Computational Restructuring: Rethinking Image Compression Using Resistive Crossbar Arrays

Baogang Zhang, Necati Uysal, *Graduate Student Member, IEEE*, and Rickard Ewetz, *Member, IEEE*

***Abstract*—Image compression is performed on billions of edge devices deployed in the Internet of Things (IoT). The bottleneck of the compression is the 2-D discrete cosine transform (2D DCT), which involves performing two matrix–matrix multiplications in series. Earlier studies have explored directly mapping the 2D DCT computation to emerging resistive crossbar arrays (RCAs), which promise to perform matrix–vector multiplication (MVM) with extremely small energy-delay product. The main drawback is that the series computation is inherently vulnerable to errors. In this article, we propose to fundamentally rethink how to perform image compression using RCAs. The key idea is to restructure the computation to natively match the properties of the underlying resistive hardware. This allows three of the main design steps within image compression (2D DCT, quantization, and zig-zag reordering) to be integrated into a single analog MVM operation. The integration is facilitated by the development of a 2D DCT reconstruction technique, a frequency spectrum optimization technique, and a quantization optimization technique. The techniques improve the robustness to errors, eliminates the storage of intermediate data, enables processing of small image blocks, facilitates the utilization of large-scale RCAs, and reduces the requirements on the expensive domain interfaces. Compared with the previous work, the experimental results demonstrate significant improvements in image quality while reducing power and latency with up to 62% and 21%, respectively.**

***Index Terms*—2D DCT, analog matrix–vector multiplication (MVM), image compression, in-memory computing, nonvolatile resistive technology.**

I. Introduction

IMAGE and video processing is a fundamental building block for emerging cyber-physical systems that are expected to have a broad impact on areas of our society as autonomous vehicles, sensor networks, and health care monitoring [1], [2]. Within these application domains, it is impossible to transmit all collected sensor data for analysis on a cloud servers. The acquired data is required to be preprocessed on the edge such that only the most important parts are transmitted. In particular, data in the form of images and videos is required to be compressed before transmission. The compression is required to be performed with low latency and high energy efficiency to enable real-time processing on power-constrained edge devices. Moreover, the demand for such low latency and high energy efficiency processing on the edge is expected to rapidly grow with the maturing of virtual reality and augmented reality systems [3], [4].

Image compression is performed by partitioning an image into smaller image blocks. Each image block is transformed from the spatial domain into the frequency domain using the 2-D discrete cosine transform (2D DCT) [5], [6]. Next, the obtained frequency coefficients are quantized using a quantization table. Subsequently, the coefficients are reordered (using a zig-zag pattern), encoded, and saved to a file. The bottleneck of the flow is the 2D DCT, which involves performing two matrix–matrix multiplications in series. Despite noteworthy efforts to accelerate image compression with algorithm innovations (as the fast Fourier transform [7], [8]) and custom digital hardware implementations [9], [10], the compression is still a limiting factor for applications with real-time processing requirements [11], [12]. Moreover, issue will not "automatically" be solved by further technology scaling, as the short-term performance gains are expected to be limited [13].

An arising solution to accelerating image compression is based on leveraging emerging restive technology [14]–[18] to perform highly energy-efficient in-memory computing [19]–[21]. Resistive technology has recently attracted significant interest due to that resistive devices arranged into crossbar array structures can natively perform analog matrix–vector multiplication (MVM). When the dimensions of the resistive crossbar arrays (RCAs) are scaled-up, the computation is projected to be orders of magnitude more energy-efficient than using digital hardware. Moreover, the latency is low because the entire computation is performed in a single time-step. Nevertheless, analog computation is vulnerable to various sources of errors [22], [23].

Directly mapping the expensive matrix–matrix multiplications within the 2D DCT to RCAs has been explored in [19]–[21]. The matrix–matrix multiplications were decomposed into multiple MVM operations, which were efficiently accelerated using RCAs. The limitation of this direct approach is that the computational structure has been optimized for digital hardware, which prohibits the full potential of the resistive hardware to be unleashed. In particular, the direct mapping results in poor image quality and significant amounts of redundant computation. The loss in image quality stems from that the series computation is vulnerable to errors, i.e., small errors in the output of the first

Manuscript received April 9, 2020; revised June 20, 2020; accepted July 12, 2020. Date of publication July 20, 2020; date of current version April 21, 2021. This work was supported in part by NSF under Award CCF-1755825 and Award CNS-1908471, and in part by Cyber-Florida under Grant 3910-1011-00. This article was recommended by Associate Editor A. Gamatie. *(Corresponding author: Baogang Zhang.)*

The authors are with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando FL 32816 USA (e-mail: baogang.zhang@knights.ucf.edu; necati@knights.ucf.edu; rickard.ewetz@ucf.edu).

Digital Object Identifier 10.1109/TCAD.2020.3010714

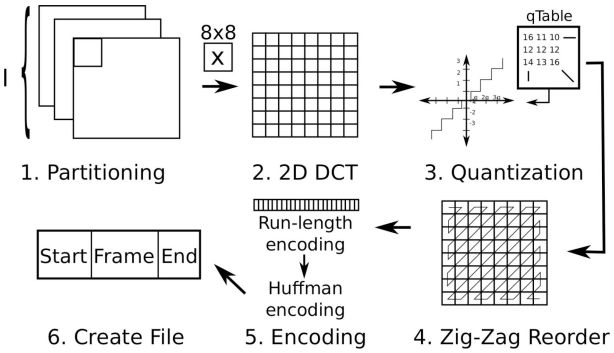


Fig. 1. Review of JPEG image compression based on 2D DCT [24].

matrix–matrix multiplication are amplified into large errors by the second matrix–matrix multiplication. The high amount of redundant computation is a result of that the RCAs were simply viewed as an accelerator for MVM operations, instead of a novel piece of hardware with unique characteristics that can be exploited to perform energy-efficient computation.

In this article, we propose to fundamentally rethink how to perform image compression using RCAs. The key idea is to restructure the computation to natively match the properties of the underlying resistive hardware. This allows three of the main design steps within image compression (2D DCT, quantization, and zig-zag reordering) to be integrated into a single analog MVM operation. The integration is facilitated by the development of: 1) a 2D DCT reconstruction technique; 2) a frequency spectrum optimization technique; and 3) a quantization optimization technique.

The 2-D reconstruction technique involves converting the series matrix–matrix multiplication into a single linear transformation (or MVM operation), where the input and output vectors are the representation of an image block in the spatial and frequency domain, respectively. The reconstruction itself reduces computation with $2\times$, improves the robustness to errors, eliminates storage of intermediate data, and allows RCAs with large dimensions to process small image blocks. Moreover, the reconstruction results in that each frequency coefficient is explicitly computed, which opens the door for the subsequent frequency spectrum optimization and quantization optimization. The frequency spectrum optimization involves reordering rows in the reconstructed DCT matrix such that the zig-zag reordering is performed for free. Next, high-frequency coefficients are pruned to allow RCAs with smaller dimensions to be utilized. Although errors are introduced by the pruning of frequency coefficients, the overall image quality is improved because smaller RCAs introduce smaller analog errors. The quantization optimization is based on configuring ADC, the domain interfaces to inherently perform the quantization step. In particular, the bit-accuracies of the ADCs are configured to mimic the quantization table. This alignment of the computational kernels with the properties of the underlying hardware results in that the requirements on the domain interfaces can be reduced to an absolute minimum, which results in further power and area savings. The experimental results demonstrate that the obtained image quality is significantly improved compared with the previous works. Moreover, the power and latency is reduced with up to 62% and 21%, respectively.

The remainder of this article is organized as follows. Preliminaries are provided in Section II. Previous work is given in Section III. The motivation for the proposed reconstruction is provided in Section IV. The details of the proposed image compression is outlined in Section V. The experimental results are given in Section VI. This article is concluded in Section VII.

II. Preliminaries

In this section, we review the basics of image compression, metrics for image compression, and the acceleration of MVM operations using emerging RCAs.

A. Review of Image Compression

Common lossy image and video compression formats as JPEG [6] and motion JPEG (MJPEG) are based on transforming an image (or video) from the spatial domain to the frequency-domain using 2D DCT and encoding the frequency coefficients. The fundamental steps of JPEG compression are partitioning, 2D DCT, quantization, zig-zag reordering, encoding, and create file, which is illustrated in Fig. 1.

The first step is to partition the input image I into small image blocks X with dimension 8×8 (or 16×16). Small block sizes are used in order to preserve high image quality. For color images, the RGB components are compressed separately. Second, each image block X is converted into the frequency domain by applying the 2D DCT as follows:

$$C = DXD' \tag{1}$$

where C is a matrix with the frequency coefficients of X. D is the standard 2D DCT matrix. Each element D_{ij} in D is defined as follows:

$$D_{ij} = \begin{cases} \frac{1}{\sqrt{N}}, & i = 1, 1 \le j \le N \\ \sqrt{\frac{2}{N}} \cos\left[\frac{\pi(2j-1)(i-1))}{2N}\right], & 2 \le i \le N, 1 \le j \le N \end{cases} \tag{2}$$

where the block size is $N\times N$. Third, the frequency coefficients are divided by each corresponding entry in a quantization table. The quantization table is designed to preserve low frequency components and discard high frequency components, as empirical studies have shown that humans are less sensitive to high frequency patterns. Moreover, the coefficients in the quantization table can be scaled with a factor q_{user} to balance image quality and compression ratio. The quantization is followed by zig-zag reordering of the frequency coefficients from into a vector (with the coefficients ordered from low to high frequency). The zig-zag reordering is performed to statistically placing the nonzero coefficients in the beginning and the zero components at the end of the vector, which allows the subsequent encoding to be performed more effectively. The encoding step consists of run-length encoding and Huffman encoding. Run-length encoding is based on storing the nonzero elements and the number of zeros that are followed by the nonzero element in the vector. In particular, each nonzero-element is stored using a triplet (r, s, v),

where r is the number of zeros before the nonzero element; v is the value of the nonzero element; and s is the number of bits required to store v. Next, the triplets are further compressed using Huffman encoding. The last step is to create file where the encoded image is appended with the information required to perform the uncompression, i.e., the quantization table and the specification of the Huffman encoding that was used. Uncompression is performed by reversing the process in Fig. 1.

