Adaptive Critic Design for Energy Minimization of Portable Video Communication Devices

Zhao Sun, Xi Chen, and Zhihai He, *Senior Member, IEEE*

***Abstract*—Portable video communication devices operate on batteries with limited energy supply. Video compression is computationally intensive and energy-demanding. Therefore, one critical issue in portable video communication system design is to minimize the energy consumption of video encoding so as to prolong the operational lifetime of portable video devices. In this paper, we explore advanced methods in adaptive system control and develop an online complexity control and energy minimization scheme for real-time video encoding. More specifically, we introduce a set of parameters to control the computational complexity and energy consumption of the video encoder. We consider this video encoder as a nonlinear control system. Based on adaptive critic design, an advanced adaptive control method recently developed in control science, we design an online control scheme which is able to select the best configuration of complexity control parameters to minimize the energy consumption under rate-distortion constraints, or equivalently maximize video quality under rate and energy constraints. Our extensive experimental results demonstrate that the proposed scheme approaches the true optimum performance. The proposed online complexity control and energy minimization scheme will provide an important tool for energy minimization of portable video devices.**

***Index Terms*—Adaptive critic design, energy minimization, nonlinear control, video coding.**

I. Introduction

WIRELESS video communication over portable devices has become the driving technology of many important applications, experiencing dramatic market growth and promising revolutionary experiences in personal communication, gaming, entertainment, military, security, environment monitoring, and more [1]. Portable devices are powered by batteries. Video data is voluminous. It has to be very efficiently compressed. Otherwise, the amount of transmission energy or required storage space will be tremendous. During the past decades, many video compression algorithms and international standards, such as MPEG-2, H.263, MPEG-4, and H.264 [2]–[4], have been developed for efficient video compression. An efficient video compression system is often computationally intensive and energy-consuming, since it involves many sophisticated operations in spatiotemporal prediction, transform, quantization, mode selection, and entropy coding [5]. Recent studies [6], [7] show that, in typical scenarios of video communication over portable devices, video encoding consumes a significant portion (up to 40–60%) of the total energy. Therefore, one critical issue in the portable video communication system design is to minimize the energy consumption of video encoding so as to prolong the operational lifetime of portable video devices.

There are two types of portable video devices: video encoders (e.g., video cell phones, wireless video cameras, etc.) and players (e.g., iPod video). In this paper, we focus on energy minimization for portable video encoding devices. This is because, on portable video devices, the fraction of energy consumption by video encoding (typically 60–85%) is much higher than that of video decoding [11].

A. Related Work

To reduce the energy consumption of video encoders, a number of algorithms, software and hardware energy minimization techniques, including low-complexity encoder design [13], [14], low-power embedded video encoding [15], adaptive power control [6], [16], [17], and joint encoder and hardware adaptation [18]–[20] have been proposed. Fast algorithms for major video encoding modules, including motion estimation, mode decision, and transform have been developed [14], [21]–[25]. A comprehensive review of fast algorithms for motion estimation in video compression is provided in [25]. Low-complexity algorithms have also been developed for spatial transforms [21]. Since there is no motion estimation for INTRA macroblocks (MBs), the INTRA ratio parameter, which is the fraction of INTRA MBs in the video frame, can be used to control the motion estimation complexity in the video encoder [6]. A parametric scheme for scalable motion estimation and DCT has been proposed in [16]. To speed-up the encoding process, a statistical modeling approach is proposed in [14] to predict the zero DCT coefficients after quantization. Based on the prediction, the DCT computation for those zero coefficients can be saved. Fast mode decision algorithms have been developed for H.264 video coding [23]. Hardware implementation technologies have been developed to improve the video encoding speed [15], [26]–[28]. Recently, researchers have realized the importance of

Manuscript received March 14, 2008; revised September 11, 2008 and January 27, 2009. First version published July 7, 2009; current version published January 7, 2010. This work was supported in part by the National Science Foundation under the Grant DBI-0529082. This paper was recommended by Associate Editor G. Wen.

Z. Sun is with the Center for Cooperative Control, National Institute of Aerospace, Hampton, VA 23666 USA (e-mail: zhaosun@nianet.org).

X. Chen and Z. He are with the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211 USA (e-mail: Xi.Chen@mizzou.edu; hezhi@missouri.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2009.2026835

cross-layer design for energy saving in multimedia systems [13], [20]. Joint adaptation of video encoder/decoder and hardware scheduling for energy saving has been studied [13], [18]–[20]. Joint encoder and wireless transmission power control schemes have also been developed; see for example [6].

B. This Paper and Major Contributions

Although a number of algorithms have been developed in the literature to reduce the computational complexity and energy consumption of video encoding, most of them have been focused on encoder complexity adaptation instead of optimization. The inherent relationship between energy consumption and rate-distortion (R-D) behaviors of a video encoding system has not been systematically analyzed. In our previous work [5], we have developed a power-rate-distortion method to characterize the R-D behavior of an MPEG-4 encoder under power consumption constraints. However, this method is not generic, relying on specific complexity control schemes implemented at the encoder. In addition, it is not suitable for online encoding complexity control and energy optimization of real-time video encoders.

In this paper, we investigate how theories and algorithms developed in adaptive system control [29] could be applied to video encoder complexity control and energy optimization. We consider the video encoder as a nonlinear dynamic system. The purpose of energy-consumption control is to find a sequence of encoding parameters such that the overall energy consumption is minimized under rate-distortion constraints. By exploring the approach of adaptive critic design (ACD), an important technique developed in adaptive system control, we develop an online complexity control and energy minimization scheme for *real-time* video encoding over portable devices. Based on neural network computations, the proposed ACD scheme simulates critical thinking of human intelligence and determines a sequence of complexity control parameters for the video sequence to maximize the overall video encoding performance under energy constraints. The proposed scheme is able to learn the behavior patterns of video encoder and can be easily extended to other image and video encoding systems. Our experimental results demonstrate that this nonlinear system control approach is very effective, being able to achieve the near-optimum performance.

The rest of this paper is organized as follows. In Section II, we discuss energy-scalable video encoding system design and operational power-rate-distortion analysis. Section III provides an overview of our proposed approach for online complexity control and energy minimization. The detailed adaptive critic system design for online complexity control and energy minimization is presented in Section IV. Section V summarizes our algorithm. Experimental results are presented in Section VI. Section VII concludes the paper.

II. Operational Power-Rate-Distortion Analysis

Our central idea is to introduce a set of complexity control parameters to control the computational complexity of major encoding operations of the video encoder. With dynamic voltage scaling (DVS), a recently developed power control technology for microprocessors [22], this complexity-scalable video encoder can be translated into an energy-scalable video encoder, since video encoding energy consumption is a function of its computational complexity [19], [22]. The energy-scalable video encoder design has the following three major steps.

In the first step, we group the encoding operations into several modules, such as motion prediction, precoding (transform and quantization), and entropy coding, and then introduce a set of control parameters $\Gamma=(\xi_1,\xi_2,\ldots,\xi_L)$ to control the power consumption of these modules. Therefore, the encoder complexity C is then a function of these control parameters, denoted by $C(\xi_1,\xi_2,\ldots,\xi_L)$. Within the DVS design framework, the encoding power consumption, denoted by P, is a function of encoder complexity C, denoted by $P=\Phi(C)$. Therefore, it is also a function of $\Gamma=(\xi_1,\xi_2,\ldots,\xi_L)$ denoted by $C(\xi_1,\xi_2,\ldots,\xi_L)$. The expression of this function depends on the power consumption model of the specific microprocessor [22].

