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

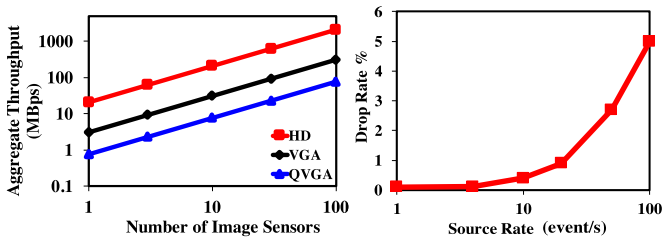
Ningyuan Cao, *Student Member, IEEE*, Saad Bin Nasir, *Student Member, IEEE*, Shreyas Sen, *Member, IEEE*, and Arijit Raychowdhury, *Senior Member, IEEE*

***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 choses 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 Thing 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 results 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 in the worst case [5]. This data back-log on cloud servers 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 both 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.




(a) (b)

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].

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

Manuscript received December 8, 2016; revised April 30, 2017 and June 3, 2017; accepted June 6, 2017. Date of publication July 7, 2017; date of current version August 28, 2017. This work was supported by the Semiconductor Research Corporation under Grant 2720.001. This paper was recommended by Associate Editor M. Alioto. *(Corresponding author: Ningyuan Cao.)*

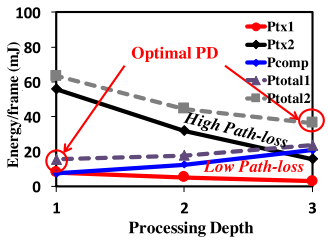
N. Cao, S. B. Nasir, and A. Raychowdhury are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: nycao@gatech.edu; saadbinnasir@gatech.edu; arijit.raychowdhury@ece.gatech.edu).

S. Sen is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: shreyas@purdue.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/TCSI.2017.2716358

| Processing Depth (PD) | Operations |
|---|---|
| 1 | Object Segmentation+ Compression |
| 2 | Object Segmentation+ Compression+ Feature Extraction |
| 3 | Object Segmentation+ Compression+ Feature Extraction+ Classification |




*(a)* *(b)*

Fig. 2. (a) Pipelined operations at different processing depth (PD), including temporal difference of consecutive frames (TD), compression (CR), feature extraction (FE) and classification (CL). (b) Power consumption changes with PD and the optimal PD for minimum-power consumption also varies under different channel conditions. For example, a noisy channel results in more embedded processing.

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. Our experimental results show measurements in a dynamic environment where both the information content of the video and the channel conditions are constantly changing. This is due to (1) a mobile sensor node and (2) time varying path-loss.

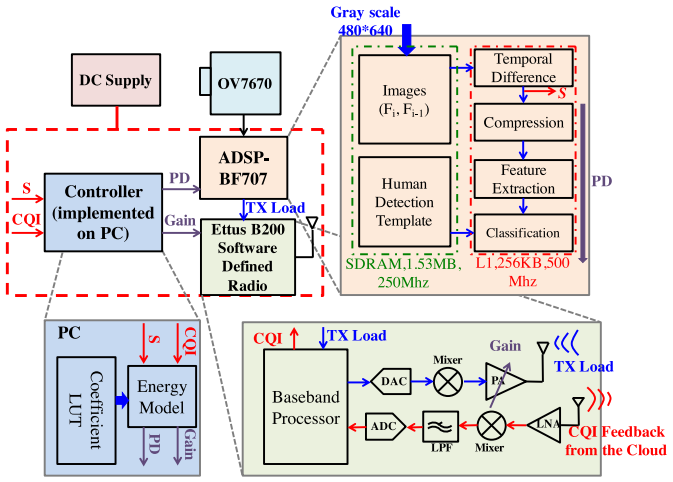



Fig. 3. End-to-end system architecture showing the different hardware components, the data processing pipeline and the software defined transceiver. CQI is the channel quality index quantified by path-loss and S is the information content size which will be defined in Section III.

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

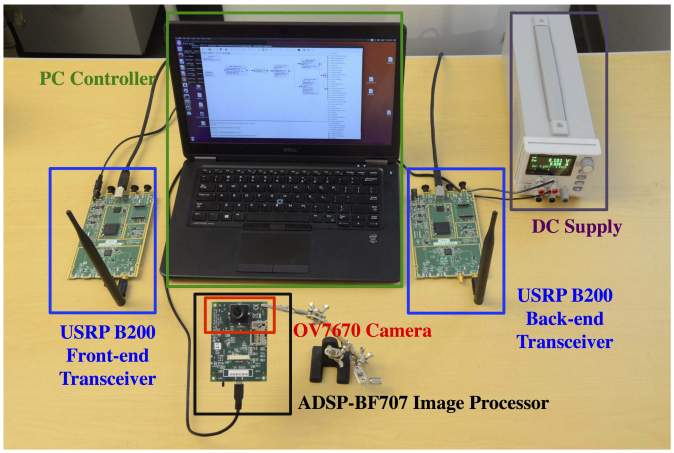


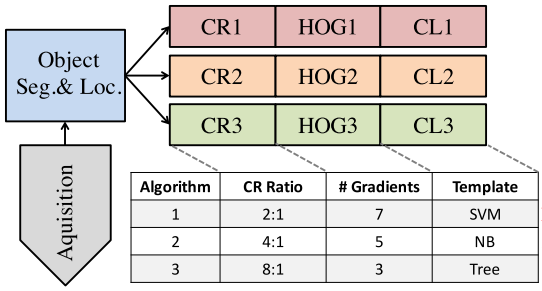

Fig. 4. Experimental setup showing the system components.