B. Image Compression Performance Metrics

In this article, the quality of the image compression is measured using mean-squared errors (MSEs), peak signal-to-noise ratio (PSNR), and structural similarity index measure (SSIM) [25]. The degree of compression (or compression ratio) is measured using bits per pixel (BPP). The MSE is computed as follows:

$$\text{MSE}\left(I, \hat{I}\right) = \frac{1}{PQ} \sum_{p=1}^{P} \sum_{q=1}^{Q} \left(I_{pq} - \hat{I}_{pq}\right) \tag{3}$$

where $\hat{I}$ is the original reference image with dimensions $P \times Q$. I is the image obtained after $\hat{I}$ has been compressed and uncompressed using the flow in Fig. 1. PSNR is computed as follows:

$$\text{PSNR}\left(I, \hat{I}\right) = 20 \cdot \log_{10}\left(\frac{I_{\text{peak}}}{\sqrt{\text{MSE}\left(I, \hat{I}\right)}}\right) \tag{4}$$

where I_{peak} is the maximum pixel value. The technical details of the SSIM metric are provided in [25]. The BPP metric for an image is computed as follows:

$$\text{BPP} = \frac{\#\text{num\_bits}}{\#\text{num\_pixels}}. \tag{5}$$

Using the basic RGB representation of an image, each RGB component is represented using eight bits. Consequently, the RGB representation results in that an image is stored using 24 BPP.

C. Acceleration of MVM Using Emerging RCAs

In this section, we outline how MVM operations can be accelerated using emerging RCAs, which is shown in Fig. 2(a). In particular, we focus on MVM multiplication ($x = Dy$), where D is a DCT matrix (or a reconstructed DCT matrix $\overline{D}$ or $\widetilde{D}$ in Section V) and x and y are the input and output vectors, respectively. An RCA consists of wordlines and bitlines with a nonvolatile resistive in each cross-point. The fabrication of nonvolatile resistive devices has been explored based on resistive random access memory (ReRAM) [14], [26], spin transfer torque magnetic random access memory (STT-MRAM) [15], [27], and phase change memory (PCM) [16], [28].

Analog MVM is performed using a one-time expensive initialization phase and a fast and efficient evaluation phase. In the initialization phase, conductance values of the resistive devices are programmed to realize a conductance matrix G. In this article, the conductance matrix G is programmed to be proportional to the DCT matrix (D) in (2). Next, analog MVM is performed by passing an input vector v_{in} to the wordlines and recording an output vector v_{out} from the transimpedance amplifiers (TIAs) attached to the bitlines, where R_s is the feedback resistance of the TIAs. $v_{\text{out}}^T = v_{\text{in}}^T G R_s$ is the computation performed in the analog domain. The digital input vector x is converted into an analog input vector v_{in} using digital-to-analog converters (DACs). Similarly, the analog output vector v_{out} is converted into a digital output vector y using analog-to-digital converters (ADCs). As conductance values cannot be negative, the common differential pair approach is used to represent negative matrix values, i.e., one bitline is, respectively, used to represent the positive and negative elements for one row in a matrix. Next, the two outputs are subtracted while being converted into the digital domain using an differential ADC. Consequently, an $N \times N$ matrix is represented using an RCA with dimensions $N \times 2N$.

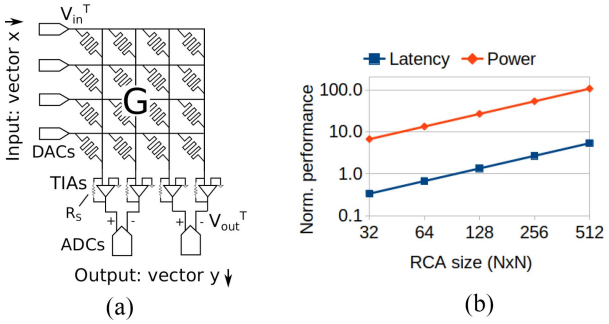


Fig. 2. (a) RCA for MVM. (b) Normalized performance of RCA hardware versus digital hardware [22].

The advantage of leveraging RCAs is that the computation is orders of magnitude more efficient than using digital hardware, which is shown in Fig. 2(b). The results in the figure are obtained with respect to custom ASIC implementation that has been optimized for high throughput. The main limitation of using RCAs to accelerate MVM operations is that the computations is vulnerable to analog errors and errors introduced by the domain interfaces. This analog errors stem from the programming accuracy of the resistive devices, the array parasitics, and random telegraph noise. The errors introduced by the domain interfaces stem from that the input and output vectors are quantized with respect to the input and output value ranges, respectively. The complexity of the domain interfaces is measured using bit-accuracy. Specifically, an ADC with a bit-accuracy of b is capable of measuring 2^b distinguishable states within a specified voltage range $[v_L, v_H]$. The domain interfaces dominate the power and area overhead of an RCA. Moreover, the overheads increase with the complexity (or bit-accuracy) of the interfaces.

III. Previous Work

In [19]–[21], image compression was accelerated by directly mapping the computation of the 2D DCT step to resistive hardware, which is illustrated in Fig. 3. The 2D DCT computation was selected because it is the bottleneck of image compression.

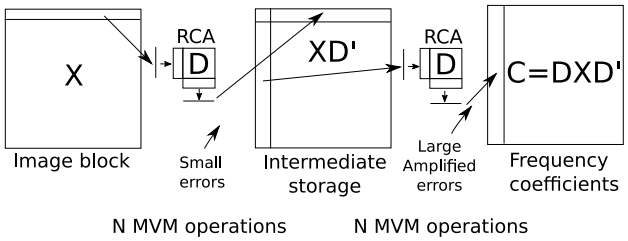


Fig. 3. Review of direct mapping in [19]–[21].

(a) (b)

Fig. 4. Image compression using (a) digital and (b) resistive hardware. The RCAs have dimensions 64×128 and parameters as in [19]–[21].

The figure shows how an image block X with dimensions $N \times N$ is processed into the corresponding frequency coefficients C using $2N$ MVM operations. The two RCAs in the figure have dimensions $N \times 2N$ as two resistive devices are used per matrix element. First, XD' is computed by passing each row from image block X as an input vector to an RCA programmed with the transpose of the DCT matrix D. The result of each MVM operation is saved as a row in a temporary storage, which is illustrated to the left in Fig. 3. Next, DXD' is computed by passing each column from the temporary storage as an input vector to a second RCA, which is programmed with the DCT matrix D. The output vector of each MVM operation is stored as a column in the final output $C = DXD'$, which is shown to the right in Fig. 3.

The two main limitations of the direct mapping are: 1) the image quality after uncompression is degraded and 2) it is difficult to scale-up the RCA dimensions, which is highly beneficial in terms of power and area. We illustrate the image quality obtained when image compression is performed by using digital and resistive hardware in Fig. 4. The images in the figure are obtained with the quantization step deactivated. It can be easily observed that the analog computation degrades the image quality. Although the image is recognizable, it is well known from adversarial learning that even minor distortions may have a devastating impact on the subsequent processing (such as classification or object detection) [24], [29]. The degraded image stems from that the series matrix–matrix multiplication is inherently sensitive to variations. Small errors introduced in the first matrix–matrix multiplication are amplified into large errors by the second matrix–matrix multiplication. Due to the inherent presence of errors and variations within analog computing, it is impossible to achieve high image quality [19]–[21]. Moreover, it is not possible to tradeoff performance (power/area) with image quality by reducing the complexity of the domain interfaces, as the uncompressed images quickly become unrecognizable.

RCA with large dimensions have to be leveraged in order to gain performance advantages (power and latency) over digital implementations, which was shown in Fig. 2(b). Consequently, large block sizes are required to be used for the compression. In [19]–[21], RCAs with dimensions of 64×128 were used to process block sizes of 64×64. In contrast, small 8×8 or 16×16 are commonly used in standard image and video compression formats. The small block sizes are needed to attain high image quality after uncompression. Nevertheless, these errors may be relatively minor for block sizes of 64×64. However, the errors are significant when the RCA dimensions are scaled to 512×512 and above.

IV. Rethinking Image Compression using RCAs

In this article, we propose to fundamentally rethink how to perform image compression using RCAs. The key idea is to restructure the computation within image compression to natively match the properties of the underlying resistive hardware, which allows full potential of the emerging hardware to be unleashed.

The proposed computational restructuring is based on two observations.

1) Any number of linear transformations performed in series can, by definition, be restructured into a single linear transformation. Consequently, the 2D DCT in (1) can be reconstructed into a single linear transformation (or MVM operation).
2) We view the quantization performed by the ADC as a "free" quantization operation that can be exploited to perform efficient computation. In contrast, quantization performed by ADCs is commonly viewed as a source of errors that should be minimized.

These two observations enable the 2D DCT step, the quantization step, and zig-zag reordering step to be integrated into a single analog MVM operation, which is illustrated in Fig. 5(a). The integration is facilitated by 2D DCT reconstruction, frequency spectrum optimization, and quantization optimization, which is shown in Fig. 5(b)–(d).

The 2D DCT reconstruction involves reconstructing the 2D DCT into a single larger linear transformation, which is illustrated in Fig. 5. The reconstruction solves the two main challenges in the previous works [19]–[21]: 1) the sensitivity to errors is reduced as the series computation is eliminated and 2) the reconstruction allows RCAs with large dimensions to efficiently process small block sizes. This translates into significant improvements in terms of latency, power, and area. The details of the reconstruction and the advantages are provided in Section V-A. The reconstruction also facilitates the explicit computation of each frequency coefficient from the spatial representation, which opens the door for frequency spectrum optimization and quantization optimization.

