Self-Optimizing IoT Wireless Video Sensor Node With In-Situ Data Analytics and Context-Driven Energy-Aware Real-Time Adaptation

Ningyuan Cao, Shreyas Sen, Member, IEEE, Saad Bin Nasir, Student Member, IEEE, and Arijit Raychowdhury, Senior Member, IEEE

Abstract—It is well understood that data-acquisition by distributed sensors and subsequent transmission of all the acquired data to the cloud will produce a “data deluge” in next-generation wireless networks leading to immense network congestion, and data back-logs on the server which will prevent real-time processing and control. This motivates in situ data analytics in energy-constrained wireless sensor nodes that can perform context-aware acquisition and processing of data; and transmit data only when required. This paper presents a camera-based wireless sensor node with a self-optimizing end-to-end computation and communication design, targeted for surveillance applications. We demonstrate support for multiple feature-extraction and classification algorithms, tunable processing depth and power amplifier gain. Depending on the amount of information content, accuracy targets and condition of the wireless channel, the system chooses the minimum-energy operating-point by dynamically optimizing the amount of processing done on the sensor itself. We demonstrate a complete system with ADI ADSP-BF707 image processor, OV7670 camera sensor, and USRP B200 software defined radio; and achieve 4.3× reduction in energy consumption compared with a baseline design.

Index Terms—IoT, adaptive radio, video surveillance, 5G.

I. INTRODUCTION

WITH the proliferation of small form factor distributed sensors and Internet of Things end-nodes, aggregate data transfer to the back-end servers in the cloud is expected to become prohibitively large. For example, 100 image sensors in a sensor network transferring HD data can result in an aggregate throughput of over 1GBps and significantly increase the network’s drop rate [1]–[4] as is shown in Fig. 1. This large amount of data transfer not only results in high energy expenditure and hence low battery life of the sensor node, but it will also result in network congestion producing severe quality of service (QoS) degradation in the form of queueing delay at best, and packet loss or blocking of new connections.

Fig. 1. (a) Aggregate throughput increases with number of sensor node in the network and the data volume the sensor acquired. (b) Drop rate of the network increased significantly with source rate [3].

In situ data analytics also precludes any real-time processing and network control, which is a requirement in a myriad of monitoring and sensing applications [6], [7]. Moreover, with the expected rapid growth in the number of sensors and raw data, the IoT network design itself will only become more complex increasing both the implementation and deployment costs.

To achieve both high energy efficiency in the end-node and seamless network operation, in-situ data analysis capability has to be enabled in the end-node itself [6]–[8]. Limited intelligence and decision making, under strict energy constraints, embedded in ubiquitous IoT sensors can reduce the volume of transmitted data by either transmitting only the data of interest or compressing raw data into features or decisions of much smaller volume. It will greatly reduce the volume of data the network has to handle and relieve bandwidth burden on the back-end servers. Although in-situ data-analytics reduces the communication energy at the sensor nodes, it places extra burden on processing. One of the key challenges in IoT nodes is power consumption and system design in pivoted upon reducing the total dissipated power [8]. As we introduce in-situ processing, the computation power increases at the sensor, as it acquires data and analyzes it for possible information content. However, the energy to compute and the energy to communicate are not constants [6], [9], [10]. Rather, they are context and environment dependent. For example, a clean wireless channel would lead to lower communication power, with channel adaptive radios. Similarly, if there is no (or little) information contained in the sensed data, then it should be detected early in the processing pipeline. Hence, an energy-optimal system should: (1) allow in-situ data-analytics to extract information from the sensed data to reduce the power overhead of communication, and (2) perform optimal trade-off...
between the depth of computation and the amount of communication to enable lowest possible power at the sensor node.