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. 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 com-




| Algorithm | CR Ratio | # Gradients | Template | |
|---|---|---|---|---|
| 1 | 2:1 | 7 | SVM | High Accuracy |
| 2 | 4:1 | 5 | NB | Nominal |
| 3 | 8:1 | 3 | Tree | Low Power |

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.

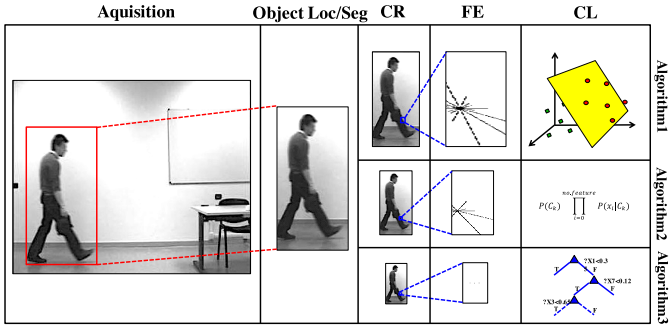



Fig. 6. Algorithm demonstration with a real video frame.

plexities 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 compresses the input frame at the least compression ratio, extracts feature with the most gradients and classifies the feature descriptor with the most computationally-intensive SVM template; and thus achieves best performance in terms of human detection accuracy.On the contrary, Algorithm-3 adopts maximum compression of the acquired frame, extracts the least number of gradient feature and applies the tree based template, which offers a low-power implementation apt for severely energy-constrained systems. Algorithm-2 is the nominal design point. The design parameters of each algorithm are also listed in the Fig. 5. Depending on the trade-off between accuracy requirement and energy budget, a particular algorithm should be selected. This offers programmability on the platform for specific applications and energy constraints. Fig. 6 demonstrates how a single frame with a moving object is processed through the IPP and each stage of the IPP are described below.

### 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]. Optical flow method can obtain complete information and detect the moving object from background better, but requires clustering, 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 energy threshold, $E_{th}$, are labeled as activated pixels with label value of 1. Otherwise, label value 0 is assigned. This can be summarized as:

$$D_i(m,n) = |F_i(m,n) - F_{i-1}(m,n)| \tag{1}$$

$$L_i(m,n) = \begin{cases} 0, |D_i(m,n) - D_{i-1}(m,n)| \le E_{th} \\ 1, |D_i(m,n) - D_{i-1}(m,n)| > E_{th} \end{cases} \tag{2}$$

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 that 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 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 $(-\pi, \pi]$ range and number of bins is $N_{bin}$. Then the spatially connected cells form a block of size

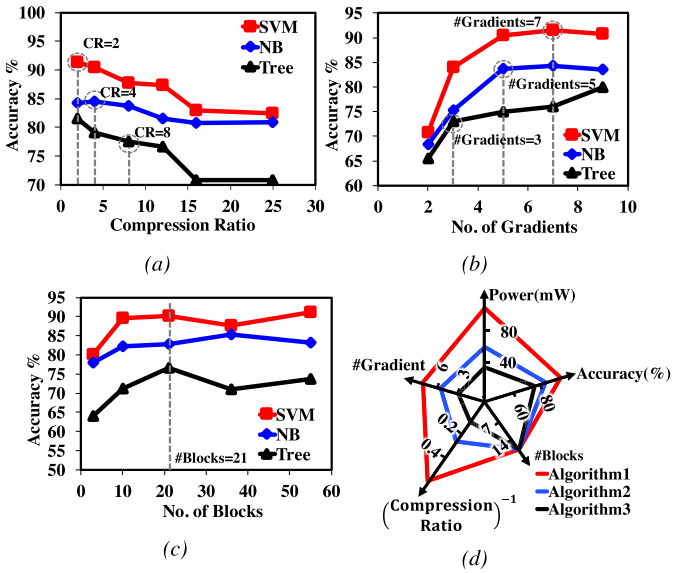



Fig. 7. (a) Measured detection accuracy vs. compression ratio. (b) Measured detection accuracy vs. number of gradients extracted from HOG feature extraction. (c) Measured detection accuracy vs. number of blocks to extract feature vectors in HOG feature extraction. (d) Power consumption and accuracy at design points in different algorithms.

