Towards Efficient Wireless Video Sensor Networks: A Survey of Existing Node Architectures and Proposal for A Flexi-WVSNP Design

Adolph Seema and Martin Reisslein

Abstract—The video capture, processing, and communication in wireless video sensor networks critically depend on the resources of the nodes forming the sensor networks. We provide a survey of wireless video sensor node platforms (WVSNPs). From a comprehensive literature review, we first select the node architectures that meet basic requirements for a WVSNP. We then introduce a classification of WVSNPs into general purpose architectures, heavily coupled architectures, and externally dependent architectures. We thoroughly survey and contrast the existing WVSNPs within this classification framework. Based on the insights from our survey we develop a novel Flexi-WVSNP design. The Flexi-WVSNP design includes dual-radio communication, a middleware for sensor operation and communication control, as well as a cohesive hardware and software design.

Index Terms—Dual-radio, middleware, in-network processing, video streaming, Zigbee.

I. INTRODUCTION

WIRELESS sensor networks capable of capturing video at distributed video sensor nodes and transmitting the video via multiple wireless hops to sink nodes have received significant interest in recent years [1]–[6]. Wireless video sensor networks have been explored for a wide range of applications, including computer vision [7], [8], video tracking [9], [10] and locating [11], video surveillance [12]–[14], remote live video and control [15], [16], and assisted living [17], [18]. Many aspects of wireless video sensor networks have been extensively researched, including multi-tier network structures, e.g., [19]–[21], multisensor image fusion [22], [23], image and video compression techniques, e.g., [24], [25], wireless communication protocols, e.g., [26], distributed algorithms, e.g., [17], light-weight operating systems and middleware, e.g., [20], [27], [28], and resource allocation strategies [29]–[31]. Generally, a large portion of the research has focused on software-based mechanisms. Several toolkits, e.g., [32]–[35], have been developed to facilitate software based video sensor network research.

In this survey, we focus on the wireless video sensor nodes forming the sensor network. We comprehensively survey the existing wireless video sensor node platforms (WVSNPs) considering the hardware and software components required for implementing the wireless video sensor node functionalities. All functional aspects of a wireless video sensor network ranging from the video capture and compression to the wireless transmission and forwarding to the sink node depend critically on the hardware and software capabilities of the sensor node platforms. Moreover, the sensor node platform designs govern to a large extent sensor network performance parameters, such as power consumption (which governs network lifetime), sensor size, adaptability, data security, robustness, and cost [36]. Also, computation capabilities, which are important for video compression, and wireless communication capabilities, which are important for the wireless transport from the source node over possibly multiple intermediate nodes to the sink node, are determined by the node platforms. An in-depth understanding of the state-of-the-art in WVSNPs is therefore important for essentially all aspects of wireless video sensor network research and operation. To the best of our knowledge, there is no prior survey of the field of wireless video sensor node platforms. Closest related to our survey are the general review articles on the components of general wireless (data) sensor networks, e.g., [5], [36]–[38], which do not consider video sensing or transmission, and the general surveys on multimedia sensor networks, e.g., [1], [4], which include only very brief overviews of sensor platforms.

Toward providing communications and networking generalists with an in-depth understanding of wireless video sensor node platforms (WVSNPs) and their implications for network design and operation, we first briefly review the requirements for WVSNPs in Section II. In Section II we also define ideal requirements for the power consumption, throughput of video frames, and cost of WVSNPs suitable for practical networks. Our exhaustive literature review revealed that currently no existing platform meets the ideal practical requirements. We therefore relax our requirements in Section III and according to the relaxed requirements select about a dozen platforms for detailed review. We introduce a classification structure of WVSNPs consisting of the categories: general purpose architectures, heavily coupled architectures, and externally dependent architectures. In Sections IV through VI we critique the existing WVSNPs following our classification structure. For each critiqued WVSNP we examine overall structure and resulting advantages and disadvantages for the wireless video sensor node functionalities, including video capture and encoding as well as wireless video transmission. In Section VII we summarize the insights gained from our detailed survey, including the key shortcomings that cause existing WVSNPs to fail the ideal practical requirements. Building on these insights, we propose in Section VIII a novel Flexi-WVSNP design that addresses the shortcomings of existing WVSNPs through a number of innovative architectural features, includ-

ing a cohesive integration of hardware and software and a dual-radio. We summarize this survey article in Section IX.

II. REQUIREMENTS FOR WIRELESS VIDEO SENSOR NODE PLATFORMS

In this section we review the sensor node requirements and define our ideal, yet reasonable practical requirements for a wireless video sensor node platform (WVSNP). From detailed reviews of the requirements for WVSNPs, e.g., [1], [3], [39], we identified three core requirements, namely power consumption, throughput, and cost and summarize these core requirements as follows. The power requirements are influenced by a wide range of design choices, including power source type, component selection, power management hardware and software, and importantly sensor node and network management algorithms, such as implemented by a real time operating system (RTOS) [28] or sensor network duty cycling schedules [40].

We define the desirable power consumption of an entire sensor node platform to be less than 100 mW when idle (also referred to in the literature as standby or deep sleep mode). We also require that a WVSNP has an instantaneous power consumption of less than 500 mW. These requirements are based on rule of thumb calculations that a node running on two AA batteries lasts a year if it consumes on average less than 0.2 mA [3], [41]. Compare this to a cell phone which typically consumes more than 4 mA.

To satisfy these stringent power consumption requirements, a sensor node has to provide most, if not all, of the following power modes. (For general background on microprocessor design and their power-efficient design and operation we refer to [42]–[46].)

On: At this fully functional state the main processor, e.g., microcontroller unit (MCU) chip/integrated circuit (IC), uses most power as all of its parts are in use. Power can only be conserved by dynamically changing the core frequency or operating voltage.

Ready: This mode saves power by shutting down a chip's core clock when not needed (also referred to as clock gating). The chip's core clock resumes when an interrupt is issued, for instance to process some input/output (I/O).

Doze: As in Ready mode, the chip's core clock is gated. Additionally, the clocks for pre-configured peripherals can be switched off. An interrupt can quickly reactivate the chip's normal functions.

Sleep: This mode switches off all clocks and reduces supply voltage to a minimum. External memory runs at a self-refreshing low-power state. Data is preserved during Sleep and hence there is no need to recover it on wake-up.

Idle: Unlike Sleep mode, data in the chip's registers is lost in Idle mode. The chip's core is turned off. An interrupt resumes the chip's normal functionality.

Hibernate: The entire chip's power supply is shut down and the chip loses all internal data. This requires a full initialize (cold-boot) resumption,

The node design and control need to generally trade off the power savings achieved by duty cycling through these power modes with the frequency of checking the radio channels and the cost of waking up for channel checking [40], [47], [48].

The throughput of a node is generally defined as the number of video frames per second received by the sink node from the source node [49]. More specifically, a frame cycle consists typically of five stages:

1) The source sensor node loads a raw frame from the attached imager into the node's memory;
2) The source node compresses the raw frame and loads the result to its output buffer;
3) The source node's radio transmits the compressed frame from the buffer to the sink node;
4) The sink node uncompresses the received frame; and
5) The sink node displays/stores the raw uncompressed frame.

We define the required throughput as a frame rate of at least fifteen common interframe format (CIF, 352×288 pixels) frames per second (fps). We choose the 15 fps as it is widely documented as an acceptable frame rate for human perception of natural motion [50]–[54].

The throughput is primarily limited by the MCU chosen as the master component of the sensor node. The choice of an MCU has implications for peripheral components and bit-width as well as the availability of power modes, multimedia processing, and memory interfaces. 32-bit MCUs are typically significantly faster and computationally more capable than the 16- or 8-bit MCUs for video; moreover, a 32-bit MCU consumes typically two orders of magnitude less power than an 8-bit MCUs for the same work load [1], [33], [55]. Therefore, we require the master processing unit to be 32-bit capable. Other main throughput limiting components are typically the radio communication and the image acquisition.

The cost of a node depends primarily on the technology chosen for the architecture, the type and maintenance cost of the selected components, the intellectual accessibility of the SW/HW components, and the scalability and upgradeability of the architecture. A low-cost platform generally has very few, if any, proprietary components. It should be possible to substitute components based on competitive pricing in a modular manner. Such substitutions require in-depth knowledge of the functions and limitations of each HW/SW component which is rarely possible for proprietary platforms. Therefore, standardized HW/SW components and well architected open source software and open hardware cores that benefit from economies of scale are important for meeting the low cost objective.

We require a fully functional sensor node platform that meets the above power and throughput requirements to cost less than $50 USD with the cost expected to decrease as standardized components get cheaper. We choose this cost requirement, as we envision a sensor node as a semi-disposable component that is widely deployed.

A sensor node can be designed to incorporate low-level input, that is, physical-layer and middleware-level input from the environment. For instance, the node can use input from other physical sensors (e.g., motion sensors) to decide when to capture a frame. We refer to a node with this capability as a *smart mote*. Smart motes can further reduce power consumption and improve effective throughput beyond the manufacturer's stated hardware capabilities for a specific application.

III. CLASSIFICATION OF WIRELESS VIDEO SENSOR NODE PLATFORMS

In the preceding section, we reviewed the main requirements for wireless video sensor node platforms (WVSNPs) and defined ideal performance requirements. We comprehensively reviewed the existing literature and found that none of the existing nodes meets the ideal requirements. In an effort to conduct an insightful survey that uncovers the underlying structural shortcomings that cause the existing nodes to fail our ideal (yet practically reasonable) requirements, we relaxed our requirements. We selected WVSNPs for our survey that meet at least three of the following rules based on the test scenarios considered in the literature about the sensor node.

1) The node has most of the power modes defined in Section II and its average power consumption is less than 2 W;
2) The node's throughput is at least two CIF fps;
3) The estimated cost of the node using current off-the-shelf technology and accounting for economies of scale projection is at most $50 USD;
4) The sensor node platform is capable of wireless transmission; and
5) The architecture implementation and major HW/SW building blocks are open to researchers without proprietary legal restrictions to educational experimentation.

Many platforms, e.g., [5], [7]–[9], [11]–[13], [15]–[17], [19], [36], [56]–[77], do not meet these relaxed requirements. For example, the platform [60] employs advanced techniques for detecting changes in brightness to achieve ultra-low-power wireless image transmission. However, the platform employs a coarse 90×90 pixel imager as well as non-standard compression that is customized for the node and test application.

Any design approach based on field programmable gate arrays (FPGA) [67] likely fails the cost rule as FPGAs have very limited off-the-shelf economies of scale; further, FPGAs have low computation performance relative to power consumption and exploit limited standardized intellectual property (IP) [78], [79]. The ScoutNode [73] embraces modularity and power mode flexibility. However, it is focused on military proprietary communication protocols, has a high cost, and a high power consumption.

