

Joint Source-Channel Coding and Channelization for Embedded Video Processing With Flash Memory Storage

Yiran Li, Guiqiang Dong, *Student Member, IEEE*, and Tong Zhang, *Senior Member, IEEE*

Abstract—This paper presents a joint source coding, channel coding, and flash memory channelization design framework to obtain optimized tradeoffs among energy consumption, bit rate, and end-to-end distortion (i.e., optimal E-R-D tradeoff space) for embedded and mobile devices with limited power source and abundant flash memory storage capacity. The optimal E-R-D tradeoff space enables embedded and mobile devices to cohesively optimize the source coding and data storage system operations subject to run-time power source, storage capacity, and/or distortion constraints. By treating flash memory as a communication channel, this work differs from classical joint source-channel coding from two perspectives: i) Classical joint source-channel coding aims to obtain an optimized R-D (bit rate and distortion) tradeoff space, while we aim to obtain an optimized E-R-D tradeoff space; ii) Flash memory can be configured (or channelized) over an energy consumption versus raw bit error rate tradeoff spectrum, and channelization is an integral part of the joint design. With the focus on video coding, this paper presents theoretical investigations and specific approaches for both scenarios where channel can and cannot contribute to end-to-end distortion. Based on detailed power estimation and representative video sequences, we quantitatively demonstrate the application of the proposed design approaches for obtaining optimized E-R-D tradeoff space.

Index Terms—Energy consumption, H.264/AVC, joint source-channel coding, NAND flash memory, video coding.

I. INTRODUCTION

CONTINUOUS technology scaling has enabled various embedded and mobile devices to incorporate an increasingly larger capacity of NAND flash memory as local data storage. However, just like hard disk drives in personal computers, it is not uncommon that a noticeable percentage of flash memory storage capacity is often left unused in embedded and mobile devices. This simple observation motivates us to investigate the possibility of exploiting the abundant flash memory storage capacity to enable certain system design innovations. Modern embedded and mobile devices typically contain several sensors, and may frequently compress/decompress and store/retrieve various sensor data, which can be energy hungry.

Manuscript received October 19, 2011; revised February 14, 2012 and April 16, 2012; accepted April 17, 2012. Date of publication May 02, 2012; date of current version July 10, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jarmo H. Takala.

The authors are with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, NY 12180 USA (e-mail: yiran.li@gmail.com, dongguiqiang@gmail.com, tzhang@ecse.rpi.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/TSP.2012.2197207

In addition, it is also not uncommon that embedded and mobile devices strive to sense, compress, and store as much data as possible subject to a limited (and dying) battery power source. Due to the inherent tradeoff between data compression energy consumption and compression efficiency, the abundant NAND flash memory storage capacity makes it possible to intentionally apply less sophisticated and lower power data compression algorithms at the penalty of compression efficiency in order to reduce system energy consumption. Nevertheless, worse compression efficiency directly results in a larger data volume and hence higher data storage/retrieval energy consumption. This clearly demands a system optimization by jointly considering data compression and data storage.

Motivated by the above discussions, we present design framework to jointly optimize data compression and NAND flash memory data storage in embedded and mobile devices. In this work, we treat NAND flash memory as a communication channel, and the system model consists of source encoding/decoding (in particular, video encoding/decoding), channel encoding/decoding, and the communication channel (i.e., NAND flash memory). We are interested in joint system optimization in terms of energy consumption, bit rate (or compression efficiency), and end-to-end source distortion. Such a joint design framework is conceptually similar to joint source-channel coding, which is a very classical research topic in communication theory and has been extensively studied. We note that the problem being addressed in this work fundamentally differs from classical joint source-channel coding since classical joint source-channel coding aims to obtain an optimal R-D (bit rate and distortion) tradeoff space [1]–[6], while we are interested in optimal E-R-D (energy, bit rate, and distortion) tradeoff space by explicitly incorporating system energy consumption. Few prior work merged energy consumption and R-D analysis together in wireless video systems. In [7], [8], dynamic voltage scaling (DVS) and complexity-scalable hardware implementation of video compression were modeled to explore E-R-D design space in a wireless video sensor network. Marijan *et al.* [9] proposed an E-R-D model for image sensor subsystem and incorporates it in power allocation for wireless video sensors. In this work flash memory channel is explicitly configured (or channelized) in terms of memory write and error correction coding (ECC) energy consumption versus raw bit error rate tradeoff, and channelization is an integral part of the joint design, while classical joint source-channel coding assumes an uncontrollable communication channel.

By explicitly incorporating system energy consumption as an optimization metric, we should be able to gracefully explore

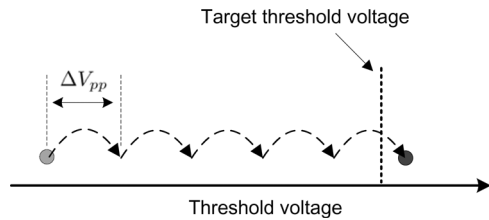


Fig. 1. Threshold voltage shift process under incremental step pulse programming. The threshold voltage is boosted by ΔV_{pp} in each program-and-verify cycle and becomes higher than target value after five cycles.

the design space of all the components in terms of energy consumption. We can combine channel coding and flash memory as a storage subsystem, and jointly adjust the flash memory channelization and error correction capability of channel coding to gracefully explore the storage subsystem energy consumption. However, with the focus on video coding in this work, it is not immediately clear how to explore the video coding design space by considering energy consumption. In this work, we first discuss how we can dynamically adjust the energy consumption versus bit rate tradeoff in video encoding through configuring interframe versus intraframe prediction. We propose a strategy that selectively switches prediction mode of macroblocks during run-time. Then, we further elaborate on obtaining optimal E-R-D tradeoff space for two different scenarios on whether errors are allowed in data storage. We further carried out simulations to demonstrate the proposed joint design framework. Using Synopsys tool set and 65 nm CMOS standard cell and SRAM libraries, we performed ASIC (application specific integrated circuit) design to estimate video coding and channel coding energy consumption. Based on open literature in device community, we developed a NAND flash memory energy consumption estimation tool. Based on these energy consumption estimation capabilities, we use representative video sequences [10] to quantitatively demonstrate the achievable optimal E-R-D tradeoff space under the two different scenarios (i.e., error-free and error-prone flash memory channel), which makes it possible to realize overall system optimization subject to run-time constraints on energy consumption, bit rate, and/or end-to-end distortion.

II. NAND FLASH MEMORY: BASICS AND MODELING

Each NAND flash memory cell is a floating gate transistor whose threshold voltage can be programmed by injecting certain amount of charges into the floating gate. To achieve tight threshold voltage distribution, incremental step pulse programming (ISPP) technique is widely employed [11], [12], i.e., memory cells are recursively programmed using a program-and-verify approach with a stair case program voltage V_{pp} . Let ΔV_{pp} denote the program voltage increment, memory cell threshold voltage can be boosted by up to ΔV_{pp} during each program-and-verify cycle, as illustrated in Fig. 1.

There is a tradeoff between memory cell raw storage reliability and programming energy consumption: If we reduce the program voltage step increment ΔV_{pp} , we can tighten each threshold voltage distribution window, leading to a larger storage noise margin between two adjacent storage levels and hence higher raw storage reliability. Meanwhile, with a smaller

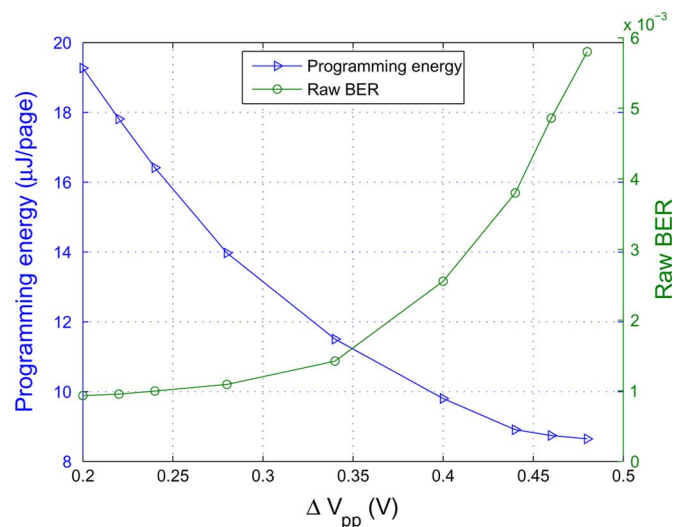


Fig. 2. Simulation results that reveals the impact of program voltage increment ΔV_{pp} on the memory cell raw BER versus programming energy consumption tradeoff.

ΔV_{pp} , we have to carry out more program-and-verify cycles, leading to more bit-line charging/discharging and hence higher energy consumption. Therefore, there is a storage reliability versus memory write energy consumption tradeoff that can be readily adjusted by a single parameter ΔV_{pp} . To enable quantitative evaluations, we use the NAND flash memory device model presented in [13] to estimate memory cell storage raw bit error rate (BER) under different ΔV_{pp} . We enhanced the NAND flash memory energy model presented in [14] in order to quantitatively reveal the impact of ΔV_{pp} on programming energy consumption. In multi-bit per cell NAND flash memory, each verify phase consecutively examines the threshold voltage of all the memory cells against several reference voltages. As a result, bit-lines are charged and discharged several times during each verify phase [15]. All these factors have been incorporated into this flash memory energy consumption model, and the energy estimation results have been verified against those presented in [14].