(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

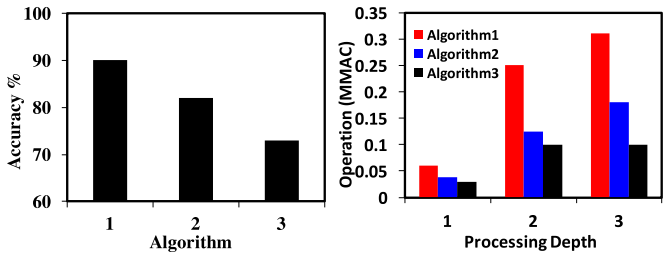
Classification is the final step in the IPP. The classifier is trained offline in testing phase and classification template is generated and stored in the SDRAM. Different machine learning classifiers have different performance-power trade-offs. We employ three different classification schemes depending 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, Naïve 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 classificaiton [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:

$$Y(\vec{x}) = sign\left(\sum_{i=0}^{N_{sup}} \alpha\_i y_i \vec{V}_i \vec{x} + \beta_i\right) \tag{3}$$

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.

*Naïve Bayes (NB) classifier* [20]–[22] assumes strong independence between individual descriptors and applies Bayes'




*(a)* *(b)*

Fig. 8. (a) Measured human detection accuracy with three different algorithms. (b) Number of estimated operations in millions of multiplication-accumulation-counts (MMAC) for different algorithms/depths.

theorem, which describes stochastic event based on related conditions, on test data to predict class:

$$P(C_k|\vec{x}) = \frac{P(C_k)}{P(\vec{x})}\prod_{i=1}^{N_{in}} p(x_i|C_k); \quad (4)$$

$$C_K = argmax_{\ C_k}\ P(C_k|\vec{x}) \quad (5)$$

where $C_k$ is the kth class and $\vec{x}$ is the input test descriptor. $P(C_k)$, $P(\vec{x})$ and $p(x_i|C_k)$ are constants and obtained from training and the predicted class, $C_K$, is the one with highest conditional probability for all classes. In the current set-up, there are only two labels: human and non-human.

*Binary tree or decision tree*, is composed of nodes, branches and leaves representing test, test-outcome and labels respectively [23], [24]. The configuration of a tree, including shape and test condition on each node is obtained by observation. The tree classifier in our application is binary, which means, starting from root node, on each node, one of the predictors in feature set is compared with a certain threshold. Then it either goes left or right depending on comparison result till it reaches the one of the leaves which contains one of the classification results.

### *E. Comparative Analysis of Classification Schemes*

The three classification schemes have been mapped to the ADI camera processor and optimized for minimum area on the on-board memory. Benchmarking is carried out on the INRIA human dataset [16], [17]. Tree classifiers use cascaded comparators of depth 10, and are the most energy efficient scheme. SVMs demonstrate highest performance but require more than 500 support vectors and hence dissipate the highest power. NB shows nominal performance and power dissipation. Fig. 7 illustrates how detection accuracy changes with CR, number of gradients and number of blocks in different classification algorithms. Accuracy improves when CR is low and more gradient features where number of blocks in feature extraction does not show strong tendency. The design parameters for each algorithm selected for our platform are also denoted in the figure. The compression ratio are designed as 2:1, 4:1 and 8:1 for three algorithm with 3, 5 and 7 gradient orientations in each block of feature extraction. Number of blocks in extracting gradient orientation is designed to be 21 for all algorithms.

Accuracy measurements in Fig. 8a were carried out on human detection database, INRIA [16] because of its relevance to surveillance. The design parameters are chosen from Fig. 7. Algorithm-1 is designed to provide a target accuracy of 91% while Algorithms-2 and 3 provide target accuracy of 83% and 77% respectively. Fig.8b illustrates how 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.

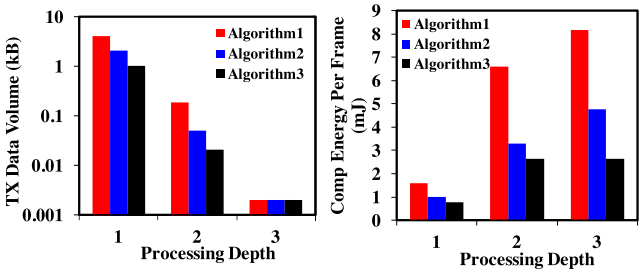
## IV. Adaptive Wireless Communication

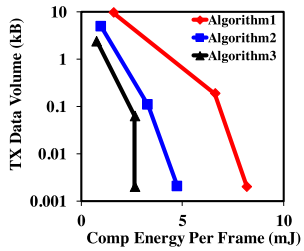
Wireless communication conventionally is the major cause of energy expenditure and shortened lifetime of wireless sensors, especially when the sensors are experiencing expanding bandwidth, rapid growth of nodes and ever-increasing data volume with the development Internet of Things [6], [7]. To implement 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]

$$Path\_loss = 20log_{10}(\frac{4\pi df}{c}) \quad (6)$$

Here d is distance, f is the carrier frequency and c is the speed of light. In our design, the carrier frequency is 985Mhz.




