A Combined Approach for QoS-Guaranteed and Low-Power Video Decoding

Jinsung Cho, Ilyong Cho, Dae-Young Kim, and Ben Lee, *Member*, IEEE

Abstract — *This paper proposes a power management technique for video playback in mobile multimedia devices, which combines low-power video decoding and QoS-guaranteed algorithms: ILI (Interval-based Linear Interpolation) and QLB (QoS-guaranteed and Low-power Buffering). First, the proposed ILI algorithm precisely estimates the decoding time of video frames through interval-based linear interpolation. Second, the QLB algorithm has two modes of operation, and dynamically switches between the modes based on feedback of decoding statistics of recent frames. The combinations ILI and QLB algorithms allow the CPU voltage and clock frequency to be adjusted to the optimal setting. Our extensive experiments show that the proposed approach can achieve about 4.8%-13.7% more power savings compared with existing methods*[1].

Index Terms — low-power video decoding, DVFS, workload prediction, ILI, QLB.

I. INTRODUCTION

Due to the increase in popularity of mobile devices, such as smartphones, portable media players, and pad/tablet computers (e.g., iPad), energy-efficient video playback schemes have recently received a lot of attention. Those schemes achieve power savings through *Dynamic Voltage and Frequency Scaling* (DVFS) by either predicting decoding times of frames [1]-[5] or using feedback of decoding statistics of recent frames [6]-[8]. Our prior study in [9] shows that DVFS based on prediction methods are superior to the feedback control methods.

DVFS exploits variable CPU frequencies and voltages to trade off between energy consumption and speed [10]. Figure 1 shows the basic idea behind DVFS. Figure on the left shows the processor activity (indicated by the shaded area) when DVFS is not used, which means that the processor runs at the highest voltage/frequency setting. After the task is completed, the processor waits until its deadline. During this idle period, the processor is still running and consuming power. These idle periods are the target for exploitation by DVFS. Figure on the right shows the ideal case where the processor scales exactly to the voltage/frequency setting required for the desired time span. Therefore, no idle time exists and power saving is maximized.

Although DVFS can be applied to any application, it is especially beneficial for video decoding because of its high computational and power requirements and frame variability [9]. Although there have been many related research efforts, low-power video decoding has yet to be implemented on mobile multimedia devices such as smartphones.

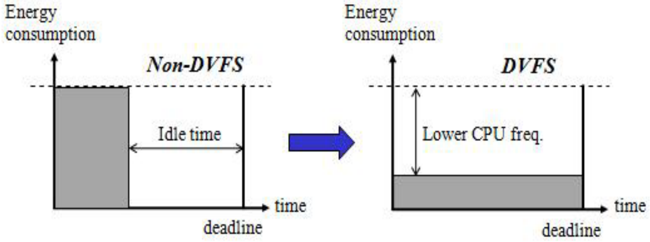


Fig. 1. A typical example of DVFS.

As mentioned earlier, both prediction-based and feedback-based methods can be applied to DVFS. Although the prediction-based DVFS method shows better performance, its effectiveness depends on the accuracy of frame decoding time predictions. There have been a number of prediction-based studies [1]-[5], but accurately predicting decoding times of frames remains a challenge. In addition, prediction-based DVFS methods may not satisfy the perceptual quality of video playback when frame deadlines are missed due to inaccurate predictions.

This paper proposes a combined approach that uses both types of DVFS methods to reduce power and guarantee QoS for video playback. The proposed method employs *Interval-based Linear Interpolation* (ILI) to more accurately estimate frame decoding times. In addition, the proposed method adopts a feedback control technique called *QoS-guaranteed and Low-power Buffering* (QLB), which adaptively switches between quality and low-power modes to guarantee the quality of video playback and increase the opportunity to apply DVFS. Therefore, the proposed approach achieves more power savings than the existing methods and at the same time satisfies QoS of video streams.

The rest of this paper is organized as follows: Section II discusses the related work. Section III presents the proposed method that combines the ILI algorithm for accurate estimation of decoding times and the QLB algorithm to tradeoff between power and video quality. Section IV presents our experimental results and conclusions are given in Section V.

[1] This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2010-(C1090-1021-0003)) .

Jinsung Cho and Dae-Young Kim are with the Department of Computer Engineering, Kyung Hee University, Korea (e-mail: chojs@khu.ac.kr).

Ilyong Cho is with the School of Computer Science & Engineering, Seoul National University, Korea (e-mail:ilyoan@soar.snu.ac.kr).

Ben Lee is with the School of Electrical Engineering and Computer Science, Oregon State University, OR (e-mail: benl@eecs.orst.edu).

