J Real-Time Image Proc
DOI 10.1007/s11554-014-0460-8

ORIGINAL RESEARCH PAPER

A dynamically reconfigurable architecture system for time-varying image constraints (DRASTIC) for motion JPEG

Yuebing Jiang · Marios S. Pattichis

Received: 21 April 2014 / Accepted: 27 September 2014

Abstract We propose a dynamically reconfigurable system for time-varying image constraints (DRASTIC) for applications in video communications. DRASTIC defines a framework for both joint and independent optimization of dynamic power, image quality, and bitrate subject to different constraint scenarios. We demonstrate DRASTIC for intra-mode video encoding for MJPEG. However, since the DCT is critical component of most video coding standards, our approach can be extended to modern standards such as AVC (H.264), and emerging standards such as HEVC (H.265), and VP9. Based on a hardware–software co-design approach, we define a family of scalable 2D DCT hardware modules that are jointly optimized with the quality factor (in software). We generate a total of 1,280 configurations of which 841 were found to be Pareto optimal. For full 2D DCT calculation, the results indicate that the proposed method is DRASTIC mode implementation at least as good or significantly better than any previously published implementation. A scalable, real-time controller is used for selecting an appropriate configuration so as to meet time-varying constraints. The real-time controller is shown to satisfy the constraints of different communications modes (e.g., minimum dynamic power, maximum image quality, etc.) as well as to adapt to mode changes. Empirically, we have found that the DRASTIC controller adapts to meet the new constraints within five video frames of a mode change. Overall, the proposed approach yields significant savings over the use of comparable static architectures.

Keywords MJPEG · FPGA · DCT · Zonal · Finite word length · Dynamic partial reconfiguration · Dynamically reconfigurable computing

1 Introduction

The performance of video communication systems depends on balancing requirements associated with the network, the user experience, and the video display device. For example, the network imposes bandwidth constraints. On the other hand, users require sufficient levels of video quality. For display on mobile devices, it is also important to conserve power. Often, the constraints can lead to opposing requirements. For example, delivering higher video quality requires higher levels of power and bandwidth. This paper describes a dynamically reconfigurable system that allows the users to meet real-time constraints on image quality, dynamic power consumption, and available bandwidth. More generally, the term dynamically reconfigurable architecture system for time-varying image constraints (DRASTIC) is used to describe a video communication system that can meet real-time constraints through dynamic reconfiguration.

We propose the use of four fundamental communications modes that can be used to summarize the requirements for optimal performance subject to real-time constraints:

- Minimum dynamic power mode: The goal for this mode is to minimize dynamic power subject to available bandwidth and a minimum level of acceptable

Y. Jiang · M. S. Pattichis (✉)
Department of Electrical and Computer Engineering,
The University of New Mexico, Albuquerque,
NM 87131-000, USA
e-mail: pattichis@ece.unm.edu

Y. Jiang
e-mail: jiangyuebing@gmail.com

Published online: 16 October 2014

image quality. In this mode, a mobile device can reduce its power requirements without sacrificing the user experience. Furthermore, in this mode, a mobile device can conserve energy and maximize its operating time.

- Minimum bitrate mode: The goal of this mode is to minimize bitrate requirements subject to a maximum level of dynamic power and a minimum level of acceptable image quality. Thus, the user can enjoy the compressed video without sacrificing video quality. Furthermore, since the bitrate is minimized, the network can accommodate a large number of users without sacrificing the service.
- Maximum image quality mode: The goal of this mode is to maximize image quality without exceeding the maximum available bandwidth or the maximum available dynamic power. In this mode, the user will be able to examine the video at the maximum possible video quality that can be delivered by all available bandwidth and computing power.
- Typical mode: In this mode, the goal is to optimize a weighted average of the required dynamic power, bitrate, and image quality within constraints on all of them. Here, we have a balanced approach that supports trade-offs between dynamic power, quality, and bitrate.

Clearly, by selecting appropriate weights in the typical mode, we can achieve the performance of the other three modes. Yet, we still focus on the different modes to emphasize the user requirements for minimizing power, bitrate, or maximizing quality.

The DCT is a critical component of most video coding standards. More specifically, AVC (H.264), the emerging HEVC (H.265), and VP9 video coding standards critically depend on DCT and quantization for compression. However, the current paper is focused on the use of DCT with MJPEG. The fundamental advantage of MJPEG comes from its very low complexity (e.g., see [1]) that makes it popular in low-profile webcams, surveillance systems [2, 3] and emerging applications in virtual network computing (VNC) [4]. Essentially, due to its low-complexity, low-power requirements, MJPEG remains popular for mobile devices with limited computational resources. Nevertheless, the system described here can be extended and applied to most video coding standard. The application to a new video coding standard would require replacing the existing DCT and quantization components with the ones developed in this paper. The adaptive controller development will be different since other coding components (intra prediction, coefficients scan process, etc.) may affect performance.

The basic system is shown in Fig. 1. The approach is demonstrated in the compression of the Y-component of color video. Here, a joint software–hardware optimization system uses a dynamic reconfiguration (DR) controller to select DCT hardware cores and quality factor (QF) values to meet constraints in bitrate, image quality, and power. To solve the optimization problem, the system relies on the use of feedback from the current bitrate, image quality measured using the structural similarity index metric (SSIM), and pre-computed dynamic power consumed by an adaptive DCT IP core. A dynamic reconfiguration (DR) controller compares the current bitrate, image quality, and power with the required levels to determine if constraints are met. Depending on each optimization mode, a suitable DCT IP core and quality factor value is selected for the next video frame. Alternatively, the dynamic reconfiguration overhead can be reduced by fixing the hardware configuration over a number of video frames or until a maximum number of reconfigurations has been met.

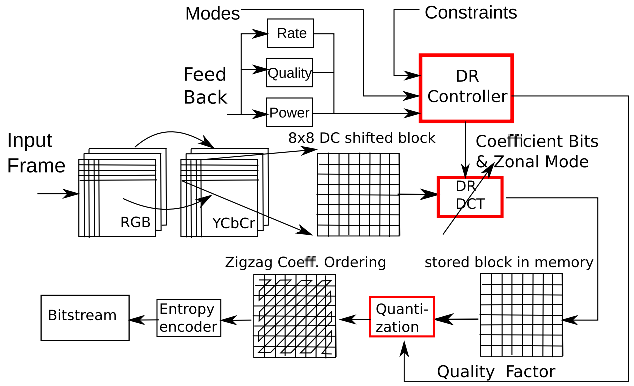


Fig. 1 Dynamically reconfigurable architecture system for time-varying image constraints (DRASTIC) for motion JPEG

The basic contributions of the paper are summarized as follows:

- DRASTIC optimization modes for video communications: The paper introduces new real-time optimization approaches that can be used to minimize dynamic power, maximize image quality, reduce bitrate, or provide balanced solutions for meeting real-time video constraints. This approach extends traditional rate–distortion optimization approaches that do not consider dynamic energy or power constraints or the use of image quality metrics (e.g., SSIM [5]). Earlier work on the modes appeared in our conference paper in [6]. However, this earlier work did not address the case when the constraints cannot be met. To avoid failure when the constraints cannot be met, this paper allows the system to use alternative configurations based on unconstrained optimization [described later with Eq. (5)]. Furthermore, our earlier conference work was never demonstrated on digital videos. In the current paper, the methodology is validated on digital videos.

- Scalable, pareto-optimal DCT cores with quantization control: A scalable architecture is used to generate a family of hardware cores that can be used to compute lower-frequency subsets of the DCT frequencies using different bit-widths. The approach is motivated by the observation that significant compression can be achieved through the effective quantization of high-frequency components. Thus, in addition to the scalable DCT cores, we investigate the use of different quality factors that control the DCT quantization tables. This results in a joint software–hardware optimization approach. Yet, not all generated configurations will necessarily be useful. We use a training set to determine the hardware configurations that are Pareto optimal. Earlier work on the Pareto front appeared in our conference papers published in [6, 7]. The current paper uses a new hardware design that combines the approaches described in [6, 7] to derive a much larger configuration and performance space that was previously considered. As a result, the new approach allows for finer optimization control.
- Scalable dynamic reconfiguration controller based on feedback: a dynamic reconfiguration controller is used for meeting real-time constraints through a joint optimization of the software–hardware configuration. In real time, the controller selects the active DCT core and quality factor from a set of pre-computed, pareto-optimal configurations. After a selection is made, real-time feedback is used for adjusting the DCT core and the quality factor. Dynamic reconfiguration overhead is controlled in a scalable fashion by adjusting the number of video frames between configurations or the total number of reconfigurations per reconfiguration duration that can be used. In the case of unrealistic constraints that cannot be met, the controller selects the best solution based on reformulation of the problem using unconstrained optimization. The controller is completely new and never appeared in our previous conference publications.

The rest of the paper is organized as follows. Related work background is given in Sect. 2. The proposed architecture is described in Sect. 3. Results and analysis are shown in Sect. 4. Section 5 gives conclusion and future work.

2 Background and related work

We provide a summary of related research in this section. Related research on dynamic partial reconfiguration for multi-objective image/video processing is discussed in Sect. 2.1. Related research on rate–distortion–complexity for video compression is discussed in Sect. 2.2. Different hardware implementations of the DCT are detailed in Sect. 2.3. Chen's algorithm is discussed in Sect. 2.4. The use of SSIM for image quality assessment and the quality factor for JPEG is discussed in Sects. 2.5 and 2.6, respectively.

2.1 Dynamic partial reconfiguration for multi-objective image/video processing/compression

Dynamic partial reconfiguration (DPR) on an FPGA system allows the modification of the functionality in real time while allowing the rest of the system to operate normally without requiring a restart [8]. Prior research focused on the computation of DCTs as reported in [9–11]. The basic idea in these papers was to avoid the computation of higher-frequency components by only computing the $N \times N$, $N = 1, 2, \ldots, 8$ lower frequency components. In [9–11], the authors demonstrated the use of this adaptive DCT in an MPEG2 system.

Some of the basic concepts behind the use of dynamic partial reconfiguration (DPR) for meeting real-time constraints have been recently presented in [12]. In [12], the focus was on the development of a video pixel processor that can be adapted to meet real-time constraints in power/energy–performance–accuracy.

To satisfy multi-objective optimization constraints in hardware, there is a need to generate a family of hardware cores that sample different points in the multi-objective space. The pareto front is computed from the family of the generated hardware realizations. The pareto front represents the set of optimal configurations. To meet real-time constraints, a dynamic reconfiguration controller selects a pareto-optimal realization and implements it in hardware using dynamic partial reconfiguration.