In the second step, for each configuration of complexity control parameters $(\xi_1,\xi_2,\ldots,\xi_L)$, we execute the video encoder over the video sequence and obtain the corresponding R-D data, denoted by $D(R;\xi_1,\xi_2,\ldots,\xi_L)$. It should be noted that this step is computationally intensive and is intended for offline analysis. Once $D(R;\xi_1,\xi_2,\ldots,\xi_L)$ is obtained, in the third step, we perform optimum configuration of the control parameters to maximize the video quality (or minimize the video distortion) under the power constraint. This optimization problem can be mathematically formulated as follows:

$$\min_{\xi_1,\xi_2,\ldots,\xi_L} D=(R;\xi_1,\xi_2,\ldots,\xi_L),\quad \text{s.t.}\quad P(\xi_1,\xi_2,\ldots,\xi_L)\le P \tag{1}$$

where P is the available power consumption for video encoding. The optimum solution, denoted by $D(R;P)$, describes the P-R-D behavior of the video encoder.

It should be noted that this operational power-rate-distortion analysis procedure is a brute-force search approach. It needs to execute the video encoder for all possible configurations of complexity control parameters. For example, in our experiment, we have two complexity control parameters. We use ten sampling points for each parameter. In total, we have $10\times 10=100$ different configurations and we need to run the video encoder 100 times to produce all the R-D points $D(R;\xi_1,\xi_2,\ldots,\xi_L)$. This procedure is extremely computation-intensive and not suitable for online complexity control and energy optimization of real-time video encoders. In this paper, based on the adaptive critic design, we develop a low-complexity and robust online complexity control and energy minimization scheme for real-time video encoding. We use the offline operational P-R-D analysis to obtain the optimum solution for the video encoding energy minimization problem outlined in (1). This ground-truth optimum solution will provide a reference point for performance evaluation of the proposed method for online complexity control and energy minimization of real-time video encoding.

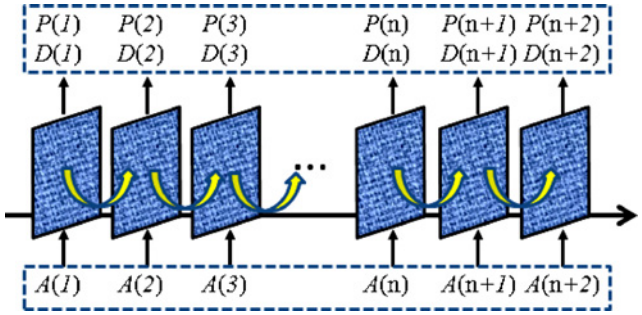

Fig. 1. Illustration of online complexity control and energy minimization.

III. Online Complexity Control and Energy Minimization for Real-Time Video Encoding Over Portable Devices

In this section, we provide an overview of the proposed method for online complexity control and energy minimization of real-time video encoding over portable devices.

A. Problem Formulation

The video encoder has a number of complexity control parameters $(\xi_1, \xi_2, \ldots, \xi_L)$. At each video frame (or time instance) n, the video encoder can adjust these parameters to control its computational complexity and energy consumption. For convenience, we refer to the encoder decision on these parameters at time instance n as an action, denoted by $A(n) = [A_1(n), A_2(n), \ldots, A_L(n)]$. Let R be the coding bit rate which is determined by the available transmission bandwidth and $D[R; A_1(n), A_2(n), \ldots, A_L(n)]$ be the corresponding video coding distortion when the encoder chooses action $A(n) = [A_1(n), A_2(n), \ldots, A_L(n)]$.

Suppose the video sequence has K video frames. During the encoding processing, the encoder needs to choose a series of actions $\{A(n) | 1 \leq n \leq N\}$, as illustrated in Fig. 1, such that the overall rate-distortion performance of the video encoder is maximized under complexity or energy constraints. Conceptually, this optimization problem can be formulated as follows:

$$\min_{\{A_1(n),\, A_2(n),\ldots,A_L(n)\}} \frac{1}{N}\sum_{n=1}^{N} D[R; A_1(n),\ A_2(n), \ldots, A_L(n)]$$

$$\text{s.t.} \quad \frac{1}{N}\sum_{n=1}^{N} P[A_1(n),\ A_2(n), \ldots, A_L(n)] \leq P_0. \qquad (2)$$

For effective online complexity control and energy minimization of real-time video encoding, the following three major issues need to be carefully addressed. First, we need to understand and model the encoder behavior, i.e., the relationship between the encoder action $A(n) = [A_1(n), A_2(n), \ldots, A_L(n)]$ and the corresponding rate-distortion performance $D(n)$ and power consumption $P(n)$. As we know, an efficient video encoder often involves highly sophisticated spatiotemporal prediction schemes, data representation algorithms, and entropy coding methods [3], [4]. Thus, the encoder behavior is often very complex and remains a "black box" to us. In addition, this encoder behavior depends on the characteristics of the input video sequence. In real-time video coding, the characteristics of the input video remain unknown to us before its encoding is completed. Therefore, in online complexity control and energy minimization, we need to learn and identify the encoder behavior on the fly by examining previous encoder actions and the corresponding system outputs $[D(n), P(n)]$. Second, in online complexity control and energy minimization of real-time video encoding, the encoder is not able to reverse previous actions once they have been taken. For example, at frame k, the encoder learns that the characteristics of the input video are different from previous learning results and realizes that those actions taken in the previous frames are not optimum. Third, the online complexity control and energy minimization is a process control problem [29]. More specifically, the encoder needs to determine a sequence of actions to maximize the rate-distortion performance under energy constraints. At each time instance, it needs to learn the encoder behavior and choose the action which can most likely optimize the overall encoder performance. Note that video frames are encoded with temporal prediction and all frames are competing for computational and energy resources. Therefore, the effect of each action is not known until the end of control process. This is one of the major challenges in online complexity control and energy minimization of real-time video coding. To address these issues and challenges, we propose to explore adaptive critic design, an advanced method recently developed for adaptive system control [29].

B. Adaptive Critic Design

Optimizing a desired metric over time is one of the central problems in control and optimization [31]. Dynamic programming (DP) uses the Bellman principle of optimality to find an optimal sequence of actions [30]. Classical DP methods discretize the state space and compare the costs associated with all feasible trajectories to find the optimum solution. However, these DP approaches suffer from curse of dimensionality, with the computational complexity growing exponentially with the number of state variables [30]. For a given series of control actions that must be taken sequentially, and not knowing the effect of these actions until the end of the sequence, it is impossible to design an optimal controller using traditional supervised learning neural networks [32]. Adaptive critic designs (ACD) is a class of approximate dynamic programming (ADP) methods that uses incremental optimization, combined with parametric structures that approximate the optimal cost and control, to reduce the required computations [29], [31]. ACD combines the concepts of reinforcement learning and approximates dynamic programming. At any moment in time, it optimizes a short-term cost metric that ensures incremental optimization of the cost over all future times. It is able to determine a sequence of actions to maximize the overall performance of the whole process. ACD simulates critical thinking of human intelligence with the ability to cope with a large number of variables in parallel, in real-time, and in a noisy nonlinear nonstationary environment [32]. ACD controllers have been successfully trained offline for two-axle vehicle steering and speed control, agile missile interception, aircraft autolanding and control, and turbo-generator control [29], [32].

