Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 234131, 14 pages
doi:10.1155/2010/234131

Research Article

ESVD: An Integrated Energy Scalable Framework for Low-Power Video Decoding Systems

Wen Ji,[1] Min Chen,[2] Xiaohu Ge,[3] Peng Li,[1] and Yiqiang Chen[1]

[1] *Institute of Computing Technology, CAS, Beijing 100190, China*
[2] *Department of Electrical and Computer Engineering, University of British Columbia, vancouver, BC, Canada V6T 1Z4*
[3] *Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, Hubei 234131, China*

Correspondence should be addressed to Xiaohu Ge, xhge@mail.hust.edu.cn

Received 1 April 2010; Accepted 6 June 2010

Academic Editor: Liang Zhou

Video applications using mobile wireless devices are a challenging task due to the limited capacity of batteries. The higher complex functionality of video decoding needs high resource requirements. Thus, power efficient control has become more critical design with devices integrating complex video processing techniques. Previous works on power efficient control in video decoding systems often aim at the low complexity design and not explicitly consider the scalable impact of subfunctions in decoding process, and seldom consider the relationship with the features of compressed video date. This paper is dedicated to developing an energy-scalable video decoding (ESVD) strategy for energy-limited mobile terminals. First, ESVE can dynamically adapt the variable energy resources due to the device aware technique. Second, ESVD combines the decoder control with decoded data, through classifying the data into different partition profiles according to its characteristics. Third, it introduces utility theoretical analysis during the resource allocation process, so as to maximize the resource utilization. Finally, it adapts the energy resource as different energy budget and generates the scalable video decoding output under energy-limited systems. Experimental results demonstrate the efficiency of the proposed approach.

1. Introduction

With the growing popularity of portable video applications, such as portable video smart phones, mobile video terminals such as PDA, and vehicle DVD devices energy consumption of video decoders becomes an important design requirement. Lots of compression codecs are issued for the several major video code standards, including MPEG4/2, H.264/3, and AVS. Generally, decoders focus on the performance while rarely support dynamic decoding process to meet the variable energy resources. However, most portable video application devices operate on batteries with limited-energy supply. The capacity of battery in portable devices is limited, as well as the usable capacity of the battery declines with using time. Thus, power should be used economically to provide longer service time. Then, how to make the video decoder adapt resource in handheld devices? How to maximum video decoding quality under battery constraint when playing on portable terminals? This paper tries to answer above-mentioned questions.

In this paper, we proposed simple, energy-scalable video decoding algorithms for energy constraint terminals to save power and improve video quality. Moreover, we complement these algorithms with device energy aware method to lengthen the available time of video services. This is implemented through maximizing the decoded available video frames at a given power budget. The algorithm, called ESVD, means an integrated energy-scalable video decoding framework for low-power video decoding applications. ESVD uses energy profiles as scalable management guideline. Each energy profile is equivalent to an energy constraint budget. On such ESVD, algorithms use utility theory to find the best energy levels for each of the subfunctions in decoding.

In ESVD system, video decoder can dynamically adapt the variable energy resources through energy aware technique. ESVD helps the decoder combine decoded data with the decoding process. Video decoder can work under variable energy resources constraint marked with different energy consumption budgets and provide a wide scope adjustable decoding energy output. Besides, it uses utility theory to solve the tradeoff between decoding effect and energy consumption, so as to obtain better performance in each energy levels.

This paper is organized as follows. Section 2 describes related work; Section 3 gives label and parsing method so as to provide a sufficient conditions for the ESVD; Section 4 describes the energy-scalable video decoding algorithms; Section 5 evaluates them; and Section 6 concludes.

2. Related Work and Backgrounds

The contributions of the paper are related to several areas of work, which we consider in turn.

2.1. Designing Low-Power Video Encoders

2.1.1. Scalable Video Decoders on Terminals. De Schrijver et al. [1] study the scalable video codec. They consider the memory, processing power, and bridge these with amount of bandwidth which comes from video fragment. Thus, scalable function is from the encoded scalable video bitstreams. Yanagihara et al. [2] propose CPU load-scalable video decoder algorithm, it uses several DCT manipulations such as low-pass filtering and resolution conversion in DCT domain. The decoder aims at the application of multichannel multicast system. Their work is rudimental to ours. Landge et al. [3] propose a systematic framework to optimize the energy consumption. They are in view of wavelet-based video decoders and use generic computational complexity metrics derived from the frequency of execution of program basic blocks. Since the decoder often does not know beforehand the encoded streams, this scalable function is obtained postmanufacturing and is unique to each codec system.

2.1.2. Designing Low-Power Video Decoders. Masselos et al. [4] design a low-power decoders based on the replacement of the image block by the selected codeword in the output image. Besides, they use efficient transformations to the codewords to compensate for the quality degradation introduced by the small codebook size in the encoder side. This method reduces its memory requirements so that it gets lower power consumption. Szu- Lee and Kuo [5] integrate the encoder selected proper interprediction modes and then generate a video bit stream. This method enables the encoder to estimate the decoding complexity and choose the best inter prediction mode to meet the complexity constraint of the target decoding platform. In a word, these methods rely on the encoder to reduce resource consumption of decoder. From integrated circuit aspect, Liu et al. [6] derive rapid algorithm in IDCT, deblocking filter and prediction, which can reduce the processing cycles and reduce the memory size and access frequency. These methods are the main measures for lowering the power consumption. The work is also complementary to ours. The other low-power design techniques include skipping computation in zero components, using lower constant multipliers, reducing transitions in the data path, and self-adaptive techniques. These methods acquire good effects in IDCT and prediction compensation modules, corresponding research examples include August and Ha [7] in IDCT and prediction, Tsung- Tsai and Fang [8] in VLC, and Xu and Choy in [9] self-adaptive prediction. We combine thoughts in scalable decoder and methods in low-power design so as to achieve integration scalability and efficiency.

2.1.3. Complexity Power Mapping in Video Decoders. From encoder aspect, researchers have developed how to measure the power consumption in video encoders. He et al. [10] analyze the rate-distortion (R-D) behavior of video encoding system under the energy constraint. Based on power-rate-distortion (P-R-D) model in [10], they prove that power is tightly coupled with rate, thus, to trade bits for joules and to perform energy minimization are rapid method to obtain minimum energy [11]. Though these models are proposed based on the encoder, they can be used for reference in low power decoding design. From decoder aspect, existing approaches use the complexity metrics as the main measure methods on the first step; these metrics include counting the number of base operations [12], and memory access frequency [13] and occupation. On the second step, use mapping relations between complexity metrics and power or energy consumption to evaluate the accurate loss value [10, 14]. We combine the complexity metrics and power mapping methods, which in turn guide the control of optimal algorithm design to optimize the energy consumption.

2.2. Complexity Metrics in Video Codec

2.2.1. Complexity Evaluation in MPEG. It is largely recognized that MPEG standards play a major role in the starting and development of multimedia communications and applications [15]. From the compression ratio point of view, MPEG possesses an important role of low-bit-rate video coding. From the complexity point of view, MPEG provides three tools to evaluate video codec complexity so that it controls the resources required at the decoder. Through these models, we can set boundaries on memory and computational requirements. The MPEG-4 standard defines video buffering verifier mechanism, which includes three virtual buffer models, named the video rate buffer verifier (VBV), the video complexity verifier (VCV), and video reference memory verifier (VMV). There, the VCV model is applied to all macroblocks in an MPEG-4 video bitstream and is used to verify the computational power required at the decoder. The model is defined in terms of the VCV MB/S decoding rate and VCV buffer size and is applied to all MBs in the scene [16, 17]. It mainly aims at the processing speed, defines in terms of the number of macroblocks (MBs) per second, and determines whether the decoding resources fit within a certain profile so as to not exceed the values specified for the corresponding profile and level.