2.2 Rate–distortion–complexity control for video compression

A relatively recent attempt to manage real-time computational complexity in video encoding has been described in [13]. In [13], to limit computational complexity, the authors recommended dropping video frames while attempting to manage image quality. Overall, this direct approach attempts to manage video quality losses by smoothing frame rates. As verified by subjective video quality measurements, the managed approach will perform better than a reference encoder.

An approach related to the research presented in this paper has been recently introduced in [14] and further developed in [15]. In [14], the authors were interested in power–rate–distortion (P-R-D) optimization for wireless video communication under energy constraints. Here, the authors use dynamic voltage scaling (DVS) to control power consumption and then investigate the rate–distortion performance under power control. The paper demonstrates how the system adjusts its complexity control to match the available energy supply while maximizing the picture quality. In [15], the authors show that they can achieve up to a 50 % reduction in power consumption by adjusting the hardware configuration to follow the non-stationary characteristics of the video. Some of the issues associated with the attempt to use complexity control with motion estimation have been the focus of more recent research in medium-granularity complexity control (MGCC) reported in [16]. In [16], the authors introduced a rate–complexity–distortion model for a group of pictures (GOP) to allocate complexity at the frame level.

The fundamental advantages of the proposed approach over previous approaches include:

- Dynamically reconfigurable hardware–software setup: DRASTIC extends the standard adjustment of software parameters to a joint software–hardware reconfiguration framework. The approach relies on the use of dynamic partial reconfiguration to adjust DCT hardware resources and the quantization parameter to meet constraints, optimize the use of resources, while optimizing objective performance.
- Dynamic multi-objective optimization and control framework: The DRASTIC approach is a multi-objective, constrained optimization approach. The four DRASTIC modes generalize prior research in this area by considering optimization of individual objectives as well as scalar combinations of the objectives. The approach is dynamic and allows fine control and switching between modes from frame to frame.

As noted earlier, for low-energy devices that use MJPEG, motion estimation is avoided. Furthermore, a scalable and parameterizable system based on the DCT and the quality factor, such as the one developed here, can also be applied for motion compensation of motion-based video coding. Clearly though, our optimization modes provide a general framework for extending this prior research of minimizing energy to new modes that support maximizing quality, minimizing bitrate and a more general typical mode. Our optimization is switchable (with different modes) and adjustable (with different constraints). The scalable DR controller also allows the user to control the reconfiguration overhead while estimating performance at the frame level.

2.3 DCT hardware implementations

This section provides a review of 2D DCT implementations. The review focuses on the complexity of each approach that suggests the need for a separable implementation that allows scalability in the number of accuracy bits and the number of DCT frequencies to be computed.

Let $X_{8\times 8}$ represent an input image block after DC shift. Here, note that the DC shift is implemented by subtracting 128 from the unsigned 8-bit input image. The DCT output image is then represented as a 16-bit signed integer given by:

$$Z_{u,v} = C_u C_v \sum_{i=0}^{7} \sum_{j=0}^{7} X_{i,j} \cos\left(\frac{u(2i+1)\pi}{16}\right) \cos\left(\frac{v(2j+1)\pi}{16}\right) \tag{1}$$

where:

$$C_u = \begin{cases} \frac{1}{2\sqrt{2}} & \text{for u} = 0 \\ \frac{1}{2} & \text{for u} > 0 \end{cases}$$

DCT implementations can be classified into the following categories:

- Direct approaches [17–23]: the 2D DCT is implemented using matrix–vector products. Direct methods based on Chen's algorithm [24] are very effective and represent a very popular choice.
- Distributed arithmetic (DA)-based designs [25–28]: the 2D DCT result is computed bit by bit by considering the products of the DCT basis functions with the input image block. DA-based designs are inherently bit-serial in nature and this issue cannot be addressed effectively except for the special cases (see [25]). Given the complexity and focus on bit-by-bit computation, DA-based approaches cannot be easily adapted for of computing a limited number of DCT frequencies, as required for DRASTIC.
- Systolic array (SA)-based designs [10, 29–31]: the DCT is computed using a relatively large array of processing elements (PEs) arranged in a systolic array pattern. Unfortunately, SA implementations require significant resources and sophisticated I/O control.
- CORDIC-based designs [32–34]: a CORDIC processor is used for computing the cosine coefficients in the DCT. Similar to SA implementations, CORDIC implementations require significant resources.
- Algebraic integer(AI)-based designs [35]: by mapping possibly irrational numbers to an array of integers, these methods can achieve high precision. However, good precision requires significant resources.

Compared to separable approaches, non-separable approaches require more resources since the number of required FIR taps grows as N^2 as opposed to N for the separable case. Furthermore, for dynamic reconfiguration, it is clear that non-separable approaches require considerable overhead since we will have store the architecture descriptions in memory. Thus, in what follows, the paper will focus on a separable approach based on Chen's algorithm [24].

2.4 A separable implementation of the 2D DCT based on Chen's algorithm

From (1), a separable implementation of the 2D DCT is given by $Z = (MX)MT$, where $M_{i,j} = C_i \cos(\frac{i(2j+1)\pi}{16})$. Here, $(MX)MT$ is implemented by first transposing (MX) and then applying M. Thus, a separable implementation is based on: $Z = (M(MX)T)T$.

To efficiently implement multiplication by M, let $D_i = \cos(i\pi/16)/2$, and define $a = D_4$, $b = D_1$, $c = D_2$, $d = D_3$, $e = D_5$, $f = D_6$, $g = D_7$, the output $Y = MX$ is computed using:

$$\begin{bmatrix} Y(0) \\ Y(2) \\ Y(4) \\ Y(6) \end{bmatrix} = \begin{bmatrix} aa \\ 00 \\ a-a \\ 00 \end{bmatrix} \begin{bmatrix} 10 \\ 01 \\ 10 \\ 01 \end{bmatrix} \begin{bmatrix} X(0)+X(7) \\ X(1)+X(6) \\ X(2)+X(5) \\ X(3)+X(4) \end{bmatrix}. \qquad (2)$$

Then, an efficient matrix decomposition can be used to implement the odd-indexed output expressed as:

$$\begin{bmatrix} Y(1) \\ Y(3) \\ Y(5) \\ Y(7) \end{bmatrix} = \begin{bmatrix} bd \\ d-g \\ e-b \\ g-e \end{bmatrix} \begin{bmatrix} X(0)-X(7) \\ X(1)-X(6) \\ X(2)-X(5) \\ X(3)-X(4) \end{bmatrix}. \qquad (3)$$

To produce a frequency-scalable representation, begin with the lower-indexed DCT coefficients given by $Y(0), Y(1), \ldots, Y(n)$, where $n \le 7$. Then, in the implementation of the DCT, the corresponding rows in Eqs. (2) and (3) need to be implemented so as to compute the required DCT coefficients. In the separable approach described here, the 2D DCT coefficients are given by $X_{u,v} \quad 0 \le u, v \le n$ [see Eq. (1)].

2.5 Video image quality assessment using SSIM

Video image quality will be assessed using the structural similarity index (SSIM) [5]. Here, note that video quality assessment is still an open problem (e.g., see [36–39]). However, SSIM provides a simple and effective method for assessing video image quality of individual frames.

Assuming that x, y represent the original and reconstructed images, SSIM is given by:

$$\mathrm{SSIM}(x,y) = l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma} \qquad (4)$$

which is expressed as the product of the luminance ($l(x,y)$), the contrast ($c(x,y)$), and structure components ($s(x,y)$), and $\alpha, \beta, \gamma > 0$ are set to the default value of 1.

2.6 Quantization table specification using a quality factor

The DCT quantization level will be controlled using the quality factor (QF). QF is given as integer value that is constrained between 1 and 100. The DCT quantization table is then given by:

$$Q_{ij} = \mathrm{Clip}_{1,255}\left[\frac{\mathrm{Q}^{*}_{\mathrm{ij}} \cdot \mathtt{scale} + 50}{100}\right]$$

clipped to stay within 1 and 255, Q^{*}_{ij} refers to the standard JPEG quantization table for $\mathtt{scale} = 1$ and the $\mathtt{scale}$ is given by:

$$\mathtt{scale} = \begin{cases} 5{,}000/\mathrm{QF}, & \text{for} \quad 1 \le \mathrm{QF} < 50; \\ 200 - 2 \cdot \mathrm{QF}, & \text{for} \quad 50 \le \mathrm{QF} \le 99; \\ 1, & \text{for} \quad \mathrm{QF} = 100. \end{cases}$$

3 A dynamically reconfigurable architecture system for time-varying image constraints (DRASTIC)

To introduce the proposed DRASTIC implementation, the constrained optimization framework for defining DRASTIC modes is discussed in Sect. 3.1. Section 3.2 describes the implementation of reconfigurable hardware DCT IP cores. The generation of the pareto and the selection of an optimal configuration are discussed in Sect. 3.3. A scalable controller to minimize the reconfiguration overhead is discussed in Sect. 3.4. Then, the proposed DRASTIC implementation for M-JPEG is given in Sect. 3.5.

3.1 Constrained optimization formulation

The optimization objectives are defined in terms of: (1) *DP* which denotes the dynamic power consumed by the FPGA (see [12]), (2) *BPS* which denotes the number of bits per sample, and (3) *Q* which denotes the image quality metric. In terms of the free parameters that we can modify to achieve our objectives, let *HW* denote the different hardware configurations, and let *QF* denote the quality factor used for controlling the quantization table (software controlled). To formulate constraints on the objectives, let $B_{\max}$ denote the maximum acceptable bitstream bandwidth, $P_{\max}$ denote the maximum dynamic power available, and $Q_{\min}$ denote the minimum level of acceptable image quality.