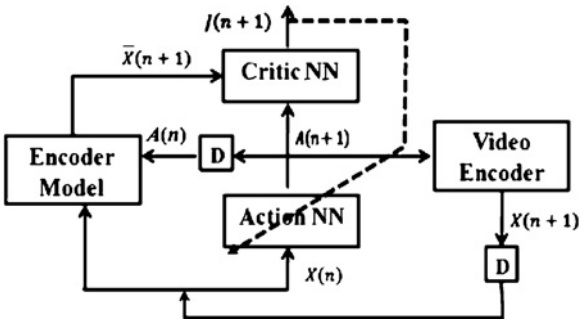

Fig. 2. Proposed ACD framework for online complexity control and optimization.

IV. Adaptive Critic Design for Energy Control and Optimization

In this paper, we observe that the problem of online complexity control and energy minimization shares similar characteristics as those control problems which have been successfully solved using ACD. Therefore, we propose to explore the ACD approach for energy minimization of real-time video encoding on portable devices. The proposed ACD framework for online complexity control and energy minimization consists of three major components: an encoder model, a critic network, and an action network, as illustrated in Fig. 2.

In ACD-based energy control and optimization, the system examines its current state in power consumption and video encoding performance and decides its *action* (complexity control parameters) for the next step. The action will change the system to its next state. For each action, the system performance gain is described by a local utility function. The ACD-based control and optimization module has three major components: a critic network to learn the global performance metric function $J(n)$, an action network to determine the next action based on current state $X(n)$, and a model network to learn the system behavior, predicting the next system state $X(n+1)$ from the current state $X(n+1)$.

The critic neural network (NN) is used to approximate the cost-to-go function (or performance metric), denoted by $J(n+1)$ in the Bellman equation of dynamic programming. Here, n is a time index. For convenience, n also represents the frame number. The inputs to this critic NN are encoder action $A(n+1)$ and encoder state $X(n+1) = [D(n+1), P(n+1)]$, where $D(n+1)$ and $P(n+1)$ are video coding distortion and encoder power consumption at time $n+1$. In this paper, we choose two complexity control parameters as the action $A(n) = [A_1(n), A_2(n)]$, where $A_1(n)$ is the number of search positions in motion estimation and $A_2(n)$ is the video coding frame rate (i.e., the number of coding frames per second). It can be seen that $A_1(n)$ and $A_2(n)$ control the computational complexity of motion search and frame-level video encoding. The action network considers the current system state $X(n) = [D(n), P(n)]$ and determines the encoder action for the next step which aims to minimize the cost-to-go function $J(n+1)$ learned by the critic network. The encoder model identifies the encoder behavior and determines how the system states will be changed if the current action is taken. Basically, it estimates the next system state $\bar{X}(n+1) = [D(n+1), P(n+1)]$ based upon the current encoder action $A(n)$. We assume a rate control algorithm operates inside the encoder such that the encoding bit rate R remains relatively constant throughout the process [8]. The interconnection between the video encoder and these three networks are shown in Fig. 2. In the following sections, we will explain the specific design of these three networks in more detail.

A. Critic Network Design

ACD approximates dynamic programming methods using incremental optimization whose parametric structures approximate the optimal cost and control. In ACD, a "primary" utility function $U(n)$ is used to describe the immediate impact of current actions of the video encoder. For example, if we aim to minimize the video coding distortion $D(n)$ under a complexity constraint $C(n) \leq C_0$, one can choose the following primary utility function:

$$U(n) = D(n)^2 + \lambda\, \phi\,(C(n) - C_0) \tag{3}$$

where $\phi(x)$ is a penalty function which remains very small when $x \leq 0$ and takes a very large value when $x > 0$. Certainly, the choice of the primary utility should depend on the control and optimization objective. Based on this primary utility function, in ACD, a process-level performance metric function is formed as follows:

$$\bar{J}(n) = \sum_{k=1}^{\infty} \gamma^k\, U(n-k) \tag{4}$$

where $0 < \gamma < 1$ is a discount factor. Unfortunately, this performance metric $\bar{J}(n)$ is not easily computable in many applications. Note that $\bar{J}(n)$ satisfies the following Bellman recursion property [29]:

$$\bar{J}(n+1) = U(n) + \gamma \bar{J} \sum_{k=1}^{\infty} \gamma^k\, U(n). \tag{5}$$

In ACD, this property is used to obtain $\bar{J}(n)$ using a critic neural network. The critic network aims to minimize the following error measure over time:

$$E_1^c(n) = \frac{1}{2}[J(n+1) - \gamma\, J(n) - U(n)]^2. \tag{6}$$

Here, $J(n)$ is the output of the critic network at time n. When the critic network converges, the error in (6) will approach zero and $J(n)$ will approach the desired performance metric $\bar{J}(n)$. This is because, when the critic neural network converges, $E_1^c(n)$ becomes zero. This implies that the relationship in (5) holds, from which the original expression of $\bar{J}(n)$ in (4) can be derived. Fig. 3 shows the basic structure of the proposed critic neural network. It has three layers: input, hidden, and output layers. In this critic neural network, the performance metric function $J(n)$ is represented by a nonlinear weighted sum of the encoder states and complexity control parameters. Through online learning [9], these weights will be adjusted such that the error function in (6) will be minimized. In this paper, we follow the convention in neural network

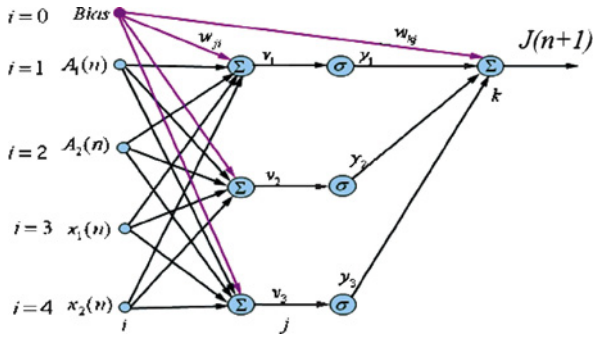

Fig. 3. Critic neural network.

literature [9] to use i to index neural network inputs, j to index hidden neurons, and k to index network outputs, as illustrated in Fig. 3. The proposed critical network takes the encoder state $X(n) = [D(n), P(n)]$ and encoder action $A(n) = [A_1(n), A_2(n)]$ as inputs and has three hidden neurons $\{v_j | \leq j \leq 3\}$. A bias term is included in the input so that a DC value can be added to $J(n)$. We index these five input variables from top to bottom by $i, 0 \leq i \leq 4$. Let $W_{ji} = [W_{j0}, W_{j1}W_{j2}W_{j3}W_{j4}], 1 \leq j \leq 3$, be the synaptic weights connecting these five input variables and three hidden neurons. At time n, the output of the hidden neurons v_j is given by

$$v_j(n) = W_{ji} \cdot [1, D(n), P(n), A_1(n), A_2(n)]'. \tag{7}$$

In the hidden layer, this hidden neuron output is passed through an active function $\sigma(v)$

$$y_j(n) = \sigma(v_j(n)) \frac{1}{1 + e^{-v_j(n).}}. \tag{8}$$

The final output of the critic neural network $j(n)$ is given by the weighted summation of $y_i(n)$

$$J(n+1) = [W_{k1}, W_{k2}, W_{k3}, W_{k4}] \cdot [1, y_1(n), y_2(n), y_3(n)]'. \tag{9}$$