From our exhaustive literature review, we found that only the platforms noted in Table I satisfy our selection criteria. As summarized in Table I, we classify the selected platforms into three main categories, namely *General Purpose Architectures*, *Heavily Coupled Architectures*, and *Externally Dependent Architectures*. In each of the Sections IV through VI we first give an overview of a category and then individually critique each of the existing platforms in the category.

Before delving into the different node platform categories, we note a common characteristic of most existing nodes, namely the use of a IEEE 802.15.4 radio, in particular the Chipcon/Texas Instruments CC2420 2.4 GHz IEEE 802.15.4 RF transceiver. (Following the common Zigbee terminology, we use the term "radio" to refer to the physical and medium access control layers.) Most nodes implement only the PHY and MAC layers of IEEE 802.15.4 and use custom protocols or Zigbee-compliant protocols for the higher protocol layers.

Nevertheless, all nodes using the CC2420 or other IEEE 802.15.4 radios are "Zigbee-ready", meaning that they can be easily made Zigbee compliant by a software update of the relevant Zigbee protocol stack. The IEEE 802.15.4 radio is readily availability, has low cost, is easy to implement, and facilitates benchmarking among nodes using the same radio. However, the IEEE 802.15.4 radio has shortcomings that can significantly weaken a node platform if the node fails to leverage the IEEE 802.15.4 advantages and does not complement its weaknesses. For instance, IEEE 802.15.4 radio transmission is limited to a 250 kbps data rate, which makes real-time video transmitting almost impossible, unless efficient supplemental architectural techniques are employed. We will comment on the specific implications of the IEEE 802.15.4 radio on each sensor node's architecture in the individual critiques.

IV. GENERAL PURPOSE ARCHITECTURES

General purpose platforms are designed similarly to a personal computer (PC), following a "catch-all" functionality approach. They attempt to cover all possible peripherals and printed circuit board (PCB) modules that an application may need. This strategy results in designs that include as many building blocks as prescribed cost limits permit. General purpose architectures are useful for fast prototyping of applications. Generally, they consist of a node (MCU) PCB to which many MCU peripherals and PCB modules are attached that highlight the capabilities of the MCU.

General purpose platforms typically suffer from high power consumption and dollar cost, as well as underutilized functional blocks despite not meeting basic WVSNP requirements. Furthermore, general purpose platforms often overuse standard interfaces, such as universal serial bus (USB), personal computer memory card international (PCMCIA), universal asynchronous receiver/transmitter (UART), and general purpose input/output (GPIO) interfaces. The disadvantage of having many I/O pins and peripherals is that the I/O subsystem can consume a disproportionately large amount of power. Powering down GPIO interfaces is not always an option as in most cases the wakeup cost negates the advantages gained from periodic shutdowns.

In Table II we summarize and contrast the considered general purpose architectures. In the first row of the table we rate the platform's flexibility from 0 to 10 (0 being functionally and architecturally inflexible and 10 being highly robust, adaptable, and extensible).

A. Stanford's MeshEye [14] and WiSN Mote [55]

1) Overview: MeshEye is a smart camera mote architecture designed for in-node processing. It selects among several available imagers based on changes in the environment. The architecture follows the philosophy that, as the level of intelligence (a priori decision making before acquiring and compressing an image) increases, bandwidth requirements on the underlying data transmission network decrease proportionally. The host processor is a 32-bit 55 MHz Atmel AT91SAM7S family MCU with an ARM7TDMI ARM Thumb RISC core. The MCU internally has up to 64 KB SRAM and 256 KB

TABLE I
SUMMARY OF CLASSIFICATION CATEGORIES FOR EXISTING WIRELESS VIDEO SENSOR NODE PLATFORMS (WVSNPs).

Architecture Categories	General Purpose	Heavily Coupled	Externally Dependent
Example Platforms	Stanford's *MeshEye* [14] and *WiSN Mote* [55], Portland State's *Panoptes* [80], Yale's *XYZ* [81]–[83], NIT-Hohai Node [84]	UC Irvine's *eCAM* and *WiSNAP* [85], UCLA's *Cyclops* [86], Philips' *Smart Camera Mote* [87], [88], CMU's *CMUcam3* [89]	CMU's *DSPCam* [90], [91] UC Berkeley's *Stargate* [18], [21], [35], [92], [93], Crossbow's *Imote2/Stargate 2* [1], [92], [94], UC's *CITRIC* [95], FU's *ScatterWeb* [96], CSIRO ICT's *FleckTM-3* [24], [97], UFranche's *Fox node* [98],
Identifying Features			
Objective	Catch-all approach. MCU centered. Many peripherals highlighting host MCU capabilities. High GPIO count.	Hardware designed to fit specific application. Highly customized. Special optimization of one of the acquisition, processing, or transmission stages but not the entire path.	Typically targeted for multi-tier networks. Modularized PCB approach. Main PCB with host MCU. Main PCB depends on external daughter boards for interfacing, power, and peripherals.
Flexibility	Flexible support for wide application range. Many interface options due to high peripheral count and GPIO count.	Very limited. Changing application requires hardware re-design. Few GPIO options.	Limited flexibility within its ecosystem of compatible daughter boards.
Extensibility	Most extensible. Standardized interfaces enable extensibility.	Very limited. Rarely accommodates a new application. Customized block to block interfacing.	Moderately extensible. Predetermined application options supported by the daughter boards.
Architecture	Similar to a PC. Medium to high MCU speed. Occasionally Co-processors. High memory and mass storage capability. Support for RTOS. Interface compatible with many imagers and radios. Assumes third-party functionality for acquisition and transmission.	Specialized hardware modules with sequential dependencies. High throughput modules offload processing from host MCU. Custom software required for external hardware block coordination. The stage-by-stage optimizations typically ignore integration of other sensor stages. Customized radio modules typical.	Medium to high speed MCU. Major application building blocks spread over daughter boards. Typical co-processor in a separate daughter board. Daughter boards customized to the host board's interfaces. High memory and mass storage options. Support for RTOS.
Performance	High performance depends on application's software design and use of available hardware.	High throughput hardware accelerator blocks. Emphasis on module image processing, filtering, and inference. Optimized custom radio protocols	Similar performance characteristics as general purpose platforms. Performance depends on the assembled parts and interboard communication.
Advantages	Most flexible. Most extensible. Potentially high performance. Enables quick application prototyping. Accepts many standardized peripheral interfaces.	Usually optimized for the target application. Saves power as there are few idle modules. Custom hardware usually faster than standard hardware.	Potentially many configurations with different daughter boards for desired functionality. Each daughter board can be separately optimized. Enables modularity of important sub-modules.
Limitations	No HW/SW integration codesign. Idle module functionality. Most functionality unused by most sensor applications. No multimedia optimization modules. Most expensive. Not necessarily suited for video processing. Transmission not accounted for in HW design. Over-reliance on standard interfaces	Not flexible. Not extensible. Costly re-designs needed for changes in application. All modules need to be active and coordinated for each task pipeline. Little opportunity for duty-cycle based power management. Few standardized modules lead to incompatibility with other sensors.	Main PCB board can rarely function stand-alone. Redundant basic PCB components on multiple daughter boards for power reliability. Overhead in coordinating daughter boards. Many idle modules within the daughter boards. Usually many boards needed for simple functionality.
Cost	Most expensive. Dollar cost proportional to System on Chip peripheral count and external interface module count.	Expensive. Hardware accelerators and hardware blocks add to the cost. Custom hardware is generally expensive.	Expensive. Daughter boards introduce hidden costs. Prices often quoted for the host MCU board only.
Power	Idle GPIOs consume high power. High clock rates proportionally costly. Unoptimized data access and transmission wasteful.	Low idle power loss. Limited power management options.	Power wasted on board to board overhead. Inter-board power management hard to implement and wasteful.

of flash memory as well as a built-in power management controller. The mote is designed to host up to eight KiloPixel imagers (Agilent Technologies ADNS-3060 high-performance optical mouse). The ADNS-3060 is a 30×30 pixel, 6-bit grayscale camera also referred to as image sensor or optical mouse sensor (due to its use in a computer mouse). The sensor node also has one programmable VGA camera module (Agilent Technologies ADCM-2700 landscape VGA CMOS module, 640×480 pixel, grayscale or 24-bit color). The dynamic use of a variety of mouse sensors and a VGA camera makes this mote "smart". The mote has a serial peripheral interface (SPI) bus attached multimedia card (MMC)/secure digital (SD) flash memory card for temporary frame buffering

or archiving of images. As illustrated in the top right part of Figure 1, a single SPI interface connects an IEEE 802.15.4 radio, up to eight KiloPixel imagers, and a flash card (on the left) to the MCU.

As shown in the bottom right part of Figure 1, the VGA camera module is controlled via a two wire interface (TWI also denoted as I^2C). The VGA camera module captures and encodes the video into CCIR (ITU-R BT.601). The encoded video data is read from the camera through general-purpose I/O pins.

The Stanford WiSN node, illustrated in Figure 2, has many similarities with MeshEye with more focus on implementing networked image sensing where multiple image sensors ob-

TABLE II
SUMMARY COMPARISON OF GENERAL PURPOSE PLATFORMS. DISTINCT CHARACTERISTICS OF THE WiSN MOTE WITH RESPECT TO THE RELATED
MESHEYE MOTE ARE GIVEN IN BRACKETS.

