A Light-Powered Smart Camera With Compressed Domain Gesture Detection

Anvesha Amaravati, Shaojie Xu, Ningyuan Cao, *Student Member, IEEE*, Justin Romberg, and Arijit Raychowdhury, *Senior Member, IEEE*

***Abstract*— This paper presents an ultralow power smart camera with gesture detection. Low power is achieved by directly extracting gesture features from the compressed measurements, which are the block averages and the linear combinations of the image sensor's pixel values. We present two classifier techniques to allow low computational and storage requirements. The system has been implemented on an analog devices BlackFin ULP vision processor. By enabling ultralow energy consumption, we demonstrate that the system is powered by ambient light harvested through photovoltaic cells whose output is regulated by TI's dc–dc buck converter with maximum power point tracking. Measured data reveals that with only 400 compressed measurements (768× compression ratio) per frame, the system is able to recognize key wake-up gestures with greater than 80% accuracy and only $95mJ$ of energy per frame. Owing to its fully self-powered operation, the proposed system can find wide applications in "always-on" vision systems, such as in surveillance, robotics, and consumer electronics with touch-less operation.**

***Index Terms*— Activity recognition, sensor systems, pattern recognition.**

I. Introduction

HUMAN machine interfaces continue to make remarkable advances as we enable new modalities of interaction and control. Beyond the traditional keyboard and mice, such smart devices enable advanced user interfaces, like voice command and control, camera and GPS based sensors and interfaces, as well as touch screens and displays. One key requirement of such interfaces is the "always on" capability, where the sensor needs to be perpetually vigilant and look out for user commands. Enabling such a capability, typically requires prohibitively high power consumption. In particular, in camera based systems, the problem is exacerbated by the high power consumption of the pixel arrays and the interface circuits. The power cost of continuously capturing and analyzing videos is so high that most systems require physical input from the user before accepting commands. To address this issue, a "wake up" camera front-end allows a sensor node to continuously acquire videos and monitor for a trigger that will wake up the back-end when necessary, thus enabling exciting new usage models. A promising "wake up" modality in "always on" cameras is hand gestures, which is presented in this paper.

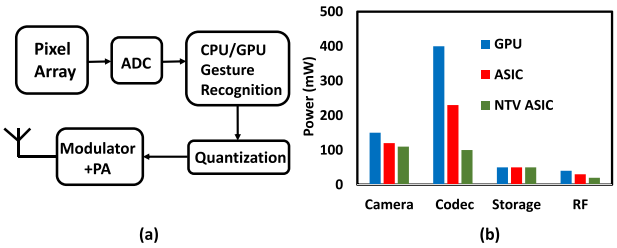


Fig. 1. (a) Typical gesture recognition system (b) Computing power across different components [5].

Traditional gesture recognition systems are power inefficient and run on batteries or even AC power supplies [1]–[3]. A non-appearance based low-power gesture recognition system [4] has been proposed. Relying on wireless signal trigger by the gestures, the performance of this system is limited by the short sensing distance and is sensitive to environmental noise. We, therefore, focus on reducing the energy consumption of the more robust appearance-based approach. With rapid advances in energy harvesting, it is enticing to think about a camera front-end which is powered by photo-voltaic cells (PV), thus paving the way for light-powered, smart, "always on" cameras.

Fig. 1 (a) shows the typical gesture recognition system. It includes pixel array followed by an analog to digital converter. Pixel array provides voltage corresponding to the intensity of the image. Digitized intensity values are fed to CPU/GPU to perform gesture recognition. If the gesture of interest is found, it is quantized and transmitted to back end server for further processing. Fig. 1 (b) shows typical power numbers for different blocks involved in gesture recognition and transmission . Digital processors like CPU/GPU dominates the power of the entire system.

Fig. 2 illustrates the landscape of self-powered sensor nodes and shows the power requirement of various electronic devices and the amount of power that can be harvested by various sources like solar energy, thermal, mechanical etc. In particular, for image/video processing and classifications we need high computational power. CPU, GPU and FPGA's are typically used to perform gesture recognition and object classification on video data [1], [6]. However, for "always on" front-ends where the objective is trigger identification and not continuous gesture recognition, high performance (and hence high power) are not optimal. Instead vision specific MCUs and

Manuscript received February 1, 2017; revised May 2, 2017 and June 29, 2017; accepted July 20, 2017. Date of publication July 27, 2017; date of current version October 24, 2018. This work was supported by the generous gift from Intel Corporation. This paper was recommended by Associate Editor W. Zuo. *(Corresponding author: Anvesha Amaravati.)*

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: aamaravati3@gatech.edu; kyle.xu@gatech.edu; ningyuan.cao@gatech.edu; justin@ece.gatech.edu; arijit.raychowdhury@ece.gatech.edu).

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

Digital Object Identifier 10.1109/TCSVT.2017.2731767

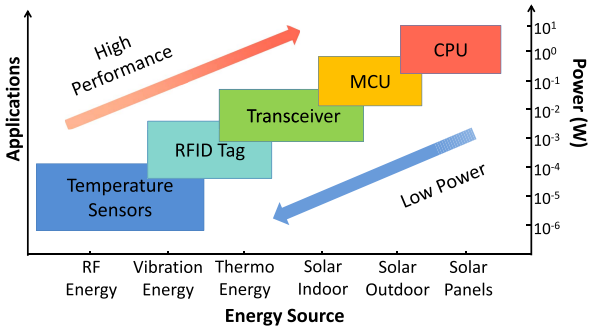


Fig. 2. The landscape of self-powered electronic devices.

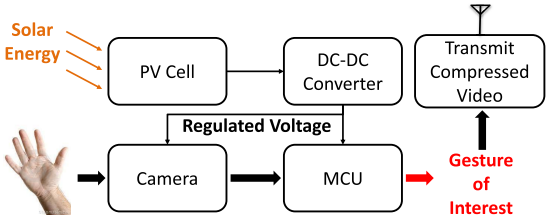


Fig. 3. Block diagram of the proposed camera front-end.

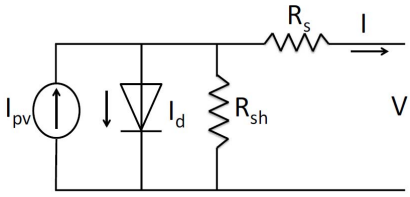


Fig. 4. Equivalent circuit representation of a PV cell.

