1,3)	(2,4)	(3,4)	(4,4)	(5,4)	(6,4)	(7,4)	(8,3)
(9,4)	(10,4)	(11,3)	(12,4)	(13,4)	(14,4)	(15,4)	(16,4)
(17,4)	(18,4)	(19,4)	(20,4)	(21,3)	(22,4)	(23,4)	(24,4)
(25,4)	(26,4)	(27,4)	(28,3)	(29,3)	(30,4)	(31,4)	(32,4)
(33,4)	(34,4)	(35,3)	(36,3)	(37,4)	(38,4)	(39,4)	(40,3)
(41,4)	(42,4)	(43,4)	(44,3)	(45,4)	(46,4)	(47,4)	(48,4)
Sports broadcasting							
(1,4)	(2,4)	(3,4)	(4,4)	(5,4)	(6,4)	(7,4)	(8,4)
(9,4)	(10,4)	(11,4)	(12,4)	(13,4)	(14,4)	(15,4)	(16,4)
(17,4)	(18,4)	(19,4)	(20,4)	(21,4)	(22,4)	(23,4)	(24,4)
(25,4)	(26,4)	(27,4)	(28,5)	(29,4)	(30,4)	(31,4)	(32,4)
(33,3)	(34,4)	(35,4)	(36,4)	(37,4)	(38,4)	(39,4)	(40,4)
(41,4)	(42,4)	(43,4)	(44,4)	(45,4)	(46,4)	(47,4)	(48,4)

B. Simulation results

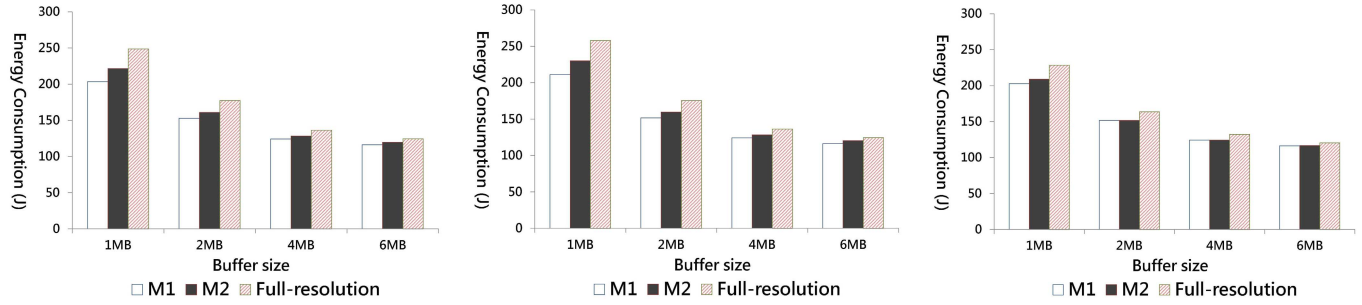
Fig.2 shows disk energy consumption depending on the buffer size when an 1.0 inch disk is used. From the figure, we obtain the following observations: first, the **M1** scheme exhibits the best performance under all workloads, saving energy between 3% and 25% more than full-resolution playback, and between 0.2% and 7% more than **M2**. This is because our schemes allow frame skipping, making a longer standby period. Second, the differential between our schemes and full-resolution playback is more pronounced when the buffer size is small; this suggests that our idea is more effective in reducing the energy consumption when a small amount of buffer space is available. Third, **M1** shows slightly better performance than **M2**, but this is at the cost of small quality degradation perceived by users, which implies that **M2** may be more desirable than **M1** as far as video quality is concerned. Fourth, for a sport movie, our methods produce relatively small energy savings compared with other video types, because it exhibits great variability, which causes users to notice the reduction in frame rates greatly as shown in Table IV.

Fig.3 shows disk energy consumption when a 2.5 inch disk is used, and demonstrates that **M1** also exhibits the best performance under all workloads, saving between 3.5% and 19% more energy than full-resolution playback and between 0.02% and 8% more energy than **M2**. We also observe the following: (1) compared with the microdrive case in Fig. 2, the energy savings over the full-resolution playback is relatively high when the buffer size is larger than or equal to 2MB. (2) Like the results in the microdrive case, **M1** shows slightly better performance than **M2**, but the energy gaps are higher than those for the microdrive. (3) The energy gap between **M1** and full-resolution playback decreases as buffer size increases.