Contributed Paper
Manuscript received 01/12/11
Current version published 06/27/11
Electronic version published 06/27/11.

II. Related Work

Bavier *et al.* provided the first study on frame decoding time prediction for video streams [1]. They found that a close relationship exists between the size of a frame and its decoding time, and thus the decoding time can be approximately modeled as a linear function of the frame size. Their study showed that predicted decoding times are accurate to within 25% of actual decoding times.

Choi *et al*. proposed another prediction method using a framed-based history [2]. They divided the required decoding time for each frame into two parts: a frame-dependent (FD) part and a frame-independent (FI) part, and observed that the FD part varies greatly according to the type of incoming frame, whereas the FI part remains constant regardless of the frame type. Therefore, the decoding time of a FD part is estimated based only on the frame type and the FI part is used to compensate for prediction errors for the FD part, which is relatively high.

An experimental verification of power savings in a DVS-based multimedia signal processor system (DVS-MSP) was performed in [3]. Assuming the computational workload required to decode each frame follows the Gaussian distribution, the authors showed that the average power savings increase when there are more voltage-frequency scaling levels.

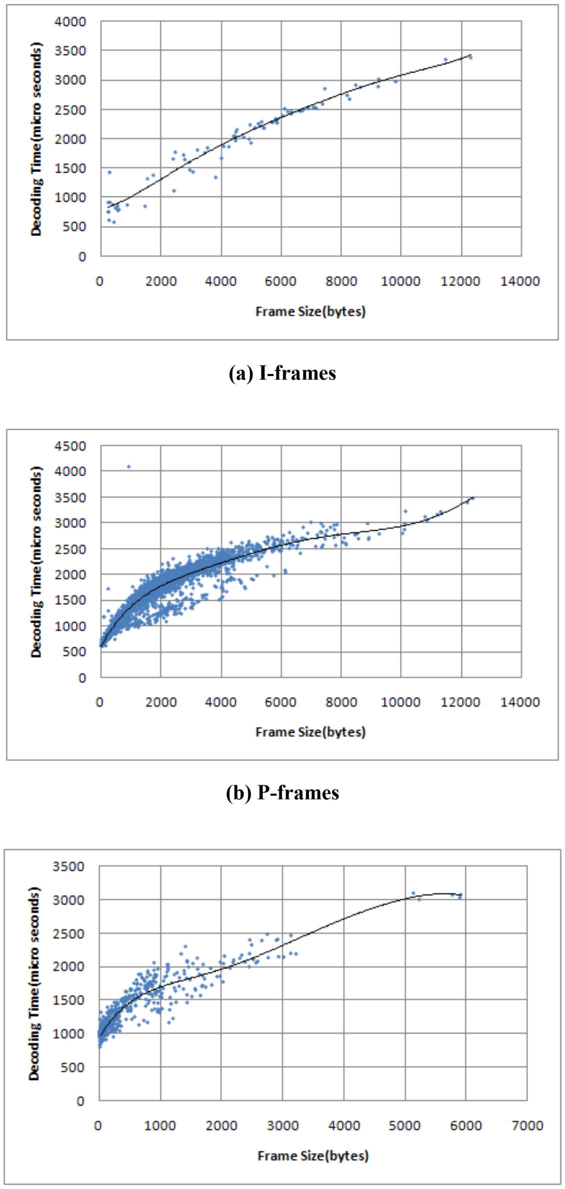
To more accurately estimate decoding times, Tan *et al.* developed a prediction model using regression analysis based on block level statistics of frames [4]. However, their model requires predefined coefficients dependent on types of movies (i.e., high and low motion, etc.), which cannot be computed dynamically.

Most recently, Liu *et al*. presented Chameleon, which is an application-level power management approach using application-domain knowledge. One such application-domain knowledge involves estimating the decoding times of video streams based simply on the average and variance calculated from last *n* frames [5]. Their estimation is simple but the accuracy has been shown to be quite good.

As for the feedback control of decoding statistics of recent frames, Son *et al*. proposed a method where the CPU clock frequency for next GOP is determined based on the drop rate and the slack time of the previous GOP [6]. However, their scheme applies the same CPU frequency across all the frames within a GOP, thus energy-efficiency and/or quality of video playback cannot be guaranteed when the decoding times of frames in a GOP fluctuate.

The dead-zone based control algorithm was presented by Lu *et al.* [7], which maintains decoded frames in a number of buffers [B_{low}, B_{high}] and adjusts the CPU clock frequency according to the number of decoded frames in the buffers. Im *et al*. provided a theoretical analysis on the minimum buffer size and the adjustment of task deadlines to reduce energy consumptions [8], but their method assumes that the best and the worst case decoding times of a frame are available a priori, which is impractical.