DSPs are more attractive for self-powered devices, since they exhibit: (1) power dissipation in the order of hundreds of mWs (a 10X reduction compared to CPUs), (2) compact size and low thermal requirements, (3) sufficient computational ability for "always on" applications, will be demonstrated here. Our system features an Analog Devices' Black Fin processor.

To enable "always on" and self-powered operation, we take advantage of recent advances in compressed domain (CD) data processing which allows trigger detection with significantly lower power and computational requirements. This is in contrast with existing algorithms which work directly in the pixel domain. Given the objective of our camera front-end, the computation complexity can be largely reduced (768× demonstrated here) from existing algorithms that are targeted for continuous gesture recognition [7], [8]. As a command to wake up the system, only a few gesture classes are needed. When the gesture is structured and contains significant motion (for example, writing a big "Z" in front of the camera), it can be readily captured by images with high compression ratios. Beyond using low-resolution images, we construct each measurement as a random linear combination of pixels in a manner compatible with compressed domain signal processing. Recent developments in compressed sensing and target recognition in the compressed domain [9], [10] further improve the accuracy and energy efficiency of the overall process of data acquisition, feature extraction and recognition. We demonstrate that we can characterize the gesture motion directly from a few compressed measurements. On the other hand, energy harvested from the environment has been used in sensor networks [11], [12] with loads that demand very low power. Here we demonstrate that an algorithm-hardware co-design enables smart camera-front ends with "always on" gesture detection.

In our system, the gesture motion is captured by a sequence of difference images between consecutive frames. Each difference image passes two layers of compression to reduce its resolution and to be transferred to the compressed domain. The parameters of the motion are directly extracted from the compressed domain. A memory/power-efficient classifier is used to recognize gestures. Our work has major contributions including:

- To the best of authors knowledge, our work is the first autonomous gesture recognition system. Previously reported works [13], [14] had always on image sensor to capture images. Our system is light-powered gesture recognition system. We achieve low power by achieving computation in compressed domain and efficient power delivery using DC-DC converter. Computation in compressed domain enables DTW co-coefficients stored in L1 cache. Light aware power management regulates V_{OUT} during low, medium and high light conditions.
- To the authors' knowledge, our work is the first in gesture recognition using compressive sensing techniques. The algorithms is designed specifically for the efficient hardware implementation. Compared with the previous DTW-based K-NN gesture classifiers [15], [16] , our classification algorithm improves both the computational and the memory efficiency.

The rest of the paper is organized as follows: in section II we describe hardware system architecture. Algorithm details are described in section III. Section IV & V presents hardware implementation, comparison between hardware power efficiencies of proposed algorithms & measurement results respectively. Conclusions are drawn in section VI.

II. Hardware System Architecture

Before describing the proposed algorithm, we brief the hardware system architecture. The proposed system consists of four main components: a PV cell array, a DC-DC converter with output voltage regulation, an MCU, and an image sensor. The block diagram of our system is shown in Fig. 3. Camera and MCU are powered by PV cell. If we find gesture of interest then we transmit the captured image/video. The PV cell converts solar energy to electrical energy. The Norton equivalent output current (Fig. 4) of PV cell is given by:

$$I = I_{pv} - I_0[exp(\frac{V + IR_s}{aN_sV_T}) - 1] - \frac{V + IR_s}{R_{sh}} \quad (1)$$

where I and V are PV cell's output current and voltage respectively; R_s and R_{sh} are the series and shunt resistances; I_0, V_T, a, N_s are dark saturation current, thermal voltage, diode ideality factor, and number of cell connected in series

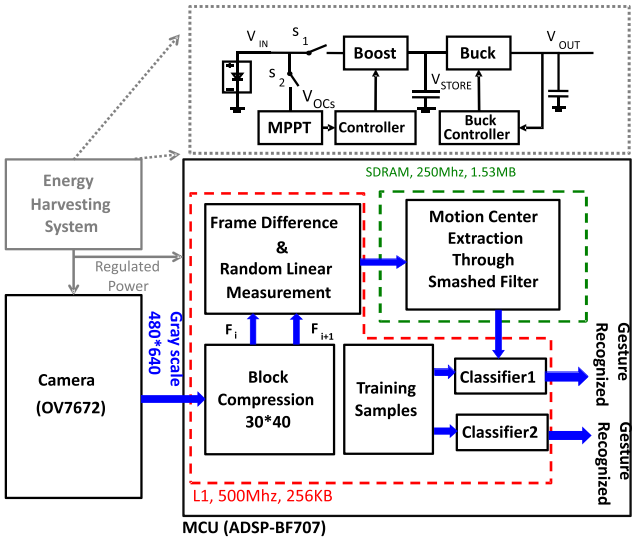


Fig. 5. Hardware system architecture demonstrating key components of power management, the MCU and the mapping of the algorithm on the MCU.

respectively; I_{pv} is the generated current whose magnitude depends on irradiation and temperature.

As the MCU and image sensors both demand regulated voltage to operate properly, the DC voltage generated by PV cells must be regulated by a DC-DC converter. For the current design, we select TI's BQ25570EVM, a two-stage DC-DC converter with Maximum Power Point Tracking (MPPT) for solar energy harvesting and for providing a regulated output supply. The block diagram of the energy harvesting system and gesture recognition flow is shown in Fig. 5.

The input image is captured by Omnivision's OV7672 sensor with a native resolution of 480 × 640. We extract only the gray-scale component of the image, which reduces the computation power without any impact on performance. The output of the pixel array is passed on to an on-board ADSP BF707 MCU using I2C interface. Once the image is received by the BF707 processor, we perform the following operations: block averaging, frame difference, random linear measurements, motion centers extraction in compressed domain followed by gesture recognition. For block compression we extract every one out of 16 pixel values in each row and column. Therefore the block compression factor is 256 (16 for every row and column). The block average, frame difference and dynamic time warping related matrices are stored in L1 cache (requires less than 128KB). Motion center extraction co-efficients are stored in external SDRAM (requires more than 1.2MB). L1 access is performed using core clock at 500MHz and SDRAM access happens at system clock with 250MHz speed. The hardware is further optimized by (1) using short integer maths and (2) optimizing memory usage that reduces total power consumption without loss of performance.