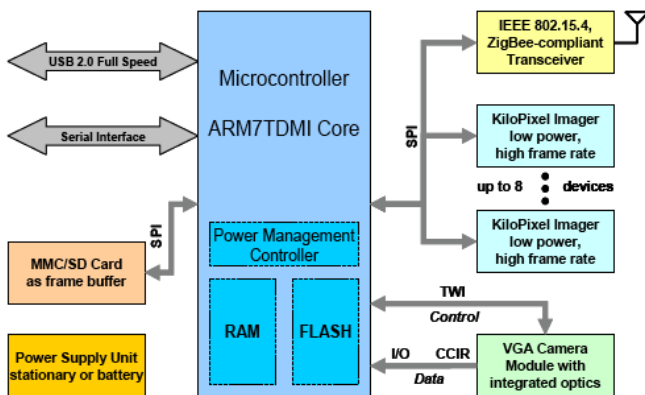|	Stanford's MeshEye and [WiSN] Motes	Portland State's Panoptes	Yale's XYZ	NIT-Hohai Node
Flexibility Rating	5.5/10 [6/10]	6/10	7/10	6.5/10
Processor(s), Core, Speed	Atmel AT91SAM7S (ARM7TDMI), 55 MHz, 32-bit	(PDA Platform) Intel StrongARM, 32-bit, 206 MHz, No Floating Point	OKI Semiconductor ML67Q500x, ARM7TDMI, 57.6 MHz, 32-bit	Intel PXA270 RISC core, 500 MHz, 32-bit
Node Power and Supply *(mW)*	DC input or AA cells [LTC3400 voltage reg. (1.8 V and 3.0 to 3.6 V)]	< 5000 mW, DC input	7 to 160 mW, 3×AA 1.2 V Ni-MH rechargeable cells, multiple voltage regulator	DC input
Supported Power Modes	Unknown	Suspend, Active	Halt, standby, deep sleep (30 μA)	Unknown
Node and Peripheral Power Management	In-built MCU power management (PM) controller, Software controlled phase locked loop (PLL)	Support for network wakeup/power mode	Power tracker, supervisor, SW controlled clock divider (57.6 to 1.8 MHz), most peripherals switch on/off	Unknown
Memory/ Storage	64 KB on-chip SRAM, 256 KB on-chip Flash, MMC/SD [2 MB off-chip Flash/32 KB FRAM]	64 MB	256 KB on-chip Flash, 32 KB on-chip RAM and 4 KB boot ROM, 2 MB off-chip SRAM	External SDRAM plus Flash (size undocumented)
I/O, Interface	USB2, UART, SPI, I2C	UART, SDLC, USB, Serial CCODEC, PCMCIA, IrDA, JTAG	SPI, I2C, 8-bit parallel port, and a DMA	USB2, UART, SPI, I2C, AC97, PCMCIA
Radio	TI CC2420 2.4 GHz Zigbee Ready	PCMCIA based 2.4 GHz (802.11b)	TI CC2420 2.4 GHz Zigbee Ready	Stand-alone 802.11
Wireless Trans. Rate	< 250 kbps	802.11b (< 11 Mbps),	< 250 kbps	802.11g (< 54 Mbps)
Imager, Max Imager Resolution, Max Frame Rate	8×ADNS-3060 (30 × 30/6-bit grayscale) and 1×ADCM-2700, VGA (640 × 480/24-bit) [2×ADCM-1670, CIF (640 × 480/24-bit) and 4×ADNS-3060 (30 × 30/6-bit)]	Logitech 3000 USB based video camera, VGA (15 fps),	Omnivision OV7649, VGA	USB based Webcam
Capture-Save Frame Rate	3 fps [Not evaluated]	< 13 CIF fps	4.1 QVGA fps	> 15 QCIF fps
HW Image Processing	None	MCU Multimedia performance primitives	None	None
SW Image Processing	None	JPEG, Differential JPEG	None	H.263
Frame Trans. Rate	Not evaluated	Not evaluated	Not evaluated	10 to 15 QCIF fps
OS / RTOS	None	Linux (kernel 2.4.19)	SOS	modified Linux 2.4.19 core.
Cost	Unknown	Unknown	Unknown	Unknown

Fig. 1. Block diagram of Stanford's MeshEye architecture [14].

serve the same object from different view points. This enables collaborative data processing techniques and applications. For its higher resolution imaging, WiSN uses two ADCM-1670 CIF (352×288 pixel) CMOS imagers, instead of MeshEye's one VGA camera. As shown in Figure 2, the node also adds a flexible expansion interface that connects to a variety of sensors, though some are not necessarily critical for a video sensor requirement. The WiSN also introduces a Linear Technology LTC3400 synchronous boost converter for regulating voltage levels (1.8 V and 3.0 to 3.6 V). The converter has a 19 μA quiescent current draw and can supply up to about 3 mA.

2) Advantages: Processing the video stream locally at the camera is advantageous as it can reduce bandwidth requirements and hence save power or improve frame rate as only necessary information is processed or transmitted. The use of more than one image sensor seems suited for distributed vision-enabled applications. The smaller imagers are used to detect some events, which removes the need to unnecessarily trigger the VGA imager for image acquisition. This saves power as the KiloPixel imagers do most of the vision monitoring whereas the slower and more power-hungry VGA imager is idle most of the time.
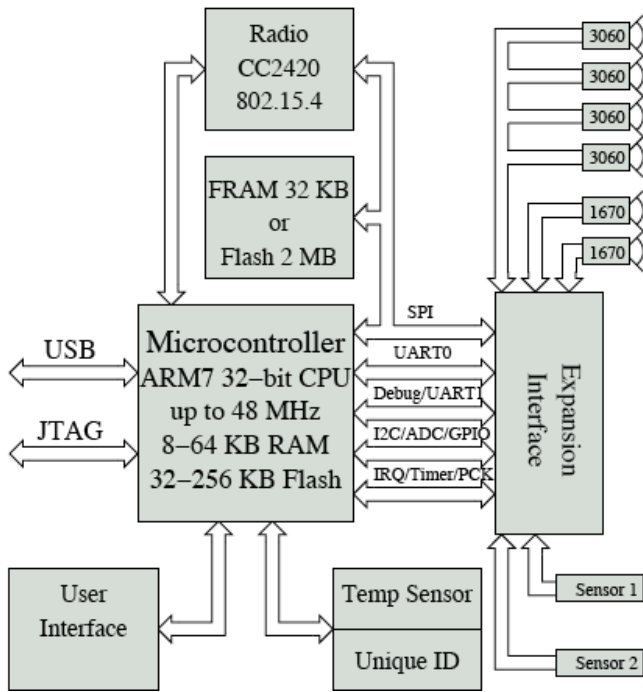
Fig. 2.   System diagram of the Stanford WiSN mote board [55].

The external MMC/SD Flash card/Flash memory gives the motes a persistent, scalable, and non-volatile memory. The ability to store files locally is helpful for debugging, logging, and data sharing.

The platforms have an option of either mains power supply or battery based supply. This makes the motes flexible for both mobile and fixed applications. The MCUs' built-in power management hardware is an efficient way of putting the MCU and its peripherals into different power-saving modes instead of depending on software managed algorithms. A programmable phase locked loop (PLL) in the MCUs allows for dynamically setting the core's clock rate to lower rates when less processing is required, which saves power.

Using a single SPI interface for several modules is an efficient use of the MCU interfaces and conserves I/O pin use. The choice of directly reading CCIR encoded video in MeshEye reduces component count, power, and cost.

WiSN's use of the expansion interface simplifies design and supports other traditional sensors. The interface also enables it to use two CIF cameras which are more useful in collaborative/stereoscopic imaging compared to having only one VGA imager. Additionally, the expansion port exposes timer inputs/outputs, and programmable clock outputs. Further, the interrupt request (IRQ) lines and standard GPIO pins are multiplexed using the remaining pins, making this platform easily expandable. Some of the GPIO pins have enough current drive (16 mA) to power attached sensors. This reduces the need to route many power lines on the board. The choice of the AT91SAM7S MCU allows an easy upgrade path as the AT91SAM7 MCU family has the same in-chip peripheral set, except for the amount of RAM and Flash memory.

Another WiSN advantage is that its LTC3400 linear regulator, which operates at low I/O voltages, protects the battery by presenting the entire circuitry as a single current sink. It

also helps reduce the sleep current draw. The LTC3400 can start up and operate from a single cell and can achieve more than 90 % efficiency over a 30 to 110 mA current draw range.

3) Disadvantages: MeshEye's capture-and-save frame rate of 3 fps is quite low. The CC2420 radio module, which is limited to 250 kbps, is the only transmission module. This requires a very high video compression ratio to be able to transmit video and limits real-time video streaming.

KiloPixel imagers are not necessarily the least energy consuming and cheapest event detectors. Events within the field of view of the VGA imager can, for instance, be sensed with infrared (IR) or ultrasound sensors, which are cheaper and consume less energy than the KiloPixel imagers.

WiSN's video capture is limited to the CIF resolution. In an attempt to support both the mouse (30×30 pixel) sensor and the CIF sensor the designers opted for a serial interface connection to the MCU. This serial connection is robust, but limits the data rate and hence the frame rate of the video.

External memory access via the serial peripheral interface (SPI) bus, due to its serial nature and its master/slave coordination, is significantly slower than on-chip memory or parallel external memory. The Ferroelectric RAM (FRAM) is currently limited to 32 KB. The off-chip Flash memory is not a direct substitute for RAM as it offers limited write/erase cycles and has slow write speeds and wait states when writing. If flash memory is used as a frame buffer, it can limit the node's lifetime depending on the frequency of data writes. For example, a 2 MB flash device designed for 100,000 write/erase cycles will last only 230 days if a 100 KB frame is written to it every 10 seconds.

B. Portland State's Panoptes [80]

1) Overview: The Panoptes video sensor captures, compresses, and transmits video at low-power levels below 5 W [80]. The tested 5 W consumption does not meet our 2 W power threshold, but the node meets most of our five criteria in Section III. The sensor node can be fine-tuned to meet the 2 W for some applications. Panoptes uses a personal digital assistant (PDA) platform called Bitsy. The platform runs Linux kernel 2.4.19 on a 206 MHz Intel StrongARM MCU and 64 MB of memory. A Logitech 3000 webcam is used to capture high-quality video and attaches to the PCB via a USB 1.0 interface. Panoptes uses spatial compression (not temporal), distributed filtering, buffering, and adaptive priorities in processing the video stream. A stand-alone third party 802.11 card attached via PCMCIA is used for wireless transmission.

2) Advantages: Panoptes is one of the few platforms with the architectural components capable of real-time video capture. It uses special multimedia instructions that are custom to this MCU for most of the video compression. These special MCU primitives enable high frame rates as they speed up multimedia processing, such as JPEG and differential JPEG compression. The Panoptes board supports network wake-up as well as optimized "wake-up-from-suspend" energy saving mechanisms. In addition to compression, Panoptes uses priority mapping mechanisms, including raw video filtering, buffering, and adaptation to locally pre-process the video stream which can be strategically used to conserve power.
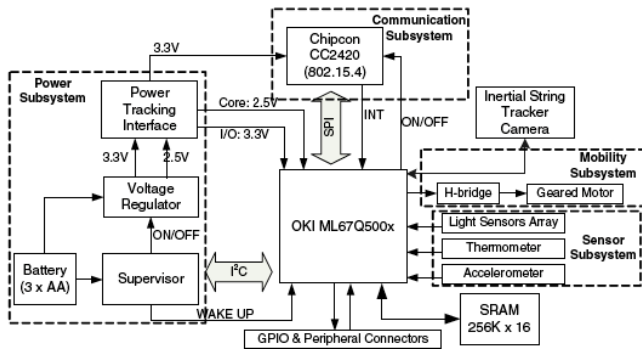
Fig. 3.   The XYZ node architecture [83].

The third party stand-alone 802.11 module makes the platform flexible as the module can be easily exchanged for more power efficient and faster modules as they become available or affordable. The use of Python scripting to connect software module objects is good for supporting a modularized system that is easily adaptable as each object can be associated with its exchangeable hardware component.