During online complexity control and energy minimization, the critic network aims to minimize the error metric in (6) by updating its weights. In this paper, we use the back-propagation method, a commonly used procedure for neural network learning [9]. The basic idea of back-propagation can be summarized as follows:

$$\begin{pmatrix} weight \\ correction \end{pmatrix} = \begin{pmatrix} learning\ rate \\ parameter \end{pmatrix} \cdot \begin{pmatrix} local \\ gradient \end{pmatrix} \cdot \begin{pmatrix} input\ signal \\ of\ neuron \end{pmatrix}. \tag{10}$$

Here, a gradient descending method is used to update the network weights so as to minimize the error function [9]. According to this back-propagation method, the critic network weights $W_{kj} = [W_{k1}, W_{k2}, W_{k3}, W_{k4}]$ at the output layers are updated as follows:

$$\begin{aligned} \Delta W_{kj}(n) &= -\eta \frac{\partial E_1^c(n)}{\partial w_{kj}} = -\eta \frac{\partial E_1^c(n)}{\partial w_{kj}} \cdot \frac{\partial E_1^c(n)}{\partial w_{kj}} \\ &= \eta[J(n+1) - \gamma J(n) - U(n)] \cdot \sigma(v_j(n)). \end{aligned} \tag{11}$$

Here, η is the positive learning rate for the back-propagation training procedure. In (11), the input signals for the output layer are generated by the hidden layer, $y_j(n) = \sigma(v_j(n))$, and the local gradient is defined by

$$\delta_j(n) = -\frac{\partial E_1^c(n)}{\partial J(n)} = -[J(n+1) - YJ(n) - U(n)]. \tag{12}$$

Similarly, at the hidden layer, the weights $W_{ji} = [W_{j0}, W_{j1}, W_{j2}, W_{j3}, W_{j4}]$ are updated as follows:

$$\Delta w_{ji} = \eta \delta_j(n) y_i(n). \tag{13}$$

Here, η is the learning rate. In this paper, we set its value to be 0.5. The local gradient $\delta_j(n)$ is given by

$$\begin{aligned} \delta_j(n) &= -\frac{\partial E_1^c(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} \\ &= -[J(n+1) - yJ(n) - U(n)] \\ &\quad \cdot \sigma_j'(v_j(n)) \cdot w_{kj}(n). \end{aligned} \tag{14}$$

In this paper, we use the sigmoid function $\sigma(\mathrm{v}) = \frac{1}{1+e^{-\mathrm{v}}}$ as the active function, which has been extensively used in neural network design [9]. Its first order and second order derivatives are given by

$$\begin{cases} \sigma'(\mathrm{v}) = \sigma(\mathrm{v}) \cdot (1 - \sigma(\mathrm{v})) \\ \sigma''(\mathrm{v}) = \sigma(\mathrm{v}) \cdot (1 - \sigma(\mathrm{v})) \cdot (1 - 2\sigma(\mathrm{v})). \end{cases} \tag{15}$$

The critic neural network connects with the action and model networks as follows: we first execute the action network to obtain $A(n+1)$ and the model network to obtain $X(n+1)$. We then use the critic network to obtain $J(n+1)$. Using (10) and (11), we update the weights of the critic neural network.

B. Action Network Design

During online complexity control and energy minimization, the action neural network determines the next encoder action $A(n+1)$ so as to minimize the performance metric function $J(n+1)$ generated by the critic network. This implies that the action network needs to let the derivative of $J(n+1)$ with respect to the encoder action $A(n+1)$ approach zero. In other words, we expect to have

$$\lambda_A(n+1) = \frac{\partial J(n+1)}{\partial A(n+1)} = 0. \tag{16}$$

To this end, we define the following error metric:

$$E_A = \frac{1}{2} \lambda_A'(n+1) \cdot \lambda_A'(n+1). \tag{17}$$

The action network aims to minimize this error metric by updating its weights. The basic observation is that, when this error metric approaches zero, $\lambda_A(n+1)$ is close to zero, and the encoder action minimizes the performance metric function $J(n+1)$.

Fig. 4 shows the structure of the action network. The inputs to the action neural network are the encoder states $[x_1(n), x_1(n)] = [D(n), P(n)]$ plus a bias term. The network output is the encoder action $A(n+1) = [A_1(n+1), A_2(n+1)]$ at the next time step. In this action network, the encoder action at the next time instance is represented using a non-linear weighted sum of the current encoder states. Our basic observation is that the encoder determines its next action

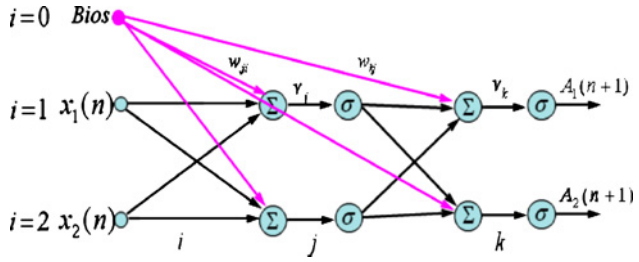

Fig. 4. Action neural network.

based on its current state. We use the action neural network to approximate the relationship between encoder action and state.

As illustrated in Fig. 4, the proposed action network has one hidden layer and one output layer. In the hidden layer, the network inputs, i.e., $[x_1(n), x_2(n)]$ plus a bias term, are aggregated into three hidden neurons ($1 \le j \le 3$) using weighted summation. The corresponding weights are denoted by w_{ji} where i and j are indices for network inputs and layer-1 hidden neurons. The summation output v_j is passed through the active function $\sigma(v)$ given in (8). At the output layer, these three action function outputs from hidden layers are aggregated into two summation outputs $v_k, 1 \le k \le 2$ using weighted summation. The corresponding weights are denoted by w_{ki}, where k is an index for network outputs. Unlike the critic network which has only one network output $J(n)$, the action network has two outputs: $A_1(n+1)$ and $A_2(n+1)$, which control the computational complexity of encoder motion search and frame rate, respectively. The summation outputs of output layer are then passed through $\sigma(v)$ again to generate the final output. Here, $\sigma(v)$ serves as a saturation function, making sure that the values of $A_1(n+1)$ and $A_2(n+1)$ are within the normalized interval of [0, 1].

In the following, we explain the weight updating in more detail. Again, we use the back-propagation method to update the network weights. More specifically, at the output layer, according to the chain rules and the back-propagation weight updating scheme described in (10), the weights are updated as follows (due to page limitation, the detailed procedure to compute the following derivatives is omitted):

$$\Delta W_{kj}(n) = -\eta \frac{\partial J(n+1)}{\partial A_k(n+1)} \cdot \frac{\partial J^2(n+1)}{\partial A_k(n+1)\,\partial A_k(n+1)} \sigma_j(v_j(n)) \cdot \sigma_k'(v_k(n)). \tag{18}$$

At the hidden layer, the weights are updated as follows:

$$\begin{cases} \Delta W_{ji}(n) = \eta\, \delta_j(n) y_i(n) \\ \delta_j(n) = -\sum_k \frac{\partial j(n=1)}{\partial A_k(n+1)} \cdot \frac{\partial j^2(n+1)}{\partial A_k(n+1)\,\partial A_k(n+1)} \\ \qquad w_{kj} \cdot \sigma_j'(v_j(n)) \cdot \sigma_k'(v_k(n)). \end{cases} \tag{19}$$

It needs to be noted that $\partial j(n)/\partial A_k(n)$ and $\partial j^2(n)/\partial A_k(n)\partial A_k(N)$ need to be derived from critic network as follows:

$$\begin{cases} \frac{\partial J(n)}{\partial A_i(n)} = \sum_j w_{kj} \cdot \sigma'(v_j(n)) \cdot w_{ji} \\ \frac{\partial J^2(n)}{\partial A_i(n)\partial A_i(n)} = \sum_j w_{kj} \cdot \sigma"(v_j(n)) \cdot w_{ji} \cdot w_{ji}. \end{cases} \quad (i = 1, 2) \tag{20}$$

In (19), $\sigma'(v_j(n))$ and $\sigma"(v_j(n))$ can be determined by (15).

