University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

2011

Impact of Channel Estimation Errors on Effectiveness of Eigenvector-Based Jamming for Physical Layer Security in Wireless Networks

James M. Taylor
University of Nebraska-Lincoln, jmtaylor@unlnotes.unl.edu

Michael Hempel
University of Nebraska-Lincoln, mhempel2@unl.edu

Hamid Sharif
University of Nebraska-Lincoln, hsharif@unl.edu

Shichuan Ma
University of Nebraska-Lincoln, sma@unlnotes.unl.edu

Yaoqing Lamar Yang
University of Nebraska-Lincoln, yyang3@unl.edu

Battery-aware Multimedia Coding Optimization by Dynamic Frequency Scaling

Jianxin Sun, Xueyi Wang, Song Ci

Department of Computer and Electronics Engineering

University of Nebraska-Lincoln

NE 68182, USA

Email : {jianxin.sun, xueyiwang}@huskers.unl.edu, sci@engr.unl.edu

Abstract— Nowadays, multimedia application and video streaming have gained a great popularity because of the boasting development on mobile hardware, and battery driven devices have emerged with a tremendous speed. Due to the important issue of battery efficiency on mobile devices, many optimization algorithms have being proposed toward various of battery powered platforms and scenarios. Most of the provided solutions choose to aim at the minimization of energy usage under a given task scheduling by adjusting parameters reside in the processes of different schemes. However, the battery discharging characteristics and its instant output pattern are still ignored if the optimization is done only from high level adjustment. In this paper, we propose a battery-aware optimization framework toward the H.264 video coding by applying dynamic frequency scaling on hardware platform. The CPU frequency can be dynamically adjusted according to the instant status of battery in order to maximize the number of the coding frame. Experimental results indicate the efficiency and effectiveness of the proposed optimization framework.

Index Terms— battery-aware, multimedia, video coding, H.264, dynamic frequency scaling.

I. Introduction

Technology advances in video compression and transmission over wireless communication networks have enabled mobile multimedia on portable wireless devices, such as cellular phones, laptop computers connected to WLANs, and cameras in surveillance and environmental tracking systems. Video coding and streaming are also envisioned in an increasing number of applications in the areas of battlefield intelligence, reconnaissance, public security, and telemedicine. Present 3G and emerging 4G wireless systems, and IEEE 802.11 WLAN/WMAN have dramatically increased the transmission bandwidth, and generated a great amount of users on video streaming applications. Although wireless video communications is highly desirable, a primary limitation in wireless systems is the basic design architecture that most mobile devices are typically powered by batteries with limited energy capacity. This limitation is of fundamental importance due to the high energy consumption rate in nowadays video encoding standard, like high definition(HD) or 3D video coding and compression. From the perspective of battery-aware design and power management, how to wisely decide the energy allocation is a critical issue in order to efficiently use battery energy to guarantee a optimal multimedia coding requirement.

In literature, complexity control parameters in those steps of a generic video encoder was investigated in [1]. However, the battery dynamics in a real system was not considered in that work. Although energy efficiency of video coding was studied in [2] [3] [4], only power was considered by defining the mathematic relationship between coding parameters and power consumption instead of battery energy consumption which is significant in revealing the battery working manner in video coding. In the literature of low level optimizations like power management and DVFS, no coding optimization is addressed in [5], and [6] presents an optimization based on task in realtime instead of dynamic changing of battery power source. [7] gives a solution under the constraint of QoS, however, battery energy consumption needs to be analyzed. Currently, no dedicated analytical framework and battery-aware optimization under the context of H.264 codec has been performed.

In this paper, we develop a systematic optimization framework for video coding under the constraint of a fixed battery with given capacity. By considering the nonlinear character of battery and applying dynamic frequency scaling based on the instant state of battery, our framework can maximize the total number of frames which can be coded under a given battery.