3) Disadvantages: The drawback for Panoptes is that it requires several watts of power, which is relatively high, compared to the similar Stargate platform, see SectionVI-B.

Similar to many other platforms, the StrongARM does not have a floating point unit. Connecting the board to the camera via a 1.0 USB interface creates a data bandwidth bottleneck, especially for 352×288 (common intermediate format, CIF) and 640×480 (video graphics array, VGA) pixel frame sizes, and increases power consumption. This is because the image data from the camera coming in over the USB needs to be decompressed from a camera-specific compression format to generic raw image data in the kernel before being sent to the host's user space and then recompressed with JPEG.

The use of polling to check whether a frame is ready is an inefficient way of acquiring video. Although using specialized built-in MCU multimedia primitives to speed frame processing is helpful, reliance on MCU specific features limits the portability of the code and complicates platform upgrades.

The 802.11 networking on Panoptes consumes about a third of the total platform power [80] and therefore needs to be optimized. The Python scripting employed for adaptability suffers from the common drawbacks of scripting engines, namely large memory space requirements and execution inefficiencies. Also, the Python interconnects result in a 5 % frame overhead [80].

C. Yale's XYZ plus OV7649 and ALOHA modules [81]–[83], [99]

1) Overview: The XYZ is a motion-enabled and power-aware sensor platform targeting distributed sensor network applications. As illustrated in Figure 3, the platform consists of several subsystems, including subsystems for sensing (light, temperature, and accelerometer), communication (TI CC2420 Zigbee radio), mobility (geared motor), power (voltage regulator, power tracker, supervisor, and three AA 1.2 V Ni-MH rechargeable battery pack) and a camera. The capacities of the batteries range from 1200 to 2000 mAh.

The XYZ node is designed around the 57.6 MHz 32-bit OKI Semiconductor ML67Q500x ARM THUMB (ARM7TDMI MCU core). The MCU has an internal 256 KB of Flash, 32 KB of RAM, and 4 KB of boot ROM as well as external SRAM. The Omnivision off-the-shelf OV7649 camera module and the 32×32 pixel event-based ALOHA CMOS imager have been connected to the XYZ node in separate research efforts [74], [82]. The OV7649 can capture VGA (640×480) and quarter VGA (QVGA, 320×240) images. The image data is transferred from the camera to the on-board SRAM with an 8-bit parallel port using direct memory access (DMA), which does not involve the MCU.

2) Advantages: The MCU provides numerous peripherals which can be turned on and off as required by the application. The on and off switching is accomplished through software enabling/disabling of clock lines to MCU peripherals. The node is therefore capable of a myriad of power management algorithms. The node provides halt and standby power saving sleep modes in addition to the internal software controlled clock divider that can halve a range of MCU speeds from 57.6 MHz down to a minimum of 1.8 MHz. During standby mode the oscillation of the MCU clock is completely stopped while the MCU still receives some power. The halt mode, on the other hand, does not stop clock oscillation, but blocks the clock from the CPU bus and several MCU peripherals.

The custom supervisor circuit supports a long-term deep sleep mode that puts the entire node into an ultra-low power mode (consumes around 30 μA) by using a real-time clock (RTC) with two interrupts. This setup adds to power management options as transitioning the node into a deep-sleep mode can be done through software control by disabling its main power supply regulator. The RTC can be scheduled to wake the node from every 1 minute up to once every 200 years.

3) Disadvantages: The XYZ uses the CC2420 radio with its limited transmission rate. The node implements the Zigbee protocol stack on the host MCU, which increases power consumption. Operating the OS and the Zigbee protocol stack on the host MCU at the maximum clock frequency is estimated to require 20 mA [81]–[83]. An independent stand-alone radio module with its own in-built protocol stack would relieve the MCU from the network management tasks and improve power savings management. Another challenge for power management is that the MCU I/O subsystem consumes between 11 and 14 mA (i.e., 35.75 to 45.5 mW) due to the high number of I/O pins and peripherals.

The node uses the SOS RTOS, which is an open-source operating system with a relatively small user base and therefore has only a small pool of available re-usable software modules.

Using the OV7649, the XYZ achieves a frame capture-and-save rate of 4.1 QVGA fps. Additionally, only 1.7 16-bit color frames, 3.4 8-bit color, or 27.3 1-bit (black and white) QVGA frames can be stored in the off-chip SRAM. The number of frames that can be stored increases 4.6 times if a platform optimized 256×64 resolution is used [74], [81]. These limited frame storage capacities can potentially reduce frame rates as application processing may require holding frames in memory, blocking the next frames.
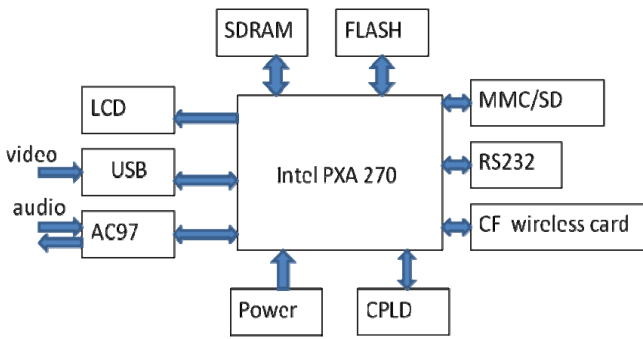
Fig. 4. NIT-Hohai node hardware architecture [84].

D. The NIT-Hohai Node [84]

1) Overview: This sensor node, designed jointly by Nanchang Institute of Technology (NIT) and Hohai University, is centered around the Intel 500 MHz 32-bit PXA270 RISC core SoC, as illustrated in Figure 4, and runs a modified Linux 2.4.19 core. Multithreading is used to multitask custom application-level streaming protocols that are layered on top of TCP/IP. The node uses IEEE 802.11 for wireless streaming, with a throughput of 10 to 15 QCIF fps. The node has external SDRAM and FLASH storage as well as a Liquid Crystal Display (LCD).

2) Advantages: The PXA27x family of processors, which is also used in IMote2 [94], has a rich set of peripheral interfaces and I/O ports, see Figure 4. The standardized ports permit use of a wide range of peripheral I/O modules, facilitating the selection of low-cost modules. The architecture is a simple plug-and-play attachment to the core SoC via standard bus protocols and has the benefits of Linux. The design uses run-time loadable module drivers to make the system flexible and scalable. The node uses an optimized H.263 video compression library and is able to transmit in real time.

3) Disadvantages: The board uses a PCMCIA compatible Compact Flash (CF) based 2.4GHz WiFi card which functions in stand-alone mode, but lacks options for independent direct power management through applications running on the attached PXA270 SoC. Significant design efforts went into the touch-capable 16-bit color 640 ×480 LTM04C380 LCD and related Graphical User Interface (GUI) components, which are not a requirement for a WVSNP. Building on the basic Linux drivers, the design is almost exclusively focused on software functionalities and lacks cohesive HW/SW optimization. All major processing, such as frame capturing, compressing, and networking management, is performed by the SoC, which limits opportunities for power saving through duty cycling. Overall, the node suffers from the disadvantages of general purpose architectures in that it is a rather general design (similar in philosophy to a personal computer) and lacks the mechanisms to achieve the low power consumption and cost required for a WVSNP.

V. HEAVILY COUPLED ARCHITECTURES

A. Overview

While general purpose platforms are designed for a wide range of applications, heavily coupled platforms are designed for a specific application and are typically over-customized and lack flexibility. The advantage of these highly customized nodes is that they can be optimized to achieve good performance for the original application that the platform has been specifically designed for (often referred to as the parent application).

On the down side, the optimization for a specific parent application often leads to over-customized architectures. For instance, in order to meet prescribed timing or cost constraints of the parent application, the hardware modules are designed to be highly dependent on each other, i.e., they are heavily coupled. The hardware is often so inflexible that any change in application load or on-site specification requires a complete hardware re-design. Similarly, the software modules are typically heavily coupled with each other and with the specific hardware such that the software modules are not reusable if some other software module or the hardware changes.

CMUcam3, for example, uses an MCU with very few GPIO pins, so that there is no extra pin to add basic next-step functionality, such as adding a second serial peripheral interface (SPI) slave. This leads to underutilization of the SPI module which is dedicated to only the MMC module, even though it is capable of supporting tens of slaves. An attempt to use SPI for any other purpose requires removing the MMC module.

In eCAM, the radio and the MCU have been merged into one module. This merged radio/MCU module speeds up data processing since the software instructions and data are co-located in the module. Thus, instructions and data do not need to be fetched from external memory or over serial buses and the module synchronization overhead is reduced. As a result, eCAM can implement a simple medium access control (MAC) protocol with increased data rate. However, this optimization prevents future expandability and compatibility with other radio standards. Moreover, in eCAM, the compression stage has been merged with the imager. Should the need for a new compression scheme or imager frame capture arise, the entire camera module will need to be replaced and re-designed.

B. UC Irvine's eCAM and WiSNAP [85]

1) Overview: The eCAM is constructed by attaching a camera module (with up to VGA video quality) to an Eco mote. As illustrated in Fig. 5, the 1 cm^3 sized Eco mote consists of a Nordic VLSI nRF24E1 System on a Chip (SoC), a chip antenna, a 32 KB external EEPROM, an Hitachi-Metal H34C 3-axial accelerometer, a CR1225 Lithium Coin battery, an LTC3459 step-up switching regulator, an FDC6901 load switch, a power path switch, a temperature sensor, and an infrared sensor. The nRF24E1 SoC contains a 2.4 GHz RF transceiver and an 8051-compatible DW8051 MCU. The MCU has a 512 Byte ROM for a bootstrap loader and a 4 KB RAM to run user programs loaded by the bootstrap from the SPI attached EEPROM. The camera module consists of the Omnivision OV7640 CMOS image sensor and OV528 compression/serial-bridge chip. The camera can function as either a video camera or a JPEG still camera. The OV528 is used as a JPEG compression engine as well as a RS-232 interface to the Eco node. The imager supports a variety of

TABLE III
SUMMARY COMPARISON OF HEAVILY COUPLED PLATFORMS.