Our proposed method incorporates both the prediction and the feedback control mechanisms. The ILI prediction algorithm leads accurate decoding times for DVFS. In addition, the QLB feedback control algorithm switches between low-power and quality modes to increase the opportunity to apply DVFS and guarantee the quality of video playback.



(a) I-frames

(b) P-frames

(c) B-frames

Fig. 2. Regression analysis on the relationship between frame sizes and decoding times: non-linear rather than linear.

III. The Proposed Method

A. The ILI Algorithm

Our ILI algorithm is based on the fact that the relationship between frame sizes and decoding times is non-linear. Figure

2 shows this relationship for I-, P-, and B-frames[2]. As the name suggests, the basic idea of ILI is to subdivide the ordered frame sizes into multiple intervals representing the non-linear size-time relationship, and then apply simple linear interpolation to each interval to obtain decoding time information. Furthermore, the sizes of the intervals are dynamically adjusted to accurately match the characteristics of video clips. The following discusses the detail of the ILI algorithm.

First, *workload*, w, is defined as the time required to decode a frame when CPU runs at the highest voltage and clock frequency, f_{max}, and *deadline*, d, is defined as the time within which decoding of a frame must be completed .

The proposed ILI scheme is based on a set of ordered frame sizes B and k^{th} interval I_k defined as follows:

$$\begin{aligned} &B = (b_1, b_2, b_3, \ldots, b_n), \forall i, j \mid i < j \Rightarrow b_i < b_j, b_0 = 0 \\ &I_k = [b_k, b_{k+1}) \end{aligned} \quad . \qquad (1)$$

Figure 3 illustrates the workload estimation. If the size s of a frame r satisfies the requirement $b_k \le s < b_{k+1}$, then r is defined as belonging to interval I_k and its workload w satisfies the relation $w_k^{avg} \le w < w_{k+1}^{avg}$. Then, linear interpolation can be performed on the interval I_k based on the following equation (see Figure 3):

$$w = \frac{w_{k+1}^{avg} - w_k^{avg}}{b_{k+1} - b_k}(s - b_k) + w_k^{avg} \qquad (2)$$

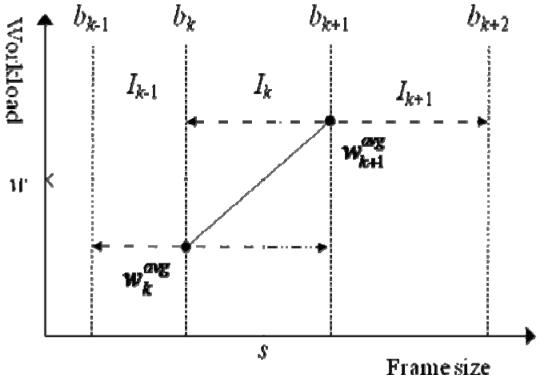


Fig. 3. Workload prediction through the proposed ILI algorithm.

After the estimated workload w for a frame r is obtained from Eq. (2), the CPU clock frequency f with deadline d can be calculated using the following equation:

$$f = \begin{cases} f_{max} & \text{if} \quad w > d \\ \dfrac{w}{d} f_{max} & \text{otherwise} \end{cases} , \qquad (3)$$

where f_{max} is the maximum clock frequency of the CPU. Since the clock frequency, and thus voltage, levels that can be chosen are discrete, the frequency selected for DFVS, f_{actual}, satisfies $f_{actual} > f$ and is the nearest discrete level.

After decoding the frame r at f_{actual}, the actual decoding time w_{actual} can be measured. Since r belongs to interval I_k, w_k^{avg} is recalculated. If w_{max} is defined to be the workload at f_{max}, w_k^{avg} should be calculated using w_{max} instead of w_{actual} because workload w is defined as the time to decode a frame at f_{max}. From w_{actual}, w_{max} is obtained using the following equation:

$$w_{max} = w_{actual} \frac{f_{max}}{f_{actual}} \ . \qquad (4)$$

Finally, w_k^{avg} is updated based on w_{max}. This calculation is performed for each picture type (i.e., I, P, or B).

After a frame is decoded, the *prediction error, e* is calculated by taking the difference between the actual and predicted decoding times. Since recent frames of the same picture type (i.e., I, P, or B) have similar characteristics, prediction errors are also similar, which has also been validated through our experiments. Hence, the average of prediction errors, e, is added to the estimated workload w given in Eq. (2). When calculating the average prediction errors, a separate exponential moving average is maintained for each picture type (i.e., I, P, or B). This compensation of prediction errors results in more accurate estimation.