In VCV model, the computational complexity of the decoder is defined by bridging the data rate, and the number of MBs per second that the decoder has to decode. Indeed, the computational power consumption required by each MB decoding may largely vary with the MB types. According to careful analysis in [16], the ways to measure the decoding complexity of the encoded video data can be associated to the rate of the following parameters, including the number of MBs, the number of MBs per shape type such as boundary or transparent, the number of MBs per combination of texture and shape coding types, and the number of arithmetic instructions and memory Read/Write operations. Therefore, the number of MBs per combined coding type is a better method to represent the major factors determining the actual decoding complexity from the compressed data. Based on this, an alternative VCV model is proposed in [18], which allows a more efficient use of the available decoding resources. The model indicates that the decoding complexity can be measured by a combination of the MB complexity types and the number of MBs in corresponding different types. Thus, the decoding complexity can be evaluated and characterized by a combination of scenes, shape, and texture coding tools. This model enhances the VCV model because of complementing some determining factors. Furthermore, simplified control method in [18] can be adopted to distinguish the various types of MBs in terms of decoding complexity, in which the complexity weights can be defined relatively to the most complex MB type in the context of each profile. This means MPEG-4 decoders in most critical cases can be a compliant decoder, making a better supplement of the video complexity verifier model.

2.2.2. Complexity Evaluation in H.264. H.264/AVC represents many advanced techniques in standard video coding technology, and promises some significant advances of the state-of-the-art video coding techniques in a broad variety of applications [19, 20]. Compared to previous standards, H.264/AVC is given with respect to the coding efficiency and hardware complexity [21]. Indeed, assessing the complexity of a video coding standard is not a straightforward task; the same is true of H.264/AVC. Though the complexity heavily depends on the characteristics of the platform on which it is implemented, there are still mapping metrics to evaluate implementation complexity. Reference [21] analyse the complexity of H.264/AVC based on the new versions of the executable H.264/AVC specification, which includes updated tool definitions and can achieve a reduced complexity [22]. This analysis divided the H.264/AVC decoder into six parts, these are CABAC, RD-Lagrangian optimization, B-frames, Hadamard transform, deblocking filter, and displacement vector resolution. And it analyzes these parts in detail from the access frequency aspect and decoding time aspect.

2.2.3. Complexity Metrics in Video Codec. Generally speaking, the VCV model and the alternative VCV model are both based on measuring the decoding complexity in terms of the number of MB. The relative complexity weight for each MB complexity type is thus obtained as the ratio between the maximum decoding time for each MB type and the highest maximum decoding time from all the MB types relevant in the decoder profile. This method is widely adopted in the video codec, such as [23].

The measurement flow of video complexity evaluation systems such as video codec can be typically divided in several main steps.

(1) Algorithmic development phase. This first step focuses on algorithmic performance. The algorithmic specification is typically released as a standard description plus a software verification model [24]. In this phase, complexity cost function in C-level analysis is needed. Efficient implementation based on each algorithm is adopted while it guarantees performance [25]. This phase focuses on deducing complexity, leading to high performance and enabling low-power realizations in algorithm-specific complexity level.

(2) Evaluation flow phase which deals with the actual system realization is based on a specific platform. The true implementation complexity of the algorithm based on universal platform can be acquired. Can this stage determine the cost of each module or each algorithm in some series terminals and, hence, its success and widespread diffusion or not?

On the other hand, memory access consumption is another key factor in power consumption. In video decoding, the primary design goal is to reduce memory transfers between large frame memories and data paths. Many researches summarize the cost of a data transfer into a function of the memory size, memory type, and the access frequency, such as [5, 13, 26]. The measure method is the number of accesses per second instead of the clock frequency [26]. To accurately calculate the dynamic cost in each frame during decoding is a difficult job. Thus, in [12], they provide the upper limit of memory consumption.

3. Parsing and Labeling Video Decoding

The main low-power techniques targeted at achieving lower consumed processing cycles and memory requirements are both described and discussed in Section 2. In this part, we address in analysis how to partition the decoder so as to provide scalable output.

In most cases, there is not enough residual capacity of battery to enable portable devices users to watch any video programs at any time as they wish, because of the exhausting battery. At the same time, in general video decoding systems, each module consumes a different amount of power and can affect a different rating of video quality. That is, the modules have different contributions in an environment with energy/battery constraint. Therefore, there is a tradeoff between maximum available lifetime of battery and minimum distortion caused by as possible as balanced decoding control.

Given that the residual capacity levels of battery can be substantial, it makes sense to schedule modules and perform power management as if the scalable affected was a heterogeneous system. On the other hand, most video decoders nowadays, especially in real time mobile video applications, are paid more efforts in improving robustness. For example, data partition techniques in H.264, decoder with little redundancy information or with little support from the encoder side. In this case, useful information can be introduced to help decoder. In this environment, there are three high-level control issues. The first is the MB types in coded data; the second is the detailed MB partition information; the third is the effect of human visual properties on single image. Based on these configurations, we present a set of energy-scalable algorithms for video decoding scheduling and energy management, aimed at minimizing power and maximizing video quality. The scheduling algorithms are intended to complement the scheduling criteria produced by the parsing and labeling control, such as priority, and fairness. In the following, we give the detailed parsing and labeling processing.

3.1. MB Type Information. In the first place, MB type information is considered as the primary criterion in decision since an intra MB is decoded without referencing any MB in another picture [27], but may be referred to by other inter MBs. Usually, intra MB is taken for more importance than inter MB. Thus, the intra MB block is marked as $L_1(0)$, the inter MB is marked as $L_1(1)$, and inter MB in B frame is marked as $L_1(2)$, which are denoted in (1). It means, from block type aspect, intrablocks and intra frames are assigned and processed in high energy profile comparing with interblocks. In fact, VCV also introduced MB type information as main decoding control term, which had been discussed in Section 2.2,

$$w_1(i,j) = \begin{cases} L_1(0), & \text{case: Intra MB}, \\ L_1(1), & \text{case: Inter MB}, \\ L_1(2), & \text{case: Inter MB in B frame}, \end{cases} \tag{1}$$

where $w_1(\cdot)$ is the results of paring MB type information. (i, j) denotes the position index of an MB.

3.2. MB Partition Information. In the second place, the MB partition information is considered as secondary criterion in decision.

Each intramacroblock could be classified into several modes including intra_4 × 4, intra_ 8 × 8, and intra_16 × 16. Each intermacroblock in P frames could be partitioned into inter_ 4×4, inter_ 8×8, inter_ 8×16, inter_ 16×8, and inter_ 16 × 16. In a word, there are following partition modes in macroblock, these are 16 × 16, 16 × 8, 8 × 16, 8 × 8, 8 × 4, 4 × 8, and 4 × 4. Among these, if a block is partitioned into 4 × 4 mode, then it is the finest block and may be assigned in top level profile; while if a block is in 16 × 16 mode, it belongs to coarse block and is in bottom level profile. The MB partition information can be easily extracted after entropy decoding. Thus, the partition information becomes a criterion in assigning the macroblock into different energy profile. Here, for simplicity, we use the energy controlling parameters to mark the blocks or macroblocks so that we can obtain a reasonable distribution in the energy profile. $L_2(x)$ denotes the controlling level while $L_2(x) \in [L_2(0), L_2(4)]$, and the values corresponding to MB partition information are in the following:

$$w_2(i,j) = \begin{cases} L_2(0), & 4 \times 4, \\ L_2(1), & 8 \times 4 \text{ or } 4 \times 8, \\ L_2(2), & 8 \times 8, \\ L_2(3), & 8 \times 16 \text{ or } 16 \times 8, \\ L_2(4), & 16 \times 16. \end{cases} \tag{2}$$

Generally, a macroblock can be regarded as a combination of basic blocks which belong to different partitions. The basic block is defined in 4 × 4 block in H.264 [19] and is defined in 8 × 8 in MPEG2 [28] and AVS [29]. Hence, the marked coefficient for a macroblock is deduced through the partition results of basic blocks. Weighted sum method is adopted in this paper. For instance, a macroblock consists of four 4 × 4 blocks in top left corner, two 4 × 8 blocks in top right corner, two 8 × 4 blocks in bottom left corner, and an 8 × 8 block in bottom right corner. Figure 1 shows the partition results. Then, the final effected coefficient which decides the macroblock into appropriate energy profile is $4 \times L_2(0) + 2 \times \sum_1^2 L_2(1) + 2 \times \sum_1^2 L_2(1) + 1 \times \sum_1^4 L_2(2)$.