| Accuracy Requirement | Algorithm | Estimated $E_{comp}$ per Frame (mJ) | | | Estimated TX Data Volume (kB) | | |
|---|---|---|---|---|---|---|---|
| | | PD=1 | PD=2 | PD=3 | PD=1 | PD=2 | PD=3 |
| Highest | 1 | 1.6 | 6.6 | 8.2 | 4 | 0.15 | 0.002 |
| Nominal | 2 | 1 | 3.3 | 4.8 | 2 | 0.11 | 0.002 |
| Lowest | 3 | 0.8 | 2.6 | 2.61 | 1 | 0.06 | 0.002 |

(d)

Fig. 9. (a) Measured transmission load vs. processing depth with different algorithms and PD. (b) Measured front-end computation energy per frame vs. processing depth. (c) Estimated Tradeoff between transmission data volume with computation energy (d) different detection accuracy requirements result in different algorithm chosen, computation energy (Ecomp) and transmission data volume.

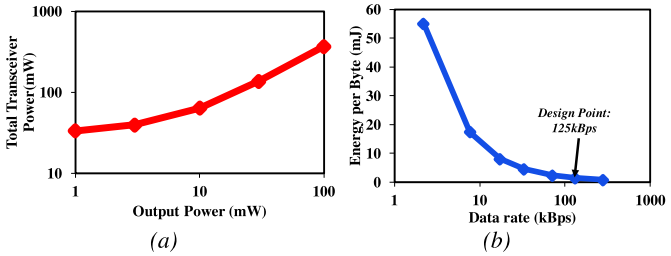



Fig. 10. Measured (a) transceiver power vs. output power. (b) energy per byte vs. data rate.

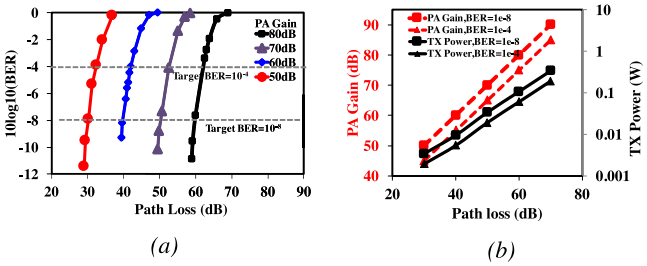



Fig. 11. Measured (a) Bit-error-rate vs. path-loss under different PA gain. (b) PA gain and transceiver power vs. path loss under BER requirement of $10^{-4}$ and $10^{-8}$.

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 BER=$10^{-4}$ 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

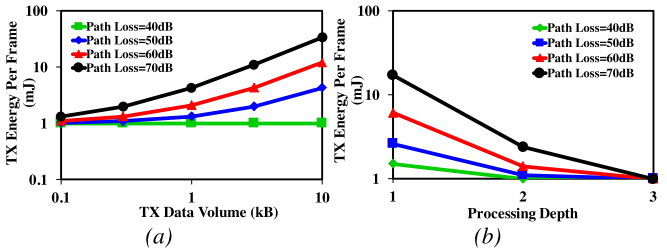



Fig. 12. Measured (a) transmission energy per frame vs. transmission data volume under various channel conditions. (b) Transmission energy per frame vs. processing depth under different path-loss conditions.

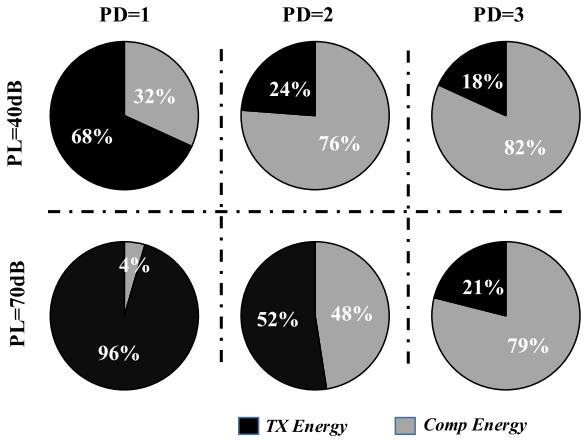



Fig. 13. Breakdown of computation energy and TX energy in different processing depth and path-loss.

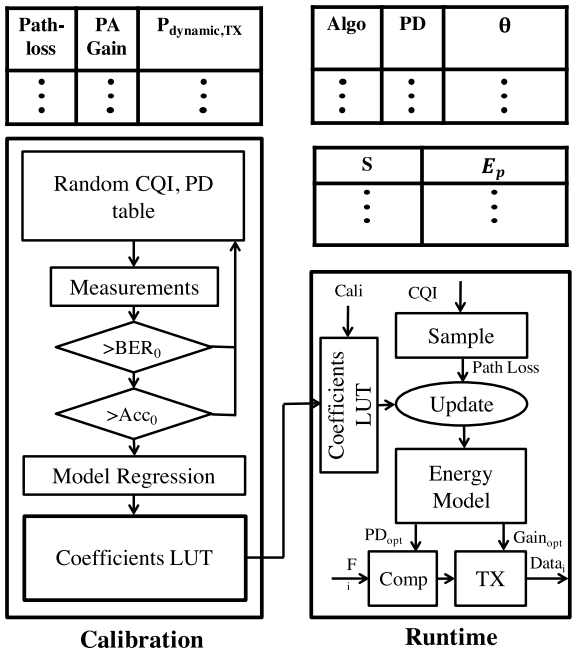



Fig. 14. Calibration and runtime self-optimization scheme.