Frequency spectrum optimization involves first reordering the rows in the reconstructed DCT matrix to perform the zig-zag reordering for free. This arranges the frequency coefficients from low to high frequency. Next, we propose to prune the less important high frequency coefficients, which is illustrated in Fig. 5(b). The optimization interestingly improves

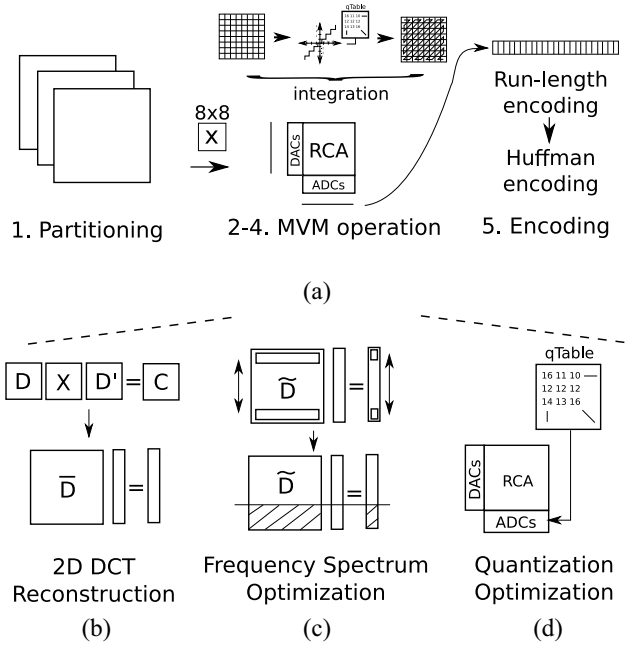


Fig. 5. (a) Flow of proposed image compression. (b) Overview of 2D DCT reconstruction, (c) frequency spectrum optimization, and (d) quantization optimization.

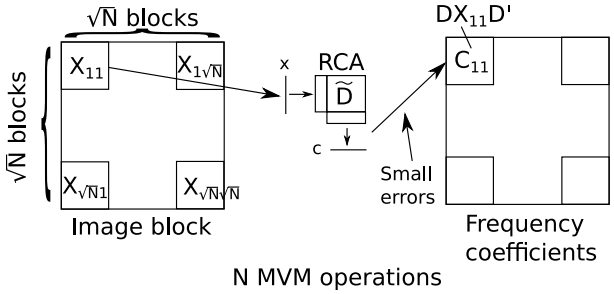


Fig. 6. Proposed 2D DCT computation using reconstructed DCT matrix.

both image quality while simultaneously reduces hardware overheads due to the array parasitics in the RCAs. The details are provided in Section V-B.

Quantization optimization is based on configuring the ADCs to perform the quantization step for free, which is shown in Fig. 5(c). In particular, the bit-accuracy of each ADC is specified based on the corresponding entry in the quantization table. This allows the requirements on the domain interfaces to be reduced to an absolute minimum, which translates into significant saving in terms of power and area. Intuitively, it is wasteful to compute each frequency coefficient with high precision and then quantize them to low precision in order to save memory. The technique is explained in Section V-C.

V. Proposed Image Compression

In this section, we provide the details of our proposed image compression consisting of 2D DCT reconstruction, frequency spectrum optimization, and quantization optimization.

A. 2D DCT Reconstruction

In this section, we explain the proposed 2D DCT reconstruction. An overview of the reconstruction is followed by an analysis of the advantages and the detailed specification of the reconstructed matrix.

1) Overview of Reconstruction: The 2D DCT reconstruction involves restructuring the series matrix–matrix multiplication in (1) into a single linear transformations as follows:

$$c = \overline{D}x \tag{6}$$

where $\overline{D}$ is a reconstructed 2D DCT matrix. x and c are columnwise vector representations of the spatial and frequency coefficients X and C, respectively. If the reconstructed DCT matrix has dimensions $N \times N$, the image block X and the block of frequency coefficients C both have dimension $\sqrt{N}$x$\sqrt{N}$.

The partitioning and the 2D DCT step performed using the proposed reconstruction is shown in Fig. 6. Given that the reconstructed DCT matrix has dimension $N \times N$, the image I is partitioned into image blocks X and frequency blocks C with dimension $\sqrt{N} \times \sqrt{N}$. In the example, it is assumed that the image I has dimension $N \times N$. Consequently, there is a total of N image and frequency blocks. Let the image and frequency blocks, respectively, be denoted X_{ij} and C_{ij} with $1 \leq i \leq \sqrt{N}$ and $1 \leq j \leq \sqrt{N}$.

The image blocks are one-by-one processed into the corresponding frequency sub-block, i.e., X_{ij} is processed into C_{ij}. Specifically, C_{ij} is obtained from X_{ij} by decomposing X_{ij} into a vector x columnwise. Next, the vector is passed to an RCA programmed with the matrix $\overline{D}$ to perform the computation $c = \overline{D}x$ using an analog MVM operation, which is shown in the middle of Fig. 6. The frequency block C_{ij} is obtained from the output vector c by organizing the elements in c into a block format.

In reality, there is no need to reorganize the vector c into the corresponding frequency block C_{ij}. Using the subsequent optimization techniques, the output vector from the RCA will be the input vector expected by the run-length encoding in the top-right of Fig. 5(a) or step 5 in Fig. 1.

2) Analysis of Reconstruction: In Table I, we analyze the number of MVM operations required to process an image of size $N \times N$ using an RCA with dimensions $N \times 2N$ (two resistive devices per matrix element). The direct mapping of DXD' to RCA hardware used in the previous works results in $2N$ MVM operations. First, N operations are used to compute XD'. Second, an additional N operations are required to compute $D(XD')$. In contrast, the proposed mapping only results in N MVM operations. The image X is decomposed into N blocks of dimension $\sqrt{N} \times \sqrt{N}$ and each block is processed using a single MVM operation. Consequently, the restructuring directly reduces the number of MVM operations by $2X$ (or from $2N$ to N), which translates into a $2X$ improvement in power, latency, and area. Moreover, no intermediate results are required to be stored. Furthermore, the robustness to errors is significantly improved because the series computation is eliminated, which results in that there are only small errors in computed frequency coefficients.

3) Reconstructed DCT Matrix $\overline{D}$: The reconstructed 2D DCT matrix $\overline{D}$ is defined with respect to a columnwise decomposition of x and c into X and C, respectively. Let the element

TABLE I
ANALYSIS OF NUMBER OF MVM OPERATIONS

	Mapping technique	
	Direct	Proposed
Block size	NxN	$\sqrt{N}$x$\sqrt{N}$
# partitions	1	N
# MVMs per partition	$2N$	1
Total # MVM operations	$2N$	N

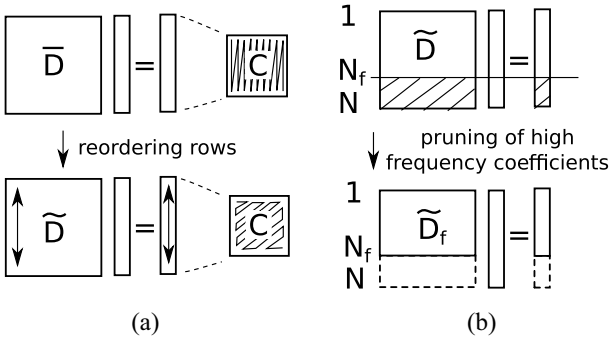


Fig. 7. (a) Frequency reordering and (b) frequency spectrum pruning.

on row i and column j in $\overline{D}$ be denoted $\overline{D}_{ij}$ and defined as follows:

$$\overline{D}_{ij} = a_p \cdot a_q \cdot \cos\left[\frac{\pi p(2t+1)}{2N}\right] \cdot \cos\left[\frac{\pi q(2r+1)}{2N}\right] \quad (7)$$

where $1 \leq i \leq N, 1 \leq j \leq N$. $q = i/N$ and $r = j/N$, where/is the integer division. $p = \text{mod}(i, N)$ and $t = \text{mod}(j, N)$ where mod is the modulus operator. The constant a_k is defined as follows:

$$a_k = \begin{cases} \frac{1}{\sqrt{N}}, & k = 0 \\ \sqrt{\frac{2}{N}}, & k \neq 0. \end{cases}$$

B. Frequency Spectrum Optimization

In this section, we explain the proposed frequency spectrum optimization. The frequency spectrum optimization consists of a frequency reordering step and frequency spectrum pruning step, which is illustrated in Fig. 7.

1) Frequency Reordering: The input to the frequency reordering step is the reconstructed DCT matrix $\overline{D}$. The output vector c contains the frequency coefficients arranged in a columnwise order with respect to the frequency block C, which is illustrated in the top-right of Fig. 7(a). However, the run-length encoding expects the frequency coefficients to be organized from low to high using the zig-zag pattern in the bottom-right of Fig. 7(a). We observe that the elements in the output vector can be reordered without any overhead by simply permuting the corresponding rows in the reconstructed DCT matrix. Consequently, the frequency reordering step consists of reordering the rows in $\overline{D}$ such that a matrix $\widetilde{D}$ is obtained where the output elements are arranged with respect to the zig-zag pattern expected by the run-length encoding. Consequently, the zig-zag reordering in Fig. 7(a) is performed for free.

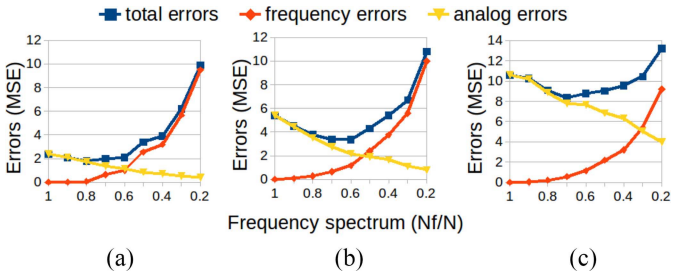


Fig. 8. The figure shows the total, analog, and frequency errors in terms of MSE with respect to only computing N_f of the N frequency coefficients. The tradeoff is shown with respect to an reconstructed DCT matrix with dimensions (a) 64×64, (b) 144×144, and (c) 256×256.