Two classifiers are used for gesture recognition once motion center is extracted. Classifier I is the traditional K nearest neighbor (K-NN) classifier using dynamic time warping (DTW) distance measurements. Classifier II is a modification of classifier one, allowing it to cooperate with clustering and dimension reduction techniques [17]. Classifier II reduces the memory requirement in the L1 cache as well as improves the computational efficiency.

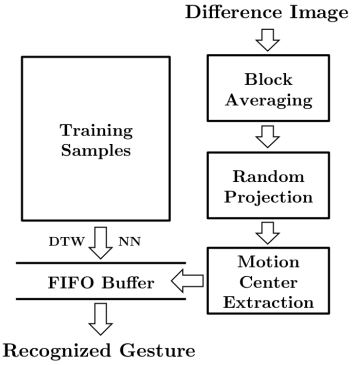


Fig. 6. Block diagram of the proposed algorithm.

III. Algorithms

Difference images are capable of capturing gestures containing significant motions. We pass each difference image through two layers of compression. In the first layer, the resolution is reduced by dividing the whole image into several blocks and taking the average of each block. In the second layer, we take coded combinations of these block-averaged pixels. We estimate the center of the motion directly from these compressed measurements. These motion centers are passed to a classifier for gesture recognition. Fig. 6 shows the block diagram of our system.

A. Two Layers of Compression

Denote F_i as the ith full resolution image output from the camera of size $W \times H$. The difference image D_i (Fig. 7 (a) of two consecutive frames is calculated as $D_i = |F_{i+1} - F_i|$.

In the first compression layer, the difference image is divided evenly into blocks of size B by B. The average of the pixel values in each block is taken, resulting in a block-compressed difference image of size W/B by H/B (Fig. 7 (b). We vectorize this low-resolution difference image and denote it as $y_i \in \mathbb{R}^N$.

In the second layer of compression, we chose a random matrix Φ of size M by N as the coded measuring matrix. Each entry of Φ is uniformly chosen from $\{+1, -1\}$. The projection of the vectorized low-resolution difference image in the compressed domain is calculated as:

$$\hat{y}_i = \Phi y_i = \Phi \Psi Y_i \tag{2}$$

Each entry in $\hat{y}_i \in \mathbb{R}^M$ is a random linear combination of all the entries in y_i. Y_i is the vectorized original difference image D_i. Ψ is the block averaging matrix of size N by $W \times H$.

B. Motion Center Extraction in the Compressed Domain

In the uncompressed low-resolution domain, the hand region in the difference image can be captured by a template shown

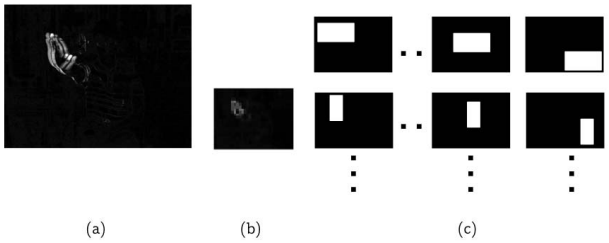


Fig. 7. (a) Full-resolution difference image D_i; (b) Block averaged difference image; (c) Matching templates in the uncompressed domain. Rectangle sizes differ among rows and centers of the rectangles differ among columns.

in Fig. 7 (c). The template (of size W/B by H/B) has uniform non-zero values within the small rectangular region and zeros elsewhere. To locate the hand region, we construct a set of vectorized templates $X(\alpha, r)$, where α represents the coordinates of the center of the small rectangle, and r represents different rectangle sizes. The variation in sizes is to adapt to the change of the hand size seen by the camera when users at different locations. The center of the hand motion is extracted by solving

$$(\alpha^*, r^*) = \arg\min \alpha, r||y_i - X(\alpha, r)||_2 \tag{3}$$

The collection of templates forms a manifold in $\mathbb{R}^N$ with intrinsic parameters α and r. Using the result from [9] and [18], we can directly extract the motion centers in the compressed domain. That is, for

$$(\hat{\alpha}^*, \hat{r}^*) = \arg\min \alpha, r||\hat{y}_i - \Phi X(\alpha, r)||_2 \tag{4}$$

$(\hat{\alpha}^*, \hat{r}^*) \approx (\alpha^*, r^*)$ with high probability for some $M \ll N$. The block averaging layer reduces the possible choices of r, and techniques such as matched filtering can be applied to efficiently solve equation (4).

C. Train the Gesture Classifier II

DTW-based classifiers perform well for dataset containing limited amount of samples [19]. Traditional DTW-based classifiers use DTW [20] as the distance measuring method between two sequences of different lengths, and use K-NN method for classification. The memory and computational requirements thus grow linearly with the size of training set.

To reduce the number of DTW calculations in the recognition stage, we perform K-means clustering in the training dataset to form "super samples". The distance between an individual sample and a super sample is measured using DTW. In each iteration, the super samples are updated as the average of all the samples within their clusters. DTW barycenter averaging (DBA) [21] is used as the averaging method.

The main difficulty of time series classification comes from the different lengths of the samples. We notice that DBA can find clustering centers of an arbitrary length set by the user. The pairwise matching information in DTW also provides a way to rescale the length of time sequences. Therefore, we propose a DTW length rescale algorithm, shown in Algorithm 1:

Using Algorithm 1, we rescale all the training sequences to the same length τ. Since each motion center in the time

Algorithm 1 DTW Length Rescaling

Require: K "super samples" $\mathbf{S_1}, \mathbf{S_2}, \ldots, \mathbf{S_K}$ of length τ, calculated using K-means with DTW and DBA.
Require: Sequence $\mathbf{T}$ to be rescaled to length τ
  $\mathbf{S}^* = \arg\min\ DTW(\mathbf{S_k}, \mathbf{T})$
  $\mathbb{M} \leftarrow$ pairwise matching information between $\mathbf{S}^*$ and $\mathbf{T}$
  Initialize $\mathbf{T}'$ of length τ
  for $i = 1$ to τ **do**
    if S_i^* is matched to multiple points $T_j, T_{j+1}, \ldots, T_{j+m}$, according to $\mathbb{M}$, **then**
      $T_i' = \underset{T_l \in T_j, \ldots, T_{j+m}}{\arg\min} ||S_i - T_l||$
    else
      $T_i' = T_j$, where T_j is the only matching point to S_i^*
    end if
  end for