Based upon the flash memory device model and memory energy consumption model, we carried out simulations to quantitatively demonstrate the memory cell raw BER versus programming energy consumption tradeoff that can be configured by the program step voltage ΔV_{pp} . As shown in Fig. 2, when we increase ΔV_{pp} , programming energy reduces linearly and memory cell raw BER increases exponentially.

III. JOINT CODING AND CHANNELIZATION FOR SYSTEM OPTIMIZATION

As flash memory technology scales down, NAND flash memory is subject to increasingly worse raw storage reliability and hence demands channel coding [16]. We are interested in the optimization of the entire datapath, as illustrated in Fig. 3, subject to run-time constraints of power source, available storage capacity, and/or end-to-end distortion. Different from conventional communication system, the characteristics of NAND flash memory channel can be configured, e.g., the adjustable tradeoff between memory cell raw BER and programming energy consumption. This channel configuration

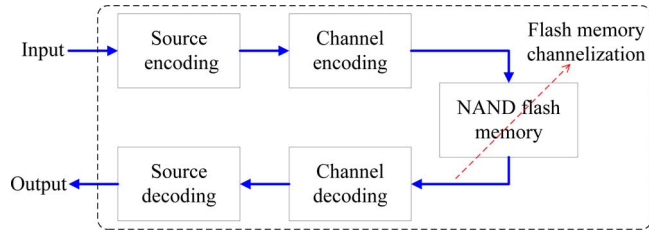


Fig. 3. Illustration of an embedded system with video processing and flash memory storage, where the channelization is enabled by tuning NAND flash memory operational parameter ΔV_{pp} .

process is called *memory channelization*. Although we can reduce NAND flash memory energy consumption by increasing program increment ΔV_{pp} , it meanwhile results in higher memory raw BER and demands stronger channel coding, which leads to higher channel coding energy consumption. Similarly, if we reduce source coding energy consumption by intentionally degrading source compression efficiency, more energy will be consumed by the channel coding and flash memory. This clearly suggests that we should *jointly* consider source coding, channel coding, and flash memory channelization.

In this work, we are particularly interested in video processing in embedded systems. Once a video stream has been compressed (or encoded) and stored in NAND flash memory, the video may be decoded (and displayed) several times. Hence, we model the energy consumption of the entire datapath as

$$E = E_{s_enc} + E_{c_enc} + E_w + N_P \cdot (E_r + E_{c_dec} + E_{s_dec}) \quad (1)$$

where E_{s_enc} and E_{s_dec} (E_{c_enc} and E_{c_dec}) denote the energy consumption of source (channel) encoding and decoding, E_w and E_r denote the energy consumption of flash memory programming and read, and $N_P \geq 0$ is the number of video playback. Let D denote the end-to-end video distortion consisting of distortions induced by both video encoding quantization errors and channel decoding failures. Let R denote the video coding bit rate in terms of bit per pixel. The three system performance metrics (i.e., energy consumption E , bit rate R , and end-to-end distortion D) are correlated. The objective of joint source-channel coding and channelization is to obtain the optimal E-R-D tradeoff space, as illustrated in Fig. 4, which reveals the minimum possible value of one metric for any combination of the other two metrics.

In order to develop the optimal E-R-D tradeoff space, we should be able to explore the design space of all the components in the system by explicitly taking into account energy consumption, bit rate, and distortion. We can combine channel coding and flash memory as a storage subsystem, and jointly adjust the flash memory program increment ΔV_{pp} and error correction capability of channel coding to gracefully explore the storage subsystem design space. However, it is not immediately clear how to address the video coding design space tradeoff in an energy-centric manner. In the following, we first discuss how we can dynamically adjust the tradeoff space in video encoding in Section IV, then further elaborate on the joint coding and channelization strategy in Section V.

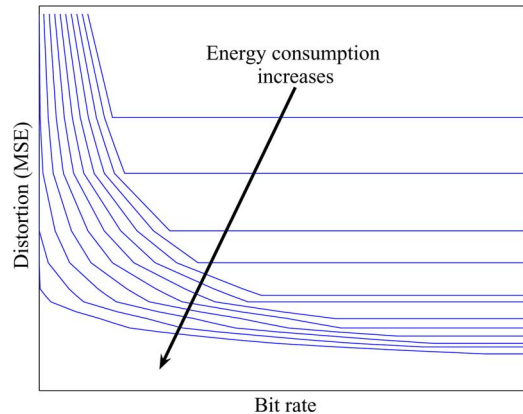


Fig. 4. Illustration of optimal E-R-D tradeoff space. The curves in the plot from top to bottom represent energy consumption from low to high.

IV. EXPLORING E-R-D TRADEOFF IN VIDEO ENCODING

Video encoding aims to remove both spatial and temporal redundancy in video sequences. In current design practice, input video frames are partitioned into macroblocks (MBs), and the encoder processes each frame in units of MBs and constructs a prediction for each MB based on the previously encoded data. A MB predicted from one or more MBs in previously encoded frames is referred as an interframe predicted MB, while a MB predicted from MBs in current frame is referred as an intraframe predicted MB. The residue is then transformed, quantized, and entropy coded.

Interframe prediction and intraframe prediction tend to result in largely different tradeoff between energy consumption and bit rate: Due to the abundant temporal redundancy in video sequences, interframe prediction can achieve (much) higher compression efficiency than intraframe prediction. Inter-frame prediction is realized by motion estimation that carries out exhaustive or non-exhaustive search within a sufficiently large region in previously encoded frame(s). As the most demanding function in video encoding, motion estimation consumes a significant amount of computation and memory access energy consumption. In comparison, intraframe prediction is carried out by searching a very small region around the present MB in the same frame, leading to much less computation and memory access energy consumption. Therefore, we can dynamically adjust the energy consumption versus bit rate tradeoff by simply tuning the interframe versus intraframe prediction ratio.

We first propose an analytical model to estimate the R-D (rate-distortion) performance with respect to the rate of switching interpredicted MBs to intrapredicted MBs, i.e., intrarefresh rate denoted by β . Let us assume the transform coefficients have a Gaussian distribution with zero mean:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2}, \quad -\infty < x < \infty. \quad (2)$$

We employ the square error distortion

$$D_s(x, \tilde{x}) = (x - \tilde{x})^2 \quad (3)$$

where \tilde{x} is quantized value of x . According to Shannon's source coding theorem [17], given a source coding distortion D_s , the minimum rate needed to represent a symbol is

$$R(D_s) = \frac{1}{2} \log_2 \frac{\sigma^2}{D_s}, \quad 0 \leq D_s \leq \sigma^2. \quad (4)$$

For the same image data, transform coefficients after intra prediction have a larger variance than inter prediction, i.e., $\sigma_P^2 < \sigma_I^2$. Denote R-D functions as $R_P(D_s)$ and $R_I(D_s)$ for inter and intra prediction, respectively. We optimize the R-D tradeoff on the frame basis and denote the distortion of a frame with inter prediction and intra prediction as D_s^P and D_s^I , respectively. Then the overall frame distortion is

$$D_s = (1 - \beta)D_s^P + \beta D_s^I, \quad (5)$$

and we have the bit rate of the frame

$$R = (1 - \beta)R_P(D_s^P) + \beta R_I(D_s^I). \quad (6)$$

Given the source coding distortion constraint \mathcal{D}_s , we would like to minimize the rate to obtain the optimal R-D tradeoff space, i.e.,

$$\min R, \text{ s.t. } D_s < \mathcal{D}_s. \quad (7)$$

In the following, we show that the rate, which results in the same distortion \mathcal{D}_s for both interpredicted and intrapredicted MBs, gives the optimal solution to (7). The Lagrangian formulation of the minimization problem is given by

$$\begin{aligned} \min J &= D_s + \lambda R \\ &= (1 - \beta)D_s^P + \beta D_s^I \\ &\quad + \lambda [(1 - \beta)R_P(D_s^P) + \beta R_I(D_s^I)]. \end{aligned} \quad (8)$$

By taking the partial derivative of D_s^P and D_s^I respectively, we have

$$\begin{aligned} \frac{\partial J}{\partial D_s^P} &= (1 - \beta) + \lambda(1 - \beta) \frac{\partial R_P}{\partial D_s^P} = 0, \\ \frac{\partial J}{\partial D_s^I} &= \beta + \lambda\beta \frac{\partial R_I}{\partial D_s^I} = 0. \end{aligned}$$

Therefore, we have that

$$\lambda = -\frac{\partial D_s^P}{\partial R_P} = -\frac{\partial D_s^I}{\partial R_I} \quad (9)$$

provides the optimal solution to (7). In other words, in order to minimize bit rate subject to a distortion constraint, we should carefully set the quantization parameters for both modes so that both inter and intra prediction have the same maximum allowable distortion. Total rate for the given distortion D_s can be calculated as

$$\begin{aligned} R &= \frac{1}{2}(1 - \beta) \log_2 \frac{\sigma_P^2}{D_s} + \frac{1}{2}\beta \log_2 \frac{\sigma_I^2}{D_s} \\ &= \frac{1}{2} \log_2 \frac{(\sigma_P^{1-\beta} \cdot \sigma_I^\beta)^2}{D_s}. \end{aligned} \quad (10)$$