2) Frequency Spectrum Pruning: The frequency spectrum pruning involves only computing the N_f lowest frequency coefficients of an image block X. The remaining $(N - N_f)$ frequency coefficients are pruned (or set to zero). Intuitively, the hardware cost is reduced when only a subset of the frequency coefficients are computed. In digital hardware, frequency spectrum pruning techniques have demonstrated a smooth tradeoff between image quality and hardware cost. Interestingly, when the image compression is performed using RCAs, substantial reductions in overheads can be obtained while at the same time improving the image quality. The savings can be significant because an expensive ADC is used to measure each frequency coefficient.

Each row in the reconstructed DCT matrix $\widetilde{D}$ is used to compute a frequency coefficient in C. Consequently, the frequency spectrum pruning involves transforming $\widetilde{D}$ into a new matrix $\widetilde{D_f}$ with dimensions N_fxN, which is illustrated in Fig. 7(b). The transformation automatically results in that only the desired N_f frequency coefficients are computed.

3) Analysis of Frequency Spectrum Pruning: In this section, we first analyze the tradeoff between *frequency errors* and *analog errors* that is introduced by the pruning of frequency coefficients. All the errors are measured in terms of MSE. Next, we analyze the implicit tradeoff between image quality and overheads in terms of power and area.

Let the errors introduced when only a subset of the frequency coefficients are computed be called frequency errors. Intuitively, the magnitude of the frequency errors are increased when the fewer number of frequency coefficients are computed. Nevertheless, the image quality is gracefully degraded with respect to the number of discarded coefficients. This stems from that we choose to discard the highest part of the frequency spectrum. In contrast, the impact of analog errors is reduced when fewer frequency coefficients are computed. The explanation is that RCAs with smaller dimension introduce smaller analog errors because there is less IR-drop over the array parasitics [30]. Consequently, there exists a tradeoff between analog errors and frequency errors that is governed by the selected number of frequency coefficients N_f. Let the combination of the frequency errors and the analog errors be equal to the *total errors*.

In Fig. 8, we plot the errors with respect to the ratio N_f/N. The total errors and the frequency errors are obtained by compressing an image using an RCA and digital hardware

TABLE II
IMAGE QUALITY AND PERFORMANCE (POWER AND AREA) WITH RESPECT TO THE SELECTED FREQUENCY SPECTRUM N_f. THE FIGURE SHOWS THAT THE IDEAL FREQUENCY SPECTRUM IS IN THE RANGE $[1, N_f^*]$

Frequency spectrum N_f with respect to N_f^*	Image Quality	Overhead (power/area)
larger	degraded	high
equal	**highest**	medium
smaller	degraded	**small**

using a subset N_f/N of the frequency components, respectively. Next, the images are uncompressed and the errors are measured using MSE using (3). The analog errors are equal to the difference between the total errors and the frequency errors. The figure shows that the frequency errors increase with the ratio N_f/N. In contrast, the analog errors are decreased with respect to N_f/N. Consequently, the total errors first decrease until a turning point from where the error start to increase rapidly, which is shown using a blue line in Fig. 8. Let N_f^*. be the number of frequency coefficients at the turning point. In other words, N_f^* is the number of frequency coefficients that maximize the image quality.

Now, we turn our attention to analyze the tradeoff between image quality and overheads based on N_f, which is shown in Table II. The key observation is that the highest image quality is obtained when N_f is equal to N_f^*. Consequently, power and area can be saved while at the same time improving the image quality. Additional power and area savings can intuitively be obtained at the expense of image quality by setting N_f to below N_f^*. However, it is never beneficial to use $N_f > N_f^*$, as both the image quality and the power/area is worse than for N_f equal to N_f^*. In our implementation, we set N_f to estimate N_f^*. The details of the estimation of N_f^* are provided in Section VI-A2.

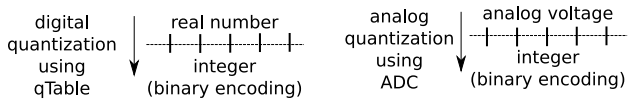
We also note that the frequency spectrum pruning can be applied using RCA with fixed dimensions. This would be realized by mapping the matrix $\widetilde{D}_f$ to the bottom-left corner of the RCA. Next, all resistive devices that are not used would be programmed to have the maximum resistance, which greatly reduces the analog errors by reducing the amount of IR-drop in the RCA.

C. Quantization Optimization

The quantization optimization consists of ADC-based quantization and hardware friendly ADC-based quantization.

1) ADC-Based Quantization: We observe that there exists an equivalency between the digital quantization performed by a quantization table and analog quantization performed by an ADC, which is shown in Fig. 9. In this section, we explain how to exploit this equivalency to perform quantization operations for free by appropriately configuring the ADCs, which intuitively allows the complexity of the ADCs to be reduced to an absolute minimum. It would obviously be very wasteful to measure the analog signal with high precision and then quantize the digital results (into low precision) to save memory (or improve the BPP metric).

The quantization step involves quantizing each frequency coefficient in a frequency block with the corresponding entry



(a) (b)

Fig. 9. (a) Quantization in the digital domain using a quantization table *qTable*. (b) Quantization in the analog domain using ADCs.

in the quantization table. The quantization is motivated by that the subsequent encoding can be more effective when most coefficients are small or preferably equal to zero. Let c_{ij}, c_{ij}^q, and q_{ij} be the frequency coefficient, the quantized frequency coefficient, and the entry in the quantization table on row i and column j with respect to a frequency block C. The quantized frequency coefficients are computed as follows:

$$c_{ij}^q = \text{round}\left(\frac{c_{ij}}{q_{ij}}\right) \tag{8}$$

where round(.) is the rounding operator. This is equivalent to defining quantization levels as follows:

$$\begin{aligned} q_k &= \frac{1}{2}q_{ij} + k\Delta q, \quad k \in \{\dots, -1, 0, 1, \dots\} \\ \Delta q &= q_{ij} \end{aligned} \tag{9}$$

where Δq is the distance between two adjacent quantization levels. q_k are the boundaries between the quantization levels. The output after quantization is k if the input number is within the range $[q_{k-1}, q_k]$.

An differential ADC with a bit-accuracy of b compares an analog input voltage with $2^b - 1$ reference voltages and output a b-bit binary number. The reference voltages are uniformly distributed between a low and high reference voltage (v_L, v_H). The reference voltage levels v_k are defined as follows:

$$\begin{aligned} v_k &= v_L + k \cdot \Delta v, \quad k \in 0 \text{ to } 2^{(b-1)} \\ \Delta v &= \frac{v_H - v_L}{2^{(b-1)}} \end{aligned} \tag{10}$$

where Δv is the distance between two adjacent voltage levels. The output from the ADC is equal to k if the reference signal is within the voltage range $[v_{k-1}, v_k]$.

It can easily be observed that there exists an equivalency between the quantization performed by an quantization table and an ADC. Therefore, by appropriately specifying the reference voltages to the ADC, the quantization operation can be performed for free. It is easy to understand that Δv is required to be specified to be proportional to Δq. The main difference between the two types of quantization is that the ADC-based quantization requires the value range of the analog input signal to be defined. We solve this issue by deriving the value range of the frequency coefficients in the digital domain. Next, the digital value range is translated into an analog value range. Given the analog value range, it is straightforward to specify the parameters v_L, v_H, and b to realize any entry in a quantization table. The technical details are specified in the Appendix. It is easy to understand that this results in that the complexity of the ADCs is reduced to an absolute minimum with respect to the specified quantization table.

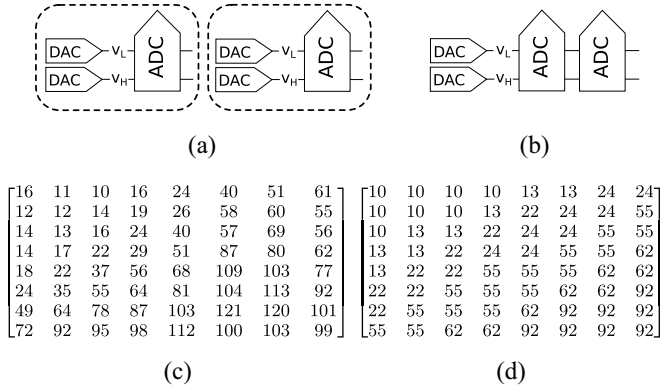


Fig. 10. (a) Two differential ADCs with individual reference voltages. (b) Two differential ADCs with shared reference voltages. (c) An ideal quantization table. (d) Shared quantization table with respect to a group size (M) of eight.

2) Hardware Friendly ADC-Based Quantization: In this section, we propose a hardware friendly implementation of the ADC-based quantization. There are two main limitations of the ADC-based quantization in the previous section. First, if the complexity of the ADCs are fabricated based on a specific quantization table, it is impossible to adjust the ADCs to obtain a higher image quality at run-time. Second, the technique requires that different reference voltages v_L and v_H are provided as an input to each differential ADC. This requires each ADC to have two internal DACs, which is illustrated in Fig. 10(a). The separate DACs naturally introduce significant power and area overheads.