C. Model Network

In our online complexity control and energy minimization, we use the model network to approximate the encoder behavior. More specifically, the task of model network is to learn a mathematical model which is able to predict the next system state $X(n+1) = [D(n+1), C(n+1)]$ for the given current system state $X(n) = [D(n), C(n)]$ and the encoder action $A(n)$. In the classical adaptive critic design, this model is implemented by a neural network [29], [32]. In this paper, we find that this approach is not effective for nonstationary input videos. When the input scene characteristics change significantly over time, the neural network is not able to quickly respond to these rapid changes by updating its network weights. On the other hand, if we choose to use a high learning rate, the neural network may not converge and become unstable.

To address this issue, in this paper, we propose to replace the model neural network in the classic adaptive critic design using an adaptive model fitting approach. During our extensive simulations, we find that there is an exponential relationship between the encoder distortion $D(n+1)$ and the complexity control parameters and an approximate linear relationship between the encoder computational complexity $C(n+1)$ and control parameters. For example, we have found that if we increase the number of motion search positions or video coding frame rate, the video coding distortion decreases exponentially and the complexity increases linearly. Based on this observation, the following model is used:

$$\begin{cases} D(n+1) = a_3 \exp(-[a_1 A_1(n) + a_2 A_2(n)]) \\ C(n+1) = b_1 A_1(n) + b_2 A_2(n). \end{cases} \tag{21}$$

Certainly, the expression of this model depends on the specific choice of complexity control parameters. If different complexity control parameters are used, one needs to examine the corresponding simulation results and use different models.

To obtain the model parameters $\{a_1, a_2, a_3, b_1, b_2\}$ in (21), we use the encoder statistics from the past Δ frames. More specifically, when an action $A(n)$ is determined by the action network, the encoder will use this action, i.e., the corresponding complexity control parameters to encode the next video frame, as illustrated in Fig. 2. The encoder will then report the coding distortion $D(n+1)$ and computational complexity (in units of microprocessor cycles) $C(n+1)$ to the model network. We then use these data points $\{A_1(n), A_2(n), D(n+1), C(n+1)\}$ of the past Δ frames to estimate the model parameters using the Levenberg–Marquardt algorithm [12]. Fig. 5 shows the fitting results for encoder complexity and distortion for *Foreman* QCIF video. Experiments over other sequences yield similar results. It can be seen that the fitting is fairly accurate. Once

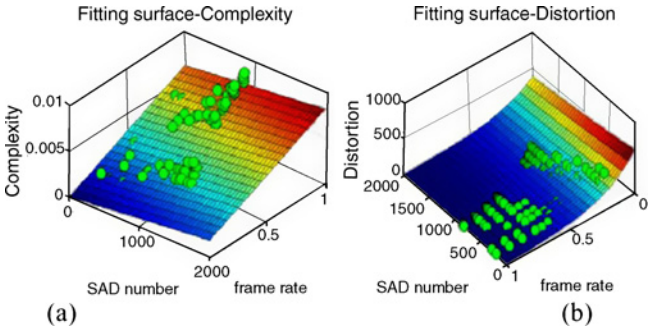

Fig. 5. (a) Surface fitting for encoder complexity. (b) Surface fitting for the encoder distortion.

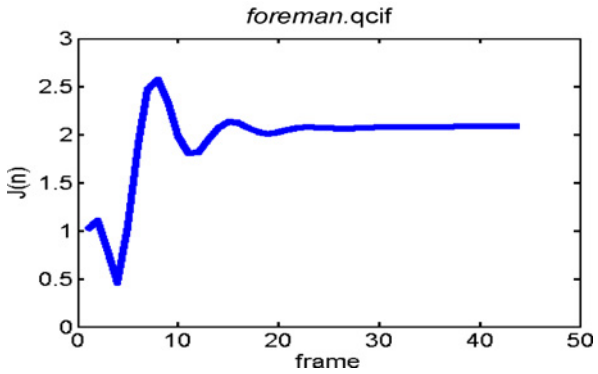

Fig. 6. Convergence of performance metric function on *Foreman* sequence.

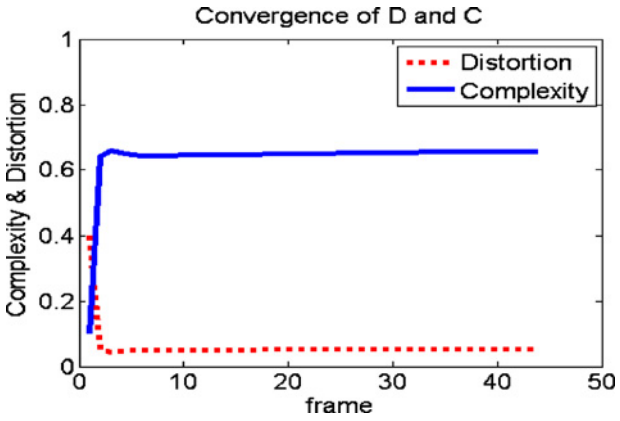

Fig. 7. Convergence of encoder state on *Foreman* sequence.

the mathematical model is obtained, it will be used in the action network to update its network weights, as shown in (17).

V. Algorithm

Critic network, action network, and encoder model are three major components of the proposed ACD algorithm for energy minimization. The interconnection between them is shown in Fig. 2. In this section, we outline the major steps in online complexity control and energy minimization algorithm for real-time video encoding.

Step-1: Initialization. Within the initial period, the ACD algorithm tries to learn the encoding behavior of the input video sequence. More specifically, within the first Δ frames, the ACD algorithm is not applied. We just use empirical values for the complexity control parameters. After frame $n\,(1 \leq n \leq \Delta_0)$ is encoded, we record the video encoding distortion $D(n)$ and computational complexity $C(n)$. Also initialize the action and critic network.

Step-2: Model Update. Based on the video encoding statistics of past Δ frames $\{[D(n), C(n)]|1 \leq n \leq \Delta\}$, update the model parameters in (21) as explained in Section IV-C. With the updated encoder model, estimate the next encoder state $X(n+1)$ with the current action $A(k)$ and current state $X(n)$.

Step-3: Action Network. Based on the current encoder state $X(n)$, the action network determines the next action $A(n+1)$ and updates its weights which aim to minimize the performance metric $J(n)$, as explained in Section IV-B.

Step-4: *Critic Network*. With the new action $A(n+1)$ and estimated encoder state $X(n+1)$, the critic network computes the performance metric $J(n+1)$ and update its weights, as explained in Section IV-A.