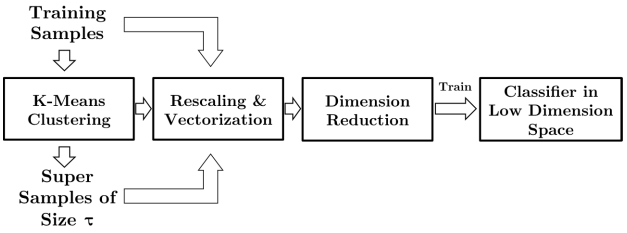


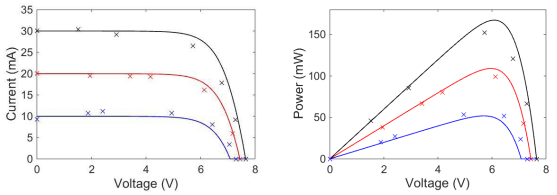
Fig. 8. Block diagram of training the gesture classifier.

sequence contains both x and y coordinates, each gesture sample is now of size $2 \times \tau$. We vectorize the samples by cascading all the y coordinates after the x coordinates, transferring the time series classification problem to a traditional classification problem in $\mathbb{R}^{2\tau}$. Various dimension reduction techniques and multi-class classification algorithms can then be implemented. Specifically, we apply PCA to the rescaled time series. In the low-dimensional subspace, we model the distribution of each gesture class as a mixture of Gaussian (GMM). The block diagram of the complete training procedure is shown in Fig. 8.

D. Gesture Recognition Using Classifier II

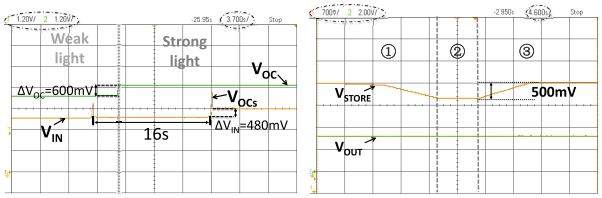
In a real-time system, the extracted motion centers are stored in a FIFO buffer of length L. To recognize the gesture, we first rescale the buffer data sequence based on the learned super samples using Algorithm 1. Within this step, open-ended DTW is used for pairwise matching in order to automatically separate the gesture-like data from the noise at both ends of the buffer. The new gesture-like sequence of length τ then goes through vectorization and dimension reduction before being sent to the trained GMM-based classifier. The likelihood of a given motion sequence belonging to each gesture class is computed, and the gesture is assigned to the most likely class, once this likelihood passes a pre-determined threshold.

Compared to the traditional DTW-based K-NN classifiers (also implemented as classifier I), our classifier significantly reduces the number of DTW calculation in the recognition stage and still being able to exploit the structure of the entire training set. In our experiments, most of the gestures can be well separated in a very low dimension, making the dimension



(a) I-V characteristics (b) P-V characteristics

Fig. 9. (a) I-V characteristics and (b) Power-Voltage characteristics of PV cells at three different irradiance levels (300W/m^2, 600W/m^2 and 1000W/m^2). Discrete points are experimental results and continuous curves are from simulations.



(a) (b)

Fig. 10. Oscilloscope captures illustrating (a) MPPT where V_{IN} tracks the open circuit voltage at 80% of V_{OC} as irradiance changes and (b) regulation of V_{OUT} under dynamically varying super-capacitor voltage (V_{STORE}). In (b) three regions are shown: (1) instantaneous load power consumption is higher than input power which reduces V_{STORE}; (2) load power and the harvested power are balanced and (3) load consumption is less than harvested power.

reduction and the low-dimensional classifier computationally efficient as well.

IV. Experimental Setup

A. Power Management Design

The overall platform is designed from COTS components and here we explain the optimal design choice. We chose omnivision OV7672 image sensor which has frame size of 480×640 pixels. The image sensor is connected to an ADSP BF707 processor using I2C interface. Measurements reveal a maximum current consumption of $170mA$ at fixed $3.3V$ power supply.

The solar cell (AM5907) produces an output voltage of 5V at the point of maximum power transfer. The I-V and P-V characteristics of each cell is shown in Fig. 9a and Fig. 9b vis-a-vis simulation results. We see a close match between experimental results and empirically fitted Eq 1. We note that for an irradiance of $600W/m^2$, the maximum power $\approx 100mW$. In the current setup, We use 6 PV cells in parallel to generate the required power that the load demands. Also, from Fig. 9b, we observe that operating voltage at maximum power point is approximately 80% of the open circuit voltage (V_{OC}). Hence, Maximum Power Point Tracking (MPPT) is achieved by regulating the output at 80% of V_{OC}.

Figure 5 shows, the MPPT block samples open circuit voltage V_{OC} every 16 seconds with S_2 on and S_1 off. This sample voltage V_{OC_sample} is sent to the boost controller to modulate the phase and frequency of the boost converter so that the PV cell operates at maximum power point, 80% of V_{OC_sample}. The sampling process is shown in oscilloscope captures in Figure 10a. It is observed that open circuit voltage is sampled and the PV cell's operating voltage changes accordingly. The energy is stored in a super-capacitor between the two converter stages. Availability of super-capacitor benefits camera-based applications whose power requirement fluctuates significantly. The output voltage is sensed and sent back to buck controller to regulate the output voltage. The output voltage is hardware programmable through programmable external resistors on the board. Fig. 10b shows how V_{STORE} varies with varying irradiance and load current conditions. Measured oscilloscope capture also reveals that V_{OUT} is well regulated under such dynamic conditions. The complete experimental setup along with the PV cells and the MCU is shown in Figure 11.

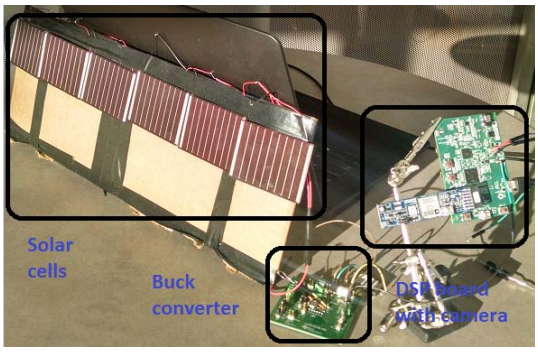


Fig. 11. The overall system demonstrating the solar cells and the MCU with the camera.