This paper presents a prototypical camera based wireless IoT sensor node for detecting the presence of human beings, with applications in video surveillance. The sensor node supports multiple machine learning algorithms to meet target accuracy requirements. The image processing pipeline (IPP) consists of hardware supported object segmentation and localization through temporal difference (TD) followed by compression (CR), feature extraction (FE) and finally classification (CL). We define processing depth (PD) as the stages of computation that are performed in the sensor node, before the data is transmitted to the cloud server. The details of the PD are tabulated in Fig. 2a. For example, PD = 1 means that only TD and CR are performed on the sensor node and then the data is transmitted. A PD = 2 means that TD, CR and FE are performed before transmission, and so on. The sensor node transmits the output of the processed data and the depth of processing (i.e., PD = 1, 2 or 3) for each video frame to the cloud. For PD < 3, the rest of the pipeline is implemented in the cloud. An adaptive radio provides power scalable transmission, depending on the signal to noise (SNR) characteristics of the channel. It is intuitive to understand that as the PD increases, the energy cost to compute increases, but the data volume required to transmit decreases, thus reducing the energy cost to communicate. As the channel condition changes (from clean to noisy channel), the minimum energy point also changes. For a clean channel, a lower PD is preferred (as the energy to communicate is low), whereas with increasing path-loss a higher PD is preferred. This is shown qualitatively in Fig. 2b, where the energy to compute and communicate (for two channel conditions) have been shown and we note that the minimum energy point is observed at two different PD points. With this motivation, we demonstrate an end-to-end self-optimizing node, which can dynamically adapt the PD depending on the channel condition, to always track the point of minimum total energy. Further, we support multiple CR, FE and CL algorithms depending on the accuracy/power consumption target set by the cloud back-end and the user.

The complete hardware system consists of an ADI ADSP-BF707 image processor, OV7670 camera sensor and USRP B200 software defined radio. The IPP is implemented on the ADSP-BF707. Measurements have been carried out with a variety of channel conditions and contexts (input image) and, compared with full-transmission and full-computation strategies, we measure a maximum of 4.3× reduction in energy consumption through end-to-end self-optimization. To the best of our knowledge, this is the first paper to report fine-grain power management between computation and communication on a self-optimizing sensor node. We have compared our design with baseline designs where (1) Full-Computation is performed on the sensor node independent of the channel conditions and (2) Full-Transmission of all the acquired data is performed at the sensor node without any “in-sensor” intelligence. The proposed system shows a peak of 4.3× improvement in energy efficiency. We have also compared the design with state-of-art camera based sensor nodes and adaptive wireless systems. These systems do not exhibit any self-optimization between computation and communication. We note 2× to 45× improvement in energy-efficiency (measured in terms of energy/frame) compared to the state-of-the-art designs.

The rest of this paper is organized as follows. In Section II, the hardware platform is described. Section III introduces the IPP and the embedded human detection algorithm(s) and the tradeoff between detection accuracy and energy-efficiency. The communication system is described in Section IV. Self-optimization between computation and communication in the end-to-end system is discussed in Section IV, followed by experimental results in Section V and finally conclusions are drawn.

**II. PROTOTYPE HARDWARE PLATFORM**

Before we dive into the algorithms and results for in-sensor processing and wireless transmission, let us discuss the hardware platform which forms the basis of the rest of the paper. In the remainder of the paper, we will present measurement results to support theory of computation/communication.
and optimization, based on this embedded platform. The proposed video based sensor platform comprises of camera, image processor, software defined radio, and a PC based controller and configuration control as is shown in Fig. 3 and Fig. 4. The camera (OV7670) captures 8-bit gray-scale VGA video frames at 10-30fps (frames per second) and consecutive frames, \( F_i \) and \( F_{i-1} \), are stored in a 1.53MB off-chip SDRAM.

Fig. 4. Experimental setup showing the system components.

Temporal difference (TD) is computed in the blackfin image processor (ADSP-BF707) with the two subsequent frames fetched from SDRAM to identify, localize and segment a moving object in the image frame. When a moving object is detected, the segmented image of interest is processed through the different IPP stages. Human detection templates are stored in off-chip SDRAM on the board and fetched during CL.