The DRASTIC modes are then defined as constrained optimization problems using:

– minimum power mode (`mode=0`):

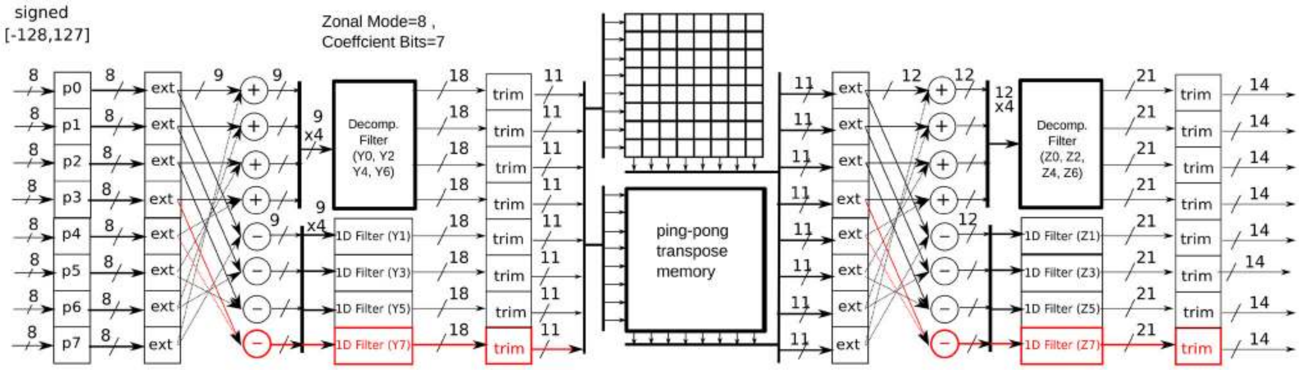


Fig. 2 Scalable data path for the 2D-DCT using ping-pong transpose memory. The number of bits used at each stage is given. The input image block is assumed to be of size 8 with signed, 8-bit integer values. The removal of the highest frequency components is highlighted in *red*

$$\min_{HW,QF} DP \text{ subj.to : } (\mathrm{SSIM} \geq \mathrm{Q}_{\min}) \text{ and } (\mathrm{BPS} \leq \mathrm{B}_{\max}).$$

- minimum bitrate mode (`mode=1`):

$$\min_{HW,QF} \mathrm{BPS} \text{ subj.to : } (\mathrm{SSIM} \geq \mathrm{Q}_{\min}) \text{ and } (\mathrm{DP} \leq \mathrm{P}_{\max}).$$

- maximum image quality mode (`mode=2`):

$$\min_{HW,QF} -\mathrm{SSIM} \text{ subj.to : } (\mathrm{BPS} \leq \mathrm{B}_{\max}) \text{ and } (\mathrm{DP} \leq \mathrm{P}_{\max}).$$

- typical mode (`mode=3`):

$$\min_{HW,QF} -\alpha \cdot \mathrm{SSIM} + \beta \cdot \mathrm{BPS} + \gamma \cdot \mathrm{DP}$$

$$\text{subj.to :}$$

$$(\mathrm{BPS} \leq \mathrm{B}_{\max}) \text{ and } (\mathrm{SSIM} \geq \mathrm{Q}_{\min}) \text{ and } (\mathrm{DP} \leq \mathrm{P}_{\max}).$$

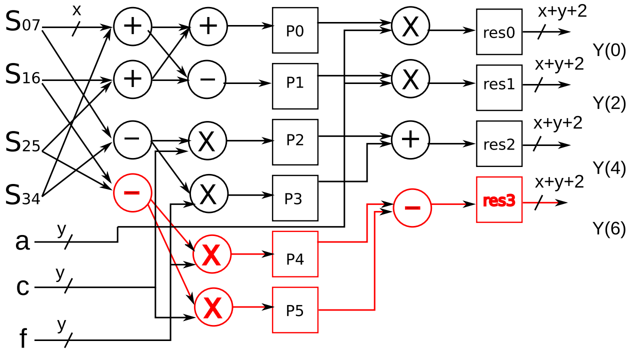


Fig. 3 Scalable decompose filter implementation of the matrix–vector product given in Eq. (2). Refer to Fig. 2 for how the decompose filter fits the DCT core. The inputs S_{ij} refer to the $X(i) + X(j)$ sum of equation (2). The outputs correspond to $Y(0), Y(2), Y(4), Y(6)$ of (2). The datapath associated with the highest frequency component is highlighted in *red*. Note how tracing backwards from each output, we can generate a scalable datapath that removes the circuitry associated with each frequency component

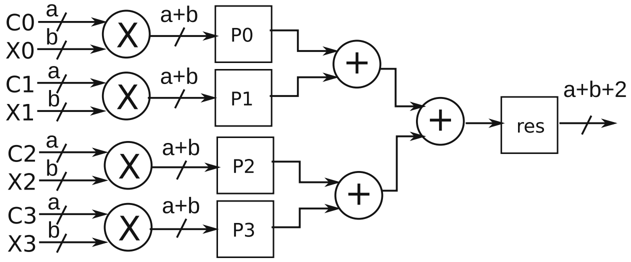


Fig. 4 Implementation of the 1D filters shown in Fig. 2. Here, `C0--C3` refer to the DCT Kernel coefficients and `X0--X3` refer to sums and differences computed on the input data (see Fig. 2)

3.2 Hardware design

A scalable and separable implementation of the DCT is shown in Fig. 2. A ping-pong memory [22] implementation is used for efficient implementation of the transpose operation. The decompose filter component shown in Fig. 2. The architecture design of the decompose filter is shown in Fig. 3. The 1D filter used in implementing (Eq. 3) is shown in Fig. 4.

The implementation in Fig. 2 represents a parallelized and pipelined implementation. In terms of parallelism, we note that the column DCTs can be implemented in parallel, followed by transposition in ping-pong memory, and then the row DCTs. Row operations in Eq. (3) are carried out in parallel using 1D filters. The trim operations implement floor operations by truncating the results towards zero as shown in Fig. 5. For each 1D DCT, we have a four-staged pipeline. The ping-pong transpose memory consists of two 8 × 8 transpose memory arrays. In pipelined operation, a row DCT is computed in each cycle. Furthermore, it takes eight cycles to complete a 2D DCT.

The scalability required for effective multi-objective optimization is implemented using zonal control [7, 9–11, 40], and output bit-width control. As stated earlier, the basic idea is to achieve perceptual scalability by keeping the lower frequency components, while eliminating the computation of higher-frequency components. We implement 8 levels associated with keeping the DCT lower-frequency subsets of the complete frequency set. We use Z0 to Z7 to denote the different zones (levels) associated with the DCT computation. Similarly, for bit-width control, we keep the most significant bits [6, 41]. This is implemented adjusting the word-length implementation of the DCT coefficients a to g in Sect. 2.3 using $WL \in [2, 9]$.

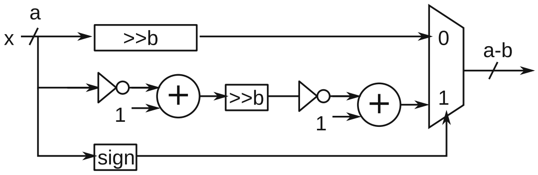


Fig. 5 Signed integer trimming of a-bit input x to an (a-b)-bit output by truncating the output towards zero (floor operation). This component is used to control the bit-width in the optimization process

3.3 Pareto front and configuration based on constraints

The selection of possible DCT hardware core needs to be jointly considered with control of the quantization table. For example, gains due to increasing the bit-width in the DCT cores may be offset by a decrease in the quality factor (see Sect. 2.6). The goal here is to eliminate configurations that are not pareto optimal [12]. In other words, we

Fig. 6 General framework for DRASTIC mode implementation

```
Input: input video, initial DRASTIC mode $\mathtt{DRASTIC_{mode}}$ with associated
    constraints as subset of $(\mathtt{B_{max}}, \mathtt{P_{max}}, \mathtt{Q_{min}})$, offline trained Pareto front, re-
    configuration period RecP, maximum allowed reconfigurations RecC per
    RecDur frames.
Output: generated compressed video stream that implements the given
    DRASTIC mode.
Initialize $\mathtt{DRASTIC_{mode}}$ as initially specified.
Initialize RecDur as initially specified.
Initialize constraints as initially specified.
Initialize counter for dyn. reconf.:  $\mathtt{RecCtr} = 0$.
Initialize frame index inside a reconf. dur. : $n = 0$.
Initialize constraint budgets:
    $\Delta_{\mathtt{BPS},0} = \Delta_{\mathtt{DP},0} = \Delta_{\mathtt{Q},0} = 0$.
while video communication holds do
  while  $\mathtt{n} < \mathtt{RecDur}$  do
    if ($\mathtt{RecCtr} < \mathtt{RecC}$ and $\mathtt{n}\%\mathtt{RecP} = 0$) then
      Search Pareto front for opt. $HW, QF$
      if no configuration satisfies constraints then
        Solve unconstr. opt. prob. for opt.
            $HW, QF$ as given by eq. (5).
      end if
      Apply DPR with $HW, QF$.
      Compress current frame.
      Update constr. budgets as given
          by eqns (6) and (7).
      Update $\mathtt{RecCtr} = \mathtt{RecCtr} + 1$.
    else
      Compress current frame.
    end if
    Update $n = n + 1$.
  end while
  Reset $\mathtt{n} = 0$, $\mathtt{RecCtr} = 0$,
      $\Delta_{\mathtt{BPS},0} = \Delta_{\mathtt{DP},0} = \Delta_{\mathtt{Q},0} = 0$.
  Check $\mathtt{DRASTIC_{mode}}$ for change.
  Check RecDur for change.
  Check constraints for change.
end while
```

eliminate configurations for which we can find another configuration that delivers performance that is at least as good in all of the objectives (quality, power, bitrate), and performs better in at least one of the objectives [12]. The remaining configurations represent the Pareto front that will be used in further optimization.

In practice, the pareto front can be computed offline using a training set. For each configuration, the average performance for each set of objectives will be used for determining the Pareto front.

We consider a direct implementation of DRASTIC modes by searching through the pareto front. In this direct approach, we select the configuration that minimizes the optimization objective while satisfying the constraints. For example, in the maximum quality mode, we would select the configuration that provides for the best (max) image quality while not exceeding constrains on power and bitrate.

Clearly, it is possible that the constraints cannot be met on the pareto front. In this case, we reformulate the problem using unconstrained optimization so that the controller will select a configuration that will be as close as possible to the desired constraints. For example, for the typical mode, when all constraints are active, we select the configuration that solves:

$$\begin{aligned} \min_{HW,QF} \quad & a(\mathtt{B}_{\mathtt{max,n}} - \mathtt{BPS}_{\mathtt{n}})^2 \\ & + b(\mathtt{P}_{\mathtt{max,n}} - \mathtt{DP}_{\mathtt{n}})^2 \\ & + c(\mathtt{Q}_{\mathtt{min,n}} - \mathtt{SSIM}_{\mathtt{n}})^2, \end{aligned} \tag{5}$$

where the weights a, b, c can be set equal or adjusted to give emphasis to different constraints. When a constraint is not active, its corresponding weight is set to zero. For example, for the maximum quality mode, we will set $c = 0$. We select the weights so as to scale each constraint violation by the user-specified range of bounds. For example, if Q_1, Q_2 represent the minimum and maximum bounds on image quality, we set c to $c = 1/(Q_2 - Q_1)^2$. We use a similar approach for a and b.