3.3. Effect of Human Visual Properties. In the third place, the effect of human visual properties is considered as third criterion in decision. In many video applications, clients would pay more attention to the regions of their interest. For example, if the shoulder and head video is always existed in video applications, the region of interest (ROI) of clients is usually the human face instead of the background. Thus, for the decoder, more resources including bits and computational power are desired to be allocated reasonably according to the human subjective effects to improve the overall visual quality [30]. From the objective aspect, [31] gave a detailed segmentation strategies for an image. The paper analyses main segmentation approaches for multimedia services from the viewpoint of their features. The first one consists in estimating segmentation scope through the position of the transitions and marks the separation between neighboring regions. This approach has been mainly successful for the temporal case and being applied to both spatial and temporal segmentation problems. The second approach consists in estimating the region through homogeneous elements according to the feature space. This approach has been mostly applied to spatial and spatial-temporal segmentation. Here, we applied the segmentation thoughts and ROI technology to the image region decision. We mark the region in image based on human's attention degree. The technology of ROI is adopted as an efficient tool for the reasonable classification of image; it could be used to divide an image into several parts into different level. When the available battery energy is not enough, the ROI information is used to optimally allocate the available energy to different parts of the image according to their relative level. Since the central region in an image will be concerned firstly according to the habit of human being,

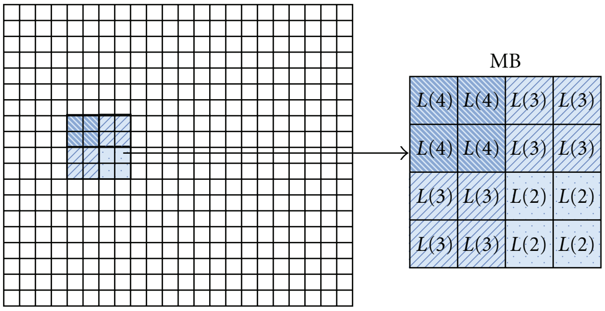


Figure 1: A example of MB partition information computing.

the blocks in central region is allocated to higher energy profile than the surrounding region. As shown in Figure 2, the marks of the human's attention degree are dispersal from central to surrounding regions, then the energy controlling parameters can be marked as (3), where (i, j) denotes the position index of an MB,

$$w_3(i,j) = \begin{cases} L_3(0), & (i,j) \in \text{interest region}, \\ L_3(1), & (i,j) \in \text{sub-interest region}, \\ L_3(2), & (i,j) \in \text{normal region}. \end{cases} \tag{3}$$

These parsing and labeling configurations provide the sufficient conditions for the following energy-scalable algorithms. Then the energy profile scheduling and energy scalable management rely on the criteria produced by the parsing and labeling control, including priority, and fairness. In the next section, we develop a model of energy-scalable video decoding (ESVD).The overall energy consumption could be optimized after these methods, at the same time the ESVD can guarantee the best video decoding quality in energy constraint circumstance.

4. Energy-Scalable Video Decoding Model

In this model, different energy profiles are equivalent to different energy consumption level, and video decoder runs at these profiles. In this scalable energy profiles, the most obvious optimization goal is to maximize performance at a given power or energy budget. Given the complexity or the power budget of this environment, to reasonably design the algorithm for scheduling and for energy or power management, a global optimization solution is required.

Section 2 shows possible algorithms to maximize performance at the target power. To simplify the problem, we construct parsing and labeling processing in video decoder in first step, which is given in details in Section 2. These provide the foundation of ESVD. On the other hand, in most video decoding systems, especially for mobile applications, there is a limited system energy supply. Most of the services or functions in mobile devices have estimable power consumption. It means that the upper bound of the consumption can be acquired. Generally speaking, the total consumption is measured by the available battery capacity, that is, the energy consumption is inverse proportion to the available battery lifetime. Strictly speaking, the energy consumption in general video processing applications results from a number of factors, including the number of functions in using regulations, operation systems, hardware, and battery life. Most researches distinguish between two types of power constraints, namely peak constraint and average constraint. Here, we propose another type of power constraints, which is a bound constraint $\mathbb{H}(\mathcal{F})$. We use $\mathcal{F} \supset \{f_n\}$ to represent a function, in which f_n represents the nth subfunction in function $\mathcal{F}$. $\mathbb{H}(\mathcal{F})$ represents the minimum energy requirement required to implement a function $\mathcal{F}$. For video decoder, a bound of energy constraints also exists. It implies that the optimal energy control method can be obtained when the total energy consumption is deduced by the method tends to the energy bound as closely as possible. Of course, the video decoding function contains many subfunctions such as interpolation (INTP), deblocking filter (DF), entropy decoding (END), and inverse transform (IDCT) [32]. According to bound constraint definition, designing an optimal energy/power consumption video decoding system can be transferred to find the best control among these subfunctions to achieve lower power/energy consumption, so that we can prolong the available battery duration.

The above discussion shows the possibility to maximize performance at different target power level. To resolve the problem, we decompose it into two steps. First, we use parsing and labeling processing to map the subfunctions in unit of MB in video decoder, so as to generate scalable video decoding output. Second, we use power management algorithm to find the best configuration in subfunctions for each power profile and at the same time, maximizes overall performance while keeping lower power consumption.

4.1. Energy Scalable Management in Video Decoder. To compute the integrated weight of MB in order to assign it into appropriate energy profile, the proposed three decision phases in Section 3 are in combined calculation. This needs a mapping bridge between the levels $L_k(i)$ in each phase. This problem is solved as follows. Given a set of N subfunctions in video decoding function $\mathcal{F}$ in unit of MB, each subfunction can run at M levels, there is $N \times M$ power consumption level, correspondingly. Then this problem can be summarized as finding the best selection of power consumption levels for the subfunctions, at the same time it can maximize the decoding quality subject to the constraints: each scalable power consumptions in whole video decoding is less than $E_{\text{profile}(k)}$ in each energy/power profile. Our approach is to reduce the problem to a linear optimization problem. Overall, from parsing and labeling procedure, we map the labeling results on energy/power profiles orderly. To be specific, from the subfunctions, we select the subfunctions in order for MBs and in round robin manner for the whole video sequences decoding.

Here, to be simplified, linear weight control

$$f(\text{MB}) = a_1 \sum L_1(k) + a_2 \sum L_2(k) + a_3 \sum L_3(k), \qquad (4)$$

where k_1, k_2 and k_3 represent the effects on the total performance for each phase, separately. We give a simple example firstly. Then the final value can assign the macroblock into appropriate energy profile. For instance, if the video is encoded in AVS, assuming that the initial (a_1, a_2, a_3) is (c_1, c_2, c_3) in empirical way and $L_j(k) = k$, thereby the maximum marked coefficient for a macroblock is $f(\text{MB}) = c_1 i + c_2 \sum i + c_3 i$ and the minimum marked one is zero. We can get the marked bound of a macroblock as $[0, \max(f(\text{MB}))]$. Suitable levels can be classified either in theoretical way or in empirical method, then there are different intervals corresponding to the levels. $(0, a_1 \cdot \max(f(\text{MB})))$ represents coarse level, $(a_1 \cdot \max(f(\text{MB})), a_2 \cdot \max(f(\text{MB})))$ denote half accurate level, and $(a_2 \cdot \max(f(\text{MB})), \max(f(\text{MB})))$ is in accurate level, for the sake of clarity, equal configuration is used, that is $a_1 = a_2 = 1/3$. For example, given a coded frame, after entropy decoding, the macroblock information is extracted as follows, the type is intra, the partition belongs to 8 × 8, the position lies in central adjacency region, and $L_j(i) = i$ still comes into existence. Then the finial marked coefficient is calculated through (4). It means that the labeling energy index of this macroblock belongs to the corresponding energy profile.

4.2. Utility Function in Power Control Scheme. As a frame decoding is composed by subdecoding in unit of MB, MB encoding is also under common resource constrained. Each MB's decoding is a competitor of battery energy for others. On the other hand, PSNR and bit rates are the measurement of decoding quality of all MBs. Ideally, an MB unit would like to achieve normal quality of decoding effect while expending a small amount of energy. In some cases, better decoding effect or long duration decoding and playing are in anticipantion even if the available battery capacity is not enough. For example, most mobile terminals can work in different battery states including "Maximum battery life mode," "Battery optimized mode," "Maximum performance mode," and "Enhanced quality mode". Each battery state corresponds to a battery working mode of the device. These states are widely used in mobile devices and terminals. It is desired that video decoder should provide corresponding decoding output to match these working states. Thus, it is necessary to optimize the video decoding process under battery resource constraint. Obviously, it can be transformed into a kind of tradeoff between obtaining better decoding effect and obtaining lower energy consumption in corresponding working state. Finding a good balance between the two conflicting objectives is the primary focus of the power control component of resource management. This tradeoff is illustrated through the conceptual line in Figure 2. If the decoder power is fixed, the terminal would experience high decoding effect which leads to increased reasonable allocation of the system resources. If the decoding effect and quality is fixed, increasing the power consumption expedites the battery drain, which reduces the effective use of the mobile terminal.