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, ($Acc_0$), as defined by application/cloud server when choosing the most-energy efficient algorithm, ALG.

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

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. $P_{dynamic,p}$ and $P_{static,p}$ 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

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

where $\theta_{ALG,PD}$ 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

$$\begin{aligned} E_{TX} &= f_2(S, PD, PL) \\ &= P_{dynamic,TX} \cdot \tau_p(S, PD) + P_{static,TX} \cdot T \\ &= P_{dynamic,TX}(PL) \cdot \frac{\Gamma_{ALG}(S, PD)}{DR} + E_{static,TX} \end{aligned} \tag{9}$$

where $\Gamma_{ALG}$(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 $P_{dynamic,TX}$ 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 processing 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 (60×40) to 21600 (180×120). 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

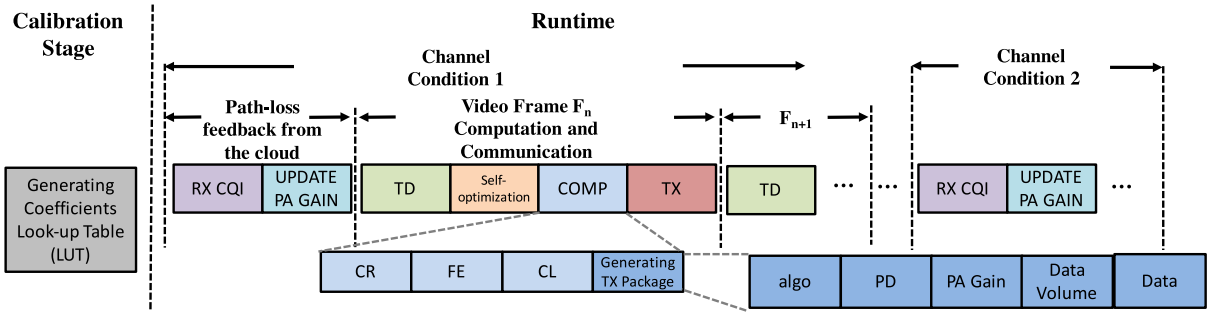



Fig. 15. Data packet configuration and modes of transmission-reception for the wireless link.

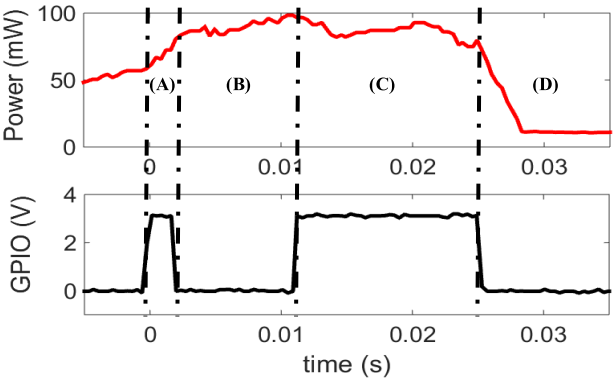



Fig. 16. Measured embedded computation power consumption where transient congrol signal of each processing stage is indicated by GPIO output voltage level: (A), object segmentation and localization through temporal difference (TD) together with compression (CR); (B), feature extraction(FE); (C), classification (CL) and finally (D), idle power down state. Note that alternative opertions have alternative active-high and active-low control signals. For example, (A) is active-high, (B) active-low and so on.

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.

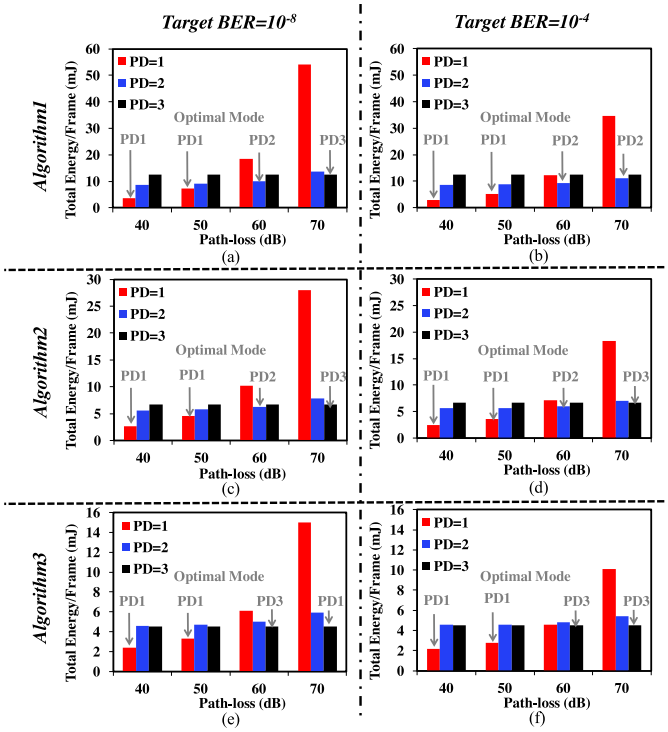



Fig. 17. Measured total energy (computation+communication) per frame for different PD with increasing path-loss. Experimental results are demonstrated for the three algorithms described here and two BER targets.When path-loss is high, the general trend is that optimal mode moves to more front-end embedded processing.