The transceiver (Ettus B200) works in half duplex mode. During transmission, it receives data from the processor (data can be the output of any PD). This data is wrapped in packages with prefix containing information of the algorithm, PD, package length and total data volume. Packages are modulated in GMSK and transmitted at 985Mhz. Channel condition (in terms of path-loss) is evaluated at cloud back-end (which also consists of an identical transceiver board) and sent to the IoT node. The transceiver at the sensor node, adjusts the power amplifier gain accordingly to meet a bit error rate (BER) target, as will be described in Section IV. The configuration settings and end-to-end controller parameters (transmitter gain, PD, choice of algorithm, energy models for each operating condition) are currently implemented in a PC; and can be ported to an embedded hardware for deployment. Platform hardware and architecture is previously discussed in [11].

### III. Embedded Computation

Our current platform is designed for detecting the presence of human beings (henceforth, called human detection) in the field of view. The IPP for human detection is composed of four processing stages: object localization and segmentation through temporal difference (TD), compression (CR), feature extraction (FE) and classification (CL). As discussed in Section I, PD is a direct control knob that allows us to trade-off computation vs. communication at the sensor node. Besides a dynamically tunable PD, the prototype platform offers three algorithm choices with different level of computation complexities and detection accuracy to provide higher level of power-performance trade-off. The target accuracy is set by the cloud back-end and is typically application specific. As is shown in Fig. 5, in our design, Algorithm-1 (highest accuracy) applies CR ratio of 2:1, 7 feature gradients and SVM classification template; Algorithm-2 (nominal) compresses input frame 4 times, extracts 5 gradients per feature and applies NB human detection template; Algorithm-3 (most energy-efficient) heavily compresses input frame 8 times, extracts 3 feature gradients and classifies with the tree template.

Fig. 5. Embedded human detection computation and design points of different algorithms/operations. Algorithm-1 (highest accuracy) applies CR ratio of 2:1, 7 feature gradients and SVM classification template; Algorithm-2 (nominal) compresses input frame 4 times, extracts 5 gradients per feature and applies NB human detection template; Algorithm-3 (most energy-efficient) heavily compresses input frame 8 times, extracts 3 feature gradients and classifies with the tree template.

![Algorithm demonstration with a real video frame.](image)

Fig. 6. Algorithm demonstration with a real video frame.

A. Objection Localization and Segmentation

Object localization and segmentation is the pre-processing stage to detect whether a certain frame contains a moving object and segment the object for further computation or transmission. The pre-processing stage prohibits unnecessary computation or communication of following stages when the field of view (FoV) is empty. As pre-processing is always on, the low-power requirement of this algorithm is a primary consideration. There are three major approaches for object activity detection and segmentation: temporal difference [12], model
based object localization [13], [14] and optical flow [15]. Opti-
cal flow method can obtain complete information and detect
the moving object from background better, but requires clus-
tering, which is computationally expensive and unsuitable for
real-time IoT operation. Model based background subtraction
relies heavily on dynamically calibrated background models,
which has a large overhead in an embedded systems, especially
under strict power constraints. Compared with optical flow
and model-based background extraction, temporal-difference
computes moving object area with the least operation and
consumes least energy. Hence, in the current implementation,
we use temporal-difference for its simplicity and high energy
efficiency [13] in the low-power pre-processing stage. In the
temporal difference method, we subtract two consecutive video
frames. The pixels whose difference is greater than a certain
temporal difference is labeled as activated pixels with label
value of 1. Otherwise, label value 0 is assigned. This can be
summarized as:
\[
D_i(m, n) = |F_i(m, n) - F_{i-1}(m, n)|
\]
(1)
\[
L_i(m, n) = \begin{cases} 
0, & |D_i(m, n) - D_{i-1}(m, n)| \leq E_{ih} \\
1, & |D_i(m, n) - D_{i-1}(m, n)| > E_{ih} 
\end{cases}
\]
(2)
The area of interest is defined as the pixels within the
rectangular boundary with label value of ‘1’. We quantify the
“information content” (S) of a frame as the number of activated
pixels (normalized to the total number of pixels) and it forms
a consistent measure of context in camera based sensor nodes.
If information content is less than 3.125% (60×40 in a QVGA
frame), we do not perform any further processing and the
entire system is gated till the next frame is captured.