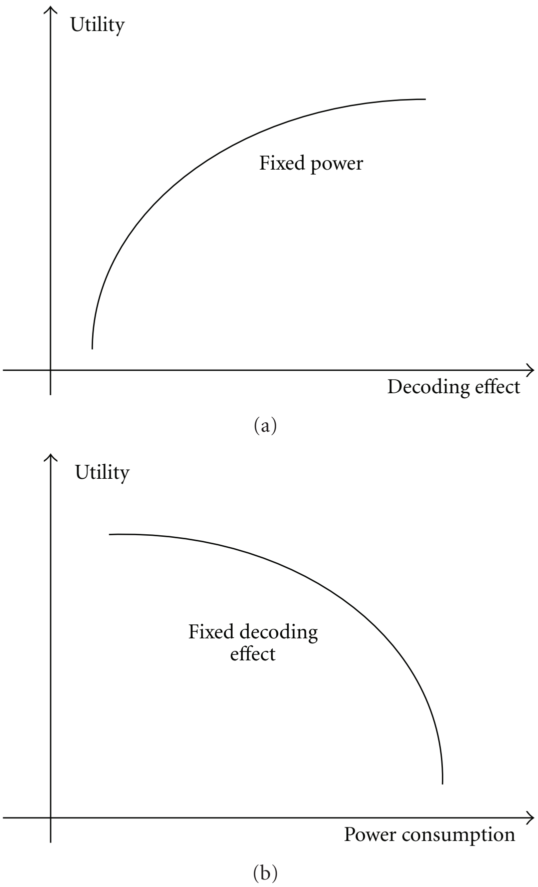


Figure 2: Utilities as a function of decoding effect and power consumption.

The optimal power control algorithm for video decoding systems should maximize the decoding quality. Traditionally, the object is to achieve acceptable PSNR as the measurement of decoding quality. However, this single target is not enough for efficient video decoding. This is because the object on power consumption is another important factor in applications. It is clear that a high PSNR level at the decoding output will result in better decoding effect. However, achieving a high PSNR level often requires the terminal to work in high power consumption state, which, in turn, results in low battery life. These issues can be quantified by defining the utility function of an MB decoding unit, which is defined as

$$u_{i,j} = \frac{\Delta e_{i,j}}{\Delta \mathrm{PSNR}_{i,j}} \frac{mwh}{\mathrm{dB}}, \tag{5}$$

where $\Delta e_{i,j} = E_{\text{normal}}(i,j) - E_{\text{profile}}(i,j)$ and $\Delta\mathrm{PSNR}_{i,j} = \mathrm{PSNR}_{\text{normal}}(i,j) - \mathrm{PSNR}_{\text{profile}}(i,j)$. For MB$(i,j)$, $E_{\text{normal}}(i,j)$ represents the battery power consumption of the decoder in normal state, while $E_{\text{profile}}(i,j)$ means the battery power consumption in corresponding energy profile. Accordingly, $\mathrm{PSNR}_{\text{normal}}(i,j)$ is the quality in full decoding state, while $\mathrm{PSNR}_{\text{profile}}(i,j)$ represents the decoding quality in corresponding energy profile. Utility as defined above combines the decoding quality and power consumption. The efficiency function yields the desirable properties. Assuming perfect case $\Delta\mathrm{PSNR}_{i,j} \rightarrow 0$ and $\Delta e_{i,j} \rightarrow 0$ means the decoder is under the full-state decoding. The mobile terminal can work in "Maximum performance mode" or "Enhanced quality mode." In this case, the decoding quality will obtain maximum value. On the other hand, $u_{i,j}$ is a monotonically increasing function of the $\Delta e_{i,j}$. That is, in case of fixed target power consumption $e_{i,j} = E_{\text{target}}$, for decoding schemes, the best strategy for MB encoding is to make a decision for each subfunction, so as to acquire maximum utility u_i. This suggests that, in order to maximize utility, all MBs in the video decoding system should try to improve the decoding effect while as possible as less consume the energy. So that the utility function is suitable for measuring power efficiency of video decoding systems.

4.3. Energy Allocation Scheme Based on Macroblock Tracking. As mentioned above, most mobile terminals provide many working states such as "Maximum battery life mode," "Battery optimized mode," "Maximum performance mode" and so on. Accordingly, supposing that video decoder can provide corresponding decoding output to match these working states. Each energy profile $E_{\text{profile}(k)}$, $k \in N$ corresponds to a decoding level. Then the goal is to adjust the decoder state in unit of MB to obtain best decoding quality under energy consumption budget $E_{\text{budget}(k)}$. Following the arguments in (5), there is

$$\begin{aligned} &\max \quad \sum_i \sum_j U(i,j) \\ &\text{s.t.} \quad \sum_i \sum_j E_{\text{MB}}(i,j) \leq E_{\text{budget}}, \end{aligned} \tag{6}$$

where $i = \text{width}_{\text{frame}}/\text{width}_{\text{mb}}$ and $j = \text{height}_{\text{frame}}/\text{height}_{\text{mb}}$. For example, if there is video decoding data in CIF format, then $\text{width}_{\text{frame}} = 352$, $\text{height}_{\text{frame}} = 288$, $\text{width}_{\text{mb}} = \text{height}_{\text{mb}} = 16$, and so forth. From the discussion above, all MBs in a frame are parsed and labeled into different scalar quantity, here we use $\widehat{w}(i,j)$ to represent the final labeling result of each MB. Then the MBs in a frame can be allocated into different energy profile levels according to their labeling results. As the decoder is divided into several levels in unit of MBs, we relate these MBs with different decoding state to realize fine allocation. Define that the number of decoder states is γ, and then it is obviously that the number of MBs is usually unequal to γ. This leads to an optimal problem. That is, we should configure these MBs into suitable decoding states to obtain better decoding quality. From (6), we have

$$\begin{aligned} &\max \quad \sum_i \sum_j U(i,j) \\ &\text{s.t.} \quad \sum_\gamma w_{\text{level}(\gamma)} \sum_i \sum_j m_{\text{level}(\gamma)} E_{\text{MB}}(i,j \mid \text{level}(\gamma)) \leq E_{\text{budget}}. \end{aligned} \tag{7}$$

As mentioned above, we classify these MBs into three levels for the sake of simplicity and define that the MBs in the same level has the same energy budget. For each MB in level 1, let the energy budget is e_1, accordingly, for each MB in level 2 and each MB in level 3, the energy budget be e_2 and e_3, separately. Then (7) can be rewritten as

$$\begin{aligned} &\max \quad \sum_i \sum_j U(i,j) \\ &\text{s.t.} \quad \sum_{\gamma=1}^{3} w_{\text{level}(\gamma)} m_{\text{level}(\gamma)} e_\gamma \leq E_{\text{budget}}. \end{aligned} \tag{8}$$

4.4. Decisions Using Learning Method. As we known, it is difficult to obtain the accurate correlation between PSNR and energy consumption level. Thus we use machine learning tools [33] to exploit the correlation and derive decision table to classify the MBs into corresponding decoding levels. Machine learning method refers to the study of decoding states to acquire knowledge from experiences. It deduces new knowledge from existing rules and uses the analysis of a set of experiments or examples, for creating a set of rules to take decisions. Thus, the correlation problem is posed into two sub-problems: one is to collection the variation of PSNR and energy consumption in each decoding state; the other is to classify these data into suitable modes according to their utilities. In next section, we give the detail of subfunctions design in unit of MB. And carry out a performance evaluation of each subfunction in terms of its variation of PSNR and the variation of PSNR and energy consumption results.