B. Mapping Proposed Gesture Recognition Algorithm on Low-Power MCU

The image sensor output at full-resolution (480×640) is captured by the MCU. The MCU performs compression on each difference image. In the block compression layer, we choose blocks of size 16×16; and hence the vectorized low-resolution difference image $y_i \in \mathbb{R}^{1200}$. The compression rate of this layer is thus 256. For low power operation and to enable a completely, self-powered system, the image sensor is operated at a maximum of 10 frames/second.

In the random projection layer, the number of compressed measurements M is a design variable. To gain better insights on the choice of M, we explore its relationship with the accuracy of motion center extraction. For a typical gesture "Z" the extraction algorithm is shown in Figure 12a. The motion centers are extracted from the block-averaged difference images by solving equation (3). As we can see in Figure 12b, the three segments of the gesture are clearly distinguished on the path of the motion centers. With $M = 250$, the motion centers are extracted in the compressed domain by solving equation (4), and are plotted in Figure 12c. The similarity between this plot and 12b demonstrates the effectiveness of the theory. For each value of M we calculate the average motion center error per frame in the compressed domain. The "L" shape of the curve indicates that $M = 250$ is the threshold for nearly error-free motion parameter estimation, granting us another factor of 5 compression rate. This "threshold" behavior is consistent with the classic results from compressed sensing presented in [9], [10], and [18]. The accurate motion center extraction in

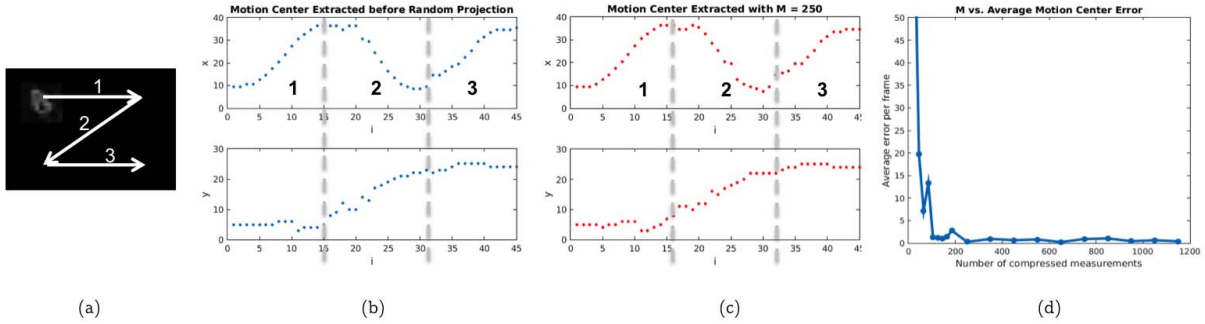


Fig. 12. Simulation results (from MATLAB) for different numbers of compressed measurements M. (a) Hand motion of gesture "Z" divided into three segments, (b) Motion center extracted before random projection by solving equation (3), (c) Motion center extracted from 250 compressed measurements by solving equation (4), (d) Average error of motion center extracted in the compressed domain compared with (b).

the compressed domain provides the foundation of preserving high recognition accuracy.

To reduce memory usage and reduce power consumption, we fix the size of the smashed filter templates (Figure 7.c) to 10×10. In other words, we construct $X(\alpha, r)$ with r fixed to 10×10 and α being every possible location in the 40×30 block-averaged difference image. Using the same Φ, we transfer all the templates into the compressed domain by calculating $\Phi X(\alpha, r)$. The buffer for storing the estimated motion centers is set to have length 40.

As proof of concept, we test the system with a variety of key gestures and in the rest of the paper, we will discuss an implementation that recognizes 5 gesture classes: "X," "+," "Z," "O," and "N." For the usage model where the key gestures are used for "wake up", this small number of gesture classes suffices. In each training example, the gesture is performed at different locations with respect to the camera, and the motion centers were extracted from the uncompressed domain by solving equation (3). When using classifier I, we provide 20 training samples per class due to the memory constraint in the hardware. Given this small number of training samples, we set up classifier I as the one-nearest-neighbor classifier with DTW distance measurement. When using classifier II, since its training stage is separated from the testing stage and neither the memory requirement nor the computational complexity depends on the number of training samples in each class, we provide 50 samples in each class. One super sample is calculated in each class, and we choose only the first 3 principle components after playing PCA. We modeled each gesture class' distribution as one multivariate Gaussian in $\mathbb{R}^3$, whose mean and variation are directly calculated from the training samples. Testing gestures are assigned to the class with highest likelihood beyond a rejecting threshold corresponding to the 85% confident region.

V. Measurement Results

A. Number of Compressed Measurements vs. Recognition Rate and Power Consumption

For different numbers of compressed measurements, we measure the energy consumption per frame and the recognition rate. We evaluate 20 gestures of each class, and the recognition rate is calculated from the total number of correctly recognized gestures. The total time per gesture is

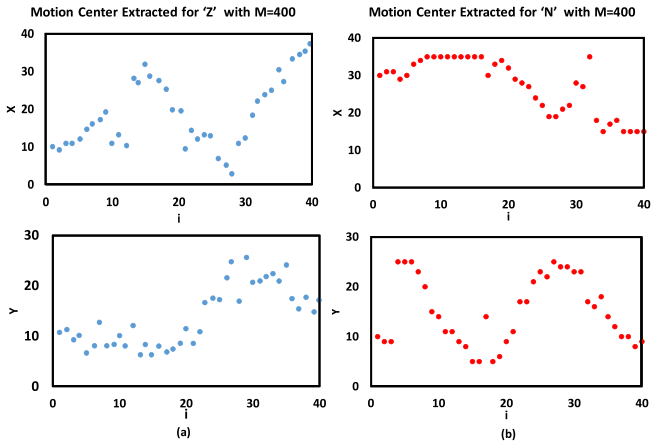
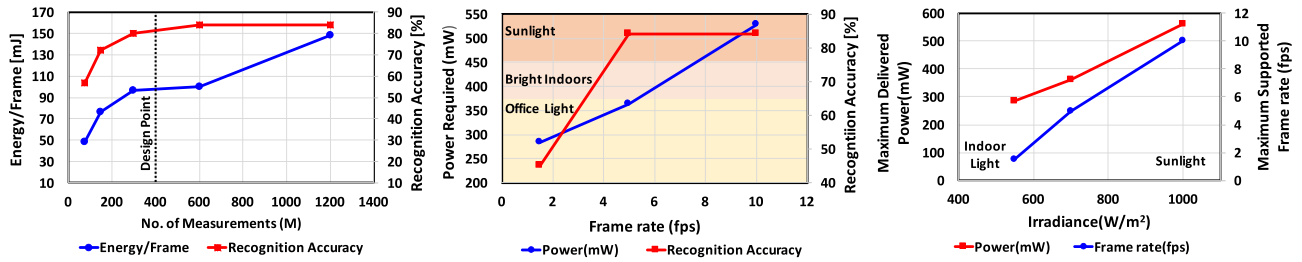