The performance of the proposed ILI algorithm is affected by the size of the intervals. This is because the number of frames that belong to each interval should be evenly distributed across all the intervals in order to more accurately compute w_k^{avg} for $0 \le k \le n$. If the size of intervals is large, the ILI algorithm may not accurately model the non-linear size-time relationship. In contrast, small interval size may lead to inaccuracy because the number of recorded frame sizes in an interval is insufficient to accurately compute w_k^{avg} .

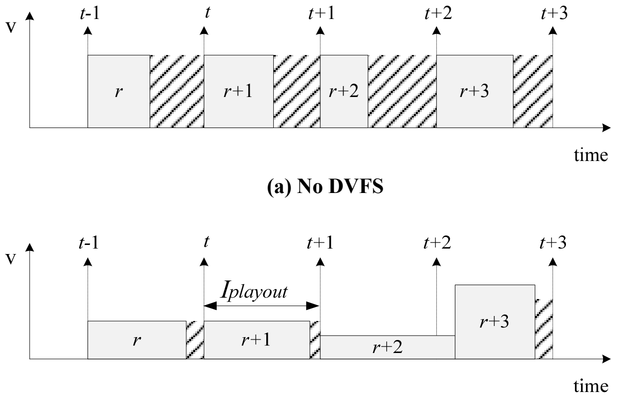
Therefore, the sizes of the intervals are dynamically adjusted to evenly distribute the number of frames that belong to each interval. This is done by subdividing the frame-size axis into small-sized steps, and maintaining information about the number of frames and the sum of workload for each step. Thus, an interval I_k consists of several steps, and the size and the number of intervals are periodically re-adjusted to evenly distribute the number of frames that belong to each interval.

B. The QLB Algorithm

The proposed QLB algorithm relies on a feedback mechanism to switch between low-power and quality modes. This not only guarantees the quality of video playback, which may not be the case for prediction only methods, but also additional energy savings are achieved compared with existing prediction only methods.

[2] These results are based on four video clips listed in Table I. Figure 2 depicts the results from GTA4 video clip but other video clips show similar non-linear relationships between frame sizes and decoding times.

Figures 4(a) and 4(b) illustrate frame decoding without and with DVFS, respectively, where t, t+1, …, t+n represent playout deadlines and $I_{playout}$ represents the playout interval, which is constant for a given video clip. Prediction-based schemes [1]-[5] adopt this concept and the accuracy of the prediction affects the perceptual quality of video playback and power consumption. Figure 4(b) shows an example of a missed deadline for $(r+2)^{th}$ frame due to decoding time prediction error, which occurs when the actual decoding time based on the adjusted clock frequency is longer than $I_{playout}$.

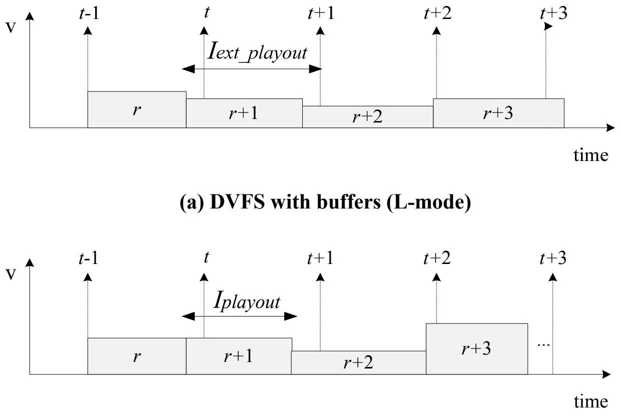


(a) No DVFS

(b) DVFS

Fig. 4. Frame decoding without DVFS vs. with DVFS.

Figures 5(a) and 5(b) illustrate the basic concept behind the QLB scheme. After frame r is decoded and stored in the buffer, decoding of frame r +1 begins immediately as shown in Figure 5(a). If this occurs before the playout time of frame r, then the playout interval of frame r+1 can be extended to $I_{ext\_playout}$, where $I_{ext\_playout} > I_{playout}$. Extended playout intervals increase the opportunity for DVFS and thus lead to lower power consumption.



(a) DVFS with buffers (L-mode)

(b) DVFS with buffers (Q-mode)

Fig. 5. The proposed QLB-based DVFS.

In order to achieve low-power and at the same time maintain quality, the QLB scheme has two modes of operation: Q-mode and L-mode. The *Q-mode* guarantees the quality of video playback, while the *L-mode* provides low power. The applications of $I_{playout}$ and $I_{ext\_playout}$ on frame r+1 in Figure 5(a) and 5(b) illustrate decoding time predictions in Q-mode and L-mode operations, respectively.