	UC Irvine's eCAM and WiSNAP	UCLA's Cyclops	Philips Smart Camera Mote	CMU's CMUCam3
Flexibility Rating	6.5/10	5.5/10	4/10	3/10
Processor(s), Core, Speed	Nordic VLSI nRF24E1 (Eco mote)	Atmel 4 MHz 8-bit Atmega128, Xilinx XC2C256 CoolRunner	Xetal IC3D SIMD and Atmel 8051	NXP LPC2106 ARM7TDMI, 32.bit, 60 MHz, No Floating Point
Node Power and Supply *(mW)*	CR1225 Lithium battery, LTC3459 switching regul.	33 mW, 2×AA cells	100 mW (typical ICD3 only)	100mW, 4×AA, DC power
Supported Power Modes	None	active, power-save, or powerdown	None	Idle (125 mW), Active (650 mW)
Node and Peripheral Power Management	FDC6901 load switch, a power path switch	External block power mode control from host	None	Software controlled frequency scaling
Memory/Storage	In-MCU 512 byte RAM and 4 KB RAM, 32 KB external EEPROM	512 KB external Flash, 64 KB external SRAM	DPRAM, 1792 bytes (inside 8051), 64 KB RAM, 2 KB EEPROM	64 KB RAM, 128 KB Flash, Up to 2 GB MMC mass storage
I/O, Interface	UART, SPI	I2C, UART, SPI, PWM	UART, OTA 8051 programming	Very few GPIO, SPI, 2×UART, I2S
Radio	2.4 GHz RF transceiver, chip antenna, 10 m range	None (depends on attached Mica Mote)	Aquis Grain ZigBee (8051 and CC2420), 5 m range	None
Wireless Trans. Rate	< 1 Mbps	38.4 kbps	< 10 kbps	None
Imager, Max Imager Resolution, Max Frame Rate	Omnivision OV7640, 30 VGA fps, 60 QVGA fps; OPTEK OP591 optic sensor	Agilent ADCM-1700, CIF	2 VGA imagers	Omnivision VGA, OV6620 (26 fps) or OV7620 (50 fps), CIF (352x288)
Capture-Save Frame Rate	Not evaluated	2 fps	Not evaluated	< 5 fps (CIF)
HW Image Processing	On camera OV528 compression/serial-bridge chip	Xilinx XC2C256 CoolRunner CPLD	ICD3 Image Processor Arrays, Line Memories and Video I/O processor blocks	Averlogic AL4V8M440 (1 MB, 50 MHz) video FIFO
SW Image Processing	None	None	None	Frame differencing, JPEG, and PNG
Frame Trans. Rate	1.5 CIF fps	Not evaluated	Not evaluated	Not evaluated
OS/RTOS	None	TinyOS	Custom RTOS on 8051	None
Cost	Unknown	Unknown	Unknown	$250

size and color formats, including VGA, CIF, and QCIF. It can capture up to 30 fps. The platform radio's transmission consumes less than 10 mA (0 dBm) whereas receiving consumes around 22 mA.

2) Advantages: The eCAM platform has a customized radio, which achieves high-speed and low-power due to a simple MAC protocol, instead of a generalized complex MAC which would consume more power. The eCAM bandwidth can theoretical peak at 1 Mbps, which is four times the theoretical peak of the 250 kbps of Zigbee. This makes the eCAM a good candidate for real-time VGA resolution video transmission. The radio's transmission output power can be configured through software to −20 dBm, −10 dBm, −5 dBm, or 0 dBm levels. The eCAM is more power efficient than Bluetooth and 802.11b/g modules, which are typically 20 dBm and 15 dBm respectively, for a 100 m range [100], [101]. The eCAM in-camera hardware JPEG compression is significantly more power efficient than software implementations [83], [85]. The camera compression engine's JPEG codec supports variable quality settings. The imager's ability to capture up to 30 fps enables considerable control of the video quality.

A shown in Figure 5, the Eco node has a 16 pin expansion port, which has been designed to use the flexible parallel male connector instead of the typical rigid PCB headers. This choice of "Flexible PCB" makes the Eco node flexible and suitable for different types of packaging, which makes it easy to customize to a variety of applications.

Additionally, the Eco node has an OPTEK OP591 optical sensor, which helps with low resolution and low power vision
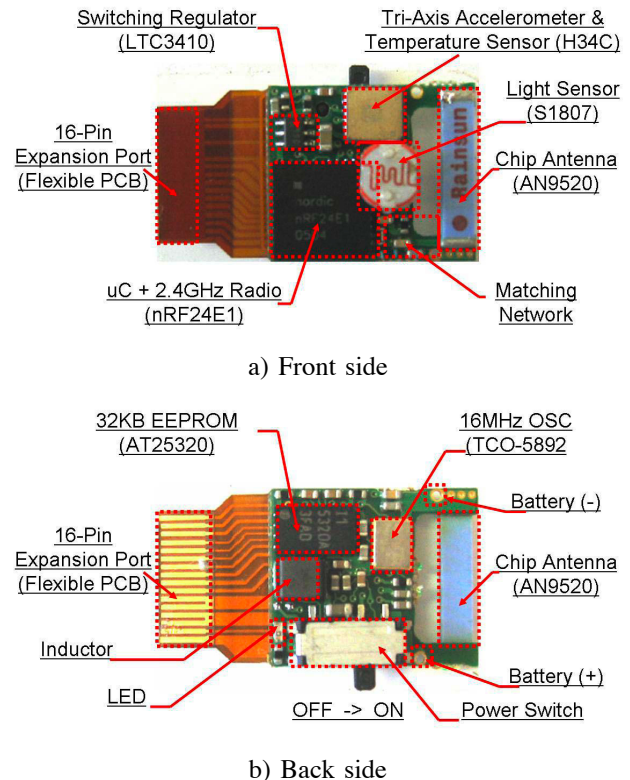

a) Front side

b) Back side

Fig. 5.   Main architecture components of the Eco board [85].

event processing. When major sensing events are detected, the VGA camera is triggered.

3) Disadvantages: The customized MAC and radio reduce the networking adaptability and compatibility with other motes. Moreover, the MAC and radio customization misses the low-cost benefit of standardized networking protocols and radio hardware, such as Zigbee compliant radios.

A further drawback of the radio is that it has a range of only about 10 m. Under a demonstration [83], [85], eCAM could only transmit relatively low resolution 320×240 (at 1.5 fps) or 160×128 video streams to the base station. This low performance suggests that the platform has a bottleneck in the video acquisition path and can not exploit its theoretical radio transmission rate of 1 Mbps. The base station then aggregates the data and transmits it to a host computer, which displays the videos in real-time. Reliance on a base station is a limitation as WVSNPs are expected to function in adhoc mode and have access to popular networks, such as WiFi, cellular, or 3G networks.

The platform is a highly optimized board-level system design that achieves a very compact form factor. However, merging MCU and radio as well as JPEG compression and the imager module makes the platform inflexible and fails to take advantage of future improvements in critical components of a mote, such as radio, MCU, compression engine, or encoder. Another concern is that the camera module attaches to the Eco via an RS232 interface, which limits the data transfer rates.

C. UCLA's Cyclops and Mica [86]

1) Overview: A typical Cyclops platform is a two-board connection between a CMOS camera module illustrated in Fig 6 with an FPGA and a wireless mote, such as a Berkeley MICA2 or MICAz mote. The camera board consists of an Agilent ADCM-1700 CIF CMOS imager with a maximum 352×288 pixel resolution. The camera has an F2.8 lens, image sensor and digitizer, image processing units and data communication units. The camera supports 8-bit monochrome, 24-bit RGB color, and 16-bit YCbCr color image formats.

The Cyclops camera module contains a Complex Programmable Logic Device (Xilinx XC2C256 CoolRunner CPLD), a 512 KB external Flash, and a 64 KB external SRAM for high-speed data communication. The CPLD provides the high speed clock, synchronization, and memory control that is required for image capture. The MCU and CPLD and both memories share a common address and data bus. The 7.3728 MHz 8-bit ATMEL ATmega128L MCU controls the imager and performs local image processing, e.g., for inference and parameter configuration. The MCU can map 60 KB of external memory into its memory space. The combination of the internal and external memory presents a contiguous and cohesive memory space of 64 KB to the node's applications.

The Cyclops design isolates the camera module's requirement for high-speed data transfer from the speed ability of the host MCU. It can optionally provide still image frames at low rates if the connecting modules are slow. The camera module is programmable through a synchronous serial I^2C port. Image data is output via an 8-bit parallel bus and three synchronization lines.
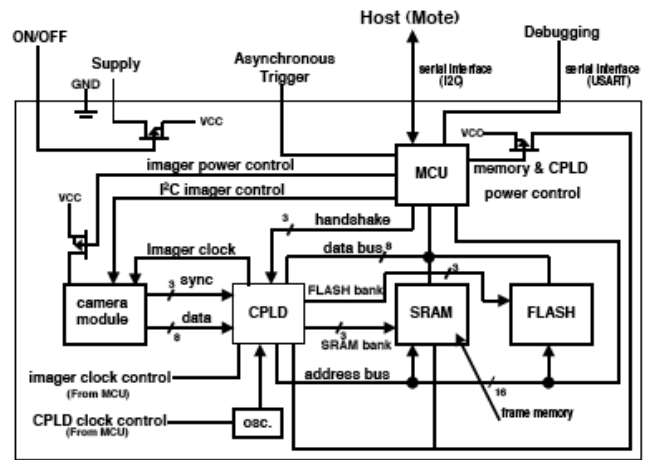

Fig. 6. Hardware architecture of Cyclops camera module [86].

2) Advantages: The modularity of Cyclops, that is, its use of a separate host mote enables "hardware polymorphism", which abstracts the complexity of the imaging device from the host mote. Moreover, the standardized interface makes the Cyclops camera module adaptable to a variety of host motes.

The dedicated image processor enables global serialization of image processing operations by offloading these image processing operations from the host MCU. The global serialization loosens the need for tight synchronization in the "acquire-process-play" path so that interrupts or handshaking signals can indicate when the dedicated image processing MCU is ready.

The dedicated image processor provides computational parallelism, such that prolonged sensing computations can be isolated to the image processor. This helps with duty cycling idle modules and saves power.

The power consumption of Cyclops is very low and enables large-scale long-term deployment. Cyclopes uses on-demand clock control of components to decrease power consumption. Moreover, to save power an external SRAM is used for storing image frames and is kept in sleep state when not needed. The camera node can automatically drive other subsystems to their lower power state. Cyclops has an asynchronous trigger input paging channel that can be connected to sensors of other modalities for event triggering. A study [21] has shown that object detection operations with Cyclops are 5.7 times more energy efficient than with CMUCam3 under the same settings and functionality.

The CPLD used by Cyclops can perform basic operations during frame capture, such as on-demand access to high speed clocking at capture time and possibly computation. In particular, the fast CPLD clock enables the camera module to carry out calculations and pixel image storage to memory while the imager is capturing. A CPLD also consumes less power than an FPGA during initial configuration reducing the overall cost of the power-down state.

3) Disadvantages: The slow 4 MHz MCU in Cyclops is not fast enough for data transfer and address generation during image capture. Therefore, the Cyclops design uses a CPLD, an additional component, to provide a high-speed clock. This

design choice increases cost, power consumption, and PCB area. Also, as noted in Section II, an 8 bit processor consumes often more power for image related algorithms than a 32 bit processor.