We propose to circumvent these two limitations by attaching an ADC with the maximum bit-accuracy (8 bits) to each bitline. This allows the ADCs to be calibrated to deliver variable image quality. Next, groups of adjacent ADCs are set to share pairs of DACs that provide the reference voltages (v_L, v_H), which is illustrated in Fig. 10(b). Specifically, we divide the ADCs into groups of M and let the ADCs in each group share the same reference voltages. The sharing intuitively reduces the power and area overheads. On the other hand, the sharing of the reference voltages results in that the corresponding entries in the quantization table must be shared. Consequently, we convert the original quantization table into an shared quantization table, which is illustrated in Fig. 10(c) and (d). The shared quantization table is constructed by setting each group of M quantization entries to be equal to the minimum of the M entries in the original quantization table. This ensures that the image quality is not degraded by the sharing. Moreover, we propose to power gate the ADC groups to save power if the full bit-accuracy is not required. An k-bit ADC requires k clock cycles to provide the k bit output. If only p bits are required to be computed, the ADC can be gated for $k - p$ cycles. For example, let the required bit-accuracy for the low and high frequency components be 8 and 5 bits, respectively. Consequently, there is an opportunity to power gate the ADCs used to compute the high frequency coefficients for 3 cycles.

In Fig. 11, we plot the power and area of the output interface, i.e., the differential ADCs and the DACs used to provide the reference voltages to the ADCs with respect to a group size of M. In Fig. 11(a), it can be observed that the power of the DACs used to provide the reference voltages is reduced when the group size is increase. At the same time, the power consumption of the ADCs is increased due to that there are fewer opportunities for power gating. Consequently, it is not surprising that the total power is reduced until a turning point from where the power starts to increase. Let the group size that minimizes the power consumption be denoted $M*$. We illustrate a breakdown of the area in Fig. 11(b). The area of the DACs providing the reference voltages is reduced when the group size is increased. However, as the area is dominated by the ADCs (constant), the total area is only slightly reduced.

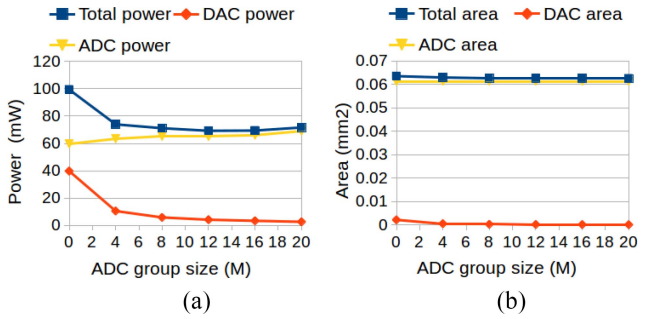


Fig. 11. The output interface (a) power and (b) area breakdown based on the ADC group size M.

TABLE III
PERFORMANCE IN POWER AND AREA WITH RESPECT TO THE GROUP SIZE (M) OF ADCS THAT SHARE REFERENCE VOLTAGES FROM THE SAME DACS. M^* IS THE GROUP SIZE THAT MINIMIZES THE POWER CONSUMPTION

Group size M with respect to $M*$	Total power	Total area
smaller	larger	high
equal	**smallest**	medium
larger	larger	**small**

We summarize our performance observations with respect to the group size M in Table III. The total power is minimum when the group size M is equal to M^*. The power consumption is degraded (or larger) if M is smaller or larger than M^*. The total area is only slightly decreased when M is increased due to the increased degree of sharing.

VI. EXPERIMENTAL RESULTS

The experimental results are obtained using a quad core 3.4-GHz Linux machine with 32 GB of memory. The images in the evaluation are subsets of the images within the Berkeley Segmentation Dataset [31] and Challenge on Learned Image Compression (CLIC) mobile and professional datasets [32]. A summary of the properties of the evaluated images is provided in Table IV.

The images in the experimental results section are obtained by performing compression using RCAs hardware using the proposed flow in Fig. 6(a) or the default flow in Fig. 1. Uncompression of the images is performed by reversing the flow using digital hardware. The compression is evaluated in terms of image quality, compression ratio, latency, power, and area. Specifically, the image quality is evaluated in terms of

TABLE IV
PROPERTIES OF THE DATA SETS OF INPUT IMAGES

Dataset (name)	Size (#images)	Average dimensions	
		rows	cols
Berkeley Segmentation	100	369	433
CLIC mobile	30	1688	1785
CLIC professional	30	1314	1961

TABLE V
PARAMETERS OF THE OF RCAS USED IN THE EXPERIMENTAL EVALUATION

Property	Value
Array block resistance	0.4Ω
Input resistance	100Ω
Output resistance	100Ω
Programmable resistance range	$[2k, 2M]\Omega$
Bit accuracy	6 bits

TABLE VI
POWER AND AREA OF CROSSBAR AND PERIPHERAL CIRCUITRY

	Bits	Power (mW)	Area (μm^2)
Differential ADC	8	1.5	1178.8
DAC	8	0.5	21.2
64x128 crossbar	n/a	4.8	400.0

MSE, PSNR, and SSIM after uncompression. Image quality degradations stem from both the quantization step and the errors introduced in the analog computation. The compression ratio is evaluated in terms of BPP after the run-length encoding. The BPP is mainly governed by the quantization table used in the quantization step.

The MVM operations that are accelerated using RCA are evaluated using circuit simulation with SPICE level accuracy. The circuit simulations explicitly capture the impact of array parasitics, programming accuracy, and domain interface quantization errors. The circuit simulation is performed using a custom simulator that exploits the sparse structure of the RCAs, which results in significant (orders of magnitude) runtime improvements over HSPICE. The accuracy has been validated to be equivalent or higher than HSPICE. Using the parameters in Table V, the experimental setup has been proven to exhibit high correlation with results obtained using hardware prototypes [20], [23]. The conductance values of the resistive devices are programmed using the program and verify techniques in [22].

The performance in terms of power, latency, and area has been obtained by carefully combining results reported in [21], [22], and [33]. The power and area for an 64×128, 8-bit DAC, and 8-bit differential ADC is shown in Table VI. The power and area for crossbars of other dimensions are obtained by scaling the crossbar parameters with the number of cells in the crossbar. The power and area of the DACs and ADCs are assumed to scale exponentially with the bit-accuracy. The latency of an MVM operation with 8-bit ADCs is 100 ns.

We evaluate the effectiveness of the different optimization techniques in Section VI-A. The performance of the entire framework is evaluated in Section VI-B.

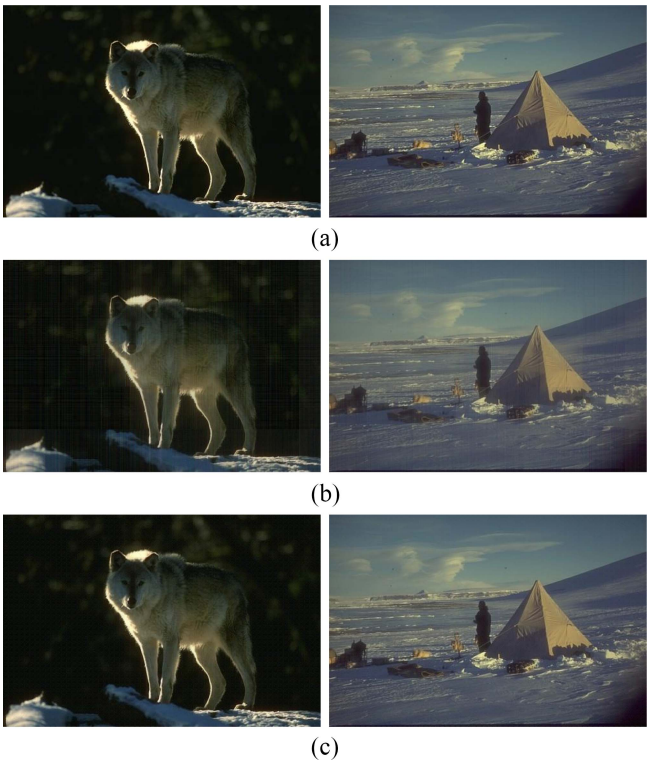


Fig. 12. (a) Reference image. (b) Images obtained using the direct mapping in [19]–[21]. (c) Images obtained using the proposed 2D DCT reconstruction.

TABLE VII
COMPARISON OF PERFORMANCE AND OVERHEADS W/O WITHOUT 2D DCT RECONSTRUCTION

2D DCT reconstruction?	Performance (power/area/latency)	Storage of intermediate data required?
No	$2X$	Yes
Yes	$1X$	No

A. Evaluation of Optimization Techniques

1) Evaluation of 2D DCT Reconstruction: We evaluate the impact of the proposed 2D reconstruction in Fig. 12. The images are obtained using RCAs with dimensions 64×128. The quantization step is disabled to demonstrate the maximum image quality that can be achieved after uncompression. The reference images are shown in Fig. 12(a). The images obtained using the direct mapping in [19]–[21] are shown in Fig. 12(b). The images obtained using the proposed 2D DCT reconstruction are shown in Fig. 12(c). The reference images are of high quality. It can be observed that the images obtained using the direct mapping are degraded by the compression. The degradation stems from the amplification of errors in the second matrix–matrix multiplication. Moreover, the impact of using the large block sizes is visible when examining the images in detail. The image obtained after the proposed 2D DCT reconstruction are just slightly degraded with respect to the reference images. This stems from that there is no amplification of errors and the reconstruction enables small block sizes can be used. In fact, the reconstruction improves the robustness to any type of errors.

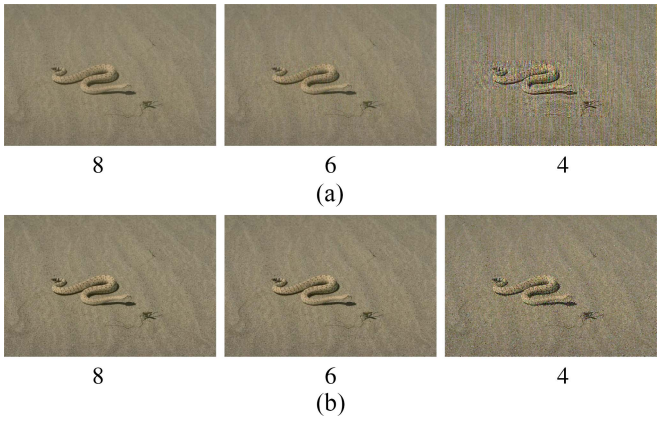


Fig. 13. Sensitivity of the image quality with respect to the bit-accuracy of the DAC and ADC domain interfaces. The images in (a) are obtained using the direct mapping and the images in (b) are obtained using the proposed 2D DCT reconstruction.