This paper is organized as follows. Section II presents the formulation of the problem to solve. Section III, presents battery discharging profile and nonlinear effect. In Section IV measurements of energy consumption of video coding under H.264 codec is introduced. A systematic optimization framework is proposed in Section V. Section VI presents simulation and experimental results. Concluding remarks are given in Section VII.

II. Problem Statement

A. Energy Consumption on Video Coding

To analyze the power consumption on video coding in a portable device, we first need to determine the computational complexity of video coding at the encoder, which is measured by the running time of processor on video coding processing. Then, based on the power management of the underlying microprocessor in the mobile device, such as DVS CMOS circuits design [8], we can measure the energy consumed by those processes.

As shown in Figure 1, the major modules in a typical video encoding system include motion estimation (ME) and
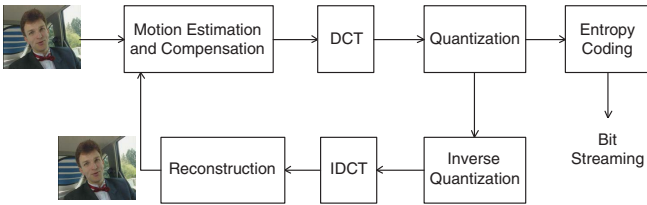
Fig. 1. Block diagram of a typical video encoder. For INTRA MB or frames, motion estimation and compensation are not needed.

compensation, DCT, quantization, entropy encoding of the quantized DCT coefficients, inverse quantization, inverse DCT, picture reconstruction, and interpolation. In literature, plenty of research results has been reported to evaluate and reduce the computational complexity and thereby the power consumption of these modules [1], [9], [10]. It has been shown that, for each module in Figure 1, a set of system parameters of video codec can be selected to control the computational complexity of that module. For example, according to [1], the ME module could use the number of sum of absolute difference (SAD) as the complexity control parameter, while the modules of DCT, quantization, inverse quantization, inverse DCT, picture reconstruction may use a same complexity control parameter - the number of macroblocks (MB), which has nonzero DCT coefficients after quantization in a video frame. Let $\Lambda = [\lambda_1, \lambda_2, \cdots, \lambda_I]$ be the set of control parameters to control the computational complexity of those modules.

Therefore, the overall codec complexity (in terms of processor workload) ξ is a function of video processing parameters Λ, denoted by $\xi(\Lambda)$. Hence, the energy consumption of the underlying microprocessor to compress a video clip, denoted by E_e, is a function of processor workload x_i, which is also a function of Λ, denoted by

$$E = \Phi(x_i) \cdot t = E(\Lambda) = E(\lambda_1, \lambda_2, \cdots, \lambda_I), \quad (1)$$

where $\Phi(\cdot)$ is the power consumption model of the microprocessor [11], which can be obtained by measurement. For example, the power consumption model of the Intel PXA255 XScale processor is well approximated by $\Phi(x_i) = \beta \times x_i^\gamma$, where $\gamma = 2.5$, and β is a constant [12].

B. Problem Formulation

In order to test the performance of video coding under different dynamic frequency scaling, we need to fix all the complexity control parameters to build a task with stable complexity. For example we set all the video precessing parameters as a set of constant.

$$\Lambda = \Lambda_0, \quad (2)$$

Let N be the total number of video frames that a specific hardware platform can code under a given battery. It has been reported that different CPU working frequency will result in different coding efficiency, which means CPU frequency can effect on the result of how many video frames a encoder can process under a battery with a fix capacity. The objective of the proposed framework is to determine a profile of optimal
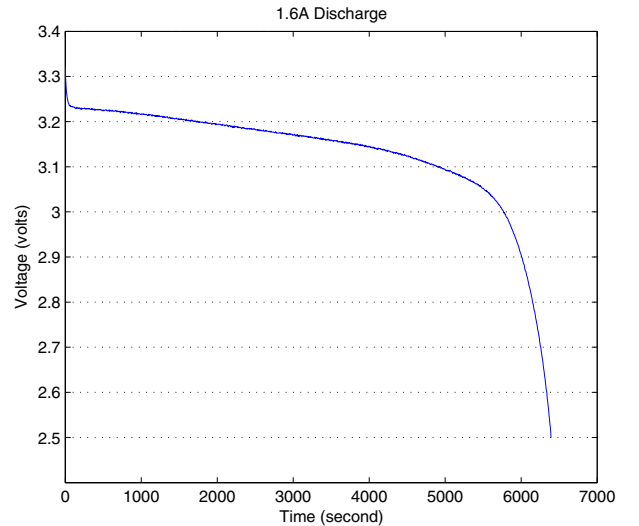
CPU frequency scaling options to perform the most encoded video frames N under the constraint of a fixed battery total energy $E_{\max}$, which can be formulated as