Fig. 13. a) Measured motion center extraction of hand gesture "Z" for $M = 400$ (Close match with simulations in Fig. 12). b) Measured motion center extraction of hand gesture "N" for $M = 400$.

kept at 0.1secs. In a typical instance, the motion centers for a gesture "Z" as extracted from the hardware is shown in Fig. 13 (a). Comparison with Fig. 12 reveals a close match between simulation and measurement. We have also plotted the measured motion centers for N gesture in Fig. 13 (b). Fig. 14a shows the measured design space exploration. We have measured the recognition accuracy as a function of M which reveals an accuracy rate of $> 80\%$ for $M > 300$, which closely matches simulation results described in the previous section. Fig. 14b shows dependence of the power consumed by the MCU and the corresponding recognition accuracy of the proposed system as a function of the frame rate. We note that a minimum frame rate of 5fps is required for maintaining a desired recognition accuracy of [$\geq$84%]. As the frame rate increases, the corresponding power consumption also increases and shows a graceful trade-off between accuracy and power consumed. Fig. 14c illustrates the efficiency of the power management system where the irradiance of the incident light is varied. The corresponding power consumed and the maximum frame rate that can be supported is also shown. It can be noted that for an irradiance of $1000 W/m^2$ (typical for outdoor sensors) a frame rate of 10fps and recognition accuracy of $> 80\%$ is achieved.

As the environmental conditions and irradiance levels change, the proposed system can scale the frame/sec accord-



(a) (b) (c)

Fig. 14. Design space exploration through measurements: (a) Number of compressed measurements (M) vs. Energy/frame and Recognition accuracy, (b) Frame rate vs. Power and Recognition accuracy, and (c) Frame rate and Power vs Irradiance.

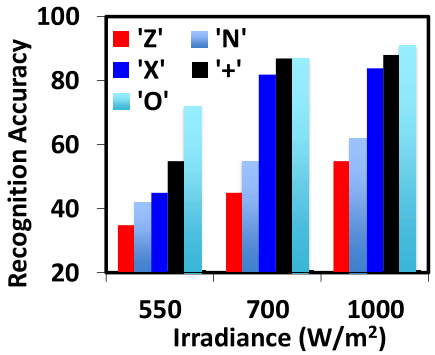


Fig. 15. Recognition accuracy vs Irradiance for Z, X, O, N and +.

TABLE I
RECOGNITION ACCURACY OF 5 GESTURE CLASSES (M = 400)

Gesture Type	+	Z	X	O	N
Recognition Rate (Classifier I)	90%	70%	85%	85%	75%
Recognition Rate (Classifier II)	90%	66%	85%	88%	72%

ingly, which gracefully trades-off recognition accuracy. Fig. 15 illustrates the trade off between recognition accuracy for different gestures as a function of Irradiance.

B. System's Multi-Class Recognition Accuracy

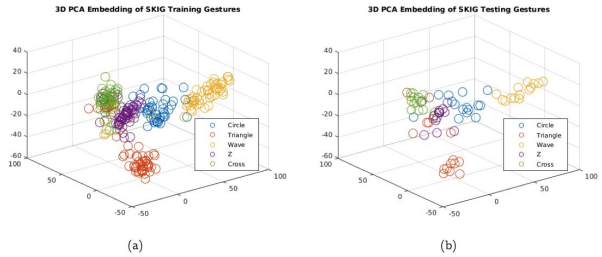
At $M = 400$ the recognition accuracy of 5 different gesture classes are shown in Table I. It can be seen that the recognition accuracy depends on complexity of the gesture. For a simple gesture, e.g., "+", a peak accuracy of 90% in a fully solar energy harvested system is measured. A comparison of the proposed system with competing hardware [2], [6], [22] based motion and gesture detection is shown in Table. II. The proposed system demonstrates more than 3× improvement compared to reported works in energy/frame for detecting "wake up" gestures. This enables a fully self-powered "always on" camera front end.

C. Comparison Between Classifier I and II

The performance of the two classifiers are first compared purely in the software level. We apply them on the

TABLE II
RECOGNITION RATES OF SKIG GESTURE DATASET

Gesture Type	Circle	Triangle	Wave	Z	Cross
Recognition Rate (Classifier II)	93.3%	73.3%	93.3%	80%	93.3%
Recognition Rate (Classifier I)	93.3%	93.3%	93.3%	100%	100%



(a) (b)

Fig. 16. (a) PCA embedding of SKIG training samples in $\mathbb{R}^3$; (b) PCA embedding of SKIG testing samples in $\mathbb{R}^3$.

public-available SKIG dataset [23] using OpenCV simulation. We select 5 classes containing significant motion in the x-y plane: Circle, Triangle, Wave, Z, and Cross. In each class, we select 70 well-illuminated samples and randomly divided them into training (55 samples) and testing (15 samples) sets. We crop the training videos so that the gesture motion filled the entire video. Since each frame is of size 240×320, in the block compression layer, we use blocks of size 10 by 10, and we set $M = 200$ in the random projection layer.

During the training stage of classifier II, we calculate 1 super sample in each gesture class. As the lengths of the gesture videos vary from 48 to 236 frame, we choose τ to be the average length 116. After rescaling and vectorizing all the training sample, we apply PCA and use the first 3 principle components. The gesture samples are well separated in $\mathbb{R}^3$, as shown in Fig. 16.a. For simplicity, we model each gesture class' distribution as one multivariate Gaussian in $\mathbb{R}^3$, whose mean and variation are directly calculated from the training samples. A testing gesture is assigned to the class with highest likelihood beyond a rejecting threshold corresponding to the 85% confident region. The resulting classifier has decision boundaries of ellipsoid shapes.