5. Implementation During Video Decoding

The overall energy consumption could be optimized after these methods; at the same time the ESVD can guarantee the best video decoding quality in energy constraint circumstance. For the sake of clarity, the whole energy constraints are summarized as the total summation of each function which the applications support. In practice, the functions cover a variety of applications. In contrast, the average power constraint can be imposed on the overall consuming power in universal user application circumstance.

Motivated by the previous discussion that all macroblocks are classified into several energy/power profiles, we

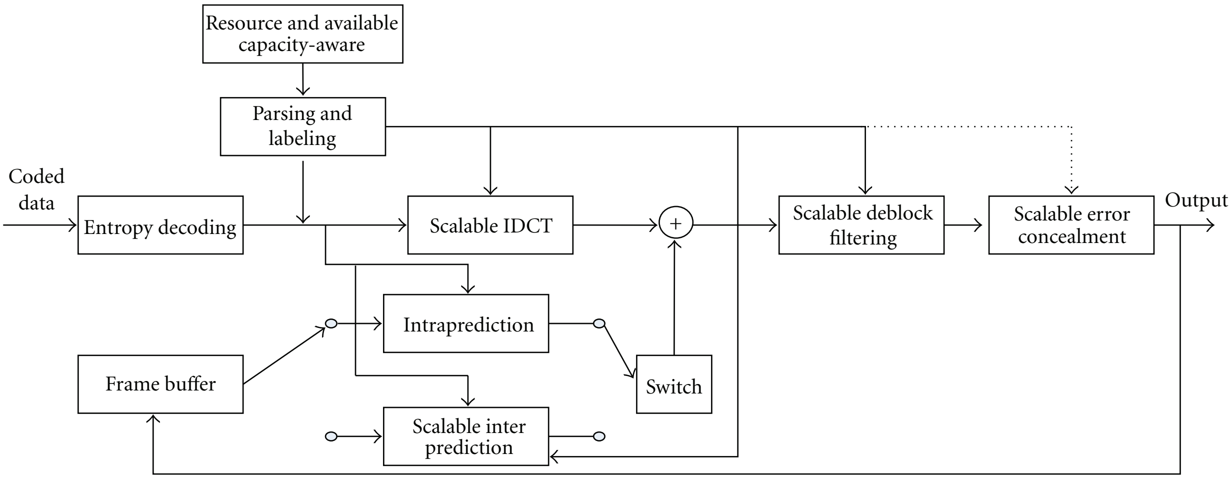

Resource and available capacity-aware
Parsing and labeling
Coded data
Entropy decoding
Scalable IDCT
+
Scalable deblock filtering
Scalable error concealment
Output
Intraprediction
Frame buffer
Switch
Scalable inter prediction

Figure 3: Illustration of the power scalable control video decoding system.

design a device resource perceptual module. This module implements a mapping bridge between the energy profile and the device available resource. This module includes two part functions. Part 1, user can specify the working state of video service. These states include "Maximum battery life mode," "Battery optimized mode," "Maximum performance mode," and "Enhanced quality mode". As mentioned above, each state corresponds to a battery working mode of the device. Part 2, to automatic adapt the working state of video service according to remaining battery capacity perception. For instance, "Maximum battery life mode" can be configured automatically when the residual capacity is under 30%, while "Enhanced quality mode" adopted automatically in the case of available battery capacity is above 80%. To be specific, when the result of part 1 and result of part 2 are not matched, that is, user configures the device as "Enhanced quality mode" but the residual capacity is under 30% at that time, the final available profile is based on perceptual remaining battery capacity results. That is to say, part 2 has higher priority than part 1, and user can manually specify the working state only when the device resource is sufficient.

It is widely accepted that END, IDCT, INTP, and DF are the four main subfunctions in universal video decoder. Consequently, the following discussion is based on these four subfunctions. The implementation of each energy profile is described in Figure 3, and the modules are listed as follows.

5.1. IDCT SubFunction. The complexity of IDCT subfunction in decoder has closed relation with the inner non-zero parameters. Researches provide many scalable methods, for example, [34] using different proportion subrectangles in blocks to output scalable computation IDCT. In general, the energy of the DCT coefficients is dissipated among the zigzag scan of the block. The low-frequency component in left-upper corner has higher energy, while the high-frequency components in right-lower corner contain lower energy. Thus, we progressively omit the data along the inverse zigzag scan, from right-lower corner to the left-upper corner, so that obtain minimal output quality degradation and at the same time achieve scalable energy consumption. Here, we classify the energy profile in IDCT subfunction into four degrees, including accurate-level, saving-level, coarse-level and non-IDCT. When accurate-level is selected, the whole parameters computation is implemented as shown in Figure 4(a). Many simplified methods can be used such as 1D IDCT optimization so as to minimize the energy consumption possible as. Figures 4(b) and 4(c) show the cases of optimal-level and matching-level, separately. The main difference between the two levels is the number of computing parameters. The number implies corresponding processing levels.

5.2. Motion Compensation and Interpolation SubFunction. Motion and residual information is generated from compressed bits after entropy decoding. Interpolation of reference samples to generate a motion-compensated prediction is generally performed for each macroblock that is intercoded [12] and occupies most complexity in motion-compensated prediction. Thus, the average time required by the interpolation subfunction is approximate to a function of the number of intercoded macroblocks. The most straightforward approach to classify this subfunction is to fully interpolate and fully compensation operations. In this level, quarter-pixel motion compensation is replaced by half-pixel operations, it forms a saving mode with little quality decline while computation is saved. Accordingly, substituted interpolation modes in unit of half-pixel and integral-pixel compensation by integer interpolation results are adopted in the other energy profile, separately.

5.3. Deblocking Filter SubFunction. Deblocking filter which is often referred to as a loop filter is the final stage of the decoding process. DF subfunction reduces the blocking effect that is introduced by encoding the process at block boundaries. Comparatively high complexity of the subfunction is in consensus. Even after a tremendous effort in speed optimization of the filtering algorithms, the filter can easily account for one-third of the computational complexity of

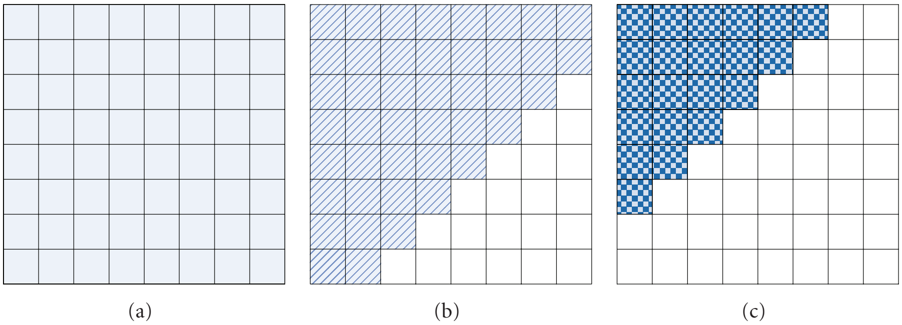
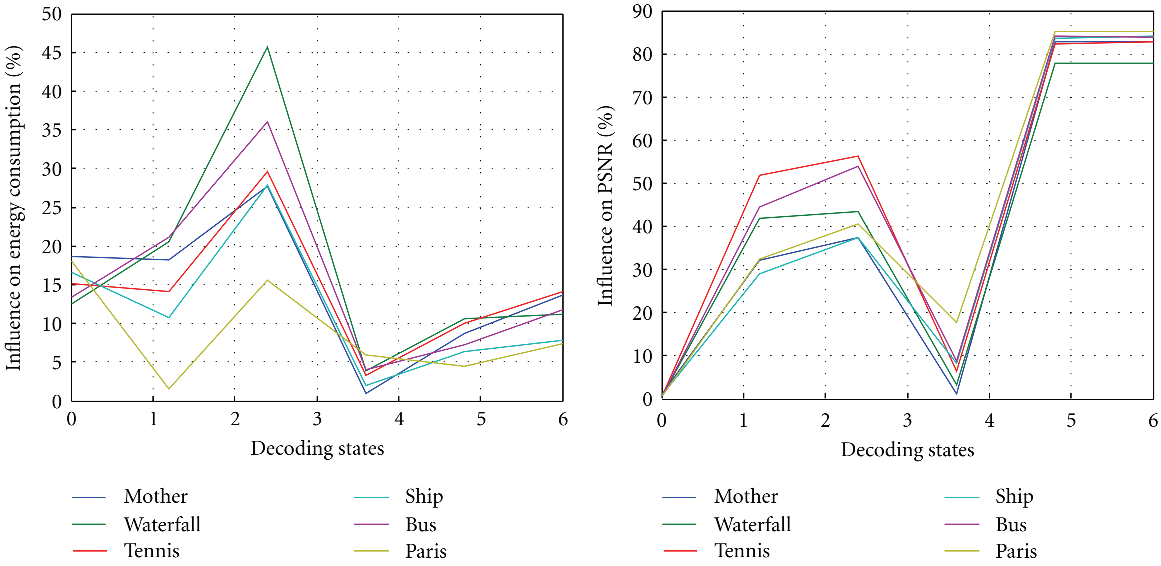


Figure 4: Data pruning patterns in IDCT.