$$\max N(f)$$
$$s.t. : E(\Lambda_0, f) = E_{\Lambda_0}(f) \leq E_{\max}. \quad (3)$$

III. BATTERY DISCHARGING PATTERN

Mobile devices used for video coding are mostly driven by battery. Once the battery becomes fully discharged, a battery-powered portable electronic system goes off-line. Available battery capacity has a nonlinear relationship with its discharging current due to the battery current effect. That means a battery tends to provide more energy at a lower discharge current. Figure 2 shows a typical battery discharging process. In this case we use 26650P 2.6Ah lithium iron battery, and discharge it with a constant current of 1.6A under the temperature of $20°C$

We can observe from the figure that battery has a nonlinear decrease of voltage after it starts to discharge, and the voltage drops very fast when it is near the limit of its capacity.

IV. MEASUREMENTS OF ENERGY CONSUMPTION OF VIDEO CODING

In order to test the execution efficiency under different CPU frequency configurations, we design several tests to get a big picture of battery energy consumption of each configuration on the same hardware platform. In this work, we have established a testbed with considerations of high-resolution battery measurements. Figure 3 shows the hardware measurement system. The main part of our hardware platform is the battery system board, which is in charge of battery power management and battery-related data collection. H.264 video codec is running on a Linux-based Imote2 wireless sensor node with a PXA271 XScale processor. In this figure, the 8-cell battery pack is used to power the platform with the output voltage of $4.2\ V$. A PC
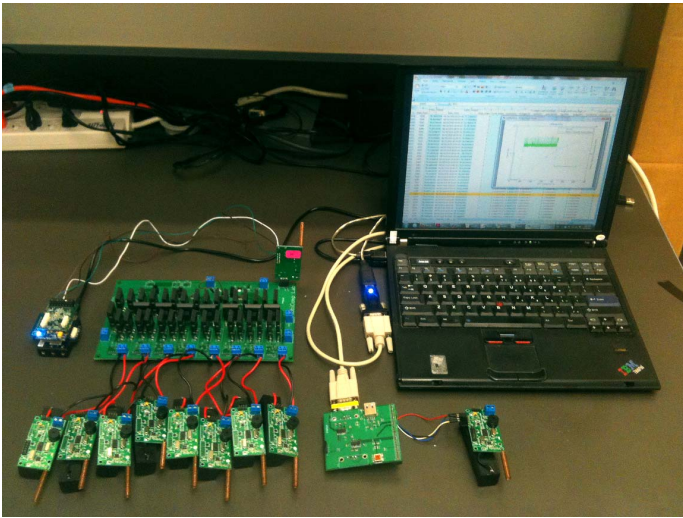

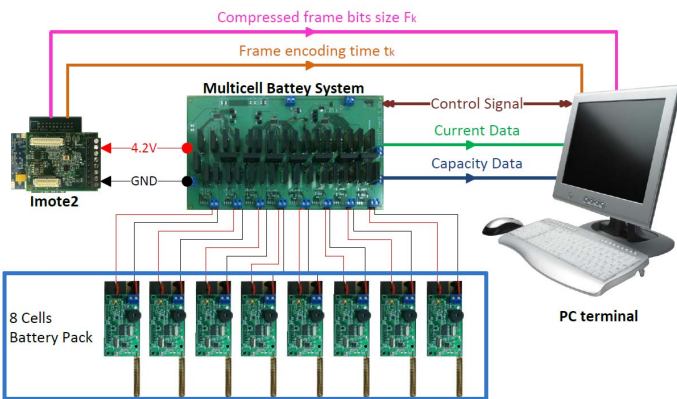Fig. 2. 26650P 2.6Ah lithium iron battery discharging with 1.6A