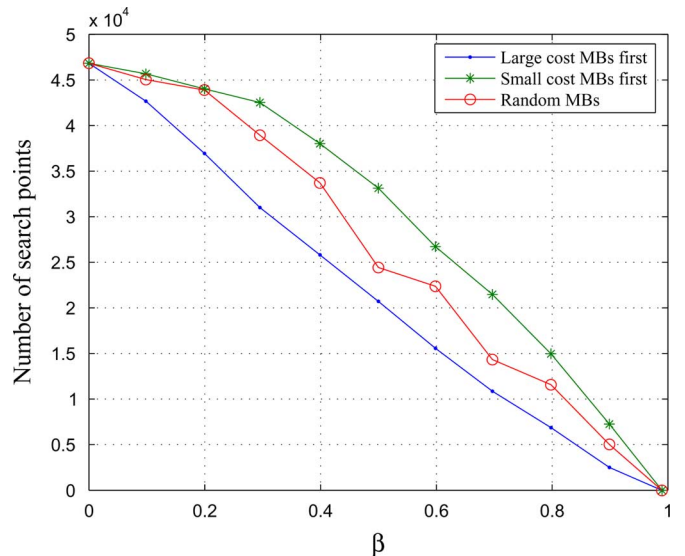


Fig. 5. Computation complexity of motion estimation decreases as β increases. Switching MBs with large estimation costs reduces the complexity substantially.

Let E_{enc}^P and E_{enc}^I denote the average energy consumption of interframe and intraframe prediction, respectively. Given the parameter β , we can estimate the total encoding energy consumption as

$$E_{s_enc} = (1 - \beta) \cdot E_{\text{enc}}^P + \beta \cdot E_{\text{enc}}^I. \quad (11)$$

Clearly, we can explore the video encoding E-R-D tradeoff space by configuring the ratio between interframe and intraframe prediction (i.e., the parameter β). If fast motion estimation algorithms are used, interframe prediction for different MBs will incur different motion estimation computational complexity and hence energy consumption. So we should apply intraframe prediction to those MBs that demand high motion estimation energy consumption. If the motion vector of a MB corresponds to a relatively large sum of absolute difference (SAD), its motion estimation tends to search more possible candidate points. Hence, we can simply use the SAD associated with the motion vector as the motion estimation cost metric. Fig. 5 demonstrates the motion estimation complexity reduction with different MBs selection schemes. The complexity is the number of search points per frame when the motion estimation uses the hexagon search [18]. In the large cost MBs first (LCMF) scheme, the motion estimation costs of all MBs are sorted in a descending order, the first β portion of all the MBs are processed by intraframe prediction and the rest are processed by interframe prediction (i.e., motion estimation); in the small cost MBs first (SCMF) scheme, the last β portion of all the MBs are processed by intraframe prediction; in the random scheme, a random β portion of all the MBs are processed by intraframe prediction.

However, it is not practical to know the motion estimation costs of all the MBs in a frame during run time unless they have been processed by interframe prediction. In practice we cannot select the MBs strictly based on their motion estimation cost, and instead we propose the following simple solution: In every group of pictures (GOP), the first frame is processed by

intraframe prediction, and the second frame are processed by interframe prediction. This provides the initial motion estimation cost statistics for all the MBs, based on which the modes of all the MBs in the successive frames are decided, and the resulting costs are used to update the motion estimation cost statistics. This strategy was used in our quantitative evaluations presented in Section VI.

V. JOINT SOURCE-CHANNEL CODING AND FLASH MEMORY CHANNELIZATION

In this section, we discuss the joint source-channel coding and flash memory channelization that aims to obtain the optimal E-R-D tradeoff space for the entire system. The source coding energy E_s consists of encoding energy given by (11), and decoding energy given by

$$E_{s_dec} = N_P \cdot [(1 - \beta) \cdot E_{dec}^P + \beta \cdot E_{dec}^I] \quad (12)$$

where E_{dec}^P and E_{dec}^I are energy of inter and intra decoding, respectively, and N_P is the number of video playbacks. Since the exact value of N_P is unknown *a priori*, the system can only rely on an estimation from either the operating systems (based on prior user's activity patterns) or directly from the user input. Combining (11) and (12), we can express the total video coding system energy consumption as

$$E_s = \Psi_s - k_s \cdot \beta \quad (13)$$

where

$$\begin{cases} \Psi_s = E_{enc}^P + N_P \cdot E_{dec}^P, \\ k_s = (E_{enc}^P - E_{enc}^I) + N_P \cdot (E_{dec}^P - E_{dec}^I). \end{cases} \quad (14)$$

Essentially, Ψ_s represents the video coding system energy consumption if all the MBs are coded in inter mode, and $k_s \cdot \beta$ represents the energy saving of inter-to-intra mode switching. Once the video sequence and computing platform are given, the values of E_{enc}^P , E_{enc}^I , E_{dec}^P and E_{dec}^I are fixed. Hence, the video coding system energy consumption E_s is the function of β , i.e., $E_s(\beta)$.

Let $E_{c_enc}^b$ and $E_{c_dec}^b$ denote the average energy consumption for ECC encoding and decoding one bit, respectively, and $E_{c_w}^b$ and $E_{c_r}^b$ denote the average energy consumption for writing/reading one bit to/from flash memory, respectively. Let r_P and r_I represent the length of compressed video bitstream if all the MBs are processed by interframe and intraframe prediction, respectively. Let η denote the ECC code rate. We can estimate the total channel energy consumption (including both ECC coding and flash memory access) as

$$E_c = E_b \cdot [(1 - \beta) \cdot r_P + \beta \cdot r_I] / \eta \quad (15)$$

where

$$E_b = (E_{c_enc}^b + E_w^b) + N_P \cdot (E_r^b + E_{c_dec}^b). \quad (16)$$

We can further rewrite (15) as

$$E_c = \Psi_c + k_c \cdot \beta \quad (17)$$

where

$$\begin{cases} \Psi_c = \frac{E_b}{\eta} \cdot r_P, \\ k_c = \frac{E_b}{\eta} \cdot (r_I - r_P). \end{cases} \quad (18)$$

Essentially, Ψ_c represents the channel energy consumption if the bitstream is generated when interframe prediction is used to all the MBs, and $k_c \cdot \beta$ represents the energy cost because of the longer bitstream length caused by inter-to-intra prediction switching. Once we choose the ECC being used (e.g., BCH code or RS code), we assume that $E_{c_enc}^b$ and $E_{c_dec}^b$ are fixed. As discussed in Section II, the parameter ΔV_{pp} determines the tradeoff between flash memory programming energy consumption $E_{c_w}^b$ and memory raw bit error rate. In addition, the allowable channel distortion D_c sets a constraint on the allowable memory raw bit error rate, and memory raw bit error rate determines ECC code rate η . Therefore, D_c and ΔV_{pp} together determine E_b and η . The parameters r_P and r_I are proportional to the R-D functions $R_P(D_s)$ and $R_I(D_s)$, and hence are also functions of the video coding distortion D_s . Therefore, the channel energy consumption E_c is a function of ΔV_{pp} , D_c , D_s , and β , i.e., $E_c(\Delta V_{pp}, D_c, D_s, \beta)$.

As proven in [6], the distortions caused by video coding and channel are uncorrelated, and the total end-to-end distortion D is simply the summation of video coding distortion and channel distortion, i.e., $D = D_s + D_c$. Based upon the above discussions, we have

$$\begin{cases} E = E_s(\beta) + E_c(\Delta V_{pp}, D_c, D_s, \beta) \\ D = D_s + D_c \\ R = (1 - \beta)R_P(D_s) + \beta R_I(D_s) \end{cases} \quad (19)$$

which can be used to jointly explore the space of the parameters β , D_s , D_c , and ΔV_{pp} to find the optimal E-R-D tradeoff space. In this section, we will discuss two possible scenarios: i) Channel does not contribute to the end-to-end distortion, i.e., ECC coding is strong enough to ensure a completely error-free channel and hence $D_c = 0$ and ii) Motivated by extensive studies on unequal error protection and error concealment for video bitstream transmission (e.g., see [19]–[26]), we also consider the scenario where distortion comes from both video encoding and channel, i.e., $D_c > 0$.