Fig 4 shows disk energy consumption when a 3.5 inch disk is used. Again **M1** shows the best performance, saving between 2% and 23% more energy than full-resolution playback and up to 9% more energy than **M2**. Comparing these results with those for the other form-factor disks, we found that the energy gap between our schemes and full-resolution playback is the highest when the buffer size is adequate (4MB). This is because 3.5 inch disk drive has a high spinup energy as shown in Table II, which requires relatively large buffer space to justify the energy cost of disk spinup.

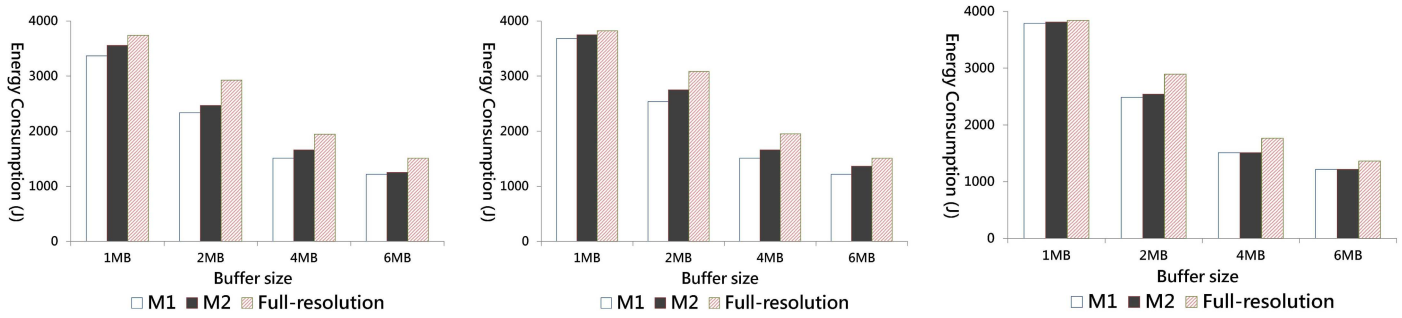
VI. CONCLUSIONS

We have presented a new disk power management scheme for SVC video playback. By exploiting temporal scalability, we proposed a new QoS model that takes users' subjective evaluation of video quality into account. Based on this, we proposed a selective frame prefetching scheme with the aim of reducing disk energy consumption while maintaining quality perceived by users. We also presented the design details of our scheme to provide real-time video playback and analyzed disk energy consumption.



(a) Animation **(b) TV Show** **(c) Sports broadcasting**

Fig. 3. Energy consumption against the buffer size when a 2.5 inch disk drive is used.



(a) Animation **(b) TV Show** **(c) Sports broadcasting**

Fig. 4. Energy consumption against the buffer size when a 3.5 inch disk drive is used.

Simulation results show that our scheme can reduce disk energy consumption by 8.4% on average without any noticeable quality degradation. These observations provide a guideline for the design of the portable media players.

As our future works, we are planning to implement our scheme on real media players, allowing us to measure actual disk energy consumption.

REFERENCES

[1] Unanue, I. Urteaga, R. Husemann, J. Ser, V. Roesler, A. Rodriguez and P. Sanchez. A tutorial on H.264/SVC scalable video coding and its tradeoff between quality, coding efficiency and performance. *Recent advances in video coding* Intech publisher, 2011.

[2] H.Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Trans. Circuits Syst. Video Technol.*, 17(9):1103-1120, 2007.

[3] Segall and G. Sullivan. Spatial scalability within the H.264/AVC scalable video coding extension. *IEEE Trans. Circuits Syst. Video Technol.*, 17(9):1121-1135, 2007.

[4] W. Cazoulat, R. Graffunder, A. Hutter and P. Amon. Real-time system for adaptive video streaming based on SVC. *IEEE Trans. Circuits Syst. Video Technol.*, 17(9):1227-1237, 2007.

[5] F. Chen and X. Zhang. Caching for bursts (c-burst): let hard disks sleep well and work energetically. in *Proc. ACM Int. Symp. Low Power Electron. Des.,* pp. 141-146, August, 2008.

[6] J. Ridenour, J. Hu, N. Pettis, and Y. Lu. Low-Power Buffer Management for Streaming Data. *IEEE Trans. Circuits Syst. Video Technol.*, 17(2):143-157, 2007.