This platform was not intended for repeated image acquisition. Instead, the Cyclops architecture targets applications that occasionally require capture of one (or a few) images. As evaluated in [21], the PCB Header-MCU architecture in Cyclops is six times slower than the FIFO-MCU architecture in CMUCam3. Cyclops also pales CMUCam3 with its 2 fps maximum capture-and-save image processing speed. It also has a low image resolution of 128×128 pixel due to its limited internal Atmega128L MCU memory (128 KB of Flash program and 4 KB of SRAM data memory). The performance analysis in [86], reveals that improving the CPLD's synchronization with the imager would significantly improve the timing (and energy cost) of the image capture. Using more parallelism in the CPLD logic could also reduce the number of CPLD clock cycles needed to perform pixel transfer to SRAM. This could also allow higher imager clock speed and facilitate faster image capture.

Another shortcoming of Cyclops is its firmware's use of the nesC language which is based on TinyOS libraries. This limits its code reusability and refinements often enjoyed by Linux targeted firmware. TinyOS does not provide a preemptive mechanism in its synchronous execution model, i.e., tasks cannot preempt other tasks.

Other key weaknesses are that the Cyclops platform does not include a radio and does not perform any on-board compression. Though Cyclops provides the ability decouple some image processing functions, it does not provide mechanisms for guaranteeing data access or modification integrity, such as semaphores or spin locks.

The Cyclops camera module relies on third-party boards to function as a complete wireless sensor node. Given the need to manage power via duty cycling, the power-aware hardware and algorithms on the camera module may need frequent adjustments to interface with a variety of third-party daughter boards with different power definitions.

D. Philips' Smart Camera Mote [87], [88]

1) Overview: The Smart Camera mote focuses mostly on reducing power consumption through low-power local image processing. Local image processing filters out unnecessary data and compresses data before transmission. As illustrated in Figure 7, the camera consists of one or two VGA image sensors, an Xetal IC3D single instruction multiple data (SIMD) processor for low-level image processing, and the ATMEL's 8051 host MCU for intermediate and high-level processing, control, and communication. The host 8051 and the IC3D share a dual port RAM (DPRAM). The platform uses a customized Aquis Grain ZigBee module made of an 8051 MCU and Chipcon CC2420 radio. The radio's software control is reprogrammable on the 8051.

A global control processor (GCP) within the IC3D system-on-chip (SoC) is used to control most of the IC3D as well as performing global digital signal processing (DSP) operations, video synchronization, program flow, and external communication. The 8051 host MCU has direct access to the DPRAM
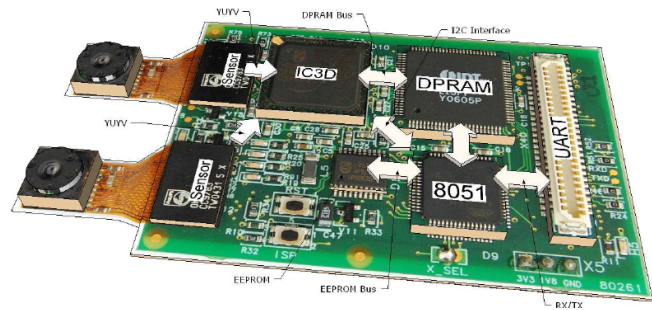

Fig. 7. Architecture of the Philips Camera Mote [87].

and has its own internal 1792 Byte RAM, 64 KB FLASH, and 2 KB EEPROM. It uses its large number of I/O pins to control the camera and its surroundings. The host has its own tiny task-switching RTOS. The radio module attaches to the platform via the 8051 host's UART.

2) Advantages: The IC3D is designed for video processing and has dedicated internal architecture blocks for video, such as linear processor arrays, line memories, and video input and output processor blocks. The video processor blocks can simultaneously handle one pixel at a time for CIF (320×240) or two at a time for VGA (640×480). Pixels of the image lines are interlaced on the memory lines. Sharing the DPRAM enables the main processors to work in a shared workspace on their own processing pace. This enables asynchronous connection between the GCP and IC3D and simple shared memory based software synchronization schemes. The DPRAM can store two images of up to 256×256 pixels and enables the IC3D to process frames at camera speed [87], [88], while a detailed evaluation of the frame capture-and-save and transmission rates remain for future research.

The SIMD based architecture of the IC3D decodes fewer instructions for more computational work and hence requires less memory access, which reduces energy consumption. In contrast, each 30×30 pixel imager of MeshEye [14], captures its own small image, loads it into memory and process the duplicate instructions on each image only to detect an event. In [87], on the other hand, a large frame is loaded to the same memory and the same "detect event" instruction is issued for each MCU core to process part of the image for an event, sequentially or in parallel. The first core to detect an event can signal the other core to stop, hence reducing not only processing time but also memory paging which conserves power.

The IC3D has a peak pixel performance of around 50 giga operations per second (GOPS). The GCP is powerful enough to perform computer vision tasks, such as face detection at power consumption levels below 100 mW.

The 8051 host's UART has its own baud rate generator which leaves the 8-bit and two 16-bit timers available for RTOS switching and user applications. The radio module's peer-to-peer structure enables point-to-point camera-to-camera communication. The camera can be remotely programmed via the radio and the in-system programmability feature of the 8051.

3) Disadvantages: The employed Zigbee module has a range of only five meters. Further, its maximum data rate
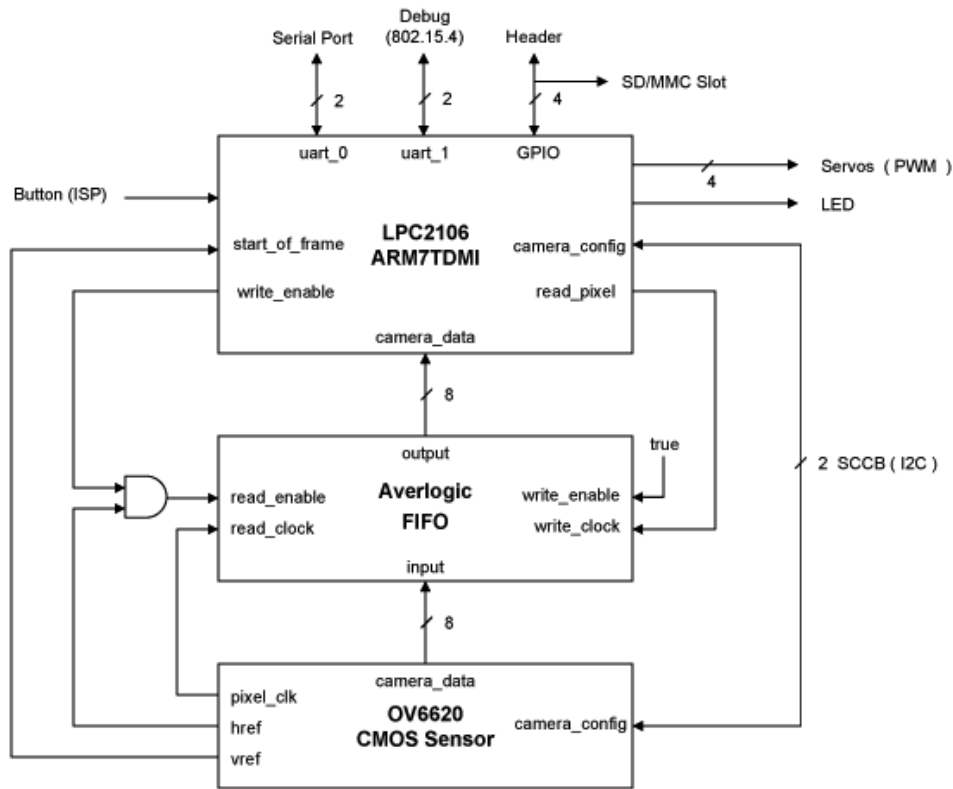
Fig. 8. CMUCam3's major block architecture [89].

of around 10 kbps makes the Zigbee module poorly suited for real-time image transmission. This low transmission rate limits the module to transmitting only meta-data of the scene's details or events.

The module has numerous major components that altogether are expensive. The power efficiency of the SIMD approach is not yet well understood and requires more research to evaluate whether the dual imagers and the parallel processing of the subsets of the VGA image for frame differencing are beneficial in typical video sensor application scenarios. Overall, the node suffers from a mismatch between the extensive image and video capture capabilities and the limited wireless transmission capability.

E. Carnegie Mellon's CMUcam3 [89] and DSPCam [90], [91]

1) Overview: Carnegie Mellon's CMUcam3 sensor node is probably the most open of the heavily coupled platforms in that all hardware schematics, software, and PCB files are freely available online for the research community. Many commercial vendors are also allowed to copy, manufacture, and sell the platform with or without design modifications. CMUcam3 is capable of RGB color CIF resolution (352×288 pixels). At its core is an NXP LPC2106, which is a 32-bit 60 MHz ARM7TDMI MCU with built-in 64 KB of RAM and 128 KB of flash memory. It uses either an Omnivision OV6620 or OV7620 CMOS camera-on-a-chip/image sensor, which can load images at 26 fps. As shown in Figure 8, CMUcam3 also uses Averlogic's AL4V8M440 (1 MB, 50 MHz) video FIFO

buffer as a dedicated frame buffer between the camera and the host MCU. Hence, the actual capture-and-save frame rate is limited by the hardware FIFO buffer between the imager and the MCU. Clocking the frames out of the FIFO buffer to the MCU memory gives the actual overall capture-and-save frame rate. CMUcam3 has software JPEG compression and has a basic image manipulation library. CMUcam3 uses an MMC card attached via SPI for mass data storage. The card uses a FAT16 file system type, which is compatible to almost all other flash card readers.

An improved follow-up to CMUcam3 is the DSPCam [90], which has the characteristics of an externally dependent architecture and is therefore included in Table IV. Nevertheless, since DSPCam grew from CMUcam3, we discuss both in this section. As illustrated in Figure 9, DSPcam uses the 32-bit RISC Blackfin DSP-MCU SoC from Analog Devices and a SXGA (1280×1024), VGA (640×480), QVGA (320×240), and CIF capable OmniVision CMOS image sensor. A stand-alone WiPort 802.11b/g module is integrated on the board. DSPCam provides an interface for third party modules for possible 802.15.4 based radios as well as other low data rate sensors. The image array's throughput can be as high as 30 VGA fps and 15 SXGA fps. The imager consumes 50 mW for 15 SXGA fps with a standby power of 30 μW. DSPCam is a smart mote, which creates metadata and tags for video to enable efficient video retrieval and transmission. DSPCam runs a uCLinux OS and a custom Time Synchronized Application level MAC (TSAM) protocol which provides quality of service (QoS) through a priority-based dynamic bandwidth allocation for the video streams. TSAM bypasses standard Linux network
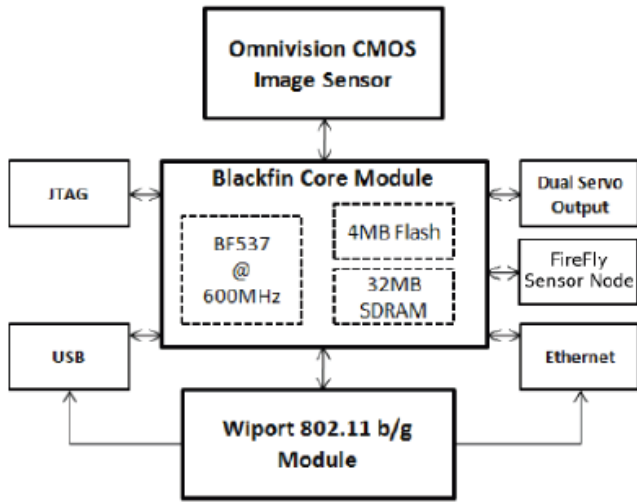
Fig. 9.   DSPCam's major block architecture [90].