B. Compression
The second stage of IPP is image compression. The pur-
purpose of compression is to reduce the amount of data to
compute or communicate while maintaining a target accuracy
requirement. This is simply performed by averaging the pixel
values over a sliding window. In our design, compression
further scales down the segmented image from pre-processing
by evenly averaging pixels at certain compression ratio. CR1,
CR2 and CR3 represents increasing compression as shown in
Fig. 5.

C. Feature Extraction
Feature extraction derives informative and non-redundant
values to facilitate the subsequent stages to generate better
classification results. In human detection, feature extraction
is crucial to discriminate human from cluttered background.
Different feature descriptors are available, including wavelets,
SIFT and HOG. Among all feature extractors, Histogram of
Gradient (HOG) is chosen for its excellent performance
and large INRIA human dataset availability [16], [17]. HOG
first divides the input image matrix evenly into M×N cells.
Gradient angle and gradient magnitude of each pixel are
computed. Each pixel within the cell votes for an orientation-
based histogram channel by comparing gradient angle with
angle bins with weight of gradient magnitude. Angle bins
evenly spread on (−π, π] range and number of bins is Nbin.
Then the spatially connected cells form a block of size
\[(M−1)×(N−1)\]
to be locally normalized to account for
changes in illumination and contrast where M and N stands
for number of rows and columns of cells. The hardware supports
three FE options, as shown in Fig. 5.

D. Classification
Classification is the final step in the IPP. The classifier is
trained offline in testing phase and classification template is
generated and stored in the SDRAM. Different machine learning
classifiers have different performance-power trade-offs.
We employ three different classification schemes depending on
the target accuracy set by the cloud back-end depending on
the application. Based on our simulations, we support
Support Vector Machine (SVM) for highest performance,
Naive Bayes classifier (NB) for nominal performance, binary
tree classifier for highest energy efficiency, as three classifiers
to offer different trade-offs of complexity/accuracy in human
detection.

In Support Vector Machine-based classification [18], [19],
a feature set with N_in predictors are mapped to a vector in
N_in-dimensional space. The space is divided into two separate
spaces by a hyper-plane obtained from training. Margins
between the two spaces are two parallel hyper-planes with
distance of 2 supported by N-dimensional support vectors.
The number of support vectors, N_sup, is determined by the training
set. The relative distances to the two marginal hyper-planes
determines which class it belongs by computing:
\[
Y(\vec{x}) = \text{sign} \left( \sum_{i=0}^{N_{\text{sup}}} \alpha_i y_i \vec{v}_i \vec{x} + \beta_i \right)
\]
(3)

Where \(Y(\vec{x})\) is the classification result of input feature vector \(\vec{x}\), and \(\alpha_i, y_i, \vec{v}_i, \beta_i \) are N_in-dimensional support vector, weight, label and offset.

Naive Bayes (NB) classifier [20]–[22] assumes strong independence between individual descriptors and applies Bayes’
theorem, which describes stochastic event based on related accumulation-counts (MMAC) for different algorithms/depths.


gives the number of computations (in terms of 10^6 MAC operations) changes with both the algorithm of choice and the PD. Higher accuracy and deep embedded processing suffer from heavy computation which is expected to result in high computation energy expenditure.

As the PD is increased, the amount of data required to transmit to the backend (including all the header information) is reduced. Fig. 9a illustrates the transmitted (Tx) load (i.e., the amount of data to be transmitted per frame) for each computation depth. Fig. 9b, illustrates the measured computation energy per frame for the three different algorithms and PDs as discussed above. We note that the lowest computational energy of 0.71mJ/frame is recorded for Algorithm-3 and PD-1 while the highest computational energy of 8.2mJ/frame is measured for Algorithm-1 and PD-3, thus showing a span of 8X/9X depending on the choice of algorithm and PD. We also note that as the computation energy at the sensor node increases (higher PD), the total data volume decreases sharply thus allowing a smooth trade-off in the cost of computation and communication. Key results are tabulated in Fig. 9d.