A. Scenario I: Error-Free Channel

Since the ECC can ensure a completely error-free flash-based data storage, we have $D_c = 0$ and $D = D_s$, i.e., the end-to-end distortion is only incurred by video encoding. As a result, both E_b defined in (16) and ECC code rate η are only dependent on the parameter ΔV_{pp} . Define

$$E_c^f = \frac{E_b}{\eta} \quad \text{and} \quad r_s = r_P + (r_I - r_P) \cdot \beta, \quad (20)$$

and we have E_c^f is a function of ΔV_{pp} and r_s is a function of D and β . Therefore, we can rewrite (19) as

$$\begin{cases} E = E_s(\beta) + E_c^f(\Delta V_{pp}) \cdot r_s(D, \beta) \\ R = (1 - \beta)R_P(D) + \beta R_I(D). \end{cases} \quad (21)$$

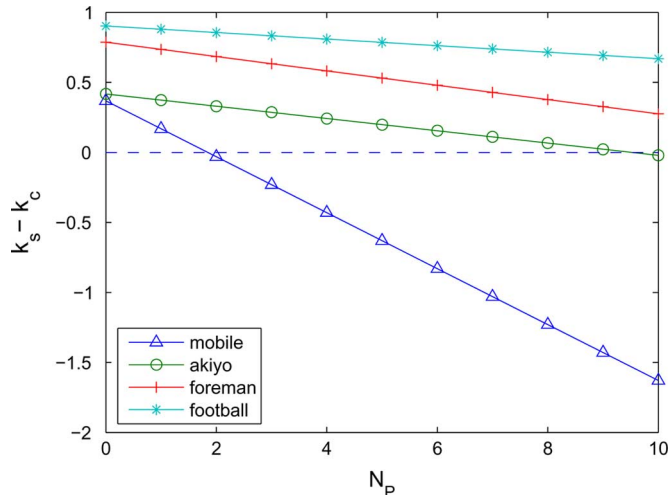


Fig. 6. Calculated $k(D) = k_s - k_c$ with a given distortion when we vary the video replay number N_P .

Since $E_c^f(\Delta V_{pp})$ is independent on other parameters, we can search for the optimal E-R-D tradeoff space in two separate steps:

- 1) We first search the space of the parameter ΔV_{pp} in order to minimize E_c^f ;
- 2) Based upon the minimized E_c^f , we then search the space of the parameter β for the optimal E-R-D tradeoff.

The first step can be simply accomplished through exhaustive search. For each given ΔV_{pp} , we can use the flash memory energy consumption model to estimate the corresponding memory write energy E_w^b and then estimate the value of E_b . Meanwhile, we can estimate the corresponding memory raw bit error rate, which will further determine ECC code rate η . In this way, for each given ΔV_{pp} , we can obtain the corresponding E_c^f . By exhaustively exploring the practically allowable region of the parameter ΔV_{pp} , we can find the minimal possible E_c^f . Once we fix E_c^f , we can rewrite (21) as

$$\begin{cases} E = \Psi(D) - k(D) \cdot \beta, \\ R = (1 - \beta)R_P(D) + \beta R_I(D) \end{cases} \quad (22)$$

where $\Psi(D) = \Psi_s + \Psi_c$ and $k(D) = k_s - k_c$. We note that Ψ_s and k_s are defined in (14), and Ψ_c and k_c are defined in (18). It essentially reveals the relation among system energy consumption E , video distortion D , and bit rate R , which can be directly used to obtain the optimal E-R-D tradeoff space. Furthermore, we note that $\beta \in [0, 1]$, and according to (6), for a given coding distortion, a smaller β results in a lower bit rate. Given a distortion D , if $k(D) \leq 0$, i.e., video encoding energy saving due to inter-to-intra prediction switch is not sufficient to offset channel coding and memory energy overhead due to increased bit rate, we should set $\beta = 0$, which can minimize both energy consumption E and bit rate R . On the other hand, if $k(D) \geq 0$, there is a tradeoff between energy consumption E and bit rate R , i.e., a smaller β results in lower bit rate R but higher energy consumption E . Fig. 6 shows the calculated $k(D)$ with a given distortion for several different video sequences when we vary the video playback number N_P . The video sequences with large $(r_I - r_P)$ are more likely to have a negative $k(D)$, e.g., the *Mobile* sequence as shown in Fig. 6.

B. Scenario II: Error-Prone Channel

In this subsection, we consider the scenario that channel also incurs distortion (i.e., $D_c > 0$) due to weak ECC code. Motivated by the fact that human beings are perceptually insensitive to minor errors in video frames and different portions in video data stream have largely different importance regarding the quality of reconstructed video frames, unequal error protection for video stream transmission has been well studied (e.g., see [19]–[23]) and has been leveraged in prior research on joint source-channel coding. Following prior work, we apply a strong ECC to frame header, prediction mode, and motion vector data to ensure their error-free storage, and apply a weak ECC to the remainder texture data at the cost of relatively high error rate. When a corrupted codeword cannot be properly decoded by the weak ECC, it will be discarded and the video decoder will skip all the bits being contained in this codeword. During video decoding, if an inter coded MB is skipped, the texture information is lost but the mode and motion vector information are retained. Hence, we can simply substitute this lost MB with the motion compensated block from the reference frame. If an intra coded MB is skipped, it can be simply substituted by the MB at the same location in the previous frame.

Inter-frame prediction offers increased coding efficiency over intraframe prediction but is susceptible to error propagation, which can be alleviated by performing intrarefresh [6], [27], [28]. Therefore, in addition to realizing the encoder complexity scalability, inter-to-intra switching can also trade compression efficiency for error resilience. As a result, the channel distortion depends on both pixel error rate p_e due to error correction failure of weak ECC and intrarefresh rate β , i.e., channel distortion D_c is a function of p_e and β . Moreover, ΔV_{pp} determines the raw flash memory bit error rate, which together with p_e determines ECC code rate η . Hence E_c^f as defined in (20) is a function of ΔV_{pp} and p_e . Therefore, we can obtain the optimal E-R-D tradeoff space based upon

$$\begin{cases} E = E_s(\beta) + E_c^f(\Delta V_{pp}, p_e) \cdot r_s(D_s, \beta) \\ D = D_s + D_c(p_e, \beta) \\ R = (1 - \beta)R_P(D_s) + \beta R_I(D_s) \end{cases} \quad (23)$$

To reduce the computational complexity, we can derive the optimal E-R-D trade space in two separate steps:

- 1) Given a pixel error rate p_e , we first search the space of the parameter ΔV_{pp} in order to minimize E_c^f .
- 2) Based upon the minimized E_c^f , we then search the space of the parameter β for overall system optimization.

Similarly, the first step can be simply accomplished through exhaustive search. Nevertheless, the second step becomes much more complicated than the scenario of error-free channel. Unequal error protection reduces the total flash memory access energy at the expense of nonzero channel distortion. Part of the flash memory access energy saving can be allocated to enable video encoder carry out more interframe prediction, which can reduce source distortion without rate penalty. Such correlations among channel distortion, source distortion, energy consumption, and bit rate make β have a much more complicated impact on the E-R-D tradeoff space.

It can be proven (please refer to the Appendix for the proof) that the asymptotic average channel distortion can be expressed as

$$\begin{aligned} \bar{D}_c = & \frac{\beta p_e}{1 - a + (a - p_e)\beta} E[\delta_{\text{rec}}^2(n)] \\ & + \frac{(1 - \beta)p_e}{1 - a + (a - p_e)\beta} E[\sigma_r^2(n)] \end{aligned} \quad (24)$$

where a is a constant describing the motion randomness of the video sequence, as defined in [6]. If $f(n, i)$, $\hat{f}(n, i)$ and $g(n, i)$ are used to denote the original value, reconstructed value without and with channel error of the i th pixel in the n th frame, respectively, then

$$a = \frac{E\{[g(n-1, j) - \hat{f}(n-1, j)]^2\}}{D_c(n-1)}. \quad (25)$$

$\sigma_r^2(n)$ and $\delta_{\text{rec}}^2(n)$ are defined as

$$\begin{aligned} \sigma_r^2(n) &= E\{[\hat{f}(n-1, j) - \hat{f}(n, i)]^2\} \\ \delta_{\text{rec}}^2(n) &= E\{[\hat{f}(n, i) - \hat{f}(n-1, i)]^2\}. \end{aligned} \quad (26)$$

It should be noted that $\sigma_r^2(n)$ is the variance of residue after motion prediction in frame n . While $\delta_{\text{rec}}^2(n)$ represents the MSE between reconstructed frames n and $n-1$, which effectively is the variance of residue if all the motion vectors are forced to point to collocated MBs. Hence, $E[\sigma_r^2(n)]$ is essentially upper bounded by $E[\delta_{\text{rec}}^2(n)]$. As we increase β , the first term and second term in (34) will increase and decrease, respectively. In addition, asymptotically, the average channel distortion caused by channel error is proportional to the frame differences and residue variance. For a specific video frame, the larger the quantization step is, the smaller the residue variance is. Fig. 7 shows that the asymptotic average channel distortion may increase or decrease with β , depending on quantized residue variance. For video sequences with relatively small $E[\sigma_r^2(n)]$, decreasing β leads to smaller D_c and thus relaxes D_s , and accordingly bit rate can be reduced. For video sequences with relatively large $E[\sigma_r^2(n)]$, decreasing β results in larger D_c and thus demands a lower D_s , which will further increase the bit rate. Due to its strong dependence on video sequence characteristics, the optimal β can only be found by extensive simulations over representative video sequences.

VI. CASE STUDIES

This section presents case studies to evaluate the above proposed joint source-channel and channelization framework for obtaining the optimal E-R-D tradeoff space.