API calls. Depending on the power states of the three major modules, the power consumptions of the DSPCam ranges from above 0.330 W (all idle) to 2.574 W (all active).

2) Advantages: The CMUCam3 hardware can carry out two modes of frame differencing. In low resolution mode, the current image of 88×143 or 176×255 pixels is converted to an 8×8 grid for differencing. In the high resolution mode, the current CIF image is converted to a 16×16 grid for differencing.

The single board FIFO-MCU architecture of CMUCam3 is faster than the PCB Header-MCU setup used in Cyclops. In particular, the FIFO buffer decouples the processing of the host MCU from the camera's pixel clock, which increases frame rates. Decoupling the MCU processing from the individual pixel access times allows the pixel clock on the camera to be set to a smaller value than the worst case per pixel processing period. As evaluated in [21], the Cyclops design is six times slower than CMUCam3. Compared to the 2 fps of Cyclops, CMUCam3 can capture and save between 2 and 5 fps. An additional advantage of the FIFO buffer is its ability to reset the read pointer, which enables basic multiple pass image processing, such as down sampling, rewinding, and windowing.

The CMUCam3's OV6620 camera supports a maximum resolution of 352×288 at 50 fps. CMUCam3 is capable of software based compression only and supports other optimized vision algorithms. The sensor node software provides the JPEG, portable network graphics (PNG), and ZIP compression libraries, which are useful for low data rate streaming.

The MCU of the CMUCam3 platform uses software controlled frequency scaling for power management. CMUCam3 has three power modes (active, idle, and power down). The camera module, for example, can be powered down separately without affecting the other two main CMUCam3 blocks.

The CMUCam3 MCU core has a memory acceleration module (MAM) for fetching data from flash memory in a single MCU cycle. The MMC option in CMUCam3 provides easy external access to its data as the data are readable by standard flash readers. The availability of serial in-system programming (ISP) provides for inexpensive built-in firmware

loading and programming as compared to many MCUs that require extra joint test action group (JTAG, IEEE 1149.1) hardware. The MCU provides a co-processor interface which can be useful for offloading some heavy computation from the host MCU. CMUcam3 provides an expansion port that is compatible with a variety of wireless sensor nodes, including the popular Berkeley sensor platforms.

DSPCam has considerably more memory than CMUCam3 with 32MB of fast SDRAM, clocked up to 133MHz, and 4MB of Flash. A new high-performance feature is the Direct Memory Access (DMA), which enables low overhead block transmission of video frames from the camera to the SoC's internal memory. This frees up the CPU core for other critical tasks. In addition to standard MCU interfaces, the Blackfin SoC provides a Parallel Peripheral Interface (PPI) which enables a direct connection to the CMOS image sensor. DSPCam accelerates video and image processing through its special video instruction architecture that is SIMD compliant. The USB-UART bridge provides useful external mass storage options for the DSPCam.

3) Disadvantages: The CMUCam3 design avoids high-cost components and hence lacks efficient storage and memory structures, such as L1 cache, memory management unit (MMU) and external direct memory access (DMA), as well as adequate random access memory (RAM) and flash memory. This shortcoming as well as the relatively slow I/O can be a throughput bottleneck. For example, reading one pixel value can take up to 14 clock cycles, of which 12 are wasted on waiting for input/output (I/O) transactions. The small memory of the "MMU less" ARM7TDMI core prohibits the use of even the tiniest Linux RTOS, such as uCLinux, which has been tested to work on other "MMU less" MCUs [102].

The coarse frame differencing leads to high object location error rates and is hence unsuitable for estimating object locations. Further, as used in [21], CMUCam3's processing and object detection algorithm (frame capture and frame differencing) were 5.67 times less energy efficient than Cyclops. The CMOS camera lacks a monochrome output mode, and hence color information must be clocked out of the FIFO. Also, the FIFO structure prevents random access to pixels.

The CMUCam3's MCU has very few I/O ports to enable extensible direct access to the MCU. That is, only a few I/O ports are configurable to be used for other I/O purposes and some bus protocols are underutilized. For example, the SPI bus has only one chip select pin, which is connected directly to the MMC card. This means that no other module can be connected to the SPI bus without first disconnecting the MMC card. This inflexibility may force designers to use alternate connectors, such as UART, which are slower and limit the throughput of the sensor node.

The optimization of the hardware architecture has focused on the video acquisition but neglected the wireless transmission and memory components critical to a WVSNP. Although a dedicated frame buffer speeds up and simplifies the camera image acquisition it is not accessible to other components when not in use. A DMA system would be more efficient and cheaper.

The CMUCam3 MCU, similar to many other low-cost systems, lacks the floating point hardware, RAM, and com-

putation speed required for many complex computer vision algorithms. Further, CMUCam3 lacks a real time clock (RTC) which could be critical in duty cycling of attached modules, global packet tracking, and time stamping of real time video.

During board power down, the RAM is not maintained. Therefore, the camera parameters must be restored by the firmware at startup. CMUCam3 takes relatively long (sometimes close to a second) to switch between power modes or to transition from off to on. These long switch times limit applications that require fast duty cycling and short startup times, for example, when alerted to capture a frame.

DSPCam depends on an external node or module, such as a Firefly sensor node, to provide access to low data rate nodes using IEEE 802.15.4-based radios. The DSPCam board includes an Ethernet module, which is operated in a bridge configuration for wireless transmissions with the attached Wiport module. The TCP/IP networking drivers thus continue sending data to the Ethernet module, which is then forwarded to the Wiport module for wireless transmission. At the same time, the core module directly controls the Wiport module via a serial port. This setup introduces inefficiencies as there is duplication in the wireless transmission path.

The DSPcam architecture does not provide mechanisms for the host SoC to control the power modes of the camera, the WiFi module, and other external nodes. This is a critical functionality for a low-power WVSNP. Future research needs to evaluate in detail the impact of the TSAM protocol and other in-node processing on the QVGA/CIF frame rate. Although DSPCam is a significant improvement over CMUCam3, it traded the highly coupled architecture of CMUCam3 for an externally dependent architectures that relies on third-party modules with no power management control.

VI. EXTERNALLY DEPENDENT ARCHITECTURES

A. Overview

Externally dependent architectures depend on a mosaic of external "daughter boards" to achieve basic functionality. The justification for this designs approach is that nodes operating at different tiers in a multi-tier network have different functionality requirements. As a result, the externally dependent architectures depend heavily on the designer's view of the sensor network and hence suffer from similar target application limitations as the heavily coupled architectures.

Nodes that depend on external PCB modules often lack a cross-platform standard interface, limiting interoperability with daughter boards. In particular, a given base platform can usually interoperate only with the daughter boards specifically designed for the base platform, limiting flexibility. This design model often hides the real cost of a node and results in cumbersome designs that are inefficient. For example, the use of basic interfaces, such as RS-232, Ethernet, USB, and JTAG on Stargate requires a daughter board. Similarly, a special daughter board is required to supply the Imote2 with battery power. Assembly of an image capable platform based ScatterWeb requires at least four different boards.

The need of externally dependent architectures for daughter boards for a basic application result often in excess power consumption. This is because each stand-alone daughter board
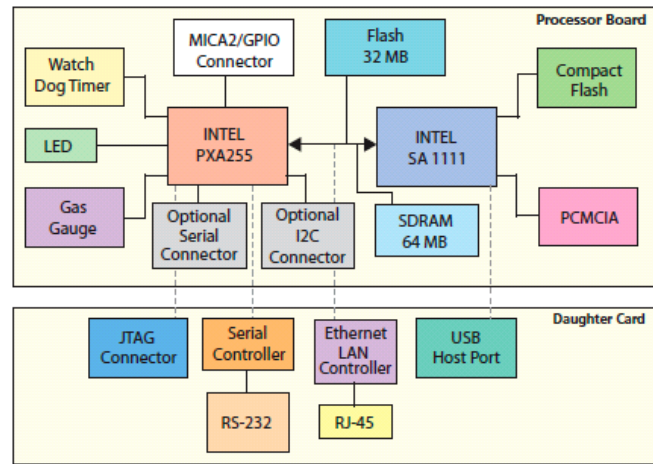

Fig. 10. Stargate architecture block diagram showing the main board and the daughter board.

needs some basic circuitry, which consumes power. This circuitry is usually duplicated on other daughter boards and hence consumes more power than reusing the same circuitry on one PCB.

B. UC Berkeley's Stargate [18], [21], [35], [92], [93]

1) Overview: Stargate is a relatively popular platform and is commercialized by Crossbow Technology Inc. The Stargate platform is capable of real-time video compression. The platform offers a wide range of interfaces, such as Ethernet, USB, Serial, compact flash (CF), and PCMCIA, making the platform suitable for residential gateways and backbone nodes in multi-tier sensor networks.

As illustrated in Figure 10, Stargate consists of an XScale PXA255 processor whose speed ranges from 100 to 400 MHz and consumes between 170 and 400 mW. The Stargate processor can be configured to have 32 to 64 MB of RAM and/or 32 MB of Flash. Energy profiling [92] shows that Stargate consumes more energy during intensive processing (e.g., FFT operations) and flash accesses than through transmissions and receptions. Interestingly, the energy consumption for data transmission was found to be 5 % less than that for data reception. This is a reversal of the typical characteristics of wireless devices and can be attributed to the specific employed duty cycling mechanisms. On average, Stargate uses about 1600 mW in active mode and around 107 mW in sleep mode.

2) Advantages: The Stargate platform is extensible enough that it can attach to other modules as needed to communicate with other wireless sensors and third-party application-specific modules. The platform has sufficient RAM and Flash memory to run a complete Embedded Linux OS. As a result, Stargate has extensive software capabilities, including support for web cams attached via USB or PCMCIA, and compact flash (CF) based 802.11b radios to communicate with higher data rate sensors.