3.4 Scalable control of reconfiguration overhead

Unlike architectures for H.264 and H.265 video encoding, for our target MJPEG application, feedback can be an expensive operation. Thus, we avoid using feedback or dynamic partial configuration (DPR) for every single video frame. Periodic updates and limited number of reconfigurations are adopted requiring the estimation of rate–distortion performance for a selected number of frames for which reconfiguration is allowed. We will next provide a description of the approach. Let `RecP` denote the reconfiguration period that describes the number of frames between two system reconfigurations. Also, let `RecC` denote the maximum number of allowed reconfigurations for a specified duration of `RecDur` (from recconfiguration duration) video frames. Unless otherwise specified, we set `RecDur` = 100 by default so that `RecC` can be interpreted as the percentage of the number of fames for which the system is allowed to reconfigure.

The proposed approach is to adjust the frame-level constraints set for each video frame for which we have rate–distortion estimates so that the mode constraints will be met on average over the processed frames. To illustrate the basic idea for the bitrate constraint, let n denote the current video frame, and recall that $B_{\max}$ denotes the maximum number of bits per pixel. After processing the video frame, let $\mathtt{BPS}_{\mathtt{n}}$ denote the measured number of bits per sample that were used in encoding the n-th frame. We then have that the remaining bits per sample that can be allocated (or deallocated) to future frames are given by

$$\Delta_{\mathtt{BPS,n}} = B_{\max} - \mathtt{BPS}_{\mathtt{n}}. \tag{6}$$

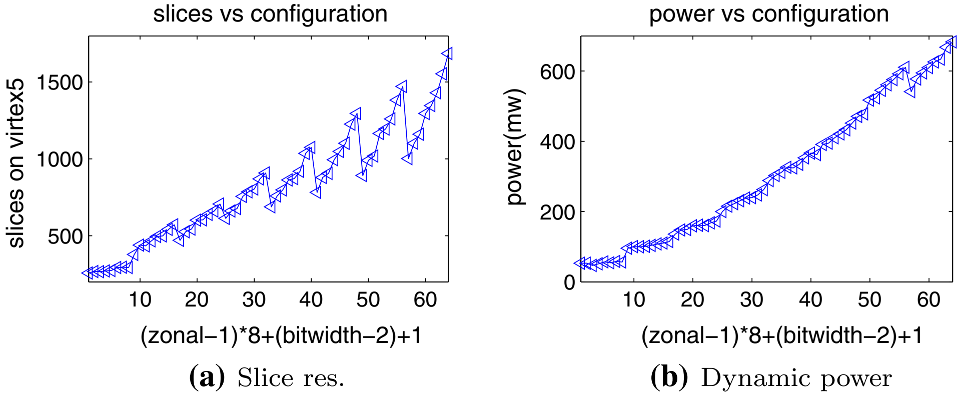
Assuming that we periodically reconfigure after `RecP` frames, we would then adjust the maximum bitrate allocated for the n-th frame $\mathtt{B}_{\mathtt{max,n}}$ using:

$$\mathtt{B}_{\mathtt{max,n}} = \begin{cases} \mathrm{B}_{\max} + \Delta_{\mathtt{BPS,n}}/\mathtt{RecP}, & \text{for}\,\mathrm{n}\,\%\,\mathtt{RecP} = 0 \\ \mathtt{B}_{\mathtt{max,n-1}}, & \text{otherwise}, \end{cases} \tag{7}$$

which allocates (or deallocates) the remaining bits over `RecP` frames. To see how the correction is applied, note that a correction of $\Delta_{\mathtt{BPS,n}}/\mathtt{RecP}$ every `RecP` frames. However, after the correction is made, the rest of the frames in the period get the same amount of correction since they use $\mathtt{B}_{\mathtt{max,n-1}}$ that already includes the additional term of $\Delta_{\mathtt{BPS,n}}/\mathtt{RecP}$. Thus, by adjusting the number of bits per sample for each video frame, we expect that the constraints will be met on average. Similarly, the approach can be applied for updating constraints on image quality $\mathtt{Q}_{\mathtt{min,n}}$ and dynamic power $\mathtt{P}_{\mathtt{max,n}}$. Furthermore, the rule in Eq. (7) can be easily extended so that it will only apply when the number of dynamic reconfigurations does not exceed a maximum bound.

3.5 Scalable DRASTIC controller

The general framework for implementing a DRASTIC mode is given in Fig. 6. Switching between DRASTIC modes requires that the system updates the $\mathtt{DRASTIC}_{\mathtt{mode}}$ variable and that the algorithm will recognize the change at the end of a reconfiguration duration. On the other hand, we note that the pareto front is applied to all of the modes and thus, it only needs to be computed once over the training data. The controller adjusts the overhead based on maximum number of reconfigurations `RecC` per `RecDur` frames and the reconfiguration period `RecP`.



(a) Slice res. **(b)** Dynamic power

Fig. 7 Resource allocation and estimated dynamic power consumption for 64 hardware configurations based on bit-width values: $2 \le WL \le 9$ and zonal values: $1 \le \mathcal{Z} \le 8$. **a** Slice resources as a function of the zonal configuration and bit-width. **b** Dynamic power consumption as a function of zonal configuration and bit-width. From the dynamic power results, it is clear that a scalable set of DCT architectures has been achieved

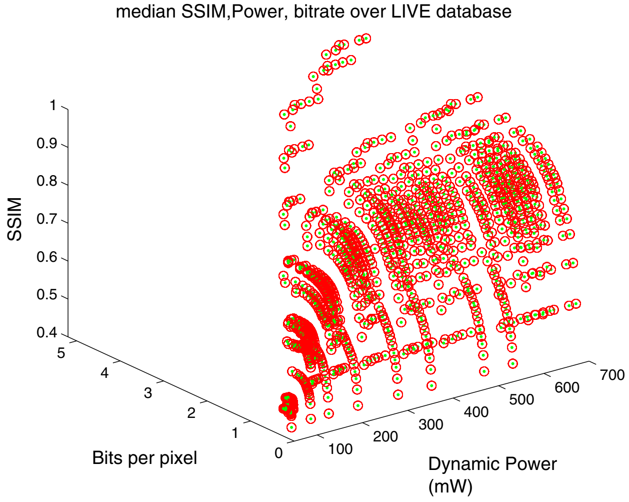


Fig. 8 Pareto front estimation for the joint space of SSIM, bitrate, and dynamic power consumption. The pareto front is estimated based on the median values for the entire UT LIVE image database. Refer to Figs. 11 and 12 for box plots of the results. Pareto optimal configurations are highlighted using *red circles*

For each reconfiguration duration, initially, a single video frame is processed to estimate the objectives based on the initial configuration. The relevant constraint budgets are then updated and used to search the pareto front for the optimal configuration [see Eq. (7)]. Failure to find a configuration that satisfies the constraints will force a reformulation of the problem as an unconstrained optimization problem. Once the optimal configuration has been found, it is used for processing the remaining $\texttt{RecP} - 1$ frames of the current period. The procedure is then repeated for the next set of $\texttt{RecP}$ video frames until *RecC* reconfigurations have been executed. Once the configuration number *RecC* is reached, rest frames in this duration will be compressed with configuration unchanged.

4 Results

In this result section, we first describe how the Pareto front is generated and compare our DCT implementation with other state-of-the-art DCT implementations in Sect. 4.1. DRASTIC implementation results and how to choose scalable parameters are analyzed in Sect. 4.2. Finally, an example of DRASTIC mode transition is discussed in Sect. 4.3.

4.1 Pareto-front estimation and comparisons of full 2D DCT implementations

To generate an estimate of the Pareto front, we use the LIVE image database as a training set [42]. For each configuration, we generate the hardware core and estimate the required bitrate, image quality, and dynamic power that is required for compressing each image. For the dynamic power, we use Xilinx's XPower tool to estimate power consumption on a Virtex-5 device (Xilinx XC5VLX110T). Then, the Pareto front is estimated based on the median value of each configuration.

We generate hardware configurations by varying: (1) the software-based quality factor $\mathrm{QF} = 5, 10, 15, \ldots, 100$ (20 settings), (2) the DCT hardware to compute $Z_{u,v}$ for $0 \le u,\ v < \mathcal{Z} = 1$ to 8 (8 settings), and (3) the DCT coefficients implemented in hardware using word length $WL = 2, 3, 4, \ldots, 9$ (8 settings). Based on the different settings, we have $20 \times 8 \times 8 = 1{,}280$ possible configurations from which only 841 were found to be Pareto optimal. The Pareto front is shown in Fig. 8.