Fig. 3.   Hardware platform at work

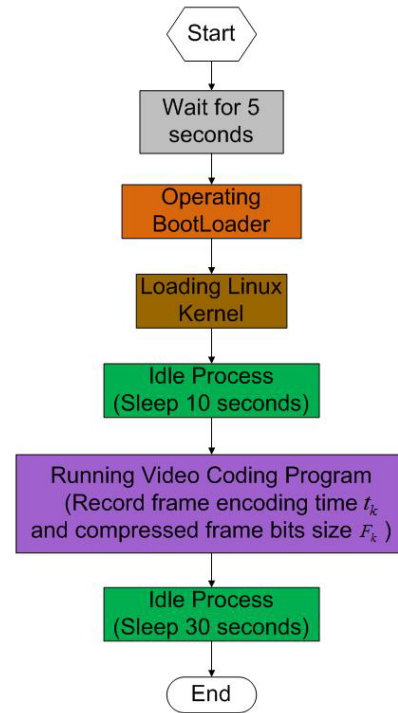Fig. 4.   Hardware measurement system

Fig. 5.   Codec running flow-process diagram

the dominate effector to determine the execution performance (task finish time) and energy consumption, that means if the frequency is fixed, performance and energy consumption is less likely to be effected by adjusting supplying voltage. As a result, higher frequency is a more optimal option to maximize the working performance, so it is resealable to chose a high CPU frequency scaling as long as the battery allows to supply the threshold voltage.

TABLE I
OPTIMIZATION ALGORITHM FOR 8 FREQUENCY OPTIONS SCALING CPU

For each frame n = $1, 2, \cdots 8$
Detect the current battery output voltage V_c
For i=1 :1 :8
If $V_c > f_i$
Execute the coding precess of current frame
If else $V_c \leq f_8$
Stop
Next i
Next n

terminal receives and records two kinds of data from the multi-cell battery system board : the current value and the battery capacity consumption. Meanwhile, during the coding process, Imote2 records and transmits the related key information to the PC terminal, such as coded frame bits F_k and frame coding time t_k. Figure 4 is the diagram of the hardware platform.

H.264 codec is stored in the flash memory of Imote2. Figure 5 shows the flow-process diagram of how a codec runs during the measurement. Five seconds after the measurement begins, the battery system board starts to power Imote2 and activates the bootloader in ROM, then linux kernel begins loading. Before the codec begin to run, the system rests for 10 seconds to initialize the time base and set the starting point as t_0 to start codec. During the video coding process, when compression of the kth video frame is finished, the codec will output the processing time of that frame t_k and the size of coded frame bits F_k.

Figure 6 is the measurement result by selecting CPU frequency to encode 50 frames Foreman video clip in QCIF formate. From figure 7 we can observe in columns that higher frequency result in less energy consumption, and that means based on Imote2 platform, higher frequency has a better energy efficiency. We can also observe that in rows frequency is