Step-5: *Video Encoding*. The video encoder uses the new action $A(n+1)$ to control its motion estimation and MB encoding modules and encodes the $(n+1)$th video frames. The corresponding video encoding distortion $D(n+1)$ and computational complexity $C(n+1)$ are then recorded.

Step-6: *Loop*. Repeat Steps 2–5 until all frames are encoded.

It can be seen that the major computational complexity of the proposed algorithm lies in updating the network weights, as discussed in Section IV. Note that the weight update only involves a small number of arithmetic operations and function evaluations. Therefore, the proposed algorithm has very low computational complexity. According to our simulation experience and rough estimation, its complexity is less than 5% of the video encoding complexity.

With slight modifications, the proposed ACD algorithm can be also extended to other video encoders with different complexity control parameters. In typical/standard video encoding, such as H.264 and MPEG-4, the number of *major* parameters that we can control and have *significant* impact on the overall encoder complexity are limited, typically less than 10. For example, one can choose the number of motion search positions, the number of reference frames, the number of coding modes, frame rates, etc., as complexity control parameters. Note that the major complexity of the proposed ACD algorithm lies in two neural networks, more specifically, the back-propagation process. According to [9], the complexity of training a neural network and updating its weights increases in a linear or quadratic fashion with the total number of network inputs (or control parameters). Therefore, we expect that the overall complexity of the ACD control algorithm will be within a reasonable range when applied to different video encoders. In practice of energy minimization, the device is operating for hours. We do not need to control and adjust the encoder at the frame-level. It will be sufficient to adjust the encoder parameters for every few seconds or even minutes, or after scene changes. In this case, the overall control complexity will be significantly reduced.

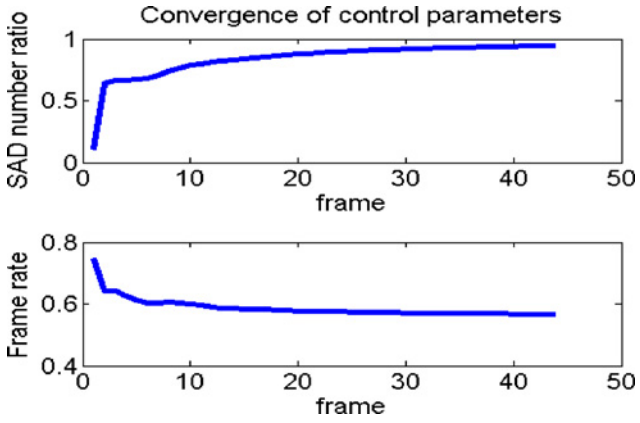

Fig. 8. Convergence of control parameters on *Foreman* sequence.

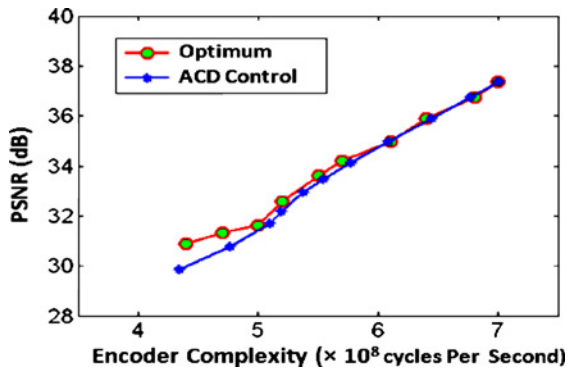

Fig. 9. Comparison between optimal operational performance and ACD control on *Foreman* sequence.

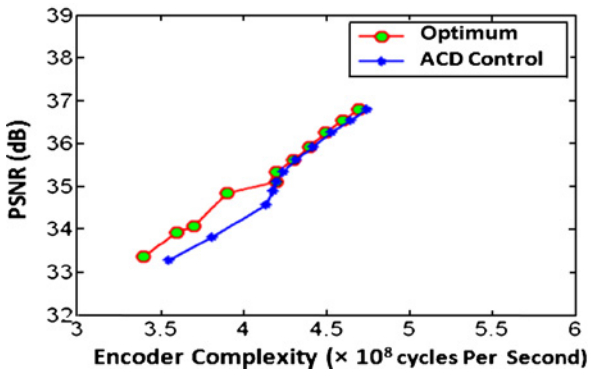

Fig. 10. Comparison between optimal operational performance and ACD control on *News* sequence.

VI. Experimental Results

We have implemented the ACD-based algorithm for online complexity control and energy minimization in MATLAB. This MATLAB control module interacts with a real-time MPEG-4 video encoder. We use a GOP (group of picture) size of 50 with the first frame as an Intra frame. This MPEG-4 encoder has an internal rate control algorithm [8] to adjust its quantization step size to achieve the target bit rate. After encoding a video frame, the encoder sends the encoder statistics, including video coding distortion $D(n)$ and power consumption $P(n)$, to the MATLAB control module.

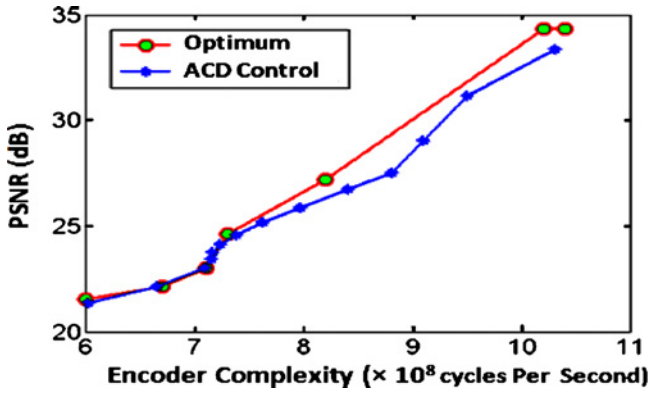

Fig. 11. Comparison between optimal operational performance and ACD control on *Football* sequence.

Fig. 12. Subjective video quality comparison between optimal operational performance (top) and online ACD control (bottom).

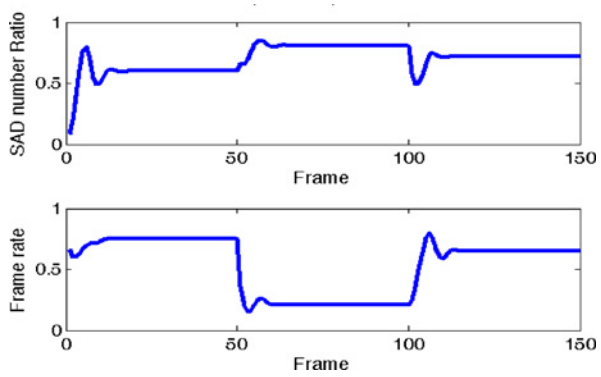

Fig. 13. Convergence of encoder states on the *Combo* video.

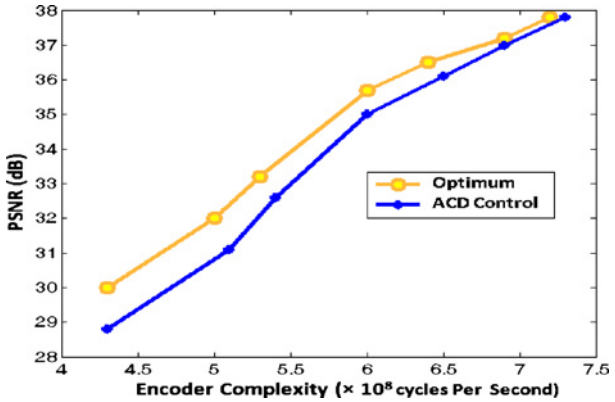

Fig. 14. Video quality comparison between the optimal control and the proposed ACD control on the *Combo* sequence.