A. Flash Memory Channelization

Table I lists the NAND flash memory configurations being used in this study. We consider two programming schemes: 1 bit/cell and 2 bits/cell. Programming 1 bit information into one cell can be very fast and energy efficient, since a larger ΔV_{pp} can be used compared with 2 bits/cell. Table II lists NAND flash memory programming energy and raw BER under different ΔV_{pp} for both 1 bit/cell and 2 bits/cell programming schemes. For each programming scheme, by increasing ΔV_{pp} ,

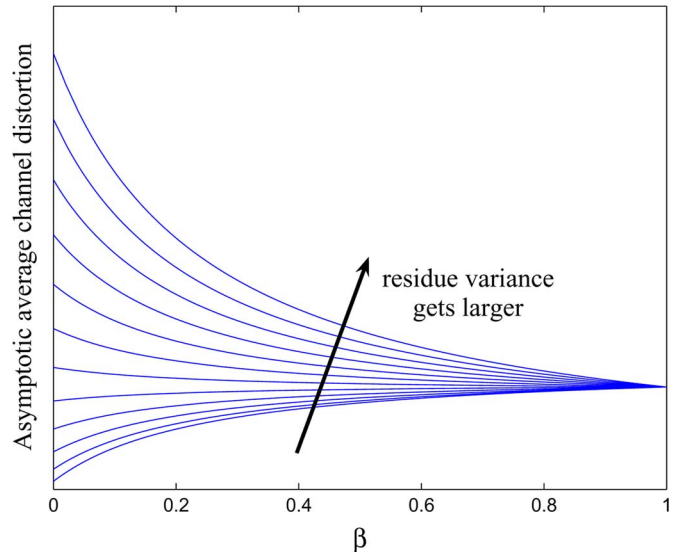


Fig. 7. The asymptotic average channel distortion may increase or decrease with β , based on residue variance values.

TABLE I
NAND FLASH MEMORY PARAMETERS USED IN SIMULATIONS

Technology Node (nm)	45
Capacity (Gbit)	8
Bits/Cell	1 or 2
Page size (Byte)	4096
Pages/Block	64
Blocks/Plane	2048

TABLE II
FLASH PROGRAMMING ENERGY AND RAW BER WITH DIFFERENT ΔV_{pp}

1 bit/cell		2 bits/cell		Raw BER
ΔV_{pp} (V)	Write energy ($\mu\text{J}/\text{page}$)	ΔV_{pp} (V)	Write energy ($\mu\text{J}/\text{page}$)	
1.10	9.46	0.20	19.27	9.30×10^{-4}
1.21	8.68	0.28	13.97	1.09×10^{-3}
1.50	7.24	0.34	11.50	1.42×10^{-3}
2.16	6.46	0.40	9.80	2.55×10^{-3}
2.52	5.32	0.44	8.91	3.80×10^{-3}
2.65	4.72	0.46	8.74	4.86×10^{-3}

we can reduce flash memory programming energy at the expense of increased raw BER. The read energy is $0.385 \mu\text{J}$ per page, which is independent on ΔV_{pp} .

Following the current design practice, we use binary BCH codes as ECC for NAND flash memory, whose construction and encoding/decoding are based on binary Galois fields [29]. A binary Galois field with degree of m is represented as $\text{GF}(2^m)$. For any $m \geq 3$ and $t \leq 2^{m-1}$, there exists a primitive binary BCH code over $\text{GF}(2^m)$, which has the code length $n = 2^m - 1$ and information bit length $k \geq 2^m - m \cdot t$ and can correct up to t errors. Under different raw BER, Table III lists the BCH code structure and decoding energy consumption with a target page error rate of 10^{-15} (corresponding to error-free storage) and 1% (corresponding to error-prone storage). The BCH decoding energy consumption is obtained by carrying out ASIC design using 65 nm CMOS standard cell and SRAM libraries, where Synopsys tools are used throughout the design hierarchy

TABLE III
 BCH CODE PARAMETERS AND DECODING ENERGY CONSUMPTION FOR DIFFERENT RAW BER

Raw BER	page error rate= 10^{-15}				page error rate= 1%			
	t	Code length n	Code rate	BCH decoding energy ($\mu\text{J}/\text{page}$)	t	Code length n	Code rate	BCH decoding energy ($\mu\text{J}/\text{page}$)
9.30×10^{-4}	84	34112	96.06%	0.121	42	33440	97.99%	0.062
1.09×10^{-3}	93	34256	95.66%	0.155	48	33536	97.71%	0.089
1.42×10^{-3}	110	34528	94.90%	0.226	61	33744	97.11%	0.102
2.55×10^{-3}	166	35424	92.50%	1.35	103	34416	95.21%	0.213
3.80×10^{-3}	223	36336	90.18%	3.86	148	35136	93.26%	0.979
4.86×10^{-3}	269	37073	88.39%	8.16	185	35728	91.72%	2.13

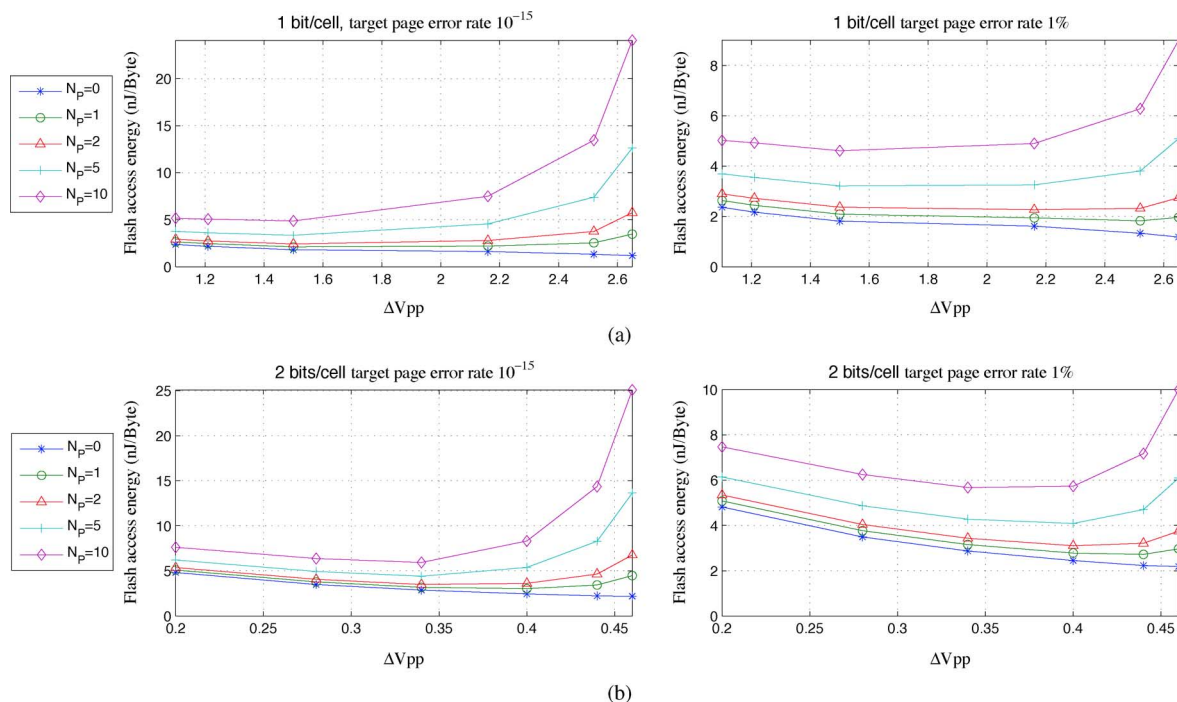


Fig. 8. Total flash access energy consumption (including flash memory write/read energy consumption and ECC decoding energy consumption) with (a) 1 bit/cell and (b) 2 bits/cell programming schemes.

down to place and route. we designed the BCH code decoder architecture based upon the well-known Berlekamp–Massey decoding algorithm [29]. We set the number of metal layers as 4 in the place and route. Post-layout results verify that the decoders can operate at 400 MHz with the power supply of 1.08 V, and the footprint is 1.49 mm². BCH encoding is very simple (i.e., only collections of shift registers), and consumes less than 5% of decoding energy. Hence, we do not explicitly take it into account in this study.

Fig. 8 shows the total flash memory access energy consumption with respect to ΔV_{pp} . Data are written to the flash memory once, but may be read out multiple times (i.e., $N_P \geq 1$). The total flash memory access energy consumption includes both flash memory write/read energy consumption and ECC decoding energy consumption. The results demonstrate the impact of the parameter ΔV_{pp} . 1 bit/cell programming outperforms 2 bits/cell programming in terms of energy efficiency, however, the capacity occupied by the same amount of data is doubled by storing only one bit per cell.