Table 1 Synthesized results for DCT Cores on XC5VLX110T −1FF1136

WL	$\mathcal{Z}$	LUT (%)	Registers (%)	Slices (%)	Max freq (MHz)	Power (mW)
2	1	622 (1)	534 (1)	257 (1)	250.376	53.03
3	1	684 (1)	540 (1)	267 (1)	249.066	52.98
4	1	659 (1)	540 (1)	266 (1)	249.066	46.67
5	1	713 (1)	545 (1)	271 (1)	245.881	51.21
6	1	756 (1)	547 (1)	274 (1)	204.834	56.87
7	1	768 (1)	548 (1)	293 (1)	204.834	54.37
8	1	799 (1)	552 (1)	295 (1)	203.832	58.22
9	1	806 (1)	553 (1)	293 (1)	203.832	56.58
2	2	847 (1)	886 (1)	378 (2)	250.689	95.85
3	2	1,074 (1)	954 (1)	439 (2)	249.066	100.28
4	2	1,099 (1)	958 (1)	434 (2)	247.463	100.06
5	2	1,314 (1)	988 (1)	467 (2)	209.293	101.09
6	2	1,361 (1)	999 (1)	495 (2)	204.834	103.57
7	2	1,403 (2)	1,006 (2)	499 (2)	204.750	108.03
8	2	1,556 (2)	1,024,(2)	541 (3)	203.832	110.65
9	2	1,662 (2)	1,038 (2)	571 (3)	203.293	114.56
2	3	951,(1)	1,137 (1)	470 (2)	250.689	135.65
3	3	1,238 (1)	1,211 (1)	526 (3)	249.066	147.82
4	3	1,238 (1)	1,215 (1)	543 (3)	247.463	148.92
5	3	1,507 (2)	1,250 (2)	601 (3)	209.293	159.57
6	3	1,597 (2)	1,263 (2)	603 (3)	204.834	161.17
7	3	1,651 (2)	1,271 (2)	639 (3)	204.750	160.50
8	3	1,835 (2)	1,293 (2)	661 (3)	203.832	167.79
9	3	1,948 (2)	1,308 (2)	706 (4)	203.293	172.91
2	4	1,242 (1)	1,454 (1)	613 (3)	250.689	199.83
3	4	1,540 (2)	1,548 (2)	661 (3)	249.066	215.33
4	4	1,573 (2)	1,555 (2)	677 (3)	247.463	222.61
5	4	1,936 (2)	1,599 (2)	756 (4)	207.684	230.89
6	4	2,070 (2)	1,619 (2)	786 (4)	203.998	238.55
7	4	2,139 (3)	1,632 (3)	804 (4)	203.832	239.61
8	4	2,416,(3)	1,672 (3)	870 (5)	203.832	247.22
9	4	2,624 (3)	1,699 (3)	908 (5)	202.143	262.42
2	5	1,367 (1)	1,701 (1)	688 (3)	250.689	288.24
3	5	1,727 (2)	1,832 (2)	759 (4)	205.170	302.56
4	5	1,844 (2)	1,849 (2)	798 (4)	202.593	311.60
5	5	2,214 (3)	1,899 (3)	863 (4)	160.668	325.87
6	5	2,341 (3)	1,917 (3)	872 (5)	160.668	324.05
7	5	2,464 (3)	1,941 (3)	919 (5)	158.078	334.35
8	5	2,771 (4)	1,989 (4)	1,036 (5)	157.953	352.93
9	5	3,040 (4)	2,019 (4)	1,074 (6)	156.519	366.55
2	6	1,560 (2)	1,983 (2)	782 (4)	250.689	362.18
3	6	1,977 (2)	2,146 (2)	879 (5)	205.170	390.94
4	6	2,125 (3)	2,166 (3)	905 (5)	202.593	394.71
5	6	2,613 (3)	2,248 (3)	994 (5)	160.668	409.80
6	6	2,742 (3)	2,277 (3)	1,048 (6)	160.668	421.03
7	6	2,880 (4)	2,306 (4)	1,101 (6)	158.078	432.90
8	6	3,315 (4)	2,363 (4)	1,226 (7)	157.953	451.66
9	6	3,659 (5)	2,412 (5)	1,295 (7)	156.519	470.98
2	7	1,685 (2)	2,230 (2)	892 (5)	250.689	478.08
3	7	2,164 (3)	2,430 (3)	990 (5)	205.170	517.46
4	7	2,396 (3)	2,460 (3)	1,018 (5)	202.593	523.60
5	7	2,891 (4)	2,548 (4)	1,165 (6)	160.668	545.27
6	7	3,013 (4)	2,575 (4)	1,194 (6)	160.668	560.92
7	7	3,205 (4)	2,615 (4)	1,260 (7)	158.078	575.57
8	7	3,670 (5)	2,680 (5)	1,382 (7)	157.953	592.27
9	7	4,075 (5)	2,732 (5)	1,471 (8)	156.519	610.75
2	8	1,895 (2)	2,510 (2)	1,002 (5)	250.689	541.17
3	8	2,431 (3)	2,742 (3)	1,101 (6)	205.170	577.46
4	8	2,694 (3)	2,775 (3)	1,160 (6)	202.593	594.46
5	8	3,338 (4)	2,898 (4)	1,295 (7)	160.668	609.64
6	8	3,481 (5)	2,934 (5)	1,347 (7)	160.668	625.68
7	8	3,686 (5)	2,979 (5)	1,430 (8)	158.078	634.09
8	8	4,212 (6)	3,046 (6)	1,553 (8)	157.953	667.75
9	8	4,716 (6)	3,122 (6)	1,686 (9)	156.519	682.88

The resulting hardware configurations are summarized in Table 1 and the corresponding estimated dynamic power is shown in Fig. 7. To visualize the scalability of the proposed approach, we index the hardware configurations using $\texttt{Config} = (\mathcal{Z} - 1) \cdot 8 + \texttt{WL} - 1$. Then, we plot the the required slices and dynamic power against Config in Fig. 7. From Fig. 7b, it is clear that the dynamic power is densely sampled in the configuration space. Returning to the pareto front results of Fig. 8, it is important to note the relatively dense sampling achieved over the pareto front for image quality levels associated with $\texttt{SSIM} > 0.7$. This observation is important since reducing image quality below this level will produce images of unacceptable quality (e.g., see Fig. 12d).

A comparison of the full DCT implementation against other FPGA implementations is given in Table 2. As a result of the parallel and pipelined implementation, the proposed DCT architecture achieves the highest throughput by only requiring 8 cycles to compute a 2D DCT. Yet, the implementation requires lower numbers of FPGA slices and consumes low levels of dynamic power. In terms of dynamic power, we note the lower results due to Huang et al. [9] were achieved using a much simpler architecture at a much lower frequency (41.79 versus 100 MHz of the proposed approach), that requires a significantly higher number of throughput cycles. In any case, the greatest advantage of the proposed approach is the fact that it is scalable in dynamic power, image quality, and bitrate while

Table 2 A comparison of FPGA implementations of 2D DCTs

	Tumeo et al. [44]	Huang et al. [9]	Madanayake et al. [35]	Sharma et al. [45]	Yuebing and Pattichis [6]	Proposed
Arch.	Single 1D DCT + ping pong TRAM	8 PE + TRAM	2 AI-DCT + TBuffer	DA based structure	Double 1D DCT+ TRAM	Double 1D DCT+ ping pong TRAM
Slices	2,823	2,944 (8 PEs)	2,377–3,618	1,701	807–1,657	257–1,686
Tech.	Xilinx	Xilinx	Xilinx	Xilinx	Xilinx	Xilinx
	Virtex II	Virtex 4	Virtex 6	Virtex II	Virtex 5	Virtex 5
	XC2VP30	XC4VSX35	XC6VLX240T	XC2VP30	XC5VLX110T	XC5VLX110T
Latency (cycles)	160	N/A	N/A	N/A	22	20
8 × 8 throughput (cycles)	64	25–102	N/A	N/A	16	8
Dyn. Power (mW)	N/A	24.03–26.27	897–1,687	751	85.2–203.19	51–683
Max. freq. (MHz)	107	N/A	123–308	45.17	200–275	157–251
Oper. freq. (MHz)	N/A	41.79	123–308	45.17	100	100

Dynamic power results are estimated for the operating frequency. Given the small number of cycles required by the proposed approach, it is clear that the proposed method yields the most energy efficient approach. For our experiments, we used a Virtex-5 XC5VLX110T device. The FPGA architecture consists of an array of configurable cell blocks (CLBs) of 160 × 54 CLBs, 17,280 slices (see http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf)

Table 3 DRASTIC constraint profiles

DRASTIC constraint	Constraints profile		
	Low	Medium	High
Image quality (SSIM)	0.7	0.8	0.9
Bitrate (bits per sample)	0.5	1.0	1.5
Power (mW)	200	300	400

The constraints represent the bounds for (1) image quality ($Q_{\min}$), (2) the bitrate ($B_{\max}$), (3) and dynamic power ($P_{\max}$) as described in Sect. 3.1

providing full DCT calculation that is at least as good or better than any previously published approach.

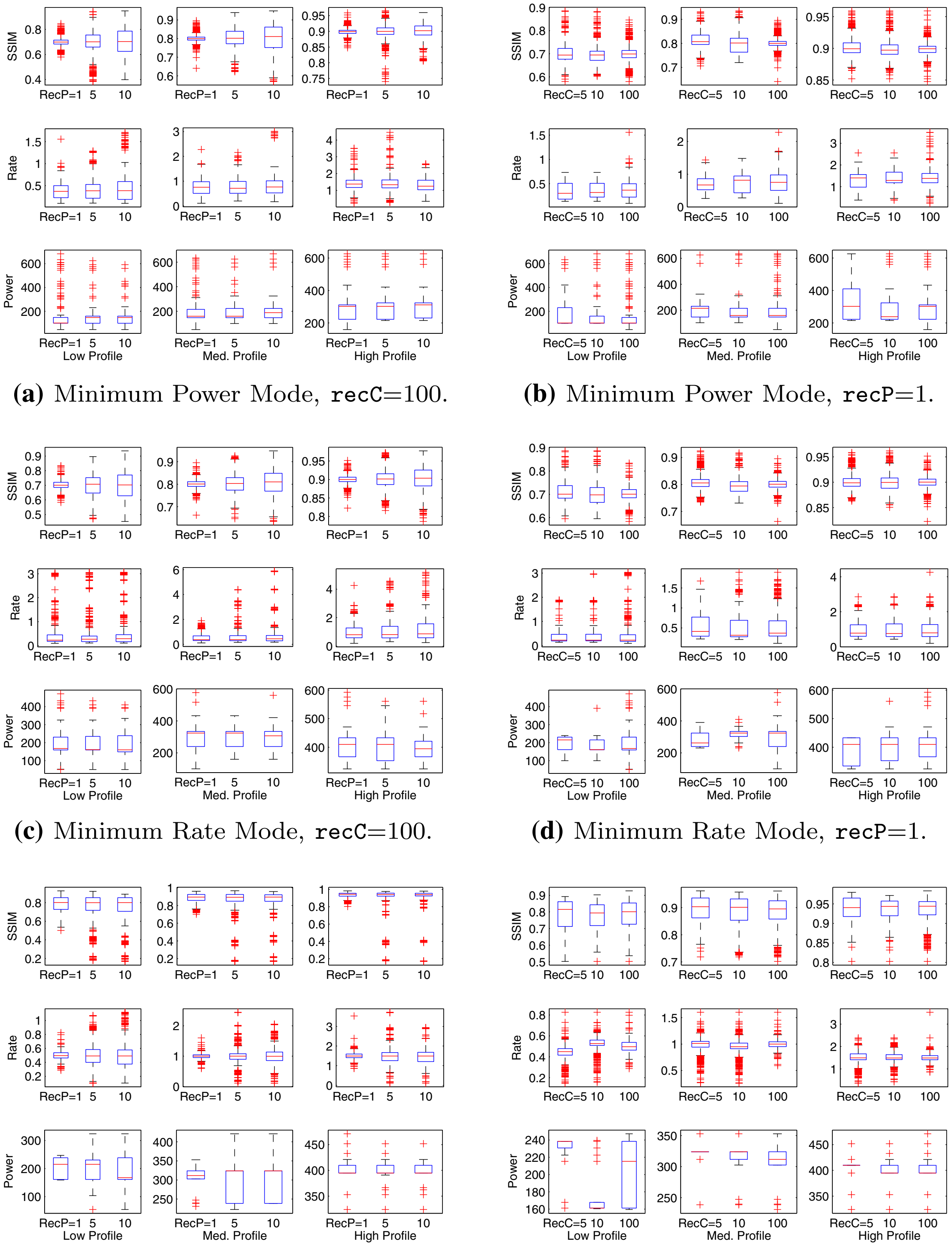
4.2 DRASTIC mode implementation and comparison to optimized static approaches

This section summarizes how the proposed approach can lead to significant savings over optimized static approaches. While training was performed on the UT LIVE image database, the system was validated on an independent testing video database of nine standard videos: city, crew, football, foreman,hall monitor, harbor, mobile, mother and daughter and soccer (see [43]).