The processor is sufficiently powerful to locally run object recognition algorithms. Studies have shown that Stargate is more energy efficient than Panoptes. It consumes 25 % less energy for some applications in spite of having twice Panoptes' processing power [21], [35], [92]. Increasing the clock speed

TABLE IV
SUMMARY COMPARISON OF EXTERNALLY DEPENDENT PLATFORMS. DISTINCT FEATURES OF CITRIC FROM THE SIMILAR IMOTE2 ARE IN BRACKETS.

	UC Berkeley's Stargate	Crossbow's Imote2/Stargate 2, [UC's CITRIC]	Freie Universität's ScatterWeb	CSIRO ICT 's FleckTM-3	CMU's DSPCam
Flexibility Rating	5/10	6/10	5/10	5/10	6/10
Processor(s), Core, Speed	XScale PXA255, 32-bit, 100 to 400 MHz, No Floating Point	XScale PXA271 SoC, 32-bit, 13 to 416 MHz, Intel Wireless MMX DSP Co-pr., No Float. Point [624 MHz, PXA270]	TI MSP430 (ESB430)	8 MHz 8-bit Atmega128, TI 32-bit DSP daughter b.	ADI 8 ADSP-BF537 Blackfin Processor, 600 MHz, DSP-MCU SoC
Node Power and Supply *(mW)*	170 to 400 mW, AA cells, DC input only via daughter board	231 mW, Liion / Li-Poly / 3×AAA NiMH / standard cells (via daughter board), via USB mini-B [428 mW (idle) to 970mW (active at 520 MHz), 4×AA, or USB, or 5V DC]	165 mW, 1 F gold-cap capacitors for energy harvesting (10 hours)	DSP daughter board (290 mA), AA cells Integrated solar charger	0.330 W (all idle) to 2.574 W (all active), 0.8 to 1.32 V, 3.3 V DC
Supported Power Modes	Sleep (107 mW), Active (1600 mW)	Deep Sleep (1.365 mW), Active Low Voltage (26.35 mW, 13 MHz), Active (231 mW, 416 MHz) [428 mW (idle) to 970mW (active at 520 MHz)]	Sleep (100 mW), Active (165 mW), Off	None	Active, idle, standby
Node and Peripheral Power Management	No support for network wake, battery monitoring utility	On PCB power management chip, frequency control from 13 MHz to 416 MHz with Dynamic Voltage Scaling [CPU speeds 208, 312, 416, and 520 MHz, External NXP-PCF50606 PMIC]	None	Board to daughter mode flexibility	Dynamic clock up to 600 MHz.
Memory / Storage	64 MB SDRAM, 32 MB Flash	256 KB in core SRAM, 32 MB in-SoC SDRAM, and in-SoC 32 MB Flash [16MB NOR FLASH eXecution-In-Place (XIP) and 64MB RAM external running at 1.8 V]	64 KB EEPROM (within camera)	128 KB on-chip, 1 MB external SRAM	32MB of SDRAM clocked up to 133MHz, and 4MB of Flash.
I/O, Interface	Ethernet, USB, UART, JTAG (on daughter board), PCMCIA, I2C	3×UART, 2×SPI, I2C, SDIO, I2S, AC97, Camera Chip Interface, JTAG, USB, Tmote Sky	UART, USB, I2C, OTA programming (via ScatterFlash board)	I2C, UART, SPI	USB, JTAG, Ethernet, PWM, UART, I2C, TWI, FireFly, SPI
Radio	PCMCIA or CF based 2.4 GHz (802.11b)	CC2420 802.15.4 radio and 2.4GHz antenna, 30 m range [Tmote Sky mote with 801.11.15 radio]	RFM TR1001 868 MHz, CC1021 434 MHz	Nordic NRF905	Stand alone 802.11b/g module, FireFly mote with 802.11.15.
Wireless Trans. Rate	802.11b (< 11 Mbps),	< 250 kbps	None	76.8 kbps	802.11g (< 54 Mbps),
Imager, Max Imager Resol., Max Frame Rate	Logitech Pro 4000 USB Webcam, VGA (640 × 480)	None [Omnivision 1.3 megapixel camera, OV9655, 15 SXGA (1280×1024) fps, 30 (640×480) VGA fps, CIF to 4×30]	COMedia C328-7640, VGA (640 × 480/16-bit)	Omnivision OV7640, VGA, 30 VGA fps, 60 QVGA fps	OmniVision OV9653 CMOS image sensor, VGA (640×480), SXGA (1280×1024), CIF.
Capture-Save Frame Rate	15 fps (CIF)	Not evaluated [OV9655, 15 SXGA (1280x1024) fps, 30 (640x480) VGA fps]	Not evaluated	Not evaluated	30 VGA fps and 15 SXGA fps.
HW Image Processing	None	MMX DSP with 30 media instructions for video [separate camera module]	In camera JPEG block	TMS320F2812 32-bit DSP	Parallel Peripheral Interface (PPI), DSP assisted SIMD, DMA
SW Image Processing	None	None. [JPEG, OpenCV]	None	None	JPEG
Frame Trans. Rate	Not evaluated	Not evaluated	0.75 to 6 fps (160 × 128/8-bit), 3–5 fr./min. (VGA)	2 QVGA fps	approx. 5 QVGA/CIF fps
OS / RTOS	Linux OS (kernel 2.4.19)	Linux, TinyOS, SOS or Microsoft .NET Micro	None	TinyOS	uCLinux
Cost	$500	$300 [Unknown, medium]	Unknown	Unknown	Unknown, medium

of the Stargate MCU by 300 % results only in a small increase of 24 % in power consumption [93], which is a desirable characteristic for a video processing MCU.

3) Disadvantages: As used in [35], Stargate operates akin to a computer networking gateway interface and is architecturally too general and not optimized for low power consumption. It uses power-inefficient interfaces, such as a personal Computer memory card international association (PCMCIA) interface based card for the 802.11b module. The PCMCIA standard is a general computer standard and not readily optimized for a low power sensor.

The webcam attached to Stargate is not suitable for a resource-constrained standalone video sensor. Stargate does not have hardware support for being woken up by other motes. Special mechanisms have to be implemented on the other connected motes to mimic the wake-up functionality. This makes Stargate dependent on the Mica-type motes for the wake-up functionality. Stargate is also dependent on the Mica-type motes for simultaneous 900 MHz low-data rate transmissions. The extra wakeup overhead adds to wakeup latency costs. The latency and power consumption further increase due to the architecture's inefficient reliance on the daughter board for Ethernet, USB, and serial connectors, see Figure 10. Though both the main and daughter boards have battery input, only the daughter board has a direct current (DC) input, which increases the main board's reliance on the daughter board.

Regarding the multimedia functionalities, the XScale MCU lacks floating-point hardware support. Floating-point operations may be needed to efficiently perform multimedia pro-
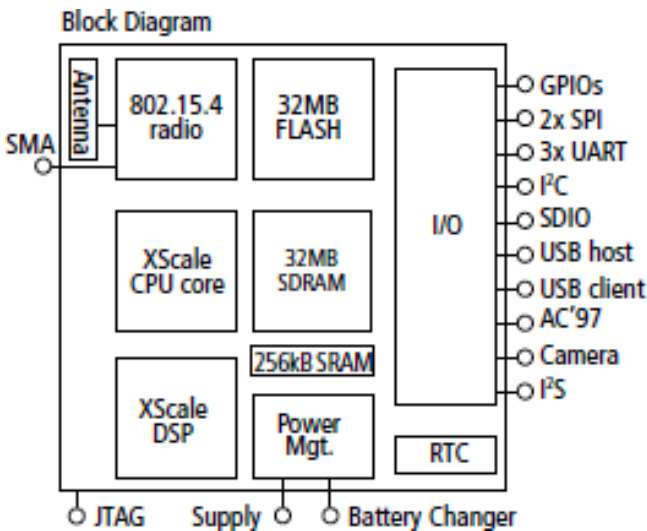
Fig. 11. Imote2 block architecture [94].

cessing algorithms. Images acquired through USB are typically transmitted to the processor in a USB compressed format. This adds to decompression overhead prior to local processing as well as loss of some image data. The employed version 1.0 USB is slow and limits image bandwidth.

C. Crossbow's Imote2/Stargate 2 [1], [92], [94] and UC's CITRIC [95].

1) Overview: Imote2 is the latest in a series of attempts to create a powerful and general sensor node by Intel and Crossbow. Its predecessors, the original trial Imotes, lacked many elements expected of a WVSNP. The first trial Imote used a slow 8-bit 12 MHz ARM7 MCU with 64 KB RAM and 32 KB Flash memory. Its successor used an ARM7TDMI MCU with 64 KB SRAM, 512 KB Flash, and speed ranging from 12 to 48 MHz. The first two Imotes had an on-board Bluetooth radio and support for the TinyOS RTOS.

Compared to its predecessors, Imote2 has substantially increased computation power and capabilities. It features a PXA271 XScale SoC. The SoC's 32-bit ARM11 core is configurable between 13 and 416 MHz clock speeds. The ARM core contains 256 KB SRAM, and is attached to a 32 MB Flash, and 32 MB SDRAM storage within the SoC. Imote2 has a Zigbee compliant IEEE 802.15.4 CC2420 radio and a surface-mount antenna, but has no default Bluetooth radio. Supported RTOSs for Imote2 are TinyOS, Linux, Microsoft's .NET Micro, and SOS. Imote2 is intended to replace the original Stargate platform and is therefore also referred to as Stargate 2.

A similar recent platform, CITRIC [95], Figure 12, by the Universities of California at Berkeley and Merced as well as the Taiwanese ITR Institute is a follow-up design to Imote2. CITRIC consists of a 624 MHz frequency-scalable XScale MCU, 256KB of internal SRAM, 16MB FLASH, and 64MB external low-power RAM running at 1.8 V. Compared to Imote2, CITRIC is more modular in its design in that it separates the image processing unit from the networking unit. CITRIC also uses a faster Omnivision 1.3 megapixel camera, OV9655, capable of 15 SXGA (1280×1024) fps, 30

(640×480) VGA fps, and a scale-down from CIF to 40×30 pixels. CITRIC runs embedded Linux. The imager has an active current consumption of 90 mW for 15 SXGA fps and a standby current of less than 20 μA. CITRIC has an overall power consumption from 428 mW (idle) to 970 mW (active at 520 MHz). This means that CITRIC can last for slightly over 16 hours with four AA batteries with a power rating of 2700 mAh.

2) Advantages: The PXA271 XScale in Imote2 is a very powerful SoC platform, combining an ARM11 Core, a DSP core, as well as Flash and RAM memories. This compact design improves data access and execution speeds and facilitates power management algorithms that use the SoC's power modes. Specifically, the clock speed of the