IV. ADAPTIVE WIRELESS COMMUNICATION

Wireless communication conventionally is the major cause of energy expenditure and shortened lifetime of wireless sensors, especially when the sensors are experiencing expanding bandwidth, rapid growth of nodes and ever-increasing data volume with the development Internet of Things [6], [7]. To implement energy-efficient wireless design on SDR (software defined radio), the power/energy characteristics of the adaptive radio is first explored. As is shown in Fig. 10a, transceiver power, first dominated by standby power at low loads, increases with output power and dynamic power gradually dominates which is generally the case with noisy channels or long-distance transmissions. In Fig. 10b, it is observed that with the increase of data rate, energy per byte transmitted decreases tremendously. In our system, data rate is set at 125kBps by GNUradio.

Traditionally, transceivers are designed for the worst-case, hence maximum power consumption, to guarantee target performance, such as bit-error-rate (BER). However, as channel condition of wireless sensors varies significantly from time to time [25], adaptive wireless communication is desired which adjusts the transceivers dynamically to operate marginally with respect to performance according to temporal channel quality to save energy [9], [10], [26]–[33]. Channel quality is affected by (1) Path Loss (2) Interference Strength. (1) can be compensated by increasing transmitted power amplifier (PA) output power, (2) can be handled by increasing receiver linearity. Since we focus on co-optimizing computation and transmitter power, we mostly focus on (1) in this work. Path-loss in dB is expressed as [34]

Path_loss = 20log_{10}(\frac{4\pi df}{c})

where d is distance, f is the carrier frequency and c is the speed of light. In our design, the carrier frequency is 985MHz.
To compensate for path-loss, the power amplifier gain is adjusted dynamically to guarantee minimum BER. Measured BER vs. path-loss for different PA gains of the SDR are shown in Fig. 11a. The PA gain and the total transmission power required to meet a target BER=$10^{-8}$ and $10^{-4}$ for different path-loss are also shown in Fig. 11b. For the rest of the paper, we will use these two target BERs. BER=$10^{-8}$ is a conservative target, which represents minimal error detection/correction and channel coding and high communication energy. On the other hand, a more relaxed BER target of $10^{-4}$, with complex channel coding employed, illustrates usage models where the energy cost of computation can dominate the energy cost of communication, particularly for cleaner wireless channels. In this paper, we have not considered the network aspect of the wireless node. Hence, we present results for both a conservative BER target and a relaxed BER
target that encompasses typical ranges for wireless nodes. We measure the total transmitted energy as a function of the total transmitted data volume (also referred to here as Tx load). For low path loss, the standby power dominates, however with increasing path loss and PA gain we see a near-linear increase in total transmission energy as a function of the data volume (Fig. 12a). Since, the volume of transmitted data decreases with PD, we can now estimate the total transmission energy per frame of video data as a function of PD, as shown in Fig. 12b. With clean channel (40dB path-loss), transmission energy per frame is 1mJ for transmission after PD1, while for noisy channel (70dB path-loss), transmission energy per frame can be as high as 17mJ.

The energy breakdown of the system is demonstrated in Fig. 13. Here, we can observe that in a noisy channel with a path-loss=70dB, transmitter energy occupies more total budget as compared to a clean channel. At the same time, with deeper processing depth, transmitter energy can be saved at the expense of computation energy. The overall self-optimization of total energy will be introduced in the section V.

V. SELF-OPTIMIZATION PROCEDURE AND SYSTEM SETUP

In the previous sections we have seen the strong trade-off between transmission energy, PD and the algorithm of choice. A self-optimizing system needs to be cognizant of this, and adjust its operating point dynamically based on the choice of algorithm and channel conditions.