The control module determines the action $A(n+1)$ for the next time step and sends this control signal to the encoder. The encoder will use this information to control the computational complexity of its motion search and MB encoding modules. Since the proposed algorithm has very low computational complexity, this MATLAB control is able to run in real-time simultaneously with the video encoder.

The test videos used in this paper are standard test video sequences, including *Foreman*, *News*, and *Football* QCIF (176×144) videos. As discussed in Sections IV and V, we use the number of SAD computations and frame rate as the complexity control parameters, namely, the encoder actions $X(n)$. We use the number of processor cycles per second used by the video encoder to measure the encoder complexity $C(n)$. At frame n, the proposed online complexity control and energy minimization algorithm will determine the complexity control parameters $X(n)$ which aim to minimize the overall distortion under a complexity constraint, or equivalently a power constraint. Using the operational power-rate-distortion analysis procedure outlined in Section II, we obtain the true optimum complexity control parameters and the corresponding minimum video distortion. We then compare the video distortion obtained by our algorithm against the true minimum distortion.

Figs. 6–9 show the experimental results for *Foreman*. In this experiment, the coding bit rate is set to be 96 kb/s (bits per second). Fig. 6 shows the performance metric function $J(n)$ generated by the critic network. We can see that after 20 frames, $J(n)$ converges to a stable condition, which implies that the effective performance metric function has been captured by the critic network. This performance metric function will be used by the action network to determine the encoder actions, which aim to maximize the overall performance over the entire process. Fig. 7 shows the convergence behavior of encoding distortion $D(n)$ and $C(n)$ of the first 45 video frames. Fig. 8 shows the sequence of encoder actions $A(n)$ determined by the action network and used by the video encoder. In Fig. 9, we compare the actual complexity–distortion curve obtained by our algorithm and the true optimum one obtained by operational P-R-D analysis. The horizontal axis represents the encoder computational complexity. It can be seen that the actual video coding distortion (measured in PSNR, peak signal-to-noise-ratio) is very close to the true optimum, especially at higher PSNR values. The average and maximum gap between them are about 0.3 dB and 1.0 dB, respectively. Figs. 10 and 11 show the experimental results for *News* encoded at 64 kb/s and *Football* sequence encoded at 256 kb/s, respectively. We observe that, in the *News* sequence, the convergence speed of encoder control parameters is slower than that of *Foreman*. This is because the *News* video has less object motion than the *Foreman* sequence and the encoder performance is much more sensitive to changes in encoder actions. Fig. 12 shows three sample video frames encoded by the MPEG-4 encoder with optimal complexity control (top) and online ACD-based complexity control. It can be seen that they have nearly the same visual quality. All the above experimental results demonstrate that the proposed online complexity control algorithm achieves a near-optimum performance.

Next, we evaluate the performance of the ACD algorithm on input videos with scene changes. To this end, we create a *Combo* video by concatenating the first 50 frames of *NBA*, *Coastguard*, and *Stefan* (176×144) videos. We set the target encoding bit rate to be 256 kb/s. Fig. 13 shows the complexity control parameters. It can be seen that the encoder is able to adapt its control parameters to scene content changes in the input video through online learning and converge to a steady state quickly. Fig. 14 shows the video quality with ACD control against the ground-truth obtained from offline operational P-R-D analysis. It can be seen that the quality gap is a little bit larger than those in the previous three experiments. This is because, during online complexity control, the algorithm does not have access to statistics of future frames. These types of decisions on encoder actions based on partial information will cause performance degradation, especially for videos with scene changes.

VII. Further Discussion and Concluding Remarks

Online complexity control and energy minimization for real-time video encoding is a complicated adaptive control process. The encoder needs to select a sequence of actions which are able to maximize the overall performance, or more specifically, minimize the overall video coding distortion, under a complexity/power consumption constraint. To solve this problem, we explored the adaptive critic design, a recently developed method for adaptive control of complex systems, for online complexity control and energy minimization of real-time video encoders. We have studied the power consumption behavior of a real-time embedded video encoding system. We present the operational P-R-D analysis which is used to obtain the true optimum performance for performance evaluation. We present the specific adaptive critic network design for real-time video encoding. Our experimental results demonstrate that the proposed algorithm achieves near-optimum performance.

In this paper, we use two parameters, i.e., the number of motion search positions and frame rate, to control the computational complexity of the video encoder. According to our simulations, these two control parameters are very effective, scaling down the overall encoder complexity by up to 10 times, i.e., nearly 10% of the original complexity. Certainly, in practical video encoding over portable devices, a video encoder will also have other energy-consumption modules, such as memory access. A recent study [10] shows that with proper encoder pipeline and data flow optimization, the energy consumption of memory access only contributes to a relatively small part (about 12%) of the total encoder energy consumption.

We do recognize that our current online energy control and optimization algorithm has an average of 0.5 dB (sometimes up to 1–2 dB) in performance loss, especially for videos with strong motion when compared to the offline brute-force optimization scheme. In our future work, we shall investigate more efficient control and optimization algorithms to reduce this performance gap. We shall also implement the proposed algorithm in a portable video encoding system and evaluate its performance in complexity control and energy minimization.

We shall also study how the algorithm is able to handle uncertainty or modeling errors of the video encoder, since sometimes it is hard to establish an accurate behavior model for complicated video encoders, such as H.264.

REFERENCES

[1] S. Ohmori, Y. Yamao, and N. Nakajima, "The future generations of mobile communications based on broadband access technologies," *IEEE Commun. Mag.*, vol. 38, no. 12, pp. 134–142, Dec. 2000.

[2] D. LeGall, "MPEG: A video compression standard for multimedia application," in *Proc. Commun. ACM*, vol. 34. Apr. 1991, pp. 46–58.

[3] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuit. Syst. Video Technol.*, vol. 7, no. 1, pp. 19–31, Feb. 1997.

[4] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuit. Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[5] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraint," *IEEE Trans. Circuit. Syst. Video Technol.*, vol. 15, no. 5, pp. 645–658, May 2005.

[6] X. Lu, Y. Wang, and E. Erkip, "Power efficient H.263 video transmission over wireless channels," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1. Rochester, NY, Sep. 2002, pp. 533–536.

[7] Z. He, W. Chen, and X. Chen, "Extending the operational lifetime of portable video communication devices using power-rate-distortion optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 5, pp. 596–608, May 2008.

[8] Z. He and S. K. Mitra, "A unified rate-distortion analysis framework for transform coding," *IEEE Trans. Circuit. Syst. Video Technol.*, vol. 11, no. 12, pp. 1221–1236, Dec. 2001.

[9] S. Haykin, *Neural Network: A Comprehensive Foundation*. 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, Jul. 1998.

[10] Y. Hu, Q. Li, and C.-C. J. Kuo, "Run-time power consumption modeling for embedded multimedia systems," in *Proc. 11th IEEE Int. Conf. Embedded Real-Time Comput. Syst. Appl.*, Hong Kong, China, Aug. 17–19, 2005, pp. 353–356.