Following the proposed recognition procedure, we show the testing samples' PCA embedding in Figure 16.b. The recognition rate is shown in Table II. The relatively low

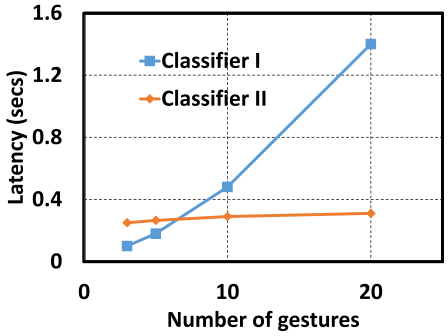


Fig. 17. Latency in last frame for classifier I and II.

TABLE III
MEMORY REQUIRED BY DTW FOR CLASSIFIER I AND II

	Classifier I	Classifier II
Memory (K=5)	14KB	1.12KB
Memory (K=10)	28KB	1.82KB

recognition rate for triangle gestures is caused by the longer sample videos that usually contain more than 200 frames. Rescaling them resulted in significant down sampling, and some of these down sampled data overlap with the "Z" gesture samples in the $\mathbb{R}^3$ embedding. Classifier II had comparable performance with Classifier I (a 5-NN classifier) for all other classes.

In the hardware design, for both the classifiers the frame rates are limited by the motion center extraction. It is mainly because for a fixed buffer length that dynamic time warping is performed at the end of each frame. Therefore, both classifiers add latency in the capture of last frame. Fig. 17 shows the latency in last frame for classifier I and II. The increases latency for classifier I is due to increased number of distance computation (proportional to number of gesture samples).

The memory required by DTW for the two classifiers is given in Table. III. The memory for classifier I grows linearly with the number of training samples. If K different gestures classes are to be recognized from and the number of training samples in each class is 20, the total number of bytes required is: $K \times 20 \times 70 \times 2$ (20 samples for each class, each sample is a sequence of size 70×2). Memory required by classifier II grows only linearly to the number of gesture classes and is independent with the number of training samples in each class. In the dimension reduction stage of classifier II, storing the pca basis (with 3 principle components) requires 420 bytes. The total memory required by classifier II is, therefore, $(420 + 140 \times K)$ bytes. As classifier II uses less cache and provides less latency, we conclude classifier II is suited for low power embedded applications. The test case recognition accuracy of classifier II is about 4% less for gestures N and Z compared to classifier I (From Table I. The transmission energy/frame from Fig. 18 is less than 10mJ. The energy/frame for sensing with computation is 95mJ. Therefore, false negatives provided by classifier II for gestures N, Z doesn't affect the overall energy consumption . Low Memory and energy/frame are the important factors for low power embedded application. Therefore, we prefer classifier II over classifier I for our application.

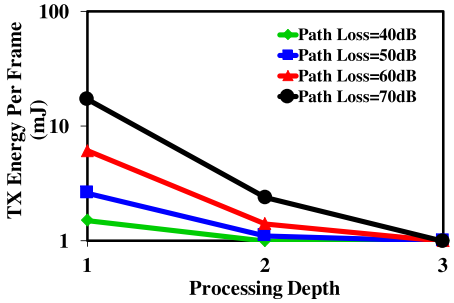


Fig. 18. Energy/frame of the transmitter vs processing depth.

Table. IV shows the comparison of the proposed system with reported research in literature. We have included an array of gesture recognition systems that have been discussed, starting from FPGAs, PCs to an Xbox console [24] in our comparison table. Our system operates with solar energy and has the lowest energy/frame for gesture recognition.

D. Improvement in Energy Efficiency of Transmitter

Once a key gesture is used to wake up the system, the captured video is expected to be transmitted to a cloud or FOG node. We realize such a system by connecting the output of the camera system to a software-defined radio. The transmitter used in the current design is Ettus USRP B200. Fig. 18 shows the energy/frame of the transmitter vs processing depth. Processing depth indicates the level of compression on the BlackFin processor. Depth 1 indicates no block averaging and compressive sensing. Depth 2 indicates transmitted frame with block averaging. Depth 3 indicates block averaging and compressive sensing (by a factor of 3). We can observe that for processing depth of 3, the energy/frame of transmitter reduces by a factor of 8X. This shows that compressed domain processing not only enables ultra-low power gesture detection, but a compressed domain image acquisition can allow significant savings in transmitted energy once the system has woken up. Image reconstruction [18] and analytics on the compressed domain image on the cloud node, is beyond the scope of this paper and has been extensively studied [25].

VI. CONCLUSIONS

This paper presents a solar powered, "always on", gesture recognition system that provides a trigger for system "wake up". The major savings of power in our system comes from the two layers of compression that reduce the resolution of the image sensor by a factor of more than 768× [256× by block averaging and 3× by random compressive measurements]. The block compression layer preserves the geometric information of the gesture and the random projection layer preserves the manifold where the motion parameters lie. These two preservation are the keys for maintaining a high recognition rate in the compressed domain. Classifier II embedding PCA is suited to reduce the latency in last frame. Hence suited for low power embedded applications. Further a hardware-algorithm co-design allows energy-efficient mapping of the recognition algorithm on a low power MCU and powered by a solar powered DC-DC converter and regulator with MPPT.

The system demonstrates an average recognition accuracy of $> 80\%$ while consuming less than $95mJ/frame$. Once the system wakes up, compressed domain image acquisition is followed by transmission via a software defined radio.

REFERENCES

[1] R. Wang, Z. Yu, M. Liu, Y. Wang, and Y. Chang, "Real-time visual static hand gesture recognition system and its FPGA-based hardware implementation," in *Proc. ICSP*, 2014, pp. 434–439.

[2] D. Lee and Y. Park, "Vision-based remote control system by motion detection and open finger counting," *IEEE Trans. Consum. Electron.*, vol. 55, no. 4, pp. 2308–2313, Nov. 2009.

[3] S. J. Desai, M. Shoaib, and A. Raychowdhury, "An ultra-low power, 'always-on' camera front-end for posture detection in body worn cameras using restricted boltzman machines," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 1, no. 4, pp. 187–194, Oct./Dec. 2015.