To define realistic DRASTIC constraint profiles, we need to select profiles that are compatible with the pareto front (see Fig. 8). Given the wide applications of the UT LIVE image quality databases, we expect that the values derived from them will be widely applicable. In general, we can see that we can achieve higher values of image quality by allocating higher bitrate and larger values of dynamic power. To understand this trend, note that higher image quality results from the need to compute higher-frequency components and longer word lengths that result in higher dynamic power. Furthermore, storing the higher-frequency components requires additional bits that raise the number of bits per sample. Realistically, image quality bounds need to have SSIM values about 0.7 to maintain a minimum level of acceptable image quality. This discussion leads to the low, medium, and high profiles given in Table 3.

The efficient implementation of the DRASTIC modes requires that we determine optimal parameters for RecC and RecP so as to minimize the reconfiguration overhead while still providing acceptable performance. We investigate the trade-off between RecC and RecP by considering all DRASTIC modes for (1) periodic update control using RecP = 1, 5, 10, while allowing the maximum number of reconfigurations per RecDur = 100 frames using RecC = 100, and (2) initial adaptation control using RecC = 5, 10, 100, while allowing the maximum number of periodic updates using RecP = 1. The results are summarized in Figs. 11 and 12.

For the typical mode plots of Fig. 12, it is clear that the constraints are met for all profiles. In other words, for most configurations, image quality remains above the minimum levels, while dynamic power and required bitrates remain below the bounds. For all of the other modes, only two of



(a) Minimum Power Mode, `recC`=100.

(b) Minimum Power Mode, `recP`=1.

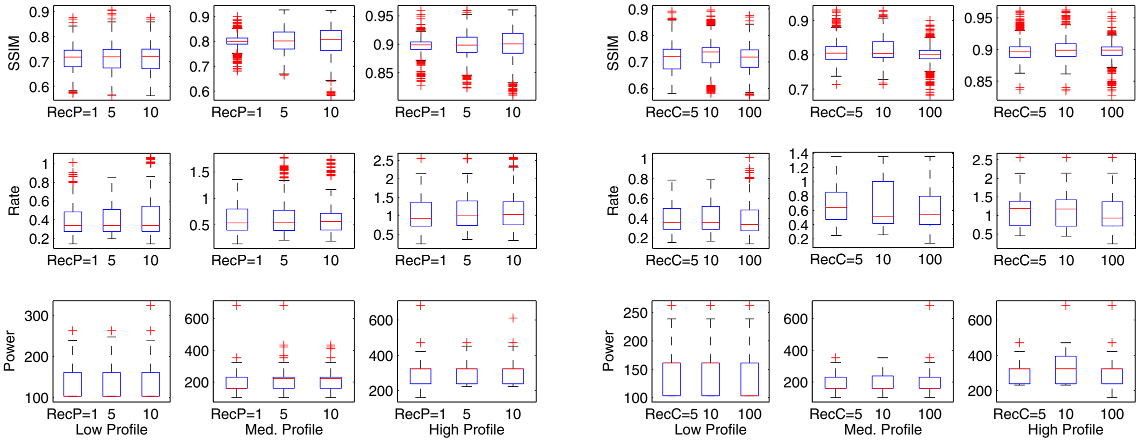
(c) Minimum Rate Mode, `recC`=100.

(d) Minimum Rate Mode, `recP`=1.

(e) Maximum Quality Mode, `recC`=100.

(f) Maximum Quality Mode, `recP`=1.

Fig. 9 DRASTIC performance for the minimum power, maximum image quality, and minimum rate modes as a function of the reconfiguration period `RecP` and the number of reconfigurations `RecC`. The box plots indicate the median (*central line*), while *box edges* represent the 25th and 75th percentiles, whiskers show the extremes of the distribution and outlier points are plotted using *plus symbol*



(a) Typical Mode, recC=100.

(b) Typical Mode, recP=1.

Fig. 10 DRASTIC performance for the typical mode as a function of the reconfiguration period RecP and the number of reconfigurations RecC

the three constraints are active, while the remaining constraint becomes an objective to be optimized. For example, for the maximum image quality mode demonstrated in Fig. 11e, f, it is clear that we have substantially higher image quality that the typical mode, which can push dynamic power consumption slightly above the constraints. On average though, it is clear that most constraints are met for the non-typical modes. Since the validation is independent of the training set, we can infer that the pareto front from the UT LIVE image database captured the constraints in more general settings.

The use of larger reconfiguration periods (larger RecP) tends to spread out the distributions of the objectives. With larger spreads, we also get an increase in constraint violations. On the other hand, when reconfiguring after each frame (RecP = 1), the number of reconfigurations (RecC) does not seem to provide significant improvements for larger values (1 to 5 to 100). Thus, by allowing an early adaptation to the input video using RecP = 1, and then limiting the number of dynamic reconfigurations (RecC = 5), we have an effective control of the overhead while still producing distributions that are centered on the desired constraints. At this setting with RecDur = 100, we only reconfigure at 5 % of the input frames, providing a significant reduction in the overhead.

To demonstrate the overall advantages of the proposed DRASTIC modes, we provide a comparison against the use of static configurations in Table 4. For the static configuration, we select the one with the maximum performance metric. For example, for the maximum image quality mode, we select the configuration that gave the maximum image quality among all video frames. Compared to the optimized static configuration, at only 5 % reconfiguration rate, we still get significant savings in dynamic power (25–37 %), bitrate (47–55 %), while reducing image quality from the maximum mode by very low percentages (3–6 %). Full reconfiguration can provide additional savings for the non-typical modes.

4.3 DRASTIC mode transition example

We consider a simple mode transition example over a video consisting of 100 frames. The transitions are provided as a means to demonstrate the capabilities of the system. The users can specify arbitrary transitions. Here, we restrict our attention to the following:

- Max im. qual. mode with high profile ($n = 1, \ldots, 25$): This mode is motivated by the need for the users to review video contents to see if there is something interesting. So, we use a maximum image quality for this initial mode.
- Typical mode with medium profile ($n = 26, \ldots, 50$): After adapting to the video, a transition to a typical mode is considered here.
- Min rate with medium profile ($n = 51, \ldots, 75$): In a limited bandwidth environment, a minimum rate mode is considered.
- Min power with low profile ($n = 76, \ldots, 100$): The need to support operations over longer periods motivates the transition to a minimum power mode as the final mode.