To further demonstrate the improvement in robustness to errors, the image quality with respect to the bit-accuracy of the domain interfaces is shown in Fig. 13. The figure shows that using the direct mapping in [19]–[21], the image quality is quickly degraded when the bit-accuracy is reduced from 8 to 4 bits. Using the proposed 2D DCT reconstruction, the image quality is more smoothly degraded when the bit-accuracy of the domain interfaces are reduced. Hence, a lower vulnerability to errors is demonstrated.

In Table VII, we compare the performance in terms of latency, power, and area. While excluding the extra overhead introduced by the storage of the intermediate data, the power, area, and latency is reduced with $2X$. It is easy to understand that these performance benefits are obtained because the number of MVM operations is reduced from $2N$ to N, which was analyzed in detail in Section V-A.

Based on the observed results, it is clearly advantageous to leverage the proposed 2D DCT reconstructions because it both improves the image quality and the performance in terms of power, area, and latency.

2) Evaluation of Frequency Optimization: In this section, we analyze the impact of the frequency spectrum optimization in Section V-B. The analysis is focused on the frequency spectrum pruning because the frequency reordering only avoids performing the zig-zag reordering using a specialized router. For the frequency spectrum pruning, the optimal $\hat{N}_f^*/N$ ratios are 0.8, 0.7, and 0.7 for $\widetilde{D}$ matrices with dimensions 64×64, 144×144, and 256×256, respectively. The ratios were determined by performing the image compression using RCAs with different dimensions and selecting the ratio that minimized MSE in (3). It is not surprising that the N_f^*/N ratio becomes smaller for RCAs with larger dimensions, as larger RCAs are more severely impacted by IR-drop over the array parasitics [30].

The frequency spectrum pruning is evaluated in terms of image quality in Fig. 14. The images in the left column are obtained using the full frequency spectrum. The images in the right column are obtained using a reduced frequency spectrum. The number of columns in the reconstructed DCT matrix is 64, 144, and 256 for the top, middle, and bottom row, respectively. The dimensions of the reconstructed DCT matrices ($\widetilde{D}$ or $\widetilde{D}_f$) and the MSE are shown below each image. It can be observed that image quality is gracefully degraded when RCAs with larger dimensions are utilized. The degradation stems from the IR-drop over the array parasitics. Note that the loss in image quality is observed although the state-of-the-art technique of tuning the memristors conductance values to compensate for the IR-drop is utilized [22]. It can also be seen that the frequency pruning improves the image quality (or reduces MSE). The image quality improvements are a result of that smaller analog errors are introduced when the size of the RCAs are scaled down.

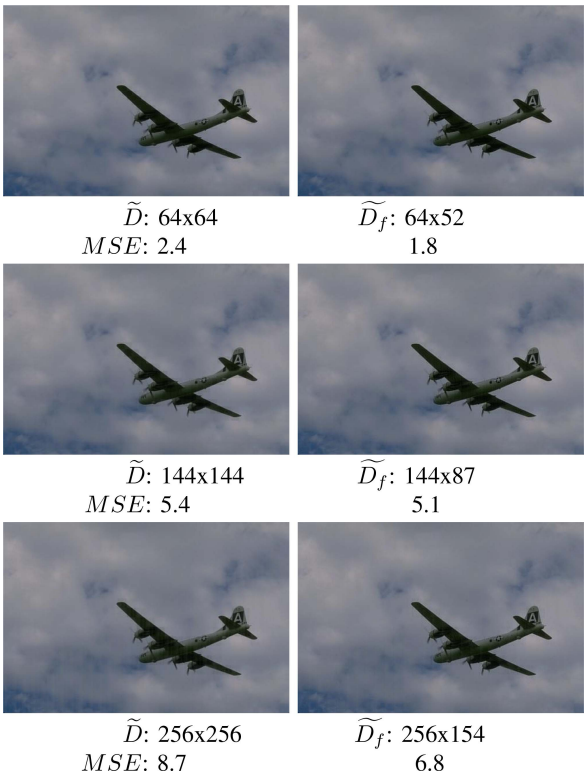
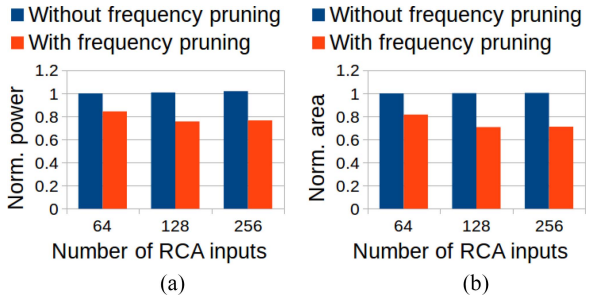


Fig. 14. The images in the left (right) column are obtained without (with) frequency spectrum optimization. The dimension of the reconstructed 2D DCT matrix and the *MSE* are shown below each figure.

Next, we focus on evaluating the frequency spectrum pruning in terms of power and area for RCAs with different dimensions. The evaluation in Fig. 15(a) and (b). The figure shows that the pruning significantly reduces the power and area while N_f is selected to maximize the image quality. The improvements are slightly smaller the N_f/N ratio because only the number of bitlines is reduced. However, large gains are still obtained because the ADCs used to measure the outputs are more expensive in terms of overheads than the DACs used to provide the inputs. For the RCAs with 144 or 256 inputs, it may be advantageous to accept a slight degradation in image quality in order to significantly reduce the hardware overheads. The tradeoff between the image quality and the number of frequency coefficients for an RCA with 144 inputs was shown in Fig. 8(b). The trends for RCAs with 256 inputs are similar.

Given that the frequency spectrum pruning provides performance benefits without any degradation in image quality,



(a) (b)

Fig. 15. Performance improvements from frequency spectrum optimization for reconstructed 2D DCT matrices with different dimensions.

it can be concluded that it is always advantageous to apply frequency spectrum optimization.

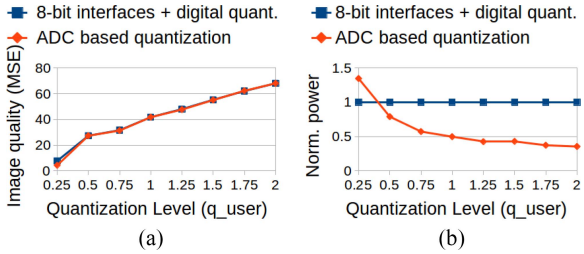
3) Evaluation of Quantization Optimization: In this section, we evaluate the quantization optimization in Section V-C.

The ADC-based quantization is examined in Fig. 16. We compare the proposed ADC-based quantization with using 8-bit ADCs and performing digital quantization. The evaluation is performed in terms of image quality with respect to q_{user} in Fig. 16(a). Recall that the quantization table is scaled with q_{user} to balance compression ratio with image quality. The figures shows that the MSE for both methods is correlated with q_{user}. For q_{user} larger than 0.25, the MSE is similar for both methods because the overall errors are dominated by the quantization specified by the quantization table. However, for q_{user} equal to 0.25, the ADC-based quantization results in smaller errors. This stems from that the 8-bit ADCs introduce larger errors than the digital quantization for small values of q_{user}. The ADC-based quantization would use ADCs with a bit-accuracy higher than 8 to circumvent this to occur. We evaluate the normalized power with respect to q_{user} in Fig. 16(b). The figure shows that the power consumption is constant when 8-bit ADCs are used and quantization is performed in the digital domain. In contrast, the power consumption of ADC-based quantization is correlated with the value of q_{user}. This is easy to understand because when q_{user} is equal to 1, the ADC-based quantization utilizes 22/10/12/8[1] ADCs with a bit-accuracy of 5/6/7/8, respectively.

In summary, the ADC-based quantization is quality configurable, i.e., the effort in power is proportional with the desired image quality. With respect to the 8-bit ADC baseline, power is saved when q_{user} is set equal or greater than 0.5 by down sizing the bit-accuracy of the ADCs. The power saving are obtained without degrading the image quality in terms of MSE. In particular, the figure shows that the technique is able to save up to 60% of the total power for larger values of q_{user}. For q_{user} equal to 0.25, the image quality is improved at the expense of increasing the power consumption by sizing up some ADCs beyond the 8-bit baseline. The main limitation of the ADC-based quantization is that q_{user} is restricted to a single at the time of fabrication.

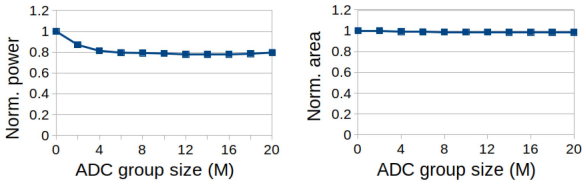
The hardware friendly ADC-based quantization is evaluated in Fig. 17. The parameter q_{user} is set to 1 in the evaluation. The technique allows the domain interfaces to be configured

[1]Frequency pruning is assumed to be used, i.e., there are only 52 ADCs.



(a) (b)

Fig. 16. Comparison between proposed ADC-based quantization and using 8-bit ADCs and performing digital quantization. The comparison is evaluated in terms of MSE in (a) and power in (b).