The QLB algorithm initially operates in Q-mode to guarantee the quality of video playback. Note that our approach is independent of any prediction method used. Thus, suppose a prediction-based DVFS method, P, is used. P predicts frame decoding times and appropriately adjusts the clock frequency and $I_{playout}$ as shown in Figure 5(b). When the number of decoded frames in the buffer is $n_{threshold}$, it means that there is a sufficient number of decoded frames to extend the decoding time of the current frame by $n_{threshold}$×$I_{playout}$. Therefore, when $I_{ext\_playout}$ satisfies the relation $I_{ext\_playout} \geq (n_{threshold}+1) \times I_{playout}$, the QLB algorithm switches to L-mode to reduce power consumptions, as in Figure 5(a).

Although low-power consumption can be achieved in L-mode, the possibility of deadline misses still exists. Therefore, the interval, $I_{ext\_playout}-n_{threshold} \times I_{playout}$ is used, which will not cause deadline miss unless the prediction error is greater than $n_{threshold}$×$I_{playout}$. When $I_{ext\_playout} < (n_{threshold}+1) \times I_{playout}$, the QLB scheme switches back to Q-mode.

IV. Performance Evaluation

The performance evaluation was conducted on Intel PXA255 board with 400MHz XScale processor running embedded Linux kernel 2.4.16. The power consumption was measured using National Instruments DAQPad. The XScale processor provides four voltage-frequency levels which are (1.0V, 99.5MHz), (1.1V, 199.5MHz), (1.2V, 298.6MHz), and (1.3V, 398.1HMz).

The proposed method was implemented and integrated into *mplayer*, which is one of the popular MPEG players for Linux. In addition, four video clips encoded for handheld devices shown in Table I were used. The performance was compared with FTL [1], which is based on simple linear interpolation, and Chameleon [5], which is simple but provides accurate estimation as mentioned in Section II. The accuracy of decoding time estimation of DVS-MSP [3] is anticipated to be lower than that of Chameleon because DVS-MSP assumes the decoding time for each frame follows the Gaussian distribution without consideration of the relationship between frame sizes and decoding times.

TABLE I
Characteristics of Video Clips used in the Experiment

Title	Genre	Size	Fps	Data rate
GTA4	3D game	240x128	25 fps	300kbps
Hellboy2	Action movie	240x128	25 fps	300kbps
Ratatouille	Animation	192x112	30 fps	250kbps
TVCall	Commercial	192x144	25 fps	300kbps

Figure 6 compares the *average prediction errors*, which represents the accuracy of the prediction-based DVFS algorithms. The prediction error, which was calculated by taking the difference between the actual and predicted decoding times, is independent of the processor used and thus is a very meaningful metric. As can be seen in the figure, FTL has the highest average prediction error, which is then followed by Chameleon. Our ILI provides the most accurate workload estimation, which leads to lower power consumption.

Besides prediction errors, the optimal selection of CPU voltage and clock frequency setting is another related factor that is crucial for obtaining both energy savings and QoS. A frequency, and its corresponding voltage, selection is defined to be *false* when it is not optimal. Furthermore, false selections can be classified into two types: pessimistic and optimistic. A *pessimistic false selection* is when the selected clock frequency is faster than the optimal frequency leading to more power consumption. In contrast, an *optimistic false selection* is when the selected frequency is slower than the optimal frequency, which may not satisfy QoS.

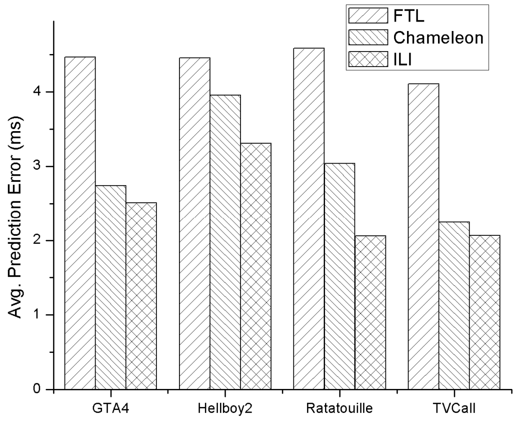


Fig. 6. Average prediction errors: interval-based linear interpolation method using non-linear relationships of frame sizes and decoding times provides the most accurate estimation.

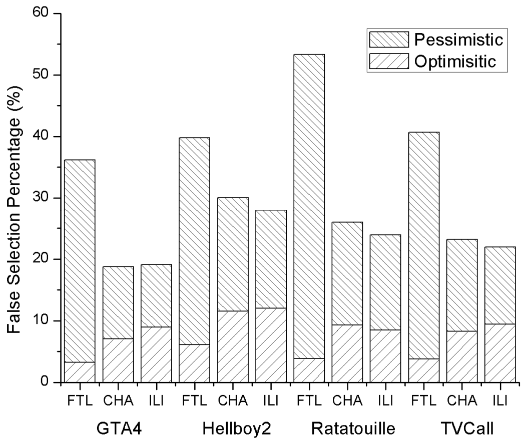


Fig. 7. False selections of CPU voltage and clock frequency: more accurate prediction leads to less false selection percentage.

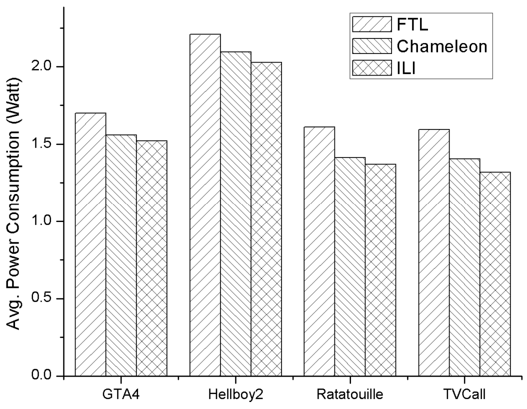


Fig. 8. Average power consumption: less pessimistic false frequency selections result in more power savings.

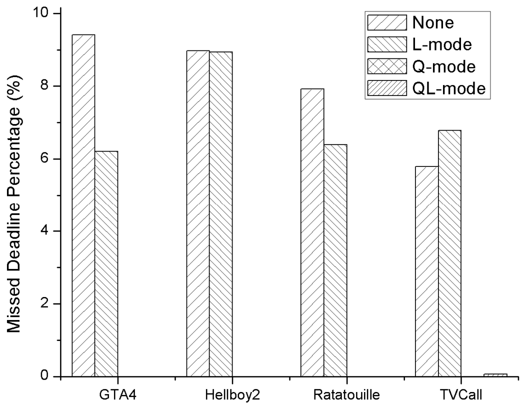


Fig. 9. Missed deadline percentage of ILI+QLB: Q-mode of QLB satisfies the QoS of video playback.

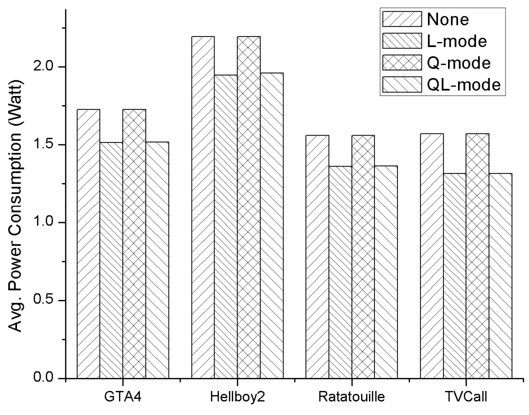
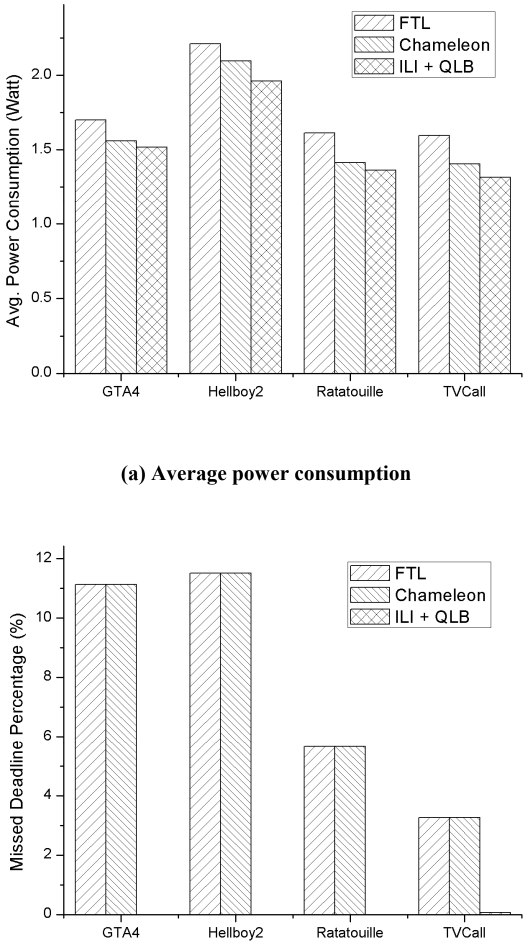


Fig. 10. Average power consumption of ILI+QLB: L-mode of QLB obtains more power savings additionally.



(a) Average power consumption

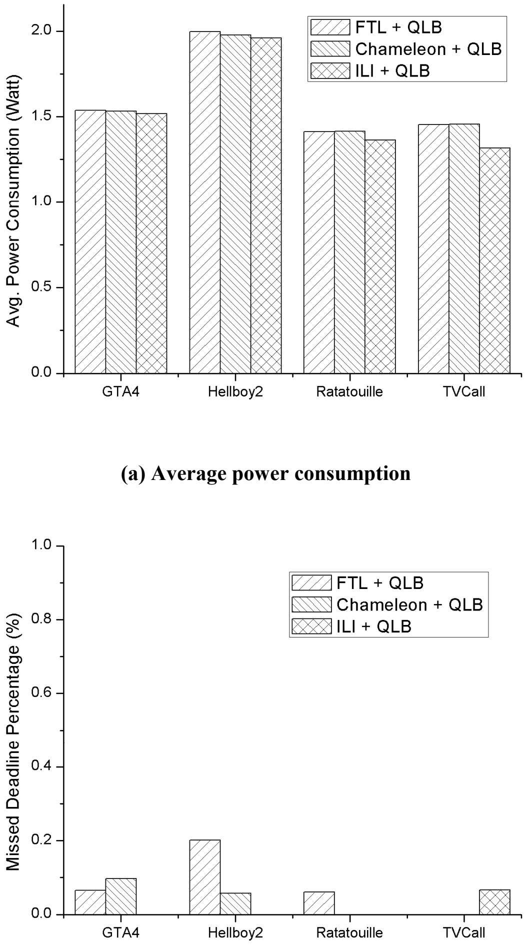
(b) Missed deadline percentage

Fig. 11. Comparison of the proposed scheme with existing schemes: Our scheme outperforms existing schemes in terms of average power consumption and missed deadline percentage.

Figure 7 analyzes the percentage of false selections. As shown in both Figures 6 and 7, the more accurate the workload prediction, the lower the percentage of false selection.

Moreover, ILI has lower percentage of pessimistic false clock frequency selection than FTL and Chameleon, which will lead to lower power consumption. This is verified in Figure 8, which shows that our ILI algorithm reduces power consumptions by 12.8% and 3.7% compared with FTL and Chameleon, respectively. Note that the average power consumption can be further reduced with processors that have more frequency/voltage settings.

Figures 9 and 10 show the performance of our QLB algorithm with ILI as the decoding time prediction, where None represents the ILI algorithm without QLB, and L-mode, Q-mode, and QL-mode apply the low-power mode, the quality mode, and both Q-mode and L-mode of the QLB algorithm, respectively.



(a) Average power consumption

(b) Missed deadline percentage

Fig. 12. Comparison of the proposed scheme with QLB-incorporated existing schemes: QLB algorithm can improve the performance of the prediction-based schemes in terms of average power consumption and missed deadline percentage.

Figure 9 shows the percentage of missed deadlines. As can be seen from the figure, None results in many deadline misses. L-mode also results in large percentage of missed deadlines since it aims for low power consumption. In contrast, both Q-mode and QL-mode result in significantly lower missed deadline percentage. This is because, Q-mode aims for quality and thus only $I_{playout}$ is used even when extended playout intervals (i.e., $I_{ext\_playout}$) are available.

Figure 10 shows the average power consumption of the proposed scheme. As described in Section II, extended playout intervals in L-mode can lead to lower clock frequency levels resulting in lower power consumption. This effect is very clearly shown in Figure 10, where both L-mode and QL-mode resulted in the most power savings.

Note that the results in Figures 9 and 10 are based on $n_{threshold} = 1$. This is because our experiments showed that on average $n_{threshold}$ is less than equal one due to the decoding speed of *mplayer* running on the XScale processor. Therefore, the proposed QLB algorithm guarantees quality and achieves low-power consumption in video playback with a buffer size equal to $n_{threshold}+1 = 2$ frames.

Figure 11 compares the performance of the proposed method (ILI+QLB) against FTL and Chameleon. In comparison to Figure 8 that showed the power consumption only with ILI, the combination of ILI and QLB leads to more power savings. The proposed method reduces the total power consumptions by 13.7% and 4.8% compared with FTL and Chameleon, respectively (see Figure 11(a)). At the same time, Figure 11(b) shows that ILI+QLB results in guaranteed QoS of video playback.

As mentioned in Section III, the QLB algorithm is independent of any prediction-based scheme. In order to examine the effectiveness of QLB alone, Figure 12 shows the performance when it was incorporated into FTL and Chameleon. With the help of QLB, all three methods show similar average power consumption results and missed deadline percentage, but the proposed method slightly outperforms the other two methods due to superior accuracy of ILI.

VI. Conclusion

This paper proposed a DVFS technique that reduces power while guaranteeing quality of video playback for mobile multimedia devices. The proposed method consists of two parts: an accurate dynamic workload prediction algorithm using interval-based linear interpolation (ILI) and a hybrid buffering algorithm called QLB. Our simulation results show that the proposed method reduces power consumption by 4.8%-13.7% compared with two existing dynamic decoding time prediction methods, and at the same time, guarantees the QoS of video playback.

References

[1] A. Bavier, A. Brady, M. Larry, and L. Peterson, "Predicting MPEG execution times," *Proc. of ACM SIGMETRICS*, 1998.

[2] K. Choi, W. Cheng, and M. Pedram, "Frame-based dynamic voltage and frequency scaling for an MPEG player," *Proc. of IEEE/ACM International Conference on Computer Aided Design*, 2002.

[3] B.-I. Kim and T.-G. Chang, "A low power multimedia system based on the microscopic dynamic voltage scaling (DVS)," *Proc. of IEEE International Conference on Consumer Electronics*, 2006.

[4] Y. Tan, P. Malani, Q. Qiu, and Q. Wu, "Workload prediction and dynamic voltage scaling for MPEG decoding," *Proc. of International Conference on Asia South Pacific Design Automation*, 2006.

[5] X. Liu, P. Shenoy, M. D. Corner, "Chameleon: application-level power management," *IEEE Transactions on Mobile Computing*, Vol. 7, No. 8, 2008.

[6] D. Son, C. Yu, and H. Kim, "Dynamic voltage scaling on MPEG decoding," *Proc. of International Conference on Parallel and Distributed Systems*, 2001.

[7] Z. Lu, J. Lach, M. Stan,, and K. Skadron, "Reducing multimedia decode power using feedback control," *Proc. of International Conference on Computer Design*, 2003.

[8] C. Im, S. Ha, and H. Kim, "Dynamic voltage scheduling with buffers in low-power multimedia applications," *ACM Transactions on Embedded Computing Systems*, Vol. 3, No. 4, 2004.

[9] B. Lee, E. Nurvitadhi, R. Dixit, C. Yu, and M. C. Kim, "Dynamic voltage scaling techniques for power efficient video coding," *Journal of Systems Architecture*, Vol. 41, No. 10-11, 2005.

[10] D. Snowdon, S. Ruocco and G. Heiser, "Power management and dynamic voltage scaling: myths and facts," *Proc. of International Workshop on Power Aware Real-time Computing*, Sep., 2005.

Biographies

Jinsung Cho (M'04) received his B.S., M.S, and Ph.D. degrees in computer engineering from Seoul National University, South Korea, in 1992, 1994, and 2000, respectively. He was a visiting researcher at IBM T.J. Watson Research Center in 1998 and a research staff at SAMSUNG Electronics in 1999-2003. Currently, he is an associate professor of Department of Computer Engineering at Kyung Hee University, Korea. His research interests include mobile networking & computing, wireless sensor & body networks, and embedded systems & software.

Ilyong Cho received his B.S. degree in computer engineering from Kyung Hee University, Korea in 2008. He is currently a master student at Seoul National University, Korea. His research interests include combinatorial optimization and computational complexity theory.

Dae-Young Kim received his B.S. degree in electronics engineering, M.S. and Ph.D. degree in computer engineering from Kyung Hee University, Korea in 2004, 2006, and 2010, respectively. He is currently with the Communication R&D Lab., LIG Nex1 Co., Ltd., Korea. His research interests include mobile networking & computing, wireless sensor networking, and embedded systems.

Ben Lee (M'91) received his B.E. degree in Electrical Engineering in 1984 from the Department of Electrical Engineering at State University of New York (SUNY) at Stony Brook, and his Ph.D. degree in Computer Engineering in 1991 from the Department Electrical and Computer Engineering at the Pennsylvania State University. He is currently an Associate Professor of School of Electrical Engineering and Computer Science at Oregon State University. He received the Loyd Carter Award for Outstanding and Inspirational Teaching and the Alumni Professor Award for Outstanding Contribution to the College and the University from the OSU College of Engineering in 1994 and 2005, respectively. He also received the HKN Innovation Teaching Award from Eta Kappa Nu, School of Electrical Engineering and Computer Science, 2008. He has been on program and organizing committees for numerous international conferences. He is currently the chair of IEEE Workshop on Pervasive Wireless Networking 2011 (PWN11), and a program committee member for International Conference on Parallel Processing, 2011 (ICPP 2011) and International Conference on ICT Convergence 2010 (ICTC 2010). He is also on the editorial board for ETRI Journal and Journal of Information Processing. His research interests include multimedia systems, wireless networks, embedded systems, computer architecture, multithreading and thread-level speculation, and parallel and distributed systems.