## 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.

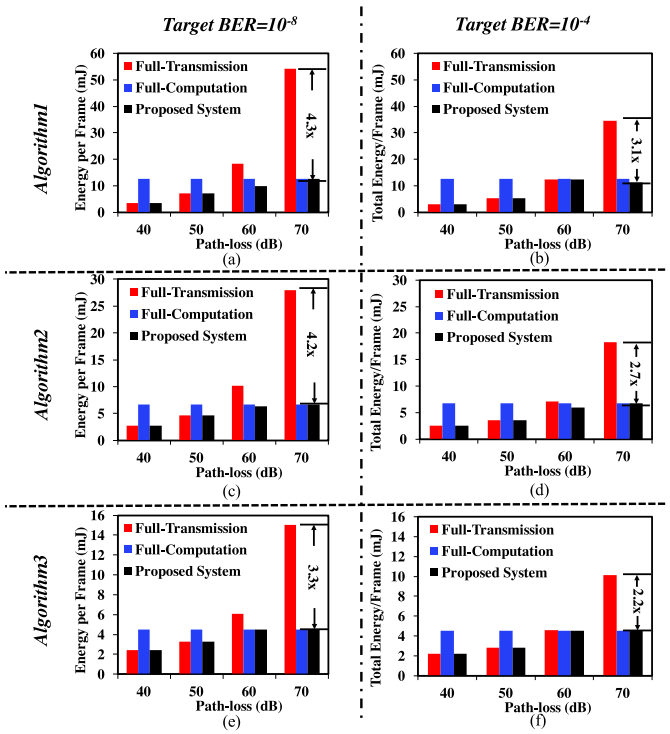



Fig. 18. Measured total energy (computation+communication) per frame for the proposed system comapred against two static designs.Experimental results are demonstrated for three algorithms and two BER targets.

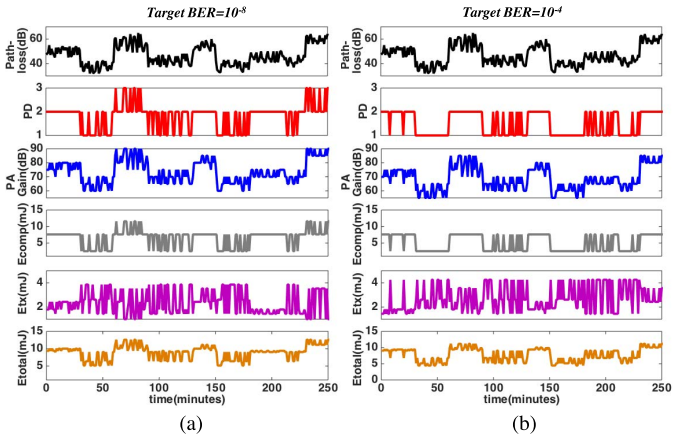



Fig. 19. Case Study: Random and dynamic path-loss condition created by a mobile IoT node and the corresponding PD, PA gain, computation, transmission and total energy per frame under BER constraints of (a) $10^{-8}$ and (b) $10^{-4}$.

Also, improved energy-efficiency will be achieved with low-power algorithm, 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.

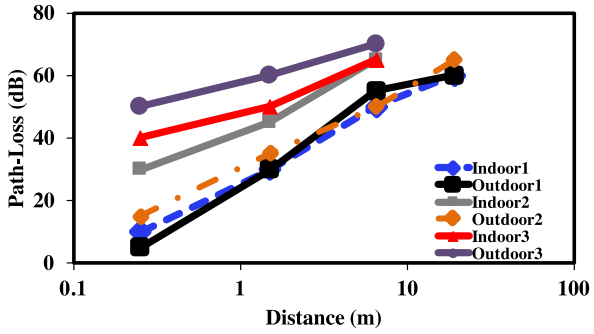



Fig. 20. Path-loss measurements under different indoor and outdoor environments.

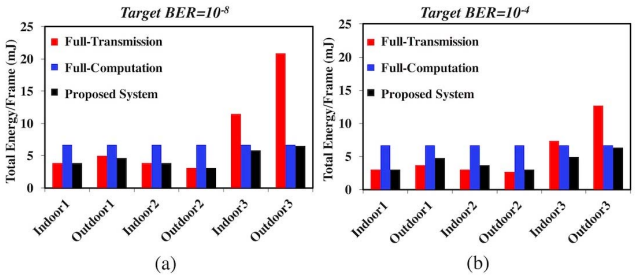



Fig. 21. Measured total energy (average) per frame in different environments vis-a-vis static designs under BER targets of (a) $10^{-8}$ and (b) $10^{-4}$.