(a) (b)

Fig. 17. Evaluation of group size selection (M) in terms of (a) power consumption and (b) area.

with respect to the image quality at run-time. The normalized performance in terms of power and area is evaluated with respect to the group size in the figure. The group size M refers to the number of ADCs that share the same reference voltages v_L and v_H. In Fig. 17(a), it can be observed that the minimum power is achieved for M equal to 14. This stems from that M equals 14 strikes a perfect balance between the power reductions obtained from the sharing of the reference voltages and the power savings obtained from the power gating. The total RCA power is reported in Fig. 17(a). The power savings are smaller than in Fig. 11 where only the power of the output interface was reported. We evaluate the total area with respect to the group size in Fig. 17(b). It can be observed in the figure that only minor savings in terms of total area are obtained with a higher degree of sharing. This stems from that the ADCs dominate the area of each RCAs. Despite that the minimum power is obtained for a group size of 14, it may be more practical to utilize a group size of 8, which would ease delivering the reference signals to the ADCs. The majority of the saving in terms of power consumption are anyways achieved.

B. Evaluation of Proposed Image Compression

In this section, we evaluate the proposed image compression as a whole and provide comparisons with previous studies. We perform the evaluation in terms of image quality, compression ratio, latency, power, and area in Table VIII. Note that a lower MSE indicates higher image quality while a higher PSNR or SSIM indicates higher image quality. The reported latency in the table is the average time for processing an image from the respective datasets. For the proposed image compression, this is equal to number of image blocks multiplied with 100 ns. In the table, we evaluate six different methodologies to clearly demonstrate the effectiveness of the proposed techniques. "Ideal" denotes the performance obtained using floating point computation in digital hardware. This method should be viewed as a reference point (or upper bound) on the

TABLE VIII
COMPARISON OF DIFFERENT IMAGE COMPRESSION TECHNIQUES

Name	Method	Image quality			Compression	Latency	Power	Area
		(MSE)	(PSNR)	(SSIM)	(BPP)	(ms)	(mW)	(mm^2)
Berkeley segmentation	Ideal	73.7	30.3	0.930	3.6	'-'	'-'	'-'
	D [20], [21]	223.5	25.1	0.757	6.3	0.61	140.8	0.077
	D-P [19]	223.5	25.1	0.757	6.3	0.31	281.6	0.154
	R	89.1	29.4	0.908	3.2	0.25	140.8	0.077
	RF	84.8	29.6	0.913	3.3	0.25	121.8	0.063
	RFQ	83.2	29.7	0.916	3.3	0.25	107.9	0.063
CLIC mobile	Ideal	22.3	35.6	0.957	2.0	'-'	'-'	'-'
	D [20], [21]	140.7	26.8	0.742	3.8	9.59	140.8	0.077
	D-P [19]	140.7	26.8	0.742	3.8	4.80	281.6	0.154
	R	26.9	34.7	0.948	1.8	4.65	140.8	0.077
	RF	25.9	34.9	0.949	1.9	4.65	121.8	0.063
	RFQ	24.9	35.1	0.952	1.9	4.65	107.9	0.063
CLIC professional	Ideal	52.2	32.1	0.938	2.9	'-'	'-'	'-'
	D [20], [21]	213.8	25.0	0.689	6.0	8.38	140.8	0.077
	D-P [19]	213.8	25.0	0.689	6.0	4.19	281.6	0.154
	R	63.7	31.1	0.922	2.6	4.07	140.8	0.077
	RF	60.7	31.3	0.925	2.6	4.07	121.8	0.063
	RFQ	59.3	31.4	0.928	2.7	4.07	107.9	0.063
Norm.	Ideal	0.88	1.02	1.01	1.08	'-'	'-'	'-'
	D [20], [21]	4.29	0.79	0.75	2.17	2.06	1.30	1.23
	D-P [19]	4.29	0.79	0.75	2.17	1.03	2.61	2.45
	R	1.08	0.99	0.99	**0.97**	**1.00**	1.30	1.23
	RF	1.03	0.99	**1.00**	0.99	**1.00**	1.13	**1.00**
	RFQ	**1.00**	**1.00**	**1.00**	1.00	**1.00**	**1.00**	**1.00**

image quality that can be achieved. The "D" method stands for the direct mapping in [20] and [21]. The "D-P" method stands for the D method but with an implementation that is pipelined to maximize throughput [19]. The "R" method denotes our framework with only the 2D DCT reconstruction applied. The "RF" method indicates the R method extended with frequency spectrum optimization. The "RFQ" method is the RF method extended with the hardware friendly quantization. The normalized performance of the different methods is shown in bold at the bottom of the table. For all the methods, we assume that the RCAs are fabricated with an ADC group sharing size of 8. The RCAs have a dimension of 64×128 and 64×104 with and without frequency pruning, respectively. Quantization is performed with respect to the quantization table in Fig. 10(c).

First, we evaluate the D method with respect to the Ideal method. The table shows that the D method has $4.33\times$ higher MSE and 23% and 26% smaller PSNR and SSIM than the ideal method. The compression rate is $2.0X$ worse in terms of BPP. The degraded image quality stems from that errors are introduced when the RCA are leveraged to perform the MVM operations. We believe that the worse compression ratio stems from that the errors introduce additional nonzero frequency coefficients. Every nonzero coefficient requires a minimum of 9 bits to be stored. Compared with the D method, the D-P method achieves the exact same performance in terms of image quality and compression. However, the latency is about two times lower and the power and area is two times higher due to a parallel implementation.

Compared with the D and D-P method, the R method improves MSE, PSNR, and SSIM with 25%, 26%, and 36%, respectively. The degree of compression is improved with 46.0%. The improvements in image quality stem from that the series matrix–matrix multiplication is circumvented by the computational reconstruction. The BPP is reduced due to the improved robustness to variations. Compared with the D method, the latency is reduced with 51%. Compared with the D-P method, power and area are reduced with 50%. The R method has slightly smaller (3%) average latency than the D-P method because the image block size is reduced from 64×64 to 8×8. Consequently, less padding is required to make the image dimensions match a multiple of the block size dimensions, i.e., the amount of redundant computation is reduced.

Compared with the R method, the RF method improves MSE, PSNR, and SSIM with 4%, 0.6%, and 0.2%, respectively. The improved image quality stems from that the frequency pruning reduces the amount of errors introduced in the analog computation. Recall that the frequency pruning reduces the dimension of the RCA, which results in that the negative impact of IR-drop is reduced. The (2%) increase in compression ratio may stem from that the image quality is improved. The power consumption is reduced with 13.2% because RCAs with 20.0% fewer bitlines are utilized.

Compared with the RF method, the RFQ method results in similar performance in terms of image quality and compression. However, the power consumption is reduced with 10.7% on the average. The savings stem from that the RF method uses 8-bit ADCs and performs quantization in the digital domain. The RFQ method performs the quantization using the ADCs, which allows the 8-bit ADCs used to compute the high frequency coefficients to be power gated in a few clock cycles.

In summary, the proposed methods result in that image compression can be performed using RCAs while only slightly degrading the image quality compared with digital hardware. The RFQ method is compared with the Ideal method in terms of image quality in Fig. 18. Despite that the MSE is slightly higher and the PSNR and SSIM are a bit lower, the image quality is very similar to the human eye. Compared with the

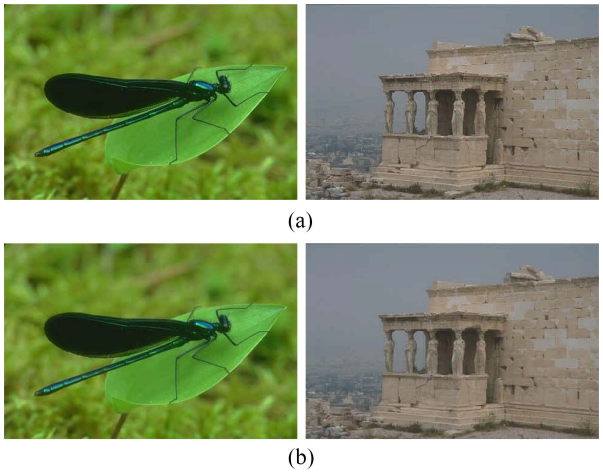


Fig. 18. Comparison of image quality obtained using (a) digital hardware and (b) resistive hardware. Quantization is performed using the table in Fig. 10.

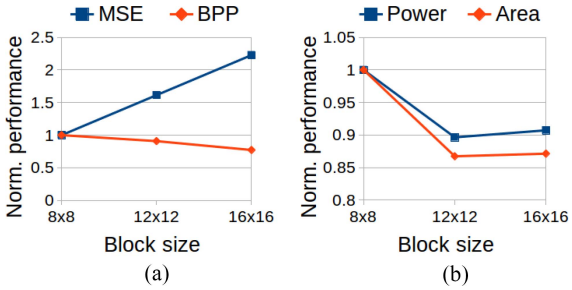


Fig. 19. The image quality and compression ratio is evaluated with respect to the processed block size in (a). The normalized power and area with respect to the block size is evaluated in (b).

previous work in [19]–[21], the image quality is improved while at the same time reducing latency and power with 51% and 24% or 3% and 61%, respectively. The benefits are obtained by reconstructing the compression such that the dominating computational kernels are aligned with the properties of the underlying hardware.

Next, we focus on evaluating the proposed compression with respect to the block sizes that are processed. We compare the normalized performance in terms of normalized MSE, BPP, power, and area in Fig. 19. A uniform quantization table consisting of only 10 is used for all the block sizes. It can be observed that the MSE is degraded and the BPP is improved when the block size is scaled up in Fig. 19(a). The power and area performance is shown in Fig. 19(b). The power and area performance is close to constant with respect to the block size. This stems from that an RCA used to process 16×16 blocks has $4X$ larger domain interfaces. At the same time, it processes a $4X$ larger block size. The power and area saving obtained when the block size is scaled from 8×8 to 12×12 stems from that additional frequency coefficients can be pruned. Nevertheless, compared with digital hardware, it is highly advantageous that the computational effort is constant (at worst) with respect the block size. The digital computational effort of 2D DCT is obviously not constant with respect to the block size.