A. Energy Model

We first develop a model for the total energy of the sensor node. The total energy, E, includes computation energy, $E_p$, and communication energy, $E_{TX}$; and is a function of temporal variables of information content (S), processing depth (PD), and path-loss, (PL), under the constraint of accuracy requirement, (Acc), as defined by application/cloud server when choosing the most-energy efficient algorithm, ALG.

$$E = E_p + E_{TX} = f(S, PD, PL), \quad Acc(ALG) > Acc_0;$$ (7)

Once the most-energy-efficient algorithm is chosen according to minimum accuracy requirement, computation energy is only a function of information content and processing depth independent of path-loss and it can be further decomposed into dynamic energy and static energy per frame. With processing period fixed at T, i.e., 1/frames per second, $E_p$ changes with processing time ($\tau_p$), a function of information content and processing depth. Large information content size, deep embedded processing and more complex algorithms will result in high computation energy. $P_{\text{dynamic, } p}$ and $P_{\text{static, } p}$ are the dynamic processing power and static processing power respectively which are obtained from the image processor measurement. The processing energy can then be expressed in terms of S, PD and other parameters as

$$E_p = f_1(S, PD) = P_{\text{dynamic, } p} \cdot \tau_p(S, PD) + P_{\text{static, } p} \cdot T$$
$$= \theta_{\text{ALG,PD}} S + E_{\text{static, } p}$$ (8)

where $\theta_{\text{ALG,PD}}$ is model coefficients of algorithm ALG at processing depth PD which is fitted via regression during pre-deployment testing and calibration.

Communication energy is modeled as a function of PL, power amplifier (PA) gain and the static power. The total energy to transmit each video frame is modeled as

$$E_{TX} = f_2(S, PD, PL)$$
$$= P_{\text{dynamic, } TX} \cdot \tau_p(S, PD) + P_{\text{static, } TX} \cdot T$$
$$= P_{\text{dynamic, } TX} (PL) \cdot \Gamma_{\text{ALG}}(S, PD) + E_{\text{static, } TX}$$ (9)

where $\Gamma_{\text{ALG}}(S, PD)$ is transmission load when processed by algorithm-ALG, processing depth of PD and information content of S, and DR is the data rate.

B. Self-Optimization Procedure

The over-all system first characterizes itself before deployment. On the test-bench, for different algorithms, PD and path loss conditions, the system performs energy calibration and determines the total energy for each IPP task and transmission. Then the system populates a look-up table (LUT) which contains information about possible operating conditions. This is currently implemented on a PC, but can be embedded if required. This calibration step can use external or embedded sensors (power/current sensors); and, in the present system we perform the calibration using external on-board sensors.

Calibration of the system is performed during test phase. This procedure is illustrated in the flow-chart shown in Fig. 14. The key algorithmic steps before the IoT node is deployed are:

1) The algorithms (combination of different compression ratios, feature extraction methods and classifiers) are characterized on a known (INRIA) data-base during design. The accuracy of the algorithms for the task at hand are determined.

2) During calibration phase, models for energy dissipation are constructed. A random value of path-loss is generated. A corresponding minimum power amplifier gain that satisfy the target BER is measured and the gain together with its $P_{\text{dynamic, } TX}$ are stored in the corresponding LUT entries.

3) LUT entries for the coefficient $\theta$ are populated for each algorithm and processing depth. Assuming a linear relationship and to avoid over-fitting, ten random testing energy measurements ($E_p$) against ten random information sizes (S) from a test video per PD and algorithm are used in the current setup. We use regression to calculate $\theta$. Videos in this calibration stage are obtained from ViSOR data-set, "Outdoor, Unimore D.I.I setup" category. It encompasses a large range of information content, from pixel sizes of 2400 (60x40) to 21600 (180x120). This allows us to obtain a comprehensive and accurate energy model which is critical for the success of the design. During run-time we test the setup with a real-time system with hours of videos obtained from the OV7670 image sensor. This allows us to obtain accurate measurements of energy consumption during
operation and perform online optimization between computation and communication energy. It should be noted that to train the system for human detection we used the INRIA image data-set, as has been mentioned, and performance/accuracy testing was done on hours of real-time videos acquired with the final system setup.