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× 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× to 3.1× 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 environment 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 conditons, 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-

| | *System/ design* | *Application/algorithm* | *Embedded Processing Unit Parameter* | *Image size* | *Performance* | *Max. output TX power* | *Energy consumption* | *Estimated energy per frame* |
|---|---|---|---|---|---|---|---|---|
| *Embedded Processing* | [35] | Object Detection & tracking | $V_{core}$=1.2V $f_{clk}$= 60-160Mhz tech. node = 45nm | 320x240 | fps=2 | N.A. | 319.9 mJ/frame | >319.9 mJ/frame |
| | [36] | Particle detection/meter/reading/ people counting | $V_{core}$=1.2V $f_{clk}$= 520Mhz tech. node = 45nm | 640x400 | fps=16-48 | 5 dBm | 17.2-22.8 mJ/frame | 17.5-29.2 mJ/frame |
| | [37] | Particle detection | $V_{core}$=1.2V $f_{clk}$= 60-160Mhz tech. node = 45nm | 640x400 | fps=48 | N.A. | 14.12 mJ/frame | >14.12 mJ/frame |
| *Adaptive Wireless* | [38] | Video offloading | N.A. | N.A. | data rate=1.5 Mb/s | N.A. | 1223 mW | 162.8 mJ/frame |
| | [10] | Sensor node | N.A. | N.A. | BW=2.2-2.6 Ghz | 14.5-23.5 dBm | 165-912 mW | 6.1-34 mJ/frame |
| *Embedded Processing + Adaptive Wireless* | Proposed Work | Human detection | $V_{core}$=1.1V $f_{clk}$= 100-400Mhz tech. node = 65nm | 320x240 | fps=10 | 20 dBm | 4-7 mJ/frame | 4.8-8.4 mJ/frame |

Fig. 22. Comparison table: The proposed system has been compared with state-of-the-art video based sensor nodes which either (1) perform "in-sensor" video processing, or (2) improve energy-efficiency of the wireless transmitter through real-time adaptation. The proposed system performs self-optimization between the computation and communication to enable the lowest power consumption in a dynamic environment.

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


[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[2] F. Ren, T. He, S. K. Das, and C. Lin, "Traffic-aware dynamic routing to alleviate congestion in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 9, pp. 1585–1599, Sep. 2011.

[3] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion detection and avoidance in sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sens. Syst. (SenSys)*, New York, NY, USA, 2003, pp. 266–279.

[4] T. Pal, S. Bandyopadhyay, and S. Dasbit, "Energy-saving image transmission over WMSN using block size reduction technique," in *Proc. IEEE Int. Symp. Nanoelectron. Inf. Syst.*, Dec. 2015, pp. 41–46.

[5] S. Ehsan and B. Hamdaoui, "A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 2, pp. 265–278, May 2012.

[6] Z. Takhirov, J. Wang, V. Saligrama, and A. Joshi, "Energy-efficient adaptive classifier design for mobile systems," in *Proc. Int. Symp. Low Power Electron. Des. (ISLPED)*, New York, NY, USA, 2016, pp. 52–57.

[7] A. Anvesha, S. Xu, N. Cao, J. Romberg, and A. Raychowdhury, "A light-powered, 'Always-On', smart camera with compressed domain gesture detection," in *Proc. Int. Symp. Low Power Electron. Des. (ISLPED)*, New York, NY, USA, 2016, pp. 118–123.

[8] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[9] A. D. Wood *et al.*, "Context-aware wireless sensor networks for assisted living and residential monitoring," *IEEE Netw.*, vol. 22, no. 4, pp. 26–33, Jul. 2008.

[10] S. Sen, R. Senguttuvan, and A. Chatterjee, "Environment-adaptive concurrent companding and bias control for efficient power-amplifier operation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 3, pp. 607–618, Mar. 2011.

[11] N. Cao, S. Nasir, S. Sen, and A. Raychowdhury, "In-sensor analytics and energy-aware self-optimization in a wireless sensor node," in *Proc. IEEE Int. Microw. Symp. (IMS)*, Jun. 2017, pp. 1–4.

[12] Z. Yi and F. Liangzhong, "Moving object detection based on running average background and temporal difference," in *Proc. IEEE Int. Conf. Intell. Syst. Knowl. Eng.*, Nov. 2010, pp. 270–272.

[13] M. K. Chowdary, S. S. Babu, S. S. Babu, and H. Khan, "FPGA implementation of moving object detection in frames by using background subtraction algorithm," in *Proc. Int. Conf. Commun. Signal Process.*, Apr. 2013, pp. 1032–1036.