Compared with performing image processing with digital hardware, we estimate the computation to be at least $44X$ more energy efficient. A detailed comparison between an RCA hardware prototype and an application-specific integrated circuit (ASIC) was performed in [21] and [34]. The study reported a $17X$ improvement in energy efficiency with obtaining similar image quality. The techniques proposed in this article further improves the energy efficiency with $2.61X$. Moreover, the image quality is improved at the same time. Therefore, the case for leveraging emerging resistive hardware is even more compelling than before.

VII. Summary and Future Work

Computation of 2D DCT is the bottleneck of real-time image and video compression. An arising solution to scalably enable 2D DCT to be performed on edge-devices is to accelerate the computation using emerging RCAs. In this article, we proposed to rethink how to perform image compression using emerging hardware by reconstructing the computational kernels to be aligned with the underlying properties of the resistive hardware. This results in significant improvements in image quality, robustness to errors, power, area, and latency. In our future work, we will investigate techniques of further mitigating the errors occurring in the analog domain. Moreover, we also plan to collaborate with device level researches to evaluate the proposed techniques using hardware prototypes.

Appendix

In this Appendix, we provide the details for how to specify parameters v_L, v_H, and b with respect to a quantization factor q in Section V-C. Let q be an arbitrary entry in a quantization table. This is performed by first determining the number of states that are required to be captured, which directly determines the bit-accuracy b of the ADCs. Next, we calculate the analog voltage step Δv that corresponds to a digital quantization step Δq. Lastly, the reference voltages (v_L, v_H) are specified based on b and Δv.

The value rage $[c^-, c^+]$ for each frequency coefficient can be determined based on the value range of the input vector x and the values in the reconstructed DCT matrix $\widetilde{D_f}$. The value range of x is $[-127.5, 127.5]$. This allows the value range $[c^-, c^+]$ to be easily computed by taking the absolute value of each entry in the reconstructed DCT matrix and computing the corresponding row-wise sum. The result is subsequently multiplied with -127.5 or 127.5 to obtain c^- or c^+, respectively. Next, the number of positive states (m^+) and negative states (m^-) are computed as follows:

$$m^+ = \text{round}\left(\frac{c^+}{q}\right) \tag{11}$$

$$m^- = \text{round}\left(\frac{c^-}{q}\right) \tag{12}$$

where q is the corresponding quantization factor. Consequently, the total number of states is $m^+ + m^- + 1$, where the 1 corresponds to the zero state. Next, the number of required bits is computed as follows:

$$b = \text{ceil}\left(\log_2\left(m^+ + m^- + 1\right)\right) \tag{13}$$

where ceil(.) is the ceiling operator. $\log_2(.)$ is the logarithm with respect to base 2. Now we turn our attention to computing

the voltage step v that corresponds to Δq, which is performed as follows:

$$\Delta v = q\frac{v_{\max}}{c^+} \cdot \alpha_{vi} \cdot R_s \tag{14}$$

where $v_{\max}$ is the maximum input voltage provided by the DAC. The DAC is assumed to provide input voltages in the range $[-v_{\max}, v_{\max}]$; α_{vi} is the scaling between an input voltage into an output current realized by the conductance matrix G; R_s is the feedback resistance of the TIAs.

Lastly, we turn our attention to specifying the reference voltages. The voltages (v_L, v_H) are defined as follows:

$$v_L = \left(m^- - \frac{1}{2}\right) \cdot \Delta v \tag{15}$$

$$v_H = v_L + \left(2^b - 1\right) \cdot \Delta v. \tag{16}$$

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswamia, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, "Wireless sensor networks: A survey on recent developments and potential synergies," *J. Supercomput.*, vol. 68, no. 1, pp. 1–48, Apr. 2014.

[3] J. Carmigniani and B. Furht, "Augmented reality: An overview," in *Handbook of Augmented Reality*. New York, NY, USA: Springer, 2011, pp. 3–46.

[4] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic, "Augmented reality technologies, systems and applications," *Multimedia Tools Appl.*, vol. 51, no. 1, pp. 341–377, Jan. 2011.

[5] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, no. 3, pp. 349–389, Mar. 1981.

[6] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 19–34, Feb. 1992.

[7] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*. Philadelphia, PA, USA: Soc. Ind. Appl. Math., 1992.

[8] F. Zhang, Z. Geng, and W. Yuan, "The algorithm of interpolating windowed FFT for harmonic analysis of electric power system," *IEEE Trans. Power Del.*, vol. 16, no. 2, pp. 160–164, Apr. 2001.

[9] A. Pedram, J. McCalpin, and A. Gerstlauer, "Transforming a linear algebra core to an FFT accelerator," in *Proc. IEEE 24th Int. Conf. Appl. Spec. Syst. Archit. Process. (ASAP)*, Washington, DC, USA, Jun. 2013, pp. 175–184.

[10] E. Konguvel and M. Kannan, "A survey on FFT/IFFT processors for next generation telecommunication systems," *J. Circuits Syst. Comput.*, vol. 27, no. 03, 2018, Art. no. 1830001.

[11] O. Fialka and M. Cadik, "FFT and convolution performance in image filtering on GPU," in *Proc. 10th Int. Conf. Inf. Vis. (IV'06)* London, U.K., Aug. 2006. pp. 609–614.

[12] P. Karas and D. Svoboda, *Algorithms for Efficient Computation of Convolution*. London, U.K.: IntechOpen, 2013.

[13] (2015). *ITRS*. [Online]. Available: https://www.semiconductors.org/

[14] H-S. Philip Wong *et al.*, "Metal–oxide RRAM," *Proc. IEEE*, vol. 100, no. 6, pp. 1951–1970, Jun. 2012.

[15] S. Parkin, X. Jiang, C. Kaiser, A. Panchula, K. Roche, and M. Samant, "Magnetically engineered spintronic sensors and memory," *Proc. IEEE*, vol. 91, no. 5, pp. 661–680, May 2003.

[16] B. G. Johnson and C. H. Dennison, "Phase change memory," U.S. Patent 6 791 102, 2004.

[17] M. Saremi, "Carrier mobility extraction method in ChGs in the UV light exposure," *Micro Nano Lett.*, vol. 11, no. 11, pp. 762–764, Nov. 2016.

[18] M. Saremi, "A physical-based simulation for the dynamic behavior of photodoping mechanism in chalcogenide materials used in the lateral programmable metallization cells," *Solid-State Ionics*, vol. 290, pp. 1–5, Jul. 2016.

[19] M. Hu and J. P. Strachan, "Accelerating discrete fourier transforms with dot-product engine," in *Proc. IEEE Int. Conf. Rebooting Comput. (ICRC)*, San Diego, CA, USA, 2016, pp. 1–5.

[20] C. Li *et al.*, "Analogue signal and image processing with large memristor crossbars," *Nat. Electron.*, vol. 1, no. 1, pp. 52–59, 2018.

[21] C. Li *et al*., "Large memristor crossbars for analog computing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Florence, Italy, 2018, pp. 1–4.

[22] M. Hu *et al*., "Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication," in *Proc. 53nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Austin, TX, USA, 2016, pp. 1–6.

[23] M. Hu *et al.*, "Memristor-based analog computation and neural network classification with a DPE," *Adv. Mater.*, vol. 30, Mar. 2018, Art. no. 1705914.

[24] Z. Liu *et al.*, "DeepN-JPEG: A deep neural network favorable jpeg-based image compression framework," in *Proc. 55th Annu. Design Autom. Conf. (DAC)*, 2018, p. 18.

[25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[26] H. Y. Lee *et al.*, "Low power and high speed bipolar switching with a thin reactive Ti buffer layer in robust HfO_2 based RRAM," in *Proc. IEEE Int. Electron Devices Meeting*, San Francisco, CA, USA, 2008, pp. 1–4.

[27] Y. Huai, "Spin-transfer torque MRAM (STT-MRAM): Challenges and prospects," *AAPPS Bull.*, vol. 18, no. 6, pp. 33–40, 2008.

[28] H.-S. P. Wong *et al.*, "Phase change memory," *Proc. IEEE*, vol. 98, no. 12, pp. 2201–2227, Dec. 2010.

[29] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," 2017. [Online]. Available: https://arxiv.org/abs/1611.01236.

[30] B. Liu *et al.*, "Reduction and IR-drop compensations techniques for reliable neuromorphic computing systems," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, 2014, pp. 63–70.

[31] (2007). *The Berkeley Segmentation Dataset and Benchmark*. [Online]. Available: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

[32] *Challenge*. Accessed: 2020. [Online]. Available: https://www.compression.cc/challenge/

[33] A. Shafiee *et al.*, "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *ACM SIGARCH Comput. Archit. News*, vol. 44, no. 3, pp. 14–26, 2016.

[34] P. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, and W. D. Lu, "Sparse coding with memristor networks," *Nat. Nanotechnol.*, vol. 12, no. 8, pp. 784–789, 2017.

Baogang Zhang received the B.S. and M.S. degrees in electrical engineering from the Florida Institute of Technology, Melbourne, FL, USA, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, USA.

His research interests are focused on in-memory computing using emerging technology and the security of nonvolatile memory systems.

Necati Uysal (Graduate Student Member, IEEE) received the B.S. degree in electrical and electronics engineering from the University of Gaziantep, Gaziantep, Turkey, in 2013, and the M.S. degree in electrical engineering from the University of Central Florida, Orlando, FL, USA, in 2017, where he is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department.

His research is focused on physical design of VLSI circuits and computer-aided design for emerging technologies.

Rickard Ewetz (Member, IEEE) received the M.S. degree in applied physics and electrical engineering from Linköping Universitet, Linköping, Sweden, in 2011, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2016.

He is currently an Assistant Professor with the Electrical and Computer Engineering Department, University of Central Florida, Orlando, FL, USA. His research interests include physical design and computer-aided design for in-memory computing using emerging technologies.

Dr. Ewetz has one best paper nomination from ASP-DAC 2019.