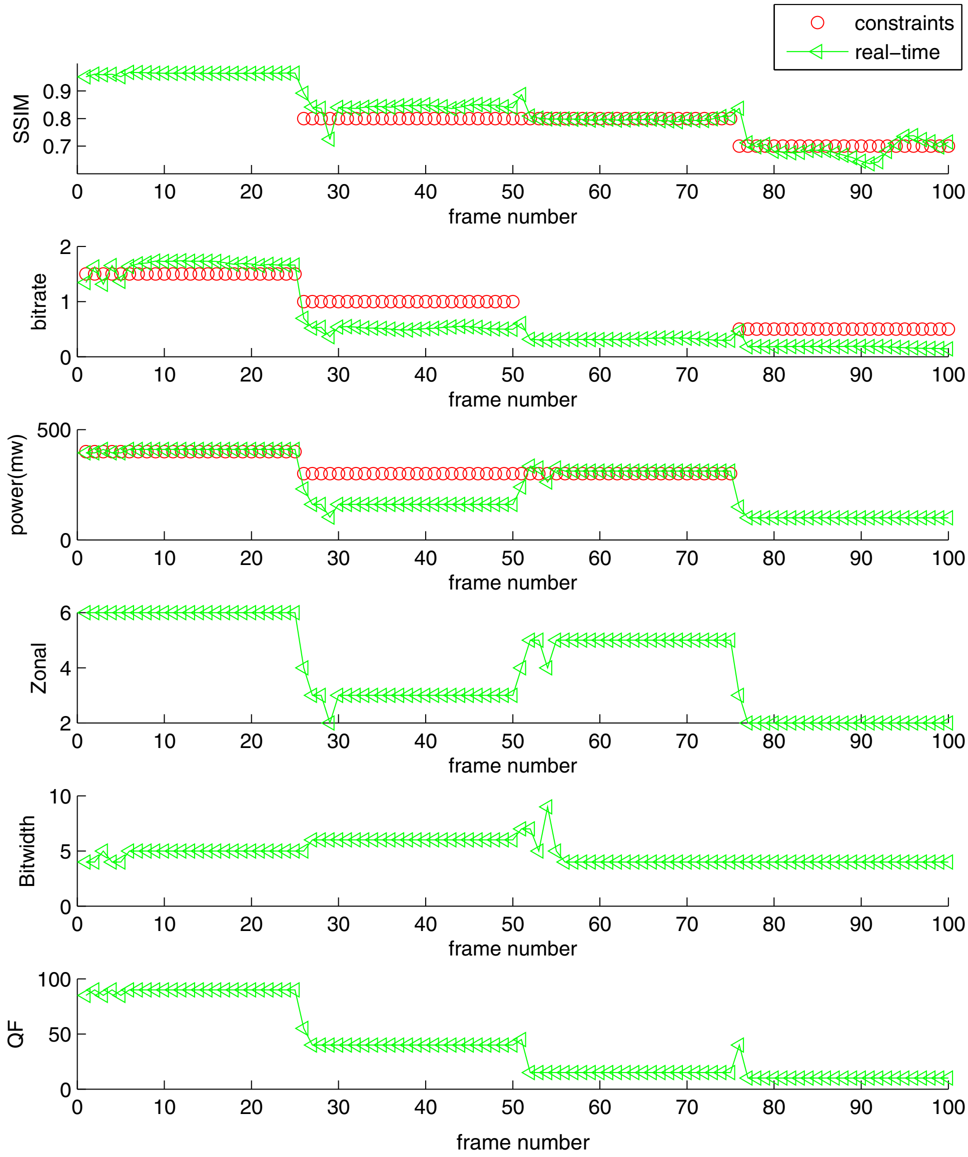
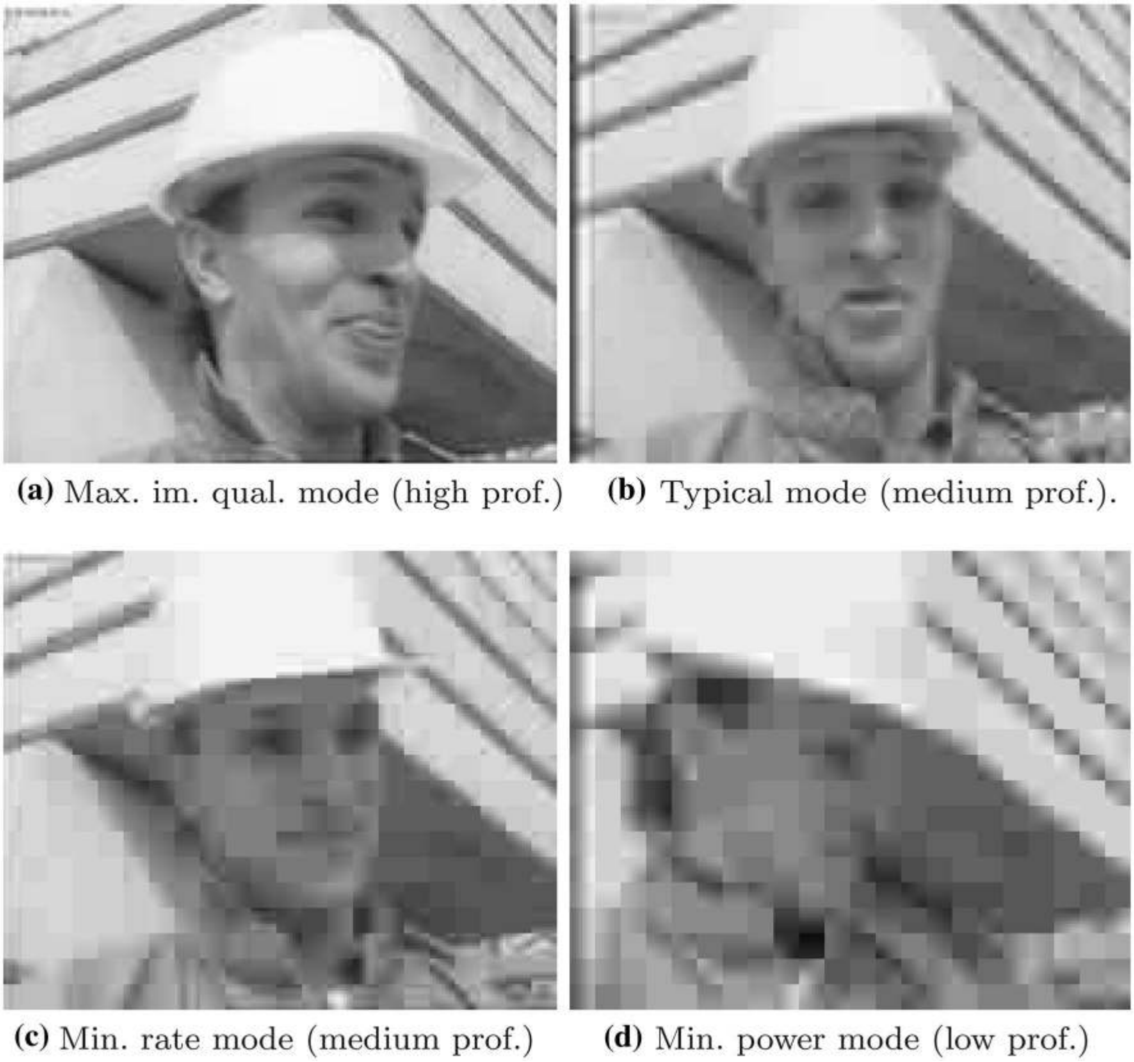


Fig. 11 DRASTIC reconfiguration results for switching between modes for the Foreman video. Here, the proposed reconfiguration settings are used (`RecC = 5, RecP = 1, RecDur = 25`)

By setting $RecDur = 25$, the proposed algorithm can respond to requirement mode transitions very rapidly. From the DRASTIC mode transitions of Fig. 9, it is clear that the dynamic reconfiguration works well. The basic idea of adjusting to the input video at the beginning of the mode does allow the system to meet the constraints. Also, from the video images of Fig. 10, we can see exceptional image quality for the maximum image quality mode (Fig. 10a), acceptable quality for the typical mode (Fig. 10b), reduced quality for the minimum rate mode (Fig. 10c), that reduces to barely acceptable quality for the minimum power mode (Fig. 10d). In terms of dynamic power consumption, it is interesting to note that the typical mode with a medium profile requires only slightly more power than the minimum power mode with a low profile.

Overall, in all of our experiments, we have found that the DRASTIC controller can adapt quickly to mode changes. After a mode change, it takes up to five frames for the DRASTIC controller adjust to meet the new constraints that come with the new mode.


(a) Max. im. qual. mode (high prof.)

(b) Typical mode (medium prof.).

(c) Min. rate mode (medium prof.)

(d) Min. power mode (low prof.)

Fig. 12 DRASTIC mode transition example results. **a** Max img qual. mode ($n = 5$): SSIM = 0.95, rate = 1.36 bps, DP = 395 mW which gives exceptional image quality while meeting the high profile constraints. **b** Typical mode ($n = 35$): SSIM = 0.84, rate = 0.51 bps, DP = 161 mW which meets all of the medium profile constraints at a much lower bitrate. **c** Min rate mode ($n = 60$): SSIM = 0.79, rate = 0.31 bps, DP = 312 mW which is right at the boundary of the image quality and dynamic power constraints (medium profile) while using significantly less bitrate. **d** Min power mode ($n = 85$): SSIM = 0.69, rate = 0.18 bps, DP = 100mW which is at the boundary of the image quality constraint for the low profile, unable to further reduce power, but still operating at a very low bitrate

5 Conclusion and future work

The paper has introduced DRASTIC modes to allow for fine optimization control for maximizing image quality, minimizing bitrate requirements, reducing dynamic power consumption, or providing a typical mode that balances constraint requirements. An efficient and scalable architecture based on the 2D DCT was used for implementing the DRASTIC modes. From the results, it is clear that the use of DRASTIC can lead to significant power and bitrate savings over optimized static approaches. Furthermore, the dynamic reconfiguration overhead can be minimized by reducing the reconfiguration rate (5 % or less).

Future work will be focused on extending the DRASTIC approach to other video processing and communications

Table 4 DRASTIC mode savings over the use of the optimized maximum setting for each mode for the nine testing videos

DRASTIC mode	Constraints profile		
	Low (%)	Medium (%)	High (%)
Min dyn. pwr (prop. rec.)	37.3	36.9	25.3
Min dyn. pwr (full rec.)	57.7	38.3	31.7
Min bitrate (prop. rec.)	51.2	46.7	55.1
Min bitrate (full rec.)	57.9	63.4	58.1
Max im. qual. (prop. rec.)	6.1	5.4	3.0
Max im. qual. (full rec.)	6.7	4.5	2.8
Typical mode (dyn. pwr, prop. rec.)	43.5	37.9	24.9
Typical mode (dyn. pwr, full rec.)	35.6	35.4	27.3
Typical mode (bitr., prop. rec.)	34.7	33.9	62.2
Typical mode (bitr., full rec.)	32.0	32.7	65.0
Typical mode (SSIM, prop. rec.)	8.3	6.1	2.4
Typical mode (SSIM, full rec.)	10.2	6.5	2.9

Here, the savings are computed as a percentage of the average performance metric. For example, for dynamic power, the percentage savings computed using $(P_{\max} - P_{\text{avg}})/P_{\text{avg}} * 100$ where $P_{\text{avg}}, P_{\max}$ are computed from the selected DRASTIC architectures. For dynamic power and bitrate constraints, higher percentages indicate higher savings. For image quality, lower percentages are preferred since they indicate that the resulting videos will be of higher quality. The proposed reconfiguration (prop. rec.) refers to `RecC = 5, RecP = 1` while full reconfiguration refers to `RecC = 100, RecP = 1`. The proposed reconfiguration requires 5 % of the overhead of the full reconfiguration. Also, note that the savings are conservative since they assume an optimal pre-selection of the static architecture

applications. Furthermore, future work will look at methods to generate constraints dynamically based on video content.

Acknowledgments This material is based upon work supported by the National Science Foundation under NSF AWD CNS-1422031.