(a) Influence on energy consumption

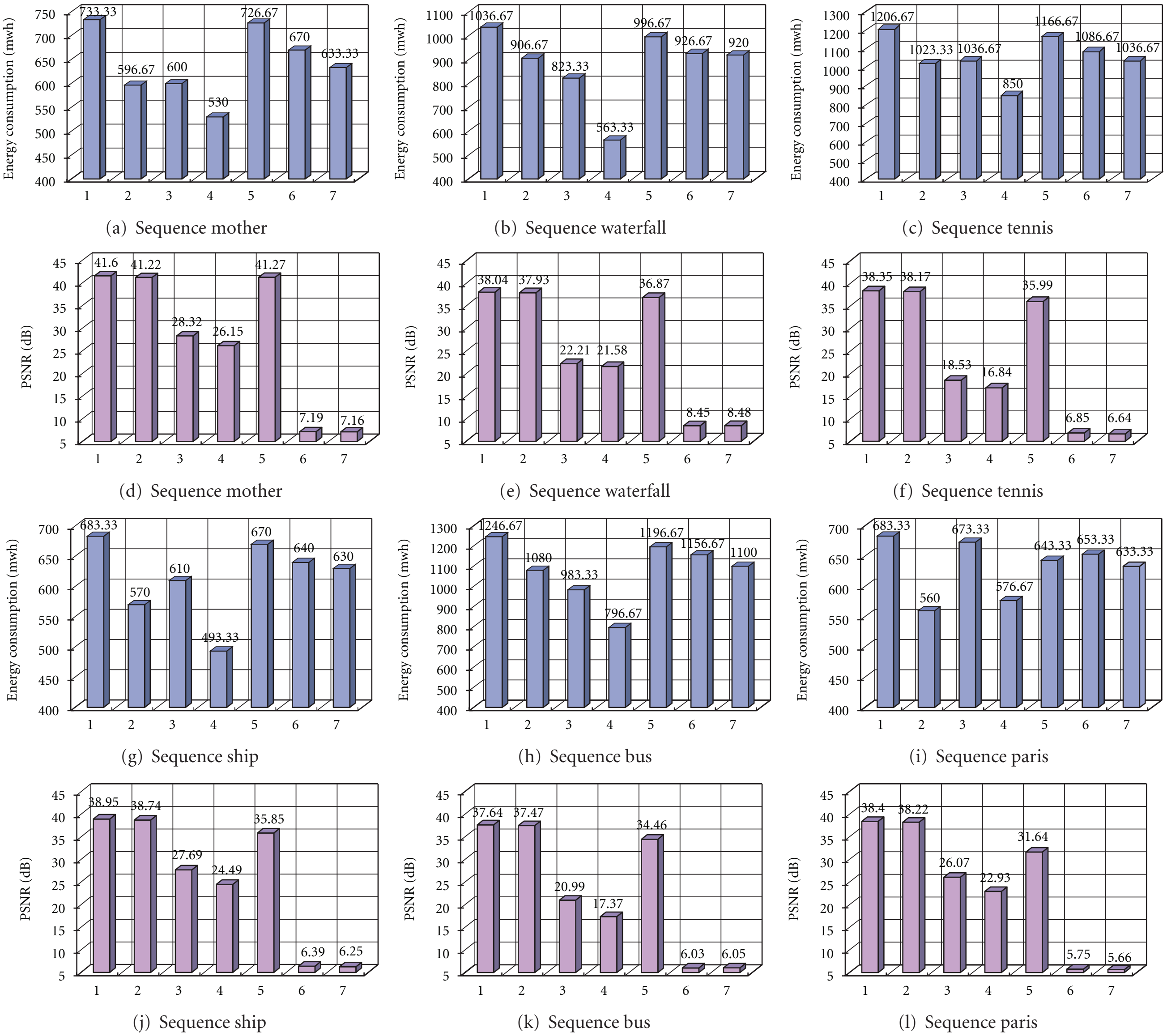
(b) Influence on PSNR

Figure 5: Influence on energy consumption and PSNR under different decoding rules.

a decoder [35]. The complexity is mainly based on the high adaptivity of the filter, which requires conditional and decisional processing on the block edge and sample levels, thus, there are many conditional branches in the filter which leads to excessive power consumption. At the same time, for a macroblock, the vertical filter begins from left-most edge and is followed from left to right by the three vertical edges; besides, the horizontal filter begins from top edge, and is followed by the three internal horizontal edges from top to bottom. Amount of relevant and candidate pixels should be loaded into the memory, this leads to additional power consumption either. Scalable energy can be achieved by classifying the filtering process into three levels, including full, half, and rough filtering. Among these, full filtering operation means that overall branch filtering is implemented for the macroblock. And, half filtering represents the operation reduced in computational complexity, which can be achieved by taking into account the fact that the image area in past frames is already filtered, and thereby optimizing or omitting the filtering process accordingly. For the rough filtering, skip operation is used with low quality degradation, while the lowest power consumption of the DF subfunction is required in this mode.

Besides, learning tools are used to analyze the data sets of decoder. The decision table will be used to determine the decoding modes of an MB. Inductive learning uses the analysis of data sets for creating a set of rules to take decisions.

Then a decision table is built as the decoding rules. This table is from a set of experiments or examples, collected as the training data set. We build information database to gather the decoding states. This set of data including the following properties: (0) full decoding mode; (1) decoding without deblocking filter mode, which corresponds to deblocking filter subfunction adjusting, (2) quarter pixel interpolation is compensated by half-pixel interpolation, (3) quarter pixel interpolation and half-pixel interpolation are both compensated by integer-pixel interpolation; these two cases are corresponding to motion compensation and interpolation subfunction adjusting, (4) data pruning pattern in IDCT complies with saving-level, (5) data pruning pattern in IDCT follows coarse-level; (6) data pruning pattern in



(a) Sequence mother (b) Sequence waterfall (c) Sequence tennis

(d) Sequence mother (e) Sequence waterfall (f) Sequence tennis

(g) Sequence ship (h) Sequence bus (i) Sequence paris

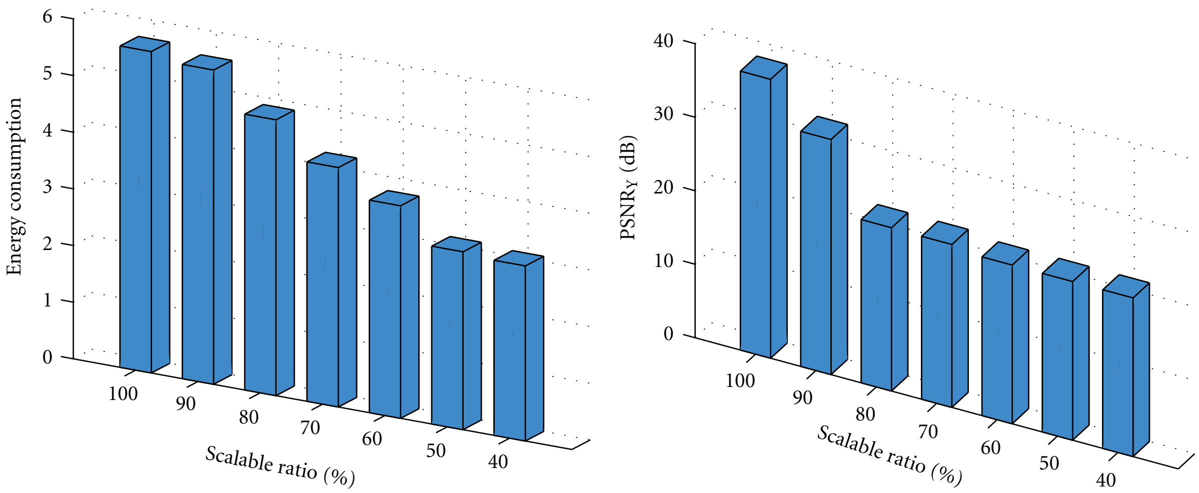
(j) Sequence ship (k) Sequence bus (l) Sequence paris

Figure 6: Stat. on energy consumption and PSNR for different video sequences in each decoding rule.