V. OPTIMIZATION FRAMEWORK

The optimization algorithm give a method to adjust CPU operating frequency on the fly according to the current status of the battery voltage. As we observed and discussed in section IV, in order to perform an efficient coding process, we need to assign a higher priority to higher CPU frequency option. Then the dynamic frequency scaling operator in the encoder automatically apply a priority oriented policy to execute the video coding. However, considering the battery discharging characteristic discussed in section III, battery can no longer
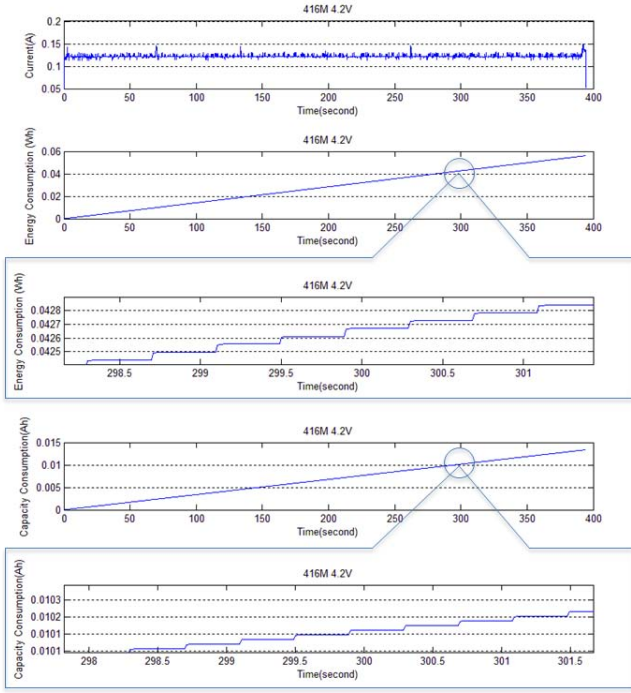
Fig. 6. Measurement result of coding 50-frames Foreman video clip with $416MHz$ frequency

supply a specific frequency when its voltage value drop bellow a certain threshold which is the minimized voltage to support this frequency. And the whole system is down and encoder stops coding the video frames. The optimization framework we propose is to let the CPU keep working at different frequency as operating time goes by, and the frequency scaling is based on the battery current voltage : If the battery voltage drop down to a level which is not sufficient to supply the current CPU frequency, the optimization algorithm in encoder automatically adjust the frequency down to the next highest level of CPU frequency. As a result, the coding process can go on operating at a lower speed. By applying this algorithm, the coding process can work to achieve the maximal workload and the battery can deplete more energy then the case without CPU frequency scaling. Table I shows the flow-process diagram of optimization algorithm applied on 8 frequency options scaling CPU. In this case, the CPU can be scaled from the highest frequency of f_1 to the lowest frequency of f_8. Before the execution of coding process of each frame, the power management component check if the current battery voltage is high enough to support the highest frequency to perform the highest efficient coding toward this frame. If it dose not fit the requirement, then the power management component choose the secondly highest frequency until the most efficient frequency can be find and set it as the CPU operating frequency for the next frame. If the case, that the current voltage cannot satisfy all the lowest frequency requirement of voltage, the system shuts down the platform.

VI. EXPERIMENTAL RESULT

We have conducted experiments to show the performance of the proposed framework. Four video sequences with varied contents (Carphone, Foreman, Coastguard, Mobile) in QCIF format are adopted in our work. An Imote2 wireless sensor node with a PXA271 XScale processor is used in experiments. A smart battery measurement system from Arbin Instruments is in charge of monitoring and recording all the desired battery operation data. The frames of each video sequence is repeatedly encoded with H.264 codec (JVT reference software, JM 16.2 [13]). The permissible QP value is set as 24, frame rate is set as 30, number of previous frames used for inter motion search is 3, search range is set as 16. All video frames except the first one are coded as inter frames. To reduce error propagation due to packet loss, ten random I Macroblocks were inserted into each frame. The video frames are packetized such that each packet/slice contains one row of MBs, which enables a good balance between error robustness and compression efficiency.

In the experiment, we choose the 3 possible scaling options of Imote2 which are $416MHz$, $208MHz$, $104MHz$ with the respective supplying voltage limits of 3.5V, 3.4V, 3.3V. In order to test and verify the improvement of the proposed algorithm, we first run the encoder under these CPU frequencies separately. Figure 8, 9, 10 are the results by using the same $750mah$ lithium iron with those different three frequencies under the temperature of $20°C$. And their respective number of coding frames are 1909, 1658, 1437. We can observe that although the low frequency option can let CPU run longer time, but it does not do much work due to its low efficiency. And high frequency option stop working due to the its quick voltage dropping down, but it perform the most coding work.