After deployment, information about path-loss is sent from back-end cloud to the front-end platform periodically (every 1s) and the minimum power amplifier gain needed to overcome path-loss is updated. Then the energy model estimates the energy for all the IPP blocks with respect to the information content. Then the system chooses the PD for minimum energy of operation. The PD information, algorithm, transmission gain and energy for IPP blocks are packed into the frame header and transmitted. This is used by cloud server for back-end processing. The calibration and run-time self-optimization scheme are shown in Fig. 14 and data/operations in time domain is shown in Fig. 15.

Upon obtaining accurate coefficients, the overhead of the self-optimized system is limited to storing the model parameters and modeling the computation/communication energy. The model, including PL-PA gain table, will consume no more than 40 bytes of memory in double-precision. For the system running at 10 frames per second, the maximum computation needed for the energy estimation is 70 MAC/second. For the overall system, both the model storage and energy estimation overheads are negligibly small.

VI. END-TO-END SYSTEM DEMONSTRATION AND MEASUREMENTS

The algorithms are implemented on ADI-BF707 image processing board and computation power consumption is measured. An example of measured power and the processing steps is shown in Fig. 16. We can observe the different processing steps through GPIO output (the IPP steps are alternatively active high and active low), and the corresponding power consumption. During pre-deployment calibration, the LUT is populated and the energy models are constructed for varying path-loss and information content of the captured video frames. Based on the LUT data, the system chooses the operating mode for minimum energy per frame. This is shown in Fig. 17 where different PL scenarios are examined. As the PL increases, the self-optimizing sensor node always chooses the most power optimal PD. We note that the increasing path-loss will result in more embedded computation and total energy is saved on the self-optimizing platform.
Fig. 18. Measured total energy (computation + communication) per frame for the proposed system compared against two static designs. Experimental results are demonstrated for three algorithms and two BER targets.

Also, improved energy-efficiency will be achieved with low-power algorithms, Algorithm-3 for example, or lower target BER, i.e. $10^{-4}$. Comparisons on total energy per frame is also demonstrated among different design strategies in Fig. 18. We compare the results of the proposed system vis-a-vis two static designs. These are:

1) **Full-Transmission**: In this design the sensor node only performs image acquisition, localization and compression, and then transmits the entire video data.

2) **Full-Computation**: In this design the sensor node performs all the tasks in the IPP without considering the energy cost of computation, independent of the channel conditions.

We note that by properly balancing the energy for computation and communication, the proposed system always operates at minimum energy point. We measure peak saving of $4.3 \times$ at 70dB path-loss, operating with Algorithm-1 and target BER of $10^{-8}$, when compared with baseline design (Full-Transmission Design). For a target BER of $10^{-4}$, the proposed system shows $2.2 \times$ to $3.1 \times$ peak savings. A random path-loss scenario is generated and its impacts on PD, PA gain, computation energy per frame, communication energy per frame and total energy is demonstrated in Fig. 19.(a). We note how in transient mode the system operated at the correct PD to track minimum overall energy by trading computation for communication energy when channel is noisy (high path-loss). Also, with lower BER requirement as is shown in Fig. 19.(b), the system performs less computation (no PD = 3 mode is observed) and operates at smaller PA gains. Energy per frame under different environments are also shown. Finally, the end-to-end system is deployed on a mobile IoT platform and various indoor and outdoor conditions are used to evaluate the potential of the design. Path-loss as a function of distance between the IoT node and the base-station for various wireless conditions are shown in Fig. 20. For these operating conditions, we compare the total energy/frame dissipated in the proposed system vis-a-vis “Full-Transmission” and “Full-Computation” designs. The comparative results for two BER targets are shown in Fig. 21. We note that the proposed system saves significant energy during run time and the optimal balance between computational energy and communication energy is obtained.