B. Exploration of Optimal System E-R-D Trade-Off

In this work, we use H.264/AVC video compression standard [30] and carry out video sequences profiling using JVT

JM 15.1 [31] codec implementation. We evaluate the optimal E-R-D tradeoff for sequences in CIF format (352 × 288 and 30 frames per second) [10]. These standard test sequences cover a wide range of video motion complexity. We show the results of *Foreman* and *Football* in this paper, and results of other video sequences are similar based on our simulation. In each group of frames, the first frame only uses intraframe prediction, followed by five frames that may use either interframe or intraframe prediction for each MB. Motion estimation uses hexagon search and search range is ± 16 , i.e., a 48×48 region. When forcing the five frames within each group only use interframe prediction, we estimate that real-time encoding of *Foreman* and *Football* consumes 1.51 mW and 1.88 mW, respectively. The average energy consumption per frame is 50.3 and 62.7 μJ , respectively. When forcing the five frames within each group only use intraframe prediction, we estimate that they consume almost the same power of 0.5 mW, i.e., 16.7 μJ per frame. Decoding power is estimated as 0.125 mW, i.e., 4.2 μJ per frame.

1) *Error-Free Channel*: Targeting at error-free channel, we carry out E-R-D space exploration following the strategy presented in Section V-A. Fig. 9 shows the optimal E-R-D tradeoff space for the four video sequences, where we set $N_P = 1$ (i.e., the video is replayed once). Sufficiently strong BCH codes are

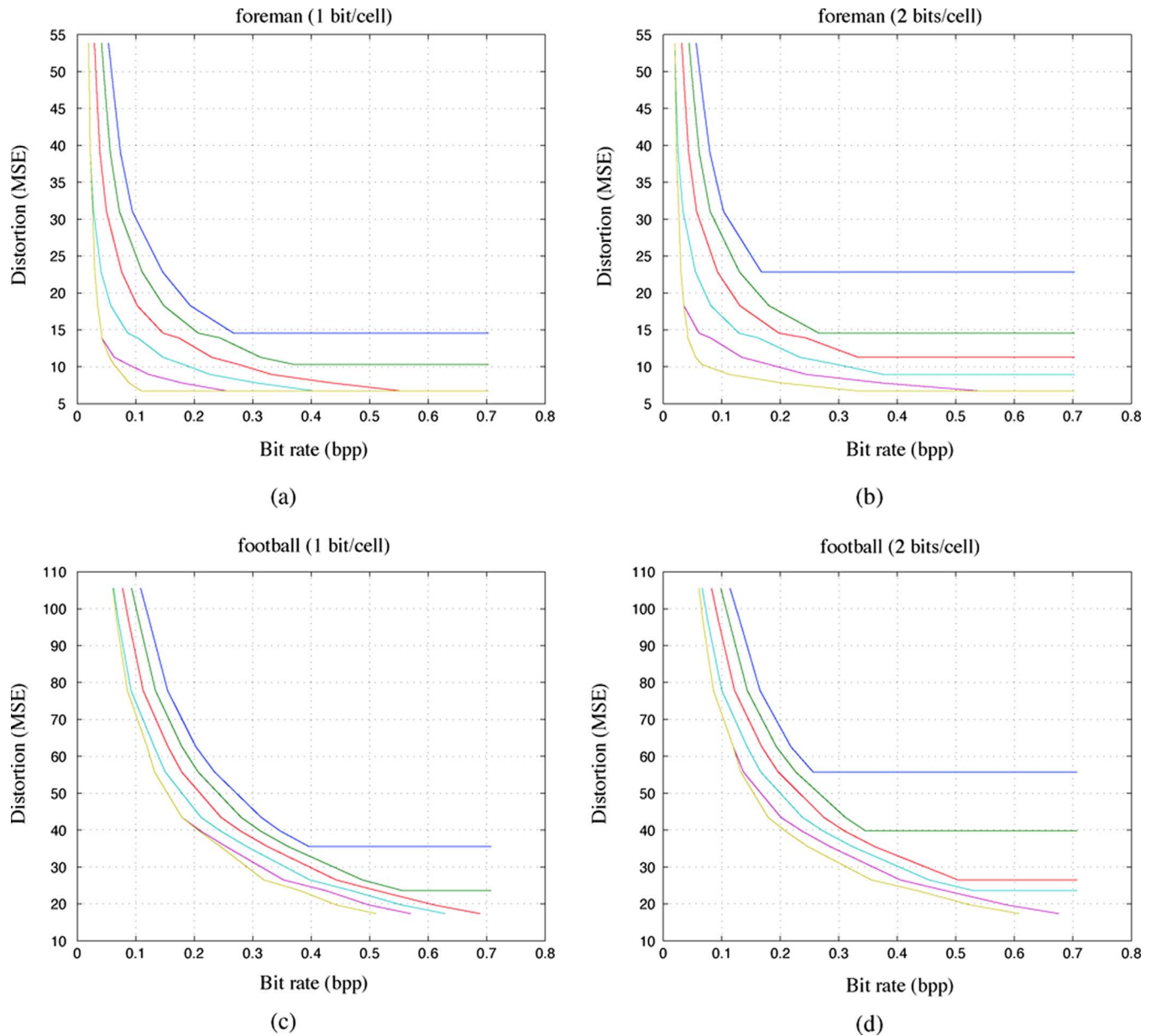


Fig. 9. Estimated optimal E-R-D tradeoff space with error-free channel for (a) *Foreman* with 1 bit/cell, (b) *Foreman* with 2 bits/cell, (c) *Football* with 1 bit/cell, and (d) *Football* with 2 bits/cell. ΔV_{pp} is carefully selected by assuming the bitstream is read once, and strong ECC is used to ensure error-free flash data storage.

used to ensure the page error rate of 10^{-15} for all the data stored in NAND flash memory. The total system energy consumption corresponding to curves from top to bottom are 33, 40, 53, 60, and 66 μJ for each frame, respectively.

When the total energy source is low, the distortion becomes flat because it cannot be pushed down beyond a certain value due to limited energy resource. With a higher energy supply source, the system achieves lower distortion and the curve becomes more steep, which means the video compression efficiency is higher. This is because the encoder has more energy resource and thus more computational capability to eliminate the temporal and spatial redundancy in the input video data. Since fast motion estimation algorithm is used, the sequence *Foreman* with little motion only requires small amount of computation to find the motion vectors, hence consumes less energy for inter encoding. While for *Football* with complex motions, it needs to search more points to find optimal motion vec-

TABLE IV
FRACTION OF RESIDUE DATA IN BITSTREAM WITH
DIFFERENT QUANTIZATION PARAMETERS (QPs)

QP	Foreman	Football
26	87.9%	93.6%
29	82.8%	91.3%
32	75.2%	86.9%
35	70.2%	82.7%
38	64.0%	77.3%
41	58.4%	69.4%

tors. Therefore, improvement of distortion for *Football* is larger than for *Foreman*, if the total energy resource is increased by same amount, as shown in Fig. 9. Since storing bitstream with 1 bit/cell is more energy efficient, it achieves better R-D performance with given energy constraint, compared to 2 bits/cell programming scheme.

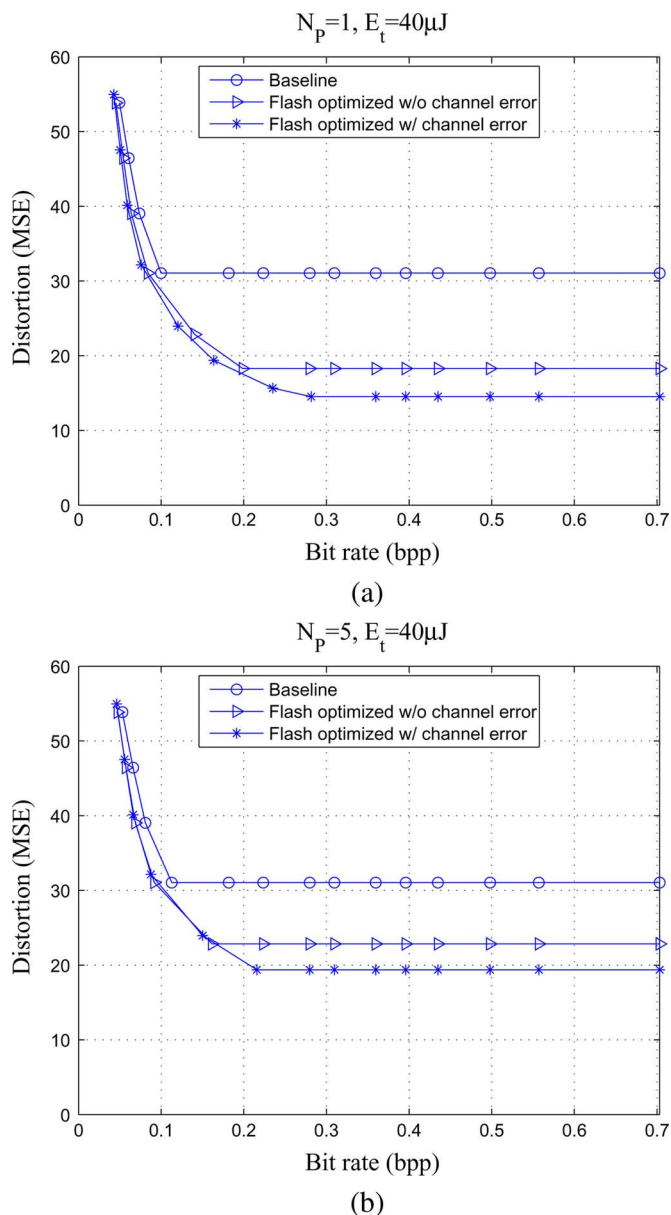


Fig. 10. Rate-distortion tradeoff under three different design scenarios subject to the energy constraint of $40 \mu\text{J}$ per frame with (a) $N_P = 1$ and (b) $N_P = 5$.