References

1. Huang, Y.-W., Hsieh, B.-Y., Chen, T.-C., Chen, L.-G.: Analysis, fast algorithm, and vlsi architecture design for h.264/avc intra frame coder. IEEE Trans. Circuits Syst. Video Technol. **15**(3), 378–401 (2005)
2. Chen, L., Shashidhar, N., Liu, Q.: Scalable secure MJPEG video streaming. In: 2012 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 111–115 (2012)
3. Du, Q., Qin, H., Tang, J., Li, X.: Design of the arm based remote surveillance system. In: 2012 3rd International Conference on System Science, Engineering Design and Manufacturing Informatization (ICSEM), vol. 1, pp. 336–338 (2012)
4. Ko, H.-Y., Lee, J.-H., Kim, J.-O.: Implementation and evaluation of fast mobile VNC systems. IEEE Trans. Consum. Electron. **58**(4), 1211–1218 (2012)
5. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
6. Jiang, Y., Pattichis, M.: Dynamically reconfigurable DCT architecture based on bitrate, power, and image quality considerations. In: 2012 International Conference on Image Processing, pp. 2465–2468 (2012)
7. Jiang, Y., Pattichis, M.: A dynamically reconfigurable DCT architecture for maximum image quality subject to dynamic power and bitrate constraints. In: 2012 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI), pp. 189–192 (2012)
8. Xilinx, Inc.; Application Note: Virtex Series; XAPP151 (v1.7); "Virtex Series Configuration Architecture User Guide", Oct. 20, 2004; available from Xilinx, Inc., 2100 Logic Drive, San Jose, CA 95124; pp. 1–45
9. Huang, J., Parris, M., Lee, J., Demara, R.F.: Scalable FPGA-based architecture for DCT computation using dynamic partial reconfiguration. ACM Trans. Embed. Comput. Syst. **9**(1), 1–18 (2009)
10. Huang, J., Lee, J.: A self-reconfigurable platform for scalable DCT computation using compressed partial bitstreams and blockram prefetching. IEEE Trans. Circuits Syst. Video Technol. **19**(11), 1623–1632 (2009)
11. Huang, J., Lee, J.: Reconfigurable architecture for ZQDCT using computational complexity prediction and bitstream relocation. IEEE Embed. Syst. Lett. **3**(1), 1–4 (2011)
12. Llamocca, D., Pattichis, M.: A dynamically reconfigurable pixel processor system based on power/energy-performance-accuracy optimization. IEEE Trans. Circuits Syst. Video Technol. **23**(3), 488–502 (2013)
13. Kannangara, C., Richardson, I., Miller, A.J.: Computational complexity management of a real-time h.264/avc encoder. IEEE Trans. Circuits Syst. Video Technol. **18**(9), 1191–1200 (2008)
14. He, Z., Liang, Y., Chen, L., Ahmad, I., Wu, D.: Power–rate–distortion analysis for wireless video communication under energy constraints. IEEE Trans. Circuits Syst. Video Technol. **15**(5), 645–658 (2005)
15. He, Z., Cheng, W., Chen, X.: Energy minimization of portable video communication devices based on power–rate–distortion optimization. IEEE Trans. Circuits Syst. Video Technol. **18**(5), 596–608 (2008)
16. Li, X., Wien, M., Ohm, J.-R.: Rate–complexity–distortion optimization for hybrid video coding. IEEE Trans. Circuits Syst. Video Technol. **21**(7), 957–970 (2011)
17. Madisetti, A., Jr Willson, A.N.: A 100 MHz 2-d 8 × 8 DCT/IDCT processor for HDTV applications. IEEE Trans. Circuits Syst. Video Technol. **5**(2), 158–165 (1995)
18. Lee, Y.-P., Chen, T.-H., Chen, L.-G., Chen, M.-J., Ku, C.-W.: A cost-effective architecture for 8 × 8 two-dimensional DCT/IDCT using direct method. IEEE Trans. Circuits Syst. Video Technol. **7**(3), 459–467 (1997)
19. Hsiao, S.-F., Shiue, W.-R., Tseng, J.-M.: A cost-efficient and fully-pipelinable architecture for DCT/IDCT. IEEE Trans. Consum. Electron. **45**(3), 515–525 (1999)
20. Cheng, K.-H., Huang, C.-S., Lin, C.-P.: The design and implementation of DCT/IDCT chip with novel architecture. In: The 2000 IEEE International Symposium on Circuits and Systems, 2000. Proceedings ISCAS 2000 Geneva, vol. 4, pp. 741–744 (2000)
21. Hsiao, S.-F., Shiue, W.-R., Tseng, J.-M.: Design and implementation of a novel linear-array DCT/IDCT processor with complexity of order log2n. IEE Proc. Vis. Image Signal Process. **147**(5), 400–408 (2000)
22. Agostini, L., Silva, I., Bampi, S.: Pipelined fast 2d DCT architecture for JPEG image compression. In: 14th Symposium on Integrated Circuits and Systems Design, pp. 226–231 (2001)

23. Kusuma, E., Widodo, T.L Fpga implementation of pipelined 2d-DCT and quantization architecture for JPEG image compression. In: 2010 International Symposium in Information Technology (ITSim), June 2010, vol. 1, pp. 1–6 (2010)
24. Chen, W.-H., Smith, C., Fralick, S.: A fast computational algorithm for the discrete cosine transform. IEEE Trans. Commun. **25**(9), 1004–1009 (1997)
25. Xanthopoulos, T., Chandrakasan, A.: A low-power DCT core using adaptive bitwidth and arithmetic activity exploiting signal correlations and quantization. IEEE J. Solid-State Circuits **35**(5), 740–750 (2000)
26. Kim, D.W., Kwon, T.W., Seo, J.M., Yu, J.K., Lee, S.K., Suk, J.H., Choi, J.R.: A compatible DCT/IDCT architecture using hardwired distributed arithmetic. In: The 2001 IEEE International Symposium on Circuits and Systems, ISCAS 2001, May 2001, vol. 2, pp. 457–460 (2001)
27. Yu, S., Swartziander, J.: DCT implementation with distributed arithmetic. IEEE Trans. Comput. **50**(9), 985–991 (2001)
28. Chungan, P., Xixin, C., Dunshan, Y., Xing, Z.: A 250 MHz optimized distributed architecture of 2d 8 × 8 DCT. In: 7th International Conference on ASIC. ASICON '07, Oct 2007, pp. 189–192 (2007)
29. Lim, H., Yim, C., Swartzlander, Jr., E.E.: Finite word-length effects of an unified systolic array for 2-d DCT/IDCT. In: Proceedings of International Conference on Application Specific Systems, Architectures and Processors. ASAP 96, Aug 1996, pp. 35–44 (1996)
30. Chiper, D., Swamy, M., Ahmad, M., Stouraitis, T.: Systolic algorithms and a memory-based design approach for a unified architecture for the computation of DCT/DST/IDCT/IDST. IEEE Trans. Circuits Syst. I: Reg. Papers **52**(6), 1125–1137 (2005)
31. Meher, P.: Systolic designs for DCT using a low-complexity concurrent convolutional formulation. IEEE Trans. Circuits Syst. Video Technol. **16**(9), 1041–1050 (2006)
32. Hu, Y.H., Wu, Z.: An efficient cordic array structure for the implementation of discrete cosine transform. IEEE Trans. Signal Process. **43**(1), 331–336 (1995)
33. Yu, S., Jr Swartzlander, E.E.: A scaled DCT architecture with the cordic algorithm. IEEE Trans. Signal Process. **50**(1), 160–167 (2002)
34. Guo, J.-I., Ju, R.-C., Chen, J.-W.: An efficient 2-d DCT/IDCT core design using cyclic convolution and adder-based realization. IEEE Trans. Circuits Syst. Video Technol. **14**(4), 416–428 (2004)
35. Madanayake, A., Cintra, R., Onen, D., Dimitrov, V., Rajapaksha, N., Bruton, L., Edirisuriya, A.: A row-parallel 8 × 8 2-d DCT architecture using algebraic integer-based exact computation. IEEE Trans. Circuits Syst. Video Technol. **22**(6), 915–929 (2012)
36. Seshadrinathan, K., Bovik, A.: Motion tuned spatio-temporal quality assessment of natural videos. IEEE Trans. Image Process. **19**(2), 335–350 (2010)
37. Seshadrinathan, K., Soundararajan, R., Bovik, A., Cormack, L.: Study of subjective and objective quality assessment of video. IEEE Trans. Image Process. **19**(6), 1427–1441 (2010)
38. Seshadrinathan, K., Soundararajan, R., Bovik, A.C., Cormack, L.K.: A subjective study to evaluate video quality assessment algorithms. In: SPIE Proceedings Human Vision and Electronic Imaging, Jan 2010 (2010)
39. Ou, Y.-F., Ma, Z., Wang, Y.: Modeling the impact of frame rate and quantization stepsizes and their temporal variations on perceptual video quality: a review of recent works. In: 2010 44th Annual Conference on Information Sciences and Systems (CISS), pp. 1–6 (2010)
40. Hsieh, C.-H.: A zonal JPEG. In: International Conference on Information Technology: Coding and Computing. ITCC 2005, April 2005, vol. 2, pp. 756–757 (2005)
41. Park, J., Choi, J.H., Roy, K.: Dynamic bit-width adaptation in DCT: an approach to trade off image quality and computation energy. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **18**(5), 787–793 (2010)
42. Sheikh, H., Sabir, M., Bovik, A.: A statistical evaluation of recent full reference image quality assessment algorithms. IEEE Trans. Image Process. **15**(11), 3440–3451 (2006)
43. Seeling, P., Reisslein, M.: Video transport evaluation with h.264 video traces", IEEE Commun. Surv. Tutor. **14**(4):1142–1165 [Online] (2011). http://trace.eas.asu.edu/yuv/
44. Tumeo, A., Monchiero, M., Palermo, G., Ferrandi, F., Sciuto, D.: A pipelined fast 2d-DCT accelerator for fpga-based socs. In: IEEE Computer Society Annual Symposium on VLSI. ISVLSI '07, March 2007, pp. 331–336 (2007)
45. Sharma, V., Mahapatra, K., Pati, U.: An efficient distributed arithmetic based vlsi architecture for DCT. In: 2011 International Conference on Devices and Communications (ICDeCom), Feb 2011, pp. 1–5 (2011)

Yuebing Jiang received the B.S. degree in microelectronics from Xi'an Jiaotong University, Xi'an, China in 2008. He is currently working towards his Ph.D. degree in Computer Engineering from the University of New Mexico (expected to be awarded in May, 2014). He is also working at Real Communications, Inc, in charge of video engine modeling and verification. His research interests include FPGA architecture for image and video applications, dynamic partial reconfiguration in video compression, and video/image compression algorithms and standards.

Marios Pattichis (M'99–SM'06) received the B.Sc. (High Hons. and Special Hons.) degree in computer sciences and the B.A. (High Hons.) degree in mathematics, both in 1991, the M.S. degree in electrical engineering in 1993, and the Ph.D. degree in computer engineering in 1998, all from the University of Texas, Austin. He is currently a Professor with the Department of Electrical and Computer Engineering, University of New Mexico (UNM), Albuquerque. His current research interests include digital image, video processing, communications, dynamically reconfigurable computer architectures, and biomedical and space image-processing applications. Dr. Pattichis is an Associate Editor for the IEEE Transactions on Image Processing, was an Associate Editor for the IEEE Transactions on Industrial Informatics, and has also served as a Guest Associate Editor for the IEEE Transactions on Information Technology in Biomedicine. He was the General Chair of the 2008 IEEE Southwest Symposium on Image Analysis and Interpretation. He was a recipient of the 2004 Electrical and Computer Engineering Distinguished Teaching Award at UNM. For his development of the digital logic design labs at UNM he was recognized by the Xilinx Corporation in 2003 and by the UNM School of Engineering's Harrison faculty excellent award in 2006. He was a founding Co-PI of COSMIAC. At UNM, he is the director of the image and video Processing and Communications Lab (ivPCL, ivpcl.org). He has published 22 book chapters, published over 200 journal and conference papers, and graduated 12 Ph.D. students and 12 M.Sc. students.