The figure 11 is the result after applying our proposed optimization framework. The two frequency turning points are marked by arrows. The first frequency scaling happened at the point when battery output voltage drop to $3.5V$, and the frequency changed from $416MHz$ to $208MHz$. The second frequency scaling happened at the point when battery output voltage drop to $3.4V$, and the frequency changed from $208MHz$ to $104MHz$. And the total number achieved by this algorithm is 2028. The improvement of the proposed algorithm against the three CPU frequencies are respectively 6.23%, 22.31%, and 41.13%.

VII. CONCLUSION

In this work, we have developed a systematic optimization framework for video coding systems powered by battery. A new battery-aware video coding scheme to maximize the total coded frames under a given constraint of a total battery energy. Experimental results verified the efficiency and effectiveness of the proposed optimization framework. The experiments and dynamic frequency scaling carried out in this work provides design insights for resource allocation in future mobile multimedia systems.
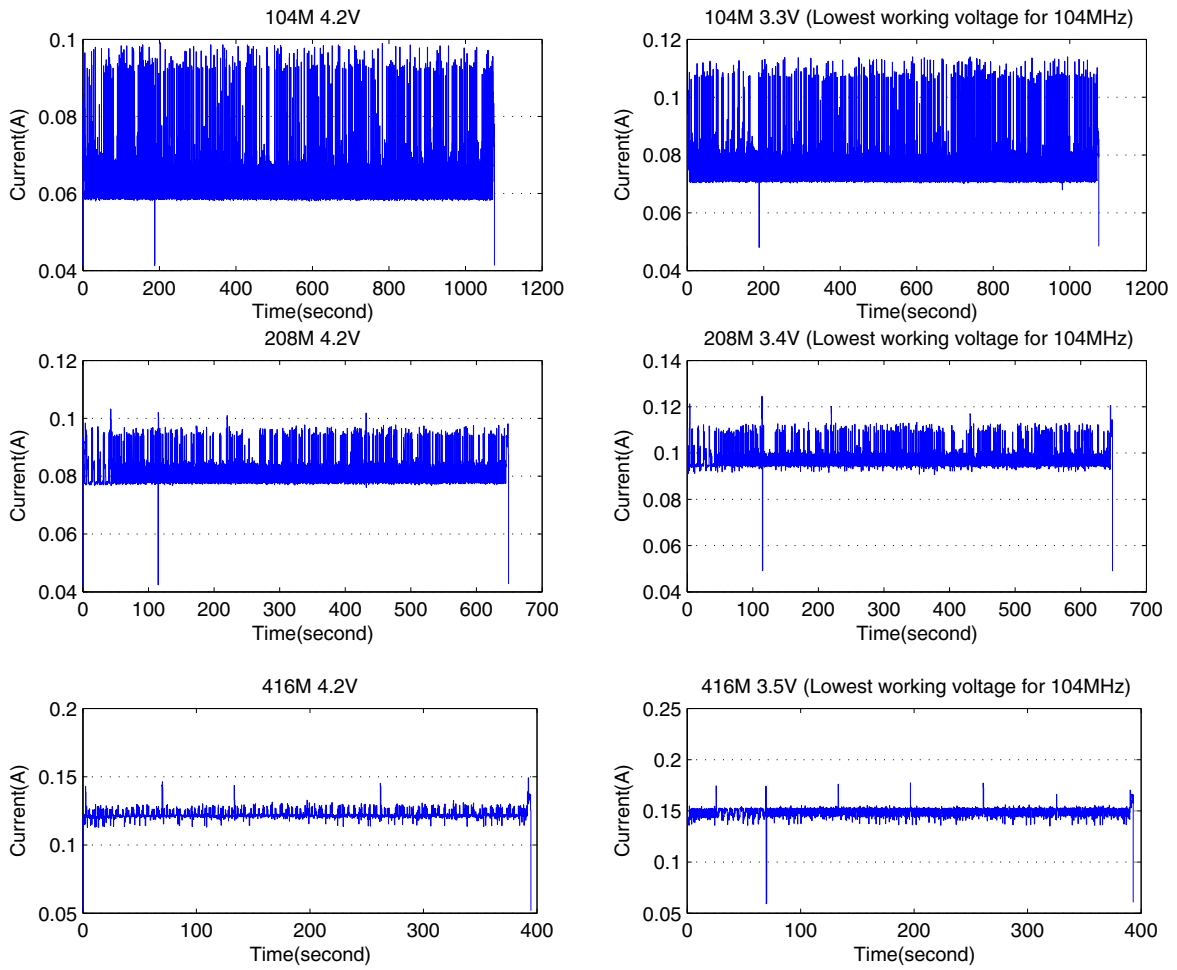
Fig. 7. Measurement result of coding 50-frames Foreman video clip with $104MHz$, $208MHz$, and $416MHz$ under their highest voltage level of $4.2V$ and lowest working voltage level of $3.3V$, 3.4 and 3.5respectively
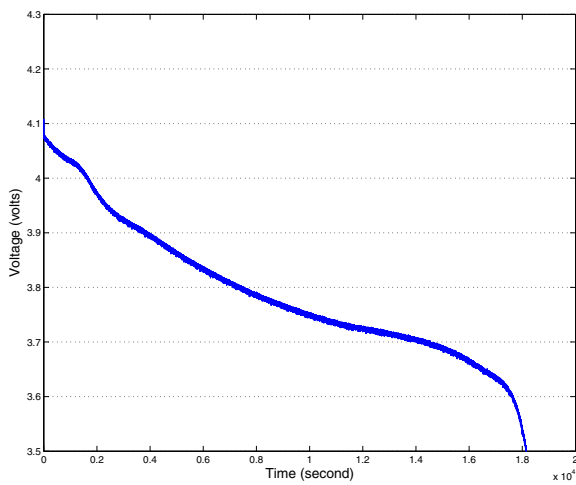