2) *Error-Prone Channel*: As pointed out earlier, following the theme of unequal error protection, we categorize the video bit stream into important data being protected by strong ECC and unimportant data being protected by weak ECC. The channel distortion and flash memory access energy consumption largely depend on the fraction of unimportant data in the entire bitstream, which is also impacted by quantization steps. Table IV shows the fraction of unimportant texture information data in bitstream with different quantization steps for the four video sequences. The texture information includes residues of both inter and intra coded MBs.

We carry out optimal E-R-D tradeoff space exploration following the strategy presented in Section V-B. Fig. 10 shows the results for the video sequence *Foreman* with 2 bits/cell flash programming. We consider the scenarios when the stored bitstream is read and played once (i.e., $N_P = 1$) and five times (i.e., $N_P = 5$), respectively, and the overall system energy con-

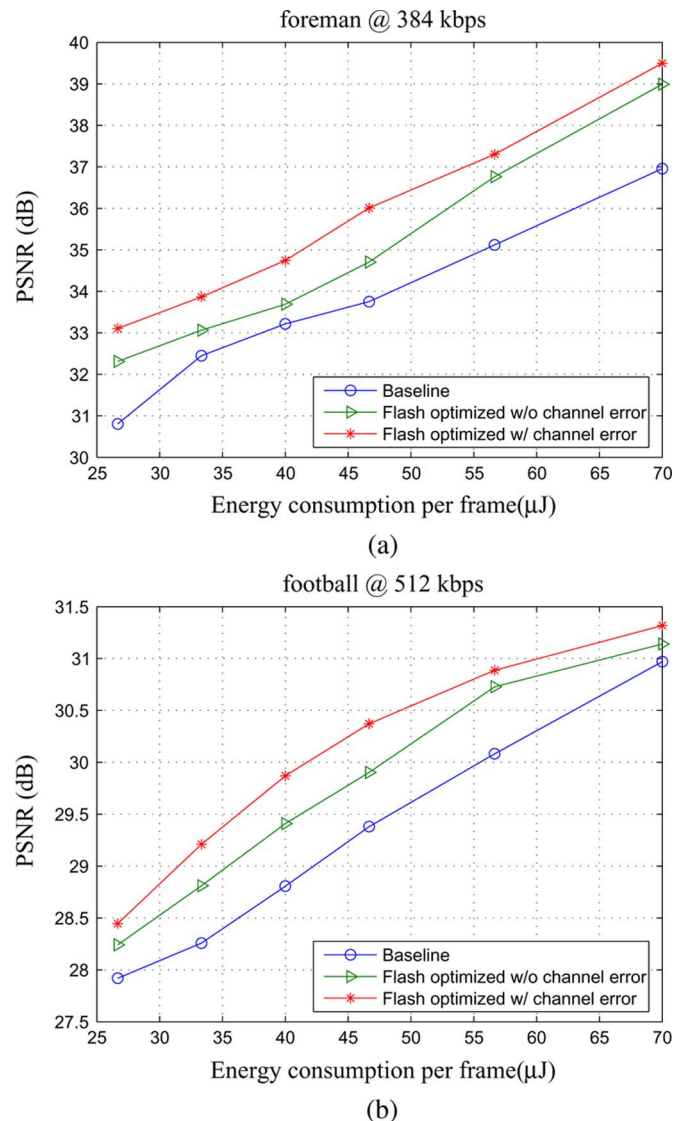


Fig. 11. PSNR improvement versus system energy consumption constraint for (a) *Foreman* (b) *Football*.

straint as $40 \mu\text{J}$ per frame. For the purpose of comparison, we also show the results when assuming an error-free channel. Results show that, by allowing channel distortion, we can achieve a slightly better E-R-D tradeoff space than enforcing an error-free channel. To further demonstrate the effectiveness of the proposed joint coding and channelization approach, we also consider a baseline scenario where we simply fix ΔV_{pp} as 0.2 V and apply a strong ECC to ensure error-free channel.

Fig. 11 shows the PSNR improvement versus system energy consumption constraint for the two video sequences under the three different design scenarios. We encode *Foreman* at a bit rate of 384 kb/s, and *Football* at a bit rate of 512 kb/s, because of their different motion and texture complexity. And 2 bits/cell programming scheme is used. The results further illustrate the potential advantages by allowing channel distortion, and the effectiveness of joint coding and channelization optimization.

VII. CONCLUSION

Targeting at embedded and mobile devices with limited power source and abundant flash memory storage capacity, this

paper presents a joint source-channel coding and flash memory channelization design framework for searching an optimal system E-R-D tradeoff space. Based upon such an E-R-D tradeoff space, embedded and mobile devices can optimally configure its operations subject to run-time power source, storage capacity, and/or source distortion constraints. Focusing on video coding and leveraging inherent programming versus reliability tradeoff of NAND flash memory, this work develops specific joint coding and flash memory channelization design approaches covering both scenarios where flash memory storage can and cannot contribute to end-to-end distortion. We further carry out quantitative evaluations to demonstrate using this joint design approach to obtain optimal system E-R-D tradeoff space.

APPENDIX

For the i th pixel in the n th frame, let $f(n, i)$ be the original value, and $\hat{f}(n, i)$ and $g(n, i)$ be the reconstructed value without and with channel error, respectively. The expected channel distortion for an intra coded frame with a pixel error rate of p_e is

$$\begin{aligned} D_c^I(n) &= E\{[g(n, i) - \hat{f}(n, i)]^2\} \\ &= p_e E\{[g(n-1, i) - \hat{f}(n, i)]^2\} \\ &= p_e E\{[g(n-1, i) - \hat{f}(n-1, i) + \hat{f}(n-1, i) - \hat{f}(n, i)]^2\} \\ &= p_e E\{[g(n-1, i) - \hat{f}(n-1, i)]^2\} \\ &\quad + p_e E\{[\hat{f}(n, i) - \hat{f}(n-1, i)]^2\} \\ &= p_e D_c(n-1) + p_e \delta_{\text{rec}}^2(n). \end{aligned} \quad (27)$$

The fourth identity in (27) is based on the assumption that the frame difference and channel distortion are uncorrelated with each other. If a pixel is predicted in inter mode, its reconstructed value at decoder is $g(n-1, j) + r(n, i)$, where pixel j in frame $n-1$ is the motion prediction of pixel i in frame n , and $r(n, i)$ is the residue. Then channel distortion for an inter coded frame is

$$\begin{aligned} D_c^P(n) &= E\{[g(n, i) - \hat{f}(n, i)]^2\} \\ &= (1-p_e)E\{[g(n-1, j) + r(n, i) - \hat{f}(n, i)]^2\} \\ &\quad + p_e E\{[g(n-1, j) - \hat{f}(n, i)]^2\} \\ &= (1-p_e)E\{[g(n-1, j) - \hat{f}(n-1, j)]^2\} \\ &\quad + p_e E\{[g(n-1, j) - \hat{f}(n-1, j) \\ &\quad + \hat{f}(n-1, j) - \hat{f}(n, i)]^2\} \\ &= (1-p_e)E\{[g(n-1, j) - \hat{f}(n-1, j)]^2\} \\ &\quad + p_e E\{[g(n-1, j) - \hat{f}(n-1, j)]^2\} \\ &\quad + p_e E\{[\hat{f}(n-1, j) - \hat{f}(n, i)]^2\} \\ &= E\{[g(n-1, j) - \hat{f}(n-1, j)]^2\} \\ &\quad + p_e E\{[\hat{f}(n-1, j) - \hat{f}(n, i)]^2\} \\ &= aD_c(n-1) + p_e \sigma_r^2(n). \end{aligned} \quad (28)$$

The overall channel distortion is given by

$$\begin{aligned} D_c(n) &= (1-\beta)D_c^P(n) + \beta D_c^I(n) \\ &= [(1-\beta)a + \beta p_e]D_c(n-1) \\ &\quad + \beta p_e \delta_{\text{rec}}^2(n) + (1-\beta)p_e \sigma_r^2(n) \\ &= \Gamma_1 \cdot D_c(n-1) + \Gamma_2 \cdot \delta_{\text{rec}}^2(n) + \Gamma_3 \cdot \sigma_r^2(n) \end{aligned} \quad (29)$$

where

$$\begin{aligned} \Gamma_1 &= (1-\beta)a + \beta p_e \\ \Gamma_2 &= \beta p_e \\ \Gamma_3 &= (1-\beta)p_e \end{aligned} \quad (30)$$

The distortion can be recursively calculated frame by frame and trace back to distortion of the first frame as

$$D_c(n) = \Gamma_1^n D_c(0) + \Gamma_2 \sum_{i=1}^n \Gamma_1^{i-1} \delta_{\text{rec}}^2(i) + \Gamma_3 \sum_{i=1}^n \Gamma_1^{i-1} \sigma_r^2(i). \quad (31)$$