IDCT follows low-level; these three cases are brought into correspondence with IDCT subfunction adjusting. Figure 5 gives the influence on energy consumption and PSNR under different decoding rules, separately.

Affiliated subfunction: discussion on error concealment subfunction. Error concealment technique aims at obtaining a close approximation of the original signal or making the output of decoder closely accepted by human eyes [36]. Most error concealment techniques are based on block matching algorithms [37] or adaptive techniques in unit of block such as [38]. It can improve the decoding quality while it leads to less computational complexity. Due to the energy consumption which lies in computation, memory occupation and memory access, the effect of error concealment on additional power consumption is more than that on complexity. Here, we classify the error concealment operation into three levels to adapt the scalable energy profiles. This classification is based on scene and region change and on the unit of block. Thus, the macroblock can belong to three energy profiles, including accurate concealment in case of scene change, half concealment in case of regional variability, and coarse concealment when few and no movements take place.

Reference [39] gives an analysis of H.264/AVC decoder in computational complexity, and [12] presents detailed analysis in both computational complexity and memory occupation complexity. For the aspect of the complexity in AVS video decoding, [32] is provided an approximate estimation. Generally speaking, for most video decoders including H.264, MPEG4, AVS, and so forth, the computational power allocation with emphasis on power-distortion (P-D) [10] can be expressed in form of cost functions. We take power consumption in video decoding into account by modifying the power-distortion-complexity (P-D-C) cost functions in processing unit of macroblock and subfunctions



(a) Energy Consumption in each decoding modes

(b) PSNR in each decoding modes

Figure 7: Influence on energy consumption and PSNR under different decoding modes.

in decoder. Through the objective function in (8), dynamic scalable assignments provide a local quality optimum in each energy profile. Consequently energy scalable video decoding (ESVD) is achieved. An undeniable fact is that scalable video decoding leads to the quality degradation. Thus minimizing this degradation is another purpose in ESVD.

6. Experimental Results

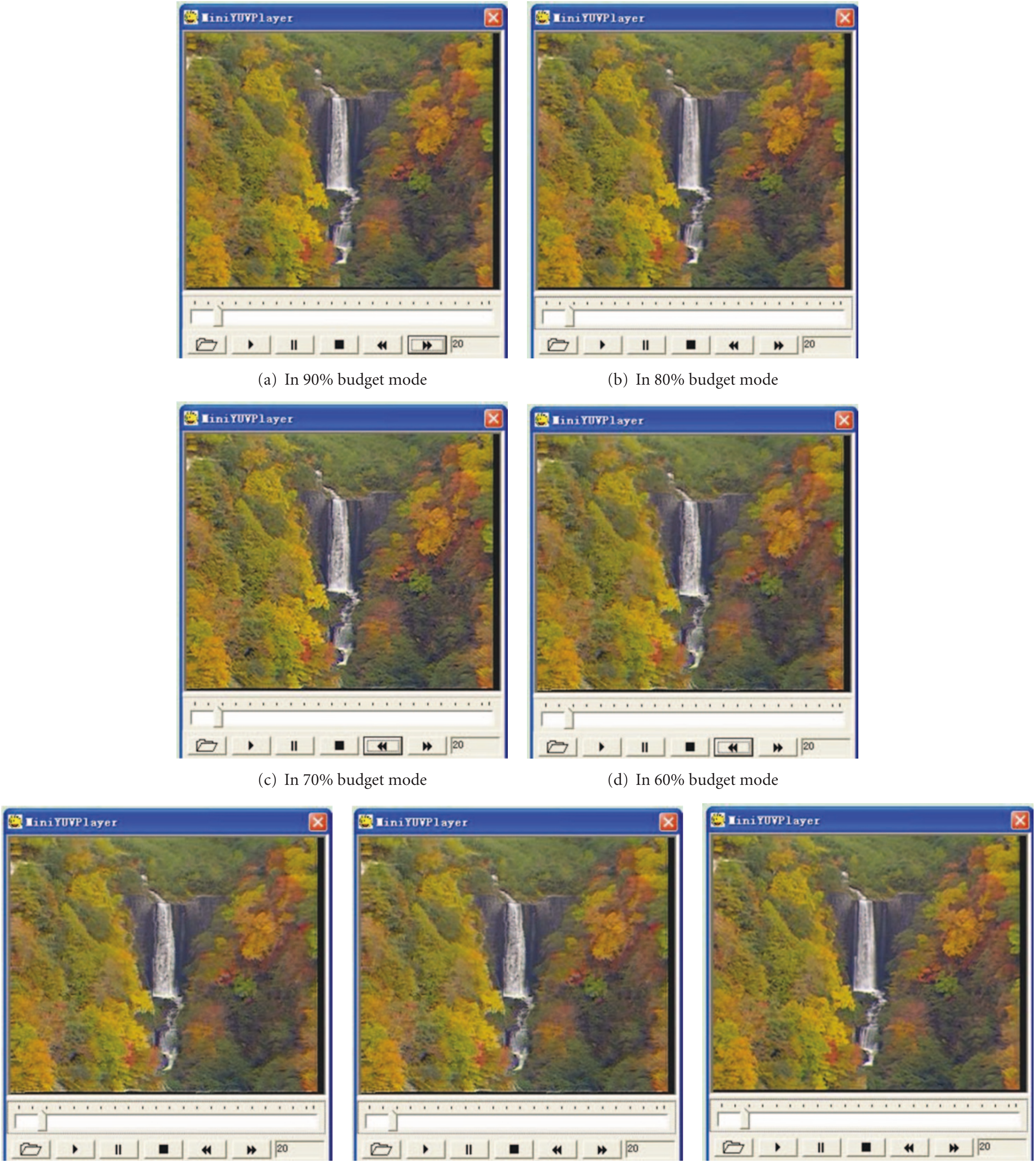
6.1. Building Energy Consumption Information Database. In this subsection, we use *Application Energy Graphing Tool* [40], which can measure the battery power consumption of an application over time, log and graph the resulting data. We use it to profile the energy distribution of the decoding modes. To calculate the energy consumption in the case of subfunctions modes, we assume that all other possible operations among the subfunctions are running, expect the testing mode. It means it will occur in power control schemes in practices that decoding data will be ergodic to all basic subfunction units in despite of some skipped or simplified operations. The reason is that compressed video data includes multifeatures, thus the decoding process varies with these features. For instance, for the same decoding program, the decoding time is different among the typical sequences such as mother, waterfall, tennis, ship, bus, and paris. Thus we use the typical video sequences as the test video set. The format is CIF and coded in AVS standard. We recycle the decoding process until the number of decoding frames obtains 15000 frames in each sequence. Figure 6 shows the total energy consumption and corresponding PSNR in each decoding rule. The results are based on statistical experimental average.

The decision table will be used to determine the decoding mode of MBs, based on the information gathered during the preanalysis of the decoder. This process can be more accurate by the information update during the decoding stage. Figure 5 depicts the process for building the decision tables from the results in Figure 6. For example, when the decoder works on mode (1), decoding without deblocking filter mode, little PSNR is lost but about 15% energy consumption saving can be obtained; when the decoder works on mode (5), data pruning pattern in IDCT follows to coarse-level, only around 10% energy consumption saving can be obtained but 85% PSNR losing occurs that is, when the energy budget is not full enough to support full-mode decoding, mode (1) is a better choice than mode (5).