[4] B. Kellogg, V. Talla, and S. Gollakota, "Bringing gesture recognition to all devices," in *Proc. Symp. Netw. Syst. Des. Implement.*, 2014, pp. 303–316.

[5] A. Amaravati, M. Chugh, and A. Raychowdhury, "A SAR pipeline ADC embedding time interleaved DAC sharing for ultra-low power camera front ends," in *Proc. IFIP/IEEE Int. Conf. Very Large Scale Integrat.-Syst. Chip*, Oct. 2015, pp. 131–149.

[6] W.-H. C. Chao-Tang Li, "A novel FPGA-based hand gesture recognition system," *J. Converg. Inf. Technol.*, no. 1, no. 9, pp. 221–229, 2012.

[7] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, Jan. 2012.

[8] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 677–695, Jul. 1997.

[9] M. A. Davenport *et al.*, "The smashed filter for compressive classification and target recognition," *Proc. SPIE*, p. 64980H, Jul. 2007.

[10] W. Mantzel, J. Romberg, and K. Sabra, "Compressive matched-field processing," *J. Acoust. Soc. Amer.*, vol. 132, no. 1, pp. 90–102, 2012.

[11] Y. Zhang *et al.*, "A batteryless 19 μW MICS/ISM-band energy harvesting body sensor node SoC for ExG applications," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 199–213, Jan. 2013.

[12] X. Liu and E. Sánchez-Sinencio, "A highly efficient ultralow photovoltaic power harvesting system with MPPT for Internet of Things smart nodes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 12, pp. 3065–3075, Dec. 2015.

[13] S. Nayar, D. Sims, and M. Fridberg, "Towards self-powered cameras," in *Proc. ICCP*, Apr. 2015, pp. 1–10.

[14] J. Choi, S. Jungsoon, D. Kand, and D.-S. Park, "Always-on CMOS image sensor for mobile and wearable devices," *IEEE J. Solid-State Circuits*, vol. 51, no. 1, pp. 130–140, Jan. 2016.

[15] G. A. ten Holt, M. J. T. Reinders, and E. A. Hendriks, "Multi-dimensional dynamic time warping for gesture recognition," in *Proc. 19th Annu. Conf. Adv. School Comput. Imag.*, vol. 300. 2007, pp. 1–8.

[16] A. Akl and S. Valaee, "Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 2270–2273.

[17] S. Xu, A. Amaravati, J. Romberg, and A. Raychowdhury, "Appearance-based gesture recognition in the compressed domain," in *Proc. ICASSP*, Mar. 2017, pp. 1722–1726.

[18] R. G. Baraniuk and M. B. Wakin, "Random projections of smooth manifolds," *Found. Comput. Math.*, vol. 9, no. 1, pp. 51–77, 2009.

[19] J. M. Carmona and J. Climent, "A performance evaluation of HMM and DTW for gesture recognition," in *Iberoamerican Congress on Pattern Recognition*. Berlin, Germany: Springer, 2012, pp. 236–243.

[20] M. Müller, "Dynamic time warping," in *Information Retrieval for Music and Motion*. New York, NY, USA: Springer, 2007, pp. 69–84.

[21] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognit.*, vol. 44, no. 3, pp. 678–693, 2011.

[22] Y. Shi and T. Tsui, "An FPGA-based smart camera for gesture recognition in HCI applications," in *Proc. 8th Asian Conf. Comput. Vis.*, 2007, pp. 718–727.

[23] L. Liu and L. Shao, "Learning discriminative representations from RGB-D video data," in *Proc. IJCAI*, vol. 1, 2013, p. 3.

[24] Y. Chen, B. Luo, Y.-L. Chen, G. Liang, and X. Wu, "A real-time dynamic hand gesture recognition system using Kinect sensor," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Mar. 2015, vol. 44. no. 3, pp. 2026–2030.

[25] A. Amaravati, S. Xu, N. Cao, J. Romberg, and A. Raychowdhury, "A light-powered, 'always-on', smart camera with compressed domain gesture detection," in *Proc. IEEE ISLPED*, 2016, vol. 1. no. 1, pp. 118–123.

Anvesha Amaravati received the M.Tech. degree in microelectronics from IIT Bombay in 2012, with a focus on PVT tolerant analog and bio-medical circuits. He is currently pursuing the Ph.D. degree with the Georgia Institute of Technology. His research interests include mixed signal circuits and systems for machine learning. He has authored/co-authored 15 international conference/journal publications and won national design contests organized by Analog Devices and Cadence.

Shaojie Xu received the B.S.E.E. degree from Rice University in 2014. He is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, Georgia Institute of Technology. His current research interests include compressive sensing, image processing, and machine learning.

Ningyuan Cao (S'17) received the B.S. degree in electrical engineering from Shanghai Jiaotong University, and the M.S. degree in electrical engineering from Columbia University. He is currently pursuing the Ph.D. degree. He has been with the Integrated Circuit and System Research Laboratory, Georgia Institute of Technology, since 2015. His current research interests include low-power wireless sensor design, power management, and energy harvesting circuit design.

Justin Romberg received the B.S.E.E., M.S., and Ph.D. degrees from Rice University, Houston, TX, USA, in 1997, 1999, and 2004, respectively. From 2003 to 2006, he was a Post-Doctoral Scholar in applied and computational mathematics with the California Institute of Technology. He is currently a Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, where he has been on the faculty since 2006.

Arijit Raychowdhury (SM'16) received the Ph.D. degree in electrical and computer engineering from Purdue University in 2007. He joined Georgia Institute of Technology in 2013, where he is currently an Associate Professor with the School of Electrical and Computer Engineering and also holds the On Semiconductor Junior Professorship. His industry experience includes five years as a Staff Scientist with the Circuits Research Laboratory, Intel Corporation, and a year as an Analog Circuit Researcher with Texas Instruments Inc. His research interests include low power digital and mixed-signal circuit design, device-circuit interactions and novel computing models and hardware realizations. He holds over 25 U.S. and international patents and has published over 100 articles in journals and refereed conferences. He was a recipient of the Intel Early Faculty Award in 2015, the NSF CISE Research Initiation Initiative Award in 2015, the Intel Labs Technical Contribution Award in 2011, the Dimitris N. Chorafas Award for Outstanding Doctoral Research in 2007, the Best Thesis Award from the College of Engineering, Purdue University, in 2007, and multiple best paper awards and fellowships.