Next we analyze the asymptotic behavior of channel distortion. Let T be the number of already coded frames. The average channel distortion of all coded frames is

$$\begin{aligned} \bar{D}_c(T) &= \frac{1}{T} \sum_{n=1}^T D_c(n) \\ &= \frac{1}{T} \sum_{n=1}^T \Gamma_1^n D_c(0) + \frac{\Gamma_2}{T} \sum_{n=1}^T \sum_{i=1}^n \Gamma_1^i \delta_{\text{rec}}^2(i) \\ &\quad + \frac{\Gamma_3}{T} \sum_{n=1}^T \sum_{i=1}^n \Gamma_1^i \sigma_r^2(i) \\ &= \frac{1}{T} \frac{1}{1-\Gamma_1} D_c(0) + \frac{1}{T} \frac{\Gamma_2}{1-\Gamma_1} \sum_{i=1}^T (1-\Gamma_1^{T-i}) \\ &\quad \cdot \delta_{\text{rec}}^2(i) + \frac{1}{T} \frac{\Gamma_3}{1-\Gamma_1} \sum_{i=1}^T (1-\Gamma_1^{T-i}) \sigma_r^2(i). \end{aligned} \quad (32)$$

Since $\delta_{\text{rec}}^2(n)$ and $\sigma_r^2(n)$ are upper bounded by $C = 255 \times 255$, we have

$$\begin{aligned} \sum_{i=1}^T (1-\Gamma_1^{T-i}) \delta_{\text{rec}}^2(i) &\leq C \frac{1}{1-\Gamma_1} \\ \sum_{i=1}^T (1-\Gamma_1^{T-i}) \sigma_r^2(i) &\leq C \frac{1}{1-\Gamma_1}. \end{aligned} \quad (33)$$

Then, we have

$$\begin{aligned} \bar{D}_c &= \lim_{T \rightarrow \infty} \bar{D}_c(T) = \frac{\Gamma_2}{1-\Gamma_1} E[\delta_{\text{rec}}^2(n)] + \frac{\Gamma_3}{1-\Gamma_1} E[\sigma_r^2(n)] \\ &= \frac{\beta p_e}{1-a+(a-p_e)\beta} E[\delta_{\text{rec}}^2(n)] \\ &\quad + \frac{(1-\beta)p_e}{1-a+(a-p_e)\beta} E[\sigma_r^2(n)] \end{aligned} \quad (34)$$

where $E[\delta_{\text{rec}}^2(n)]$ and $E[\sigma_r^2(n)]$ are average values of reconstructed frame difference and residue variance over the whole video sequence, respectively.

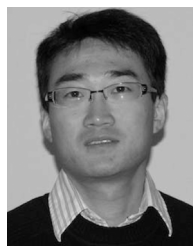
ACKNOWLEDGMENT

The authors would like to thank Prof. J. Woods for valuable discussions that helped to improve the quality of this paper.

REFERENCES

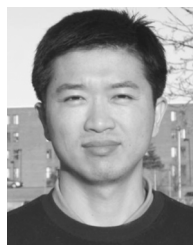
- [1] G. Cheung and A. Zakhor, "Bit allocation for joint source/channel coding of scalable video," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 340–356, Mar. 2000.

- [2] M. Gastpar, B. Rimoldi, and M. Vetterli, "To code, or not to code: Lossy source-channel communication revisited," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1147–1158, May 2003.
- [3] R. Motwani and C. Guillemot, "Tree-structured oversampled filterbanks as joint source-channel codes: Application to image transmission over erasure channels," *IEEE Trans. Signal Process.*, vol. 52, no. 9, pp. 2584–2599, Sep. 2004.
- [4] F. Behnamfar, F. Alajaji, and T. Linder, "Image transmission over the poly channel via channel-optimized quantization," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 728–733, Feb. 2005.
- [5] L. Liu, G. Cheung, and C. Chuah, "Rate-distortion optimized joint source/channel coding of WWAN multicast video for a cooperative peer-to-peer collective," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 1, pp. 39–52, Jan. 2011.
- [6] Z. He, J. Cai, and W. Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 6, pp. 511–523, May 2002.
- [7] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 5, pp. 645–657, May 2005.
- [8] Z. He and D. Wu, "Resource allocation and performance analysis of wireless video sensors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 590–599, May 2006.
- [9] M. Marijan, I. Demirkol, D. Maricic, G. Sharma, and Z. Ignjatovic, "Adaptive sensing and optimal power allocation for wireless video sensors with sigma-delta imager," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2540–2550, Oct. 2010.
- [10] Xiph.org Test Media Repository [Online]. Available: <http://media.xiph.org/video/derf/>
- [11] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti, "Introduction to flash memory," *Proc. IEEE*, vol. 91, no. 4, pp. 489–502, Apr. 2003.
- [12] K.-D. Suh *et al.*, "A 3.3 V 32 Mb NAND flash memory with incremental step pulse programming scheme," *IEEE J. Solid-State Circuits*, vol. 30, no. 11, pp. 1149–1156, Nov. 1995.
- [13] Q. Wu, G. Dong, and T. Zhang, "Exploiting heat-accelerated flash memory wear-out recovery to enable self-healing SSDs," in *Proc. 3rd USENIX Conf. on Hot Topics in Storage and File Syst.*, Jun. 2011, pp. 4–4.
- [14] V. Mohan, S. Gurumurthi, and M. R. Stan, "Flashpower: A detailed power model for nand flash memory," in *Proc. Conference on Design, Autom., Test in Eur.*, 2010, pp. 502–507.
- [15] R. Micheloni, L. Crippa, and A. Marelli, *Inside NAND Flash Memories*. New York: Springer, 2010.
- [16] K. Prall, "Scaling non-volatile memory below 30 nm," in *Proc. IEEE 2nd Non-Volatile Semiconductor Memory Workshop*, Aug. 2007, pp. 5–10.
- [17] A. Viterbi and J. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.
- [18] Z. Chen, P. Zhou, Y. He, and J. Zheng, "Fast integer-pel and fractional-pel motion estimation for H.264/AVC," *Elsevier J. Vis. Commun. Image Represent.*, vol. 17, no. 2, pp. 264–290, Apr. 2006.
- [19] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Elsevier J. Signal Process., Image Commun.*, vol. 15, no. 1-2, pp. 77–94, Sep. 1999.
- [20] J. Kim, R. Mersereau, and Y. Altunbasak, "Error-resilient image and video transmission over the internet using unequal error protection," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 121–131, Feb. 2003.
- [21] M. Gallant and F. Kossentini, "Rate-distortion optimized layered coding with unequal error protection for robust internet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 357–372, Mar. 2001.
- [22] C. Pandana, Y. Sun, and K. Liu, "Channel-aware priority transmission scheme using joint channel estimation and data loading for OFDM systems," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3297–3310, Aug. 2005.
- [23] R. Hormis, E. Linzer, and X. Wang, "Adaptive mode- and diversity-control for video transmission on MIMO wireless channels," *IEEE Trans. Signal Process.*, vol. 57, no. 9, pp. 3624–3637, Sep. 2009.
- [24] Y. Wang and Q. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.
- [25] R. Zhang, S. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 966–976, Jun. 2000.
- [26] M. Sabir, R. Heath, and A. Bovik, "Joint source-channel distortion modeling for MPEG-4 video," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 90–105, Jan. 2009.
- [27] A. Katsaggelos, Y. Eisenberg, F. Zhai, R. Berry, and T. Pappas, "Advances in efficient resource allocation for packet-based real-time video transmission," *Proc. IEEE*, vol. 93, no. 1, pp. 135–147, Jan. 2005.
- [28] M. Etoh and T. Yoshimura, "Advances in wireless video delivery," *Proc. IEEE*, vol. 93, no. 1, pp. 111–122, Jan. 2005.
- [29] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 2004.
- [30] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, July 2003.
- [31] H.264/AVC Reference Software JM15.1. Jan. 2009 [Online]. Available: <http://iphome.hhi.de/suehring/tml/download/>



Yiran Li received the B.S. degree in electronics engineering from Shanghai Jiaotong University, China, in 2003 and the M.S. degree in electrical engineering from the University of Florida, Gainesville, in 2007. He is currently working toward the Ph.D. degree at the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY.

From 2003 to 2006, he was with SONY Corporation, Mobile Products R&D Group, Tokyo, Japan. His research interests include image and video processing, memory architecture design, VLSI architecture, and circuit design for multimedia processing.



Guiqiang Dong (S'09) received the B.S. and M.S. degrees from the University of Science and Technology of China, Hefei, China, in 2004 and 2008, respectively. He is currently working toward the Ph.D. degree in the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY.

His research interests include coding theory, signal processing for data storage systems, and system design for various digital memory.



Tong Zhang (M'02–SM'08) received the B.S. and M.S. degrees in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 1995 and 1998, respectively, and the Ph.D. degree in electrical engineering from the University of Minnesota, Minneapolis, in 2002.

He is currently an Associate Professor with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY. His research activities span over circuits and systems for various data-storage and computing applications.

He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II and the IEEE TRANSACTIONS ON SIGNAL PROCESSING.