[14] I. Haritaoglu, D. Harwood, and L. S. Davis, "$W^4$: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[15] K. Kinoshita, M. Enokidani, M. Izumida, and K. Murakami, "Tracking of a moving object using one-dimensional optical flow with a rotating observer," in *Proc. 9th Int. Conf. Control, Auto. Robot. Vis.*, Dec. 2006, pp. 1–6.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, vol. 1. no. 1, pp. 886–893.

[17] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Jun. 2006, pp. 1491–1498.

[18] E. Osuna, R. Freund, and F. Girosit, "Training support vector machines: An application to face detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 130–136.

[19] P. Michel and R. El Kaliouby, "Real time facial expression recognition in video using support vector machines," in *Proc. 5th Int. Conf. Multimodal Interfaces (ICMI)*, New York, NY, USA, 2003, pp. 258–264.

[20] S. McCann and D. G. Lowe, "Local naive Bayes nearest neighbor for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3650–3656.

[21] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proc. 11th ACM Int. Conf. Multimedia (MULTIMEDIA)*, New York, NY, USA, 2003, pp. 2–10.

[22] D. J. Moore, I. A. Essa, and M. H. Hayes, "Exploiting human actions and object context for recognition tasks," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 1. Sep. 1999., pp. 80–86

[23] B. Wu and R. Nevatia, "Cluster boosted tree classifier for multi-view, multi-pose object detection," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[24] M. A. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote Sens. Environ.*, vol. 61, pp. 399–409, Sep. 1997.
[25] D. Banerjee, S. Devarakond, S. Sen, and A. Chatterjee, "Real-time use-aware adaptive MIMO RF receiver systems for energy efficiency under BER constraints," in *Proc. 50th ACM/EDAC/IEEE Des. Auto. Conf. (DAC)*, May 2013, pp. 1–7.
[26] D. Banerjee, S. Sen, and A. Chatterjee, "Self learning analog/mixed-signal/RF systems: Dynamic adaptation to workload and environmental uncertainties," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD)*, Nov. 2015, pp. 59–64.
[27] S. Sen, D. Banerjee, M. Verhelst, and A. Chatterjee, "A power-scalable channel-adaptive wireless receiver based on built-in orthogonally tunable LNA," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 5, pp. 946–957, May 2012.
[28] S. Sen, V. Natarajan, S. Devarakond, and A. Chatterjee, "Process-variation tolerant channel-adaptive virtually zero-margin low-power wireless receiver systems," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 33, no. 12, pp. 1764–1777, Dec. 2014.
[29] S. Sen, V. Natarajan, R. Senguttuvan, and A. Chatterjee, "Pro-VIZOR: Process tunable virtually zero margin low power adaptive RF for wireless systems," in *Proc. 45th ACM/IEEE Des. Auto. Conf.*, Jun. 2008, pp. 492–497.
[30] S. Sen, "Invited: Context-aware energy-efficient communication for IoT sensor nodes," in *Proc. 53rd Annu. Des. Auto. Conf. (DAC)*, New York, NY, USA, 2016, pp. 67:1–67:6.
[31] D. Banerjee, B. Muldrey, S. Sen, X. Wang, and A. Chatterjee, "Self-learning MIMO-RF receiver systems: Process resilient real-time adaptation to channel conditions for low power operation," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD)*, Nov. 2014, pp. 710–717.
[32] D. Banerjee, B. Muldrey, X. Wang, S. Sen, and A. Chatterjee, "Self-learning RF receiver systems: Process aware real-time adaptation to channel conditions for low power operation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 1, pp. 195–207, Jan. 2017.
[33] S. Sen, R. Senguttuvan, and A. Chatterjee, "Concurrent PAR and power amplifier adaptation for power efficient operation of WiMAX OFDM transmitters," in *Proc. IEEE Radio Wireless Symp.*, Jan. 2008, pp. 21–24.
[34] S. Y. Seidel and T. S. Rappaport, "914 MHz path loss prediction models for indoor wireless communications in multifloored buildings," *IEEE Trans. Antennas Propag.*, vol. 40, no. 2, pp. 207–217, Feb. 1992.
[35] M. Imran, K. Shahzad, N. Ahmad, M. O'Nils, N. Lawal, and B. Oelmann, "Energy-efficient SRAM FPGA-based wireless vision sensor node: SENTIOF-CAM," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 12, pp. 2132–2143, Dec. 2014.
[36] M. Casares and S. Velipasalar, "Adaptive methodologies for energy-efficient object detection and tracking with battery-powered embedded smart cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1438–1452, Oct. 2011.
[37] M. Imran, K. Khursheed, N. Lawal, M. O'Nils, and N. Ahmad, "Implementation of wireless vision sensor node for characterization of particles in fluids," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 11, pp. 1634–1643, Nov. 2012.
[38] L. Zhang, D. Fu, J. Liu, E. C.-H. Ngai, and W. Zhu, "On energy-efficient offloading in mobile cloud for real-time video applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 1, pp. 170–181, Jan. 2017.

**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 AFOSR Young Investigator Award 2017, 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 experience 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 recepient 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.