6.2. The Performance of the ESVD Model. To evaluate the performance of the ESVD model and the energy scalable video decoding system, we implement the proposed ESVD model and energy scalability scheme in the AVS decoder software. The ESVD model is not limited to the video coding standards, and thus similar performance can be expected for other coding systems, such as H.264 and MPEG-4. We select stochastically "waterfall" CIF sequence at 128 kB/s and 25 fps as the testing sequence. We performed two sets of evaluations—one is for evaluating decoding scalability and the other for evaluating scalability quality. We let the decoder work under four modes. The energy consumption budgets are descending. The scalable results including PSNR and energy consumption shown in Figures 7 and 8 show the subjective quality in different decoding modes, separately. Each mode corresponds to energy consumption budget ratio compared to the full decoding mode. These experiments show the scalability and efficiency of ESVD.

7. Conclusion and Future Work

This paper proposed ESVD framework in power control video decoding systems. It aims at providing the scalable decoding output which is adaptive to energy resource.



(a) In 90% budget mode

(b) In 80% budget mode

(c) In 70% budget mode

(d) In 60% budget mode

(e) In 50% budget mode

(f) In 40% budget mode

(g) In full mode mode

Figure 8: Subjective quality in the different scalable modes.

It proposed a method to make the video decoder adapt resource under battery constraint, which can be widely used in handheld devices. At the same time, it gives a method to maximum video decoding quality when playing on portable terminals, through building a decoding information database. The experiments demonstrate the efficiency of ESVD. In future research, we will try to study fine-grained energy scalable control in energy consumption through improving the scalability of each decoding module.

Acknowledgments

The authors acknowledge the support from the National Natural Science Foundation of China (NSFC), contract/grant number: 60872007; National 863 High Technology Program of China, contract/grant number: 2009AA01Z239; The Ministry of Science and Technology (MOST), International Science and Technology Collaboration Program, contract/grant number: 0903; The NAP of Korea Research Council of Fundamental Science & Technology; 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-(C1090-1011-0004))".

References

[1] D. De Schrijver, C. Poppe, S. Lerouge, W. De Neve, and R. Van De Walle, "MPEG-21 bitstream syntax descriptions for scalable video codecs," *Multimedia Systems*, vol. 11, no. 5, pp. 403–421, 2006.

[2] H. Yanagihara, M. Sugano, A. Yoneyama, and Y. Nakajima, "Scalable video decoder and its application to multi-channel multicast system," in *Proceedings of the International Conference on Consumer Electronics (ICCE '00)*, pp. 232–233, June 2000.

[3] G. Landge, M. Van Der Schaar, and V. Akella, "Complexity metric driven energy optimization framework for implementing MPEG-21 scalable video decoders," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 2, pp. 1141–1144, March 2005.

[4] K. Masselos, P. Merakos, T. Stouraitis, and C. E. Goutis, "A novel algorithm for low-power image and video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 3, pp. 258–263, 1998.

[5] S.-W. Lee and C.-C. J. Kuo, "Complexity modeling for motion compensation in H.264/AVC decoder," in *Proceedings of the 14th IEEE International Conference on Image Processing (ICIP '07)*, vol. 5, pp. v-313–v-316, September 2007.

[6] T.-M. Liu, T.-A. Lin, S.-Z. Wang, and C.-Y. Lee, "A low-power dual-mode video decoder for mobile applications," *IEEE Communications Magazine*, vol. 44, no. 8, pp. 119–126, 2006.

[7] N. J. August and D. S. Ha, "Low power design of DCT and IDCT for low bit rate video codecs," *IEEE Transactions on Multimedia*, vol. 6, no. 3, pp. 414–422, 2004.

[8] T.-H. Tsai and D.-L. Fang, "A novel design of CAVLC decoder with low power consideration," in *Proceedings of the IEEE Asian Solid-State Circuits Conference (A-SSCC '07)*, pp. 196–199, November 2007.

[9] K. Xu and C.-S. Choy, "A power-efficient and self-adaptive prediction engine for H.264/AVC decoding," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 3, pp. 302–313, 2008.

[10] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 5, pp. 645–658, 2005.

[11] Z. He, W. Cheng, and X. Chen, "Energy minimization of portable video communication devices based on power-rate-distortion optimization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 596–607, 2008.

[12] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC baseline profile decoder complexity analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 704–716, 2003.

[13] T.-H. Wang and C.-T. Chiu, "Low power design of high performance memory access architecture for HDTV decoder," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '07)*, pp. 699–702, July 2007.

[14] L. Nachtergaele, D. Moolenaar, B. Vanhoof, F. Catthoor, and H. De Man, "System-level power optimization of video codecs on embedded cores: a systematic approach," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 18, no. 2, pp. 89–109, 1998.

[15] F. Pereira, "MPEG multimedia standards: evolution and future developments," in *Proceedings of the 15th ACM International Conference on Multimedia (MM '07)*, pp. 8–9, September 2007.

[16] J. Valentim, P. Nunes, and F. Pereira, "Evaluating MPEG-4 video decoding complexity for an alternative video complexity verifier model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 11, pp. 1034–1044, 2002.

[17] J. Valentim, P. Nunes, and F. Pereira, "Evaluating MPEG-4 video decoding complexity," in *Proceedings of the 2nd Workshop and Exhibition on MPEG-4*, March 2001.

[18] J. Valentim, P. Nunes, and F. Pereira, "An alternative complexity model for the MPEG-4 video verifier mechanism," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '01)*, vol. 1, pp. 461–464, October 2001.

[19] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[20] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657–673, 2003.

[21] J. Ostermann, J. Bormans, P. List et al., "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7–28, 2004.

[22] http://iphome.hhi.de/suehring/tml.

[23] R. Karri, D. J. Goodman, and R. Karri, *System-Level Power Optimization for Wireless Multimedia Communication*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2006.

[24] S. Saponara, K. Denolf, G. Lafruit, C. Blanch, and J. Bormans, "Performance and complexity co-evaluation of the advanced video coding standard for cost-effective multimedia communications," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 2, pp. 220–235, 2004.

[25] K. Denolf, P. Vos, J. Bormans, and I. Bolsens, "Cost-efficient C-level design of an MPEG-4 video," in *Proceedings of the 10th International Workshop on Integrated Circuit Design, Power and Timing Modeling, Optimization and Simulation*, pp. 233–242, 2000.

[26] L. Nachtergaele, F. Catthoor, B. Kapoor, S. Janssens, and D. Moolenaar, "Low-power data transfer and storage exploration for H.263 video decoder system," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 1, pp. 120–128, 1998.

[27] N. Ling and N.-T. Wang, "Real-time video decoding scheme for HDTV set-top boxes," *IEEE Transactions on Broadcasting*, vol. 48, no. 4, pp. 353–360, 2002.

[28] L. Chiariglione, "MPEG and multimedia communications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 5–18, 1997.

[29] W. Gao, C. Reader, F. Wu, et al., "AVS—The Chinese Next-Generation Video Coding Standard," http://www.avs.org.cn/reference.asp.

[30] Y. Liu, Z. G. Li, and Y. C. Soh, "Region-of-interest based resource allocation for conversational video communication of H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 1, pp. 134–139, 2008.

[31] P. Salembier and F. Marques, "Region-based representations of image and video: segmentation tools for multimedia services," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1147–1169, 1999.

[32] W. Ji, Y. Chen, C. Lei et al., "Module and distortion analysis for video decoding on mobile devices," in *Proceedings of the 7th International Conference on Networking (ICN '08)*, pp. 681–685, April 2008.

[33] S. Russell and P. Norving, *Artificial Intelligence: A Modern Approach*, Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 2006.

[34] S. Peng, "Complexity scalable video decoding via IDCT data pruning," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE '01)*, pp. 74–75, June 2001.

[35] P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 614–619, 2003.

[36] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: a review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, 1998.

[37] M.-J. Chen, C.-S. Chen, and M.-C. Chi, "Temporal error concealment algorithm by recursive block-matching principle," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 11, pp. 1385–1393, 2005.

[38] S.-C. Hsia, S.-C. Cheng, and S.-W. Chou, "Efficient adaptive error concealment technique for video decoding system," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 860–868, 2005.

[39] V. Lappalainen, A. Hallapuro, and T. D. Hamalainen, "Complexity of optimized H.26L video decoder implementation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 717–725, 2003.

[40] http://software.intel.com/en-us/articles/application-energy-graphing-tool-users-guide.