[7] J. Go and M. Song. Adaptive disk power management for portable media players. *IEEE Trans. Consum. Electron.*, 54(4):409-428, November 2008.

[8] N. Pettis, L. Cai, and Y. Lu. Statistically optimal dynamic power management for streaming data. *IEEE Trans. Comput.*, 55(7):800-814, July 2006.

[9] Y. Won, J. Kim, and W. Jung. Energy-aware disk scheduling for soft real-time I/O requests. *ACM/Springer Multimedia Syst. J.*, 13(5):409-428, February 2008.

[10] M. Khatib, P. Hartel and H. Dijk. Energy-efficient streaming using non-volatile memory. *Signal Process. Syst. J.,* vol. 60, pp.149-168, 2010.

[11] J. Kim, A. Yang and M. Song. Exploiting flash memory for reducing disk power consumption in portable media players, *IEEE Trans. Consum. Electron.*, 55(4):1997-2004, 2009.

[12] X. Ji, J. Huang, M. Chiang, G. Lafruit, and F. Cathoon. Scheduling and resource allocation for SVC streaming over OFDM downlink systems. *IEEE Trans. Circuits Syst. Video Technol.*, 19(10):1549-1555, 2009.

[13] J. Lee and C. Yoo. Scalable ROI algorithm for H.264/SVC-based video streaming. *IEEE Trans. Consum. Electron.*, 57(2):882-887, 2011.

[14] H. Lee and H. Shin. Luminance scalable coding using H.264/AVC SVC extensions for mobile video applications. in *Proc. IEEE Int. Conf. Multimedia*, pp 1025-1028, July 2008.

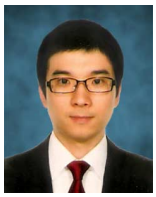
[15] M. Blestel and M. Raulet. Open SVC decoder: a flexible SVC library, in *Proc. ACM Multimedia*, pp. 1463-1466, October, 2010.

[16] Moal, D. Molaro, and J. Campello. Power-efficient real-time disk scheduling. in *Proc. ACM Workshop Netw. Oper. Syst. Support Digital Audio Video*, pp. 55-60, June 2009.

[17] T. Bisson and S. Brandt. A hybrid disk-aware spin-down algorithm with I/O subsystem support. in *Proc. IEEE Int. Performance, Comput. Commun. Conf.,* pp. 236-245, April 2007.

[18] F. Chen, S Jiang and X Zhang. Smart Saver: turning flash drive into a disk energy saver for mobile computers. in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Des.*, pp. 412-417, October 2006.
[19] Papathanasious and L. Scott. Energy efficient prefetching and caching. in *Proc. USENIX Annu. Tech. Conf.*, pp. 255-268, June 2004.
[20] W. Ryu and M. Song. Design and implementation of a disk energy saving scheme for media players which user hybrid disks. *IEEE Trans. Consum. Electron.*, 56(4):2382-2386, 2010.
[21] X. Liu, P. Shenoy and W. Gong. A time series-based approach for power management in mobile processors and disks. in *Proc. ACM Workshop Netw. Oper. Syst. Support Digital Audio Video,* pp. 74-79, 2004.
[22] R. Rao, S. Vrudhula, and M. Krishnan. Disk drive energy optimization for audio-video applications. in *Proc. ACM Conf. Compilers, Architect. Syn. Embedded Syst.*, pp. 93–103, 2004.
[23] W. Song, D. Tjondronegoro, S. Azad. User-Centered Video Quality Assessment for Scalable Video Coding of H.264/AVC Standard. in *Proc. Int. Conf. Multimedia Modeling*, pp. 55-65, 2010.

BIOGRAPHIES



Seungsoon Lee received his B.S. degree in Computer Science and Information Engineering from Inha University, Korea, in 2011. He is currently an M.S. student in the School of Computer Science and Information Engineering at Inha University, Korea. His research interests include embedded system and multimedia systems.

Minseok Song (M'07) received his B.S. and M.S. degrees in Computer Engineering from Seoul National University, Korea, in 1996 and 1998, respectively. He received Ph.D. degree in Electrical Engineering and Computer Science from Seoul National University, Korea, in 2004. Since September 2005, he has been with the School of Computer Science and Information Engineering at Inha University, Incheon, Korea, where he is currently an associate professor. His research interests include embedded systems and multimedia systems