Fig. 22 shows the comparison with state-of-art designs on low-power wireless video applications. Previous research efforts have been focused on either (1) embedded low-power video processing [35]–[37], such as SRAM-FPGA based on-board object detection, or (2) adaptive wireless communi-
cation which adjusts the PA power and transmitter linearity with the dynamic wireless channel conditions [10], [38]. To the best of, our knowledge this is the first reported work where the computational and communication energies are being co-optimized for achieve the highest energy efficiency. To compare the proposed system with published results, the power numbers reported are normalized to the image size (320 × 240), maximum TX output power (20dBm) to estimate the final metric of energy per frame. The comparison shows that the proposed system outperforms state-of-art design by more than 2 × .

VII. CONCLUSION

This paper presents a video IoT sensor node which performs self-optimization between the amount of computation (for human detection) and the total data volume to be transmitted. As the information content and the channel conditions change, the system tracks the minimum energy point. Hardware measurements show 4.3 × reduction of the total energy/frame compared to a baseline design. Comparisons with state-of-the-art video based sensor nodes, we note more than 2 × reduction in energy/frame.

REFERENCES


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 joined 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.

Saad Bin Nasir (S’13) received the B.S. degree in electrical engineering from the National University of Sciences and Technology, Islamabad, Pakistan, in 2010, and the M.S. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, in 2014, where he is currently pursuing the Ph.D. degree. His industry experience includes 3+ years as a Design Engineer with the Center for Advanced Research in Engineering, Islamabad, and as a Research Intern with Intel Labs and Qualcomm. He has authored or co-authored over 20 journal and conference publications. His current research interests include analog/digital/mixed-signal circuit design for power management in high-performance servers, mobile devices, and Internet of Things. He was a recipient of the 2013–2014 Fulbright Fellowship, the 2016–2017 International Solid-State Circuits Society Pre-Doctoral Achievement Award, and the Best Student Paper Award at HOST 2017 and TECHCON 2016 conferences. He is a finalist of the DAC Ph.D. Forum 2015 and the Qualcomm Innovation Fellowship 2015.

Shreyas Sen (S’06–M’11) received the Ph.D. degree in electrical and computer engineering from Georgia Tech, Atlanta, USA, in 2011. He has 5+ years of industry experience as a Research Scientist with the Circuit Research Laboratory and Wireless Communication Research, Intel Labs, and as a Research Intern in Qualcomm and Rambus. He is currently an Assistant Professor with the School of Electrical and Computer Engineering, Purdue University. He has authored or co-authored two book chapters, over 85 journal and conference papers, and has 11 patents granted/pending. His current research interests include mixed-signal circuits/systems for Internet of Things, biomedical and security. He was a recipient of the NSF CISE CRII Award 2017, the AFSOR Young Investigator Award 2015, the Google Faculty Research Award 2017, the Intel Labs Divisional Recognition Award 2014 for industry wide impact on USB-C type, the Intel Ph.D. Fellowship 2010, the IEEE Microwave Fellowship 2008, the GSRC Margarida Jacome Best Research Award 2007, the HOST Best Student Paper Award 2017, the ICCAD Best-in-Track Award 2014, the VTS Honorable Mention Award 2014, the RWS Best Paper Award (Silver) 2008, the Intel Labs Quality Award 2012, the SRC Inventor Recognition Award 2008, and the Young Engineering Fellowship in 2005. He serves has served as an Associate Editor of the IEEE Design & Test and Technical Program Committee Member of DATE, ICCAD, ITC, VLSI Design, and VDAT.

Arijit Raychowdhury (M’07–SM’13) received the Ph.D. degree in electrical and computer engineering from Purdue University, in 2007. He joined Georgia Tech in 2013. He is an Associate Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, where he currently holds the ON Semiconductor Junior Professorship. His industry experiences includes five years as a Staff Scientist with the Circuits Research Lab, Intel Corporation, and a year as an Analog Circuit Researcher with Texas Instruments Inc. He holds over 25 U.S. and international patents and has authored or co-authored over 100 articles in journals and refereed conferences. His current research interests include low-power digital and mixed-signal circuit design, device-circuit interactions and novel computing models and hardware realizations. 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, College of Engineering, Purdue University, in 2007; and multiple best paper awards and fellowships.