[11] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *IEEE Circuit. Syst. Mag.*, vol. 4, no. 1, pp. 7–28, Jan.–Mar. 2004.

[12] K. Levenberg, "A method for the solution of certain nonlinear problems in least squares," *Quart. Appl. Math.*, vol. 2, pp. 164–168, 1944.

[13] D. S. Turaga, M. van der Schaar, and B. Pesquet-Popescu, "Complexity-scalable motion compensated wavelet video encoding," *IEEE Trans. Circuit. Syst. Video Technol.*, vol. 15, no. 8, pp. 982–993, Aug. 2005.

[14] I. M. Pao and M. T. Sun, "Statistical computation of discrete cosine transform in video encoders," *J. Visual Commun. Image Representation*, vol. 9, no. 2, pp. 163–170, Jun. 1998.

[15] J. Villasenor, C. Jones, and B. Schoner, "Video communications using rapidly reconfigurable hardware," *IEEE Trans. Circuit. Syst. Video Technol.*, vol. 5, no. 6, pp. 565–567, Dec. 1995.

[16] W. P. Burleson, P. Jain, and S. Venkatraman, "Dynamically parameterized architecture for power-aware video coding: Motion estimation and DCT," in *Proc. 2nd USF Int. Workshop Digital Comput. Video*, 2001, pp. 8–12.

[17] P. Agrawal, J.-C. Chen, S. Kishore, P. Ramanathan, and K. Sivalingam, "Battery power sensitive video processing in wireless networks," in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun.*, vol. 1. Boston, MA, Sep. 1998, pp. 116–120.

[18] W. Yuan, K. Nahrstedt, S. V. Adve, R. H. Kravets, and D. L. Jones, "GRACE-1: Cross-layer adaptation for multimedia quality and battery energy," *IEEE Trans. Mobile Comput.*, vol. 5, no. 7, pp. 799–815, Jul. 2006.

[19] D. G. Sachs, S. Adve, and D. L. Jones, "Cross-layer adaptive video coding to reduce energy on general-purpose processors," in *Proc. Int. Conf. Image Process. (ICIP '03)*, vol. 3. Barcelona, Spain, Sep. 2003, pp. 109–112.

[20] V. Akella, M. van der Schaar, and W.-F. Kao, "Proactive energy optimization algorithms for wavelet-based video codecs on power-aware processors," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2005, pp. 566–569.

[21] A. N. Skodras, "Fast discrete cosine transform pruning," *IEEE Trans. Signal Process.*, vol. 42, no. 7, pp. 1833–1837, Jul. 1994.

[22] J. Lorch and A. Smith, "Improving dynamic voltage scaling algorithms with PACE," in *Proc. 2001 ACM SIGMETRICS Conf.*, Jun. 2001, pp. 50–61.

[23] F. Pan, X. Lin, R. Susanto, K. P. Lim, Z. G. Li, G. N. Feng, D. J. Wu, and S. Wu, "Fast mode decision algorithm for intra-prediction in H.264/AVC video coding," *IEEE Trans. Circuit. Syst. Video Technol.*, vol. 15, no. 7, pp. 813–822, Jul. 2005.

[24] Z. He, J. Eggert, X. Zhao, R. Moll, and J. Millspaugh, "Energy-aware portable video communication system design for wildlife activity monitoring," *IEEE Circuit. Syst. Mag.*, vol. 8, no. 1, pp. 25–37, May 2008.

[25] B. Zeng, R. Li, and M. L. Liou, "Optimization of fast block motion estimation algorithms," *IEEE Trans. Circuit. Syst. Video Technol.*, vol. 7, no. 6, pp. 833–844, Dec. 1997.

[26] P.-C. Tseng, Y.-C. Chang, Y.-W. Huang, H.-C. Fang, C. Huang, and L.-G. Chen, "Advances in hardware architectures for image and video coding: A survey," *Proc. IEEE*, vol. 93, no. 1, pp. 184–197, Jan. 2005.

[27] P. Pirsch and H.-J. Stolberg, "VLSI implementations of image and video multimedia processing systems," *IEEE Trans. Circuit. Syst. Video Technol.*, vol. 8, no. 7, pp. 878–891, Nov. 1998.

[28] Z.-L. He, C.-Y. Tsui, K.-K. Chan, and M. Lion, "Low-power VLSI design for motion estimation using adaptive pixel truncation," *IEEE Trans. Circuit. Syst. Video Technol.*, vol. 10, no. 5, pp. 669–678, Aug. 2000.

[29] S. Ferrari and R. F. Stengel, "On-line adaptive critic flight control," *J. Guidance Control Dynamics*, vol. 27, no. 5, pp. 777–786, Sep.–Oct. 2004.

[30] R. Bellman, "The structure of dynamic programming processes," *Dynamic Programming* (1957). Dover Paperback ed. Princeton, NJ: Princeton Univ. Press, ch. 3, 2003.

[31] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modelling," in *Handbook of Intelligent Control*, D. A. White and D. A. Sofge, Eds. New York: Van Nostrand Reinhold, 1992, pp. 493–525.

[32] G. K. Venayagamoorthy and R. G. Harley, "Adaptive critic designs for optimal control of power systems," in *Proc. 13th Int. Conf. Intell. Syst. Appl. Power Syst.*, Nov. 2005, pp. 136–148.

Zhao Sun received the B.S. degree from Nanjing University of Science and Technology, Nanjing, China, in 1989, the M.S. degree from Southeast University, Nanjing, China, in 1997, and the Ph.D. degree from North Carolina Agricultural and Technical State University, Greensboro, in 2006.

She is currently a Post-Doctorial Research Associate at the Center for Cooperative Control, National Institute of Aerospace, Hampton, VA. Her research interest includes intelligent system control.

Xi Chen received the B.S. degree in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 1998, the M.E. degree from Aerospace Institute, Beijing, China, in 2001, and the M.S. degree in electrical and computer engineering from the New Jersey Institute of Technology, Newark, in 2004. He is currently working toward the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Missouri, Columbia.

His main research interests include video compression/coding, video surveillance, and intelligent video processing. Currently, he is working on the optimized power-rate-distortion model for video encoders.

Zhihai He (S'98–M'01–SM'06) received the B.S. degree in mathematics from Beijing Normal University, Beijing, China, in 1994, the M.S. degree in mathematics from the Institute of Computational Mathematics, Chinese Academy of Sciences, Beijing, China, in 1997, and the Ph.D. degree in electrical engineering from the University of California, Santa Barbara, in 2001.

In 2001, he joined Sarnoff Corporation, Princeton, NJ, as a Member of Technical Staff. In 2003, he joined the Department of Electrical and Computer Engineering, University of Missouri, Columbia, as an Assistant Professor. His current research interests include image/video processing and compression, network transmission, wireless communication, computer vision analysis, sensor networks, and embedded system design.

Dr. He received the 2002 IEEE Transactions on Circuits and Systems for Video Technology Best Paper Award, and the SPIE VCIP Young Investigator Award in 2004. Currently, he serves as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and *Journal of Visual Communication and Image Representation*. He is also a Guest Editor for the IEEE TCSVT SPECIAL ISSUE ON VIDEO SURVEILLANCE. He is a Member of the Visual Signal Processing and Communication Technical Committee of the IEEE CIRCUITS AND SYSTEMS SOCIETY, and serves as a Technical Program Committee Member or Session Chair for a number of international conferences.