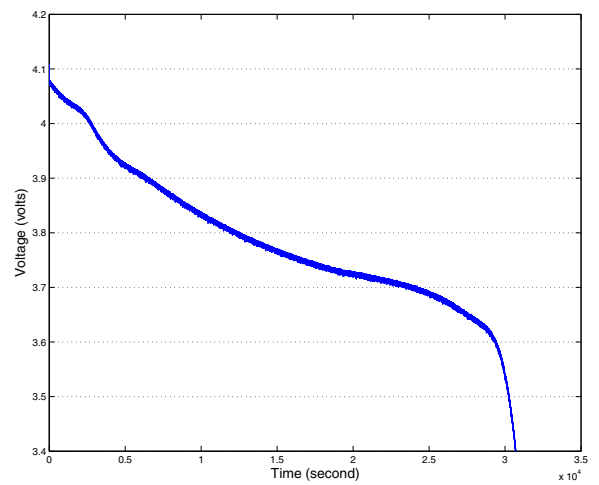Fig. 8. Battery voltage curve with $416MHz$ frequency

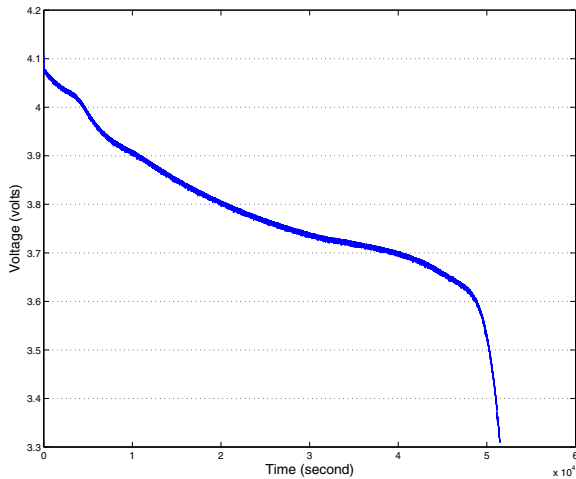Fig. 9. Battery voltage curve with $208MHz$ frequency
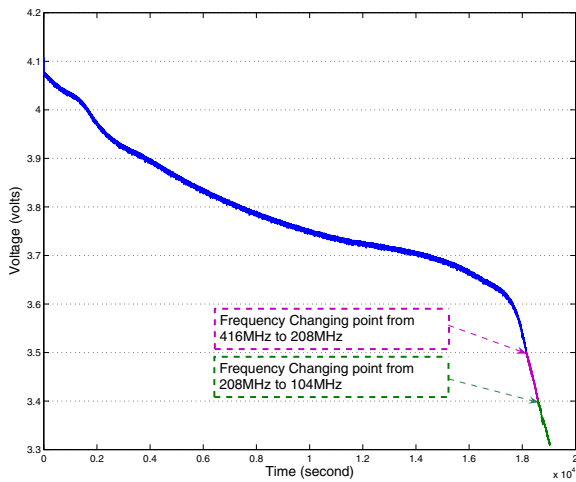
Fig. 10.   Battery voltage curve with $104MHz$ frequency

Fig. 11.   Battery voltage curve after applying optimization algorithm

REFERENCES

[1] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-Rate-Distortion Analysis for Wireless Video Communication Under Energy Constraints," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 5, pp. 645–658, May 2005.

[2] Z. He and D. Wu, "Resource Allocation and Performance Analysis of Wireless Video Sensors," *Circuits and Systems for Video Technology*, vol. 15, no. 5, pp. 645–658, May 2005.

[3] A. K. Katsaggelos, F. Zhai, Y. Eisenberg, and R. Berry, "Energy-efficient wireless video coding and delivery," *IEEE Journal Wireless Communications*, vol. 12, pp. 24–30, Aug. 2005.

[4] X. Lu, Y. Wang, and E. Erkip, "Power efficient h.263 video transmission over wireless channels," in *International Conference Image Processing*, Dec. 2002, pp. I–533–I–536.

[5] Y. Tan, P. Malani, Q. Qiu, and Q. Wu, "Workload Prediction and Dynamic Voltage Scaling for MPEG Decoding," p. 6 pp, Mar. 2006.

[6] S. Liu, Q. Qiu, and Q. Wu, "Energy Aware Dynamic Voltage and Frequency Selection for Real-Time Systems withEnergy Harvesting," pp. 236–241, Apr. 2008.

[7] Q. Qiu, Q. Wu, and M. Pedram, "Energy Aware Dynamic Voltage and Frequency Selection for Real-Time Systems withEnergy Harvesting," pp. 834–839, May 2005.

[8] R. Min, T. Furrer, and A. Chandrakasan, "Dynamic voltage scaling techniques for distributed microsensor networks," in *Proc. IEEE Computer Society Workshop VLSI*, Apr. 2000, pp. 43–46.

[9] D. N. Kwon, P. F. Driessen, A. Basso, and P. Agathoklis, "Performance and computational complexity optimization in configurable hybrid video coding system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 1, pp. 31–42, Jan. 2006.

[10] C.-J. Lian, S.-Y. Chien, C.-P. Lin, P.-C. Tseng, and L.-G. Chen, "Power-aware multimedia : Concepts and design perspectives," *IEEE Circuits and Systems Magazine*, pp. 26–34, 2007.

[11] T. Burd and R. Broderson, "Processor design for portable systems," *J. VLSI Signal Process*, vol. 13, no. 2, pp. 203–222, Aug. 1996.

[12] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Energy Minimization of Portable Video Communication Devices Based on Power-Rate-Distortion Optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 5, pp. 596–608, May 2008.

[13] H.264 JM 12.2 reference software [Online]. Available : http ://iphome.hhi.de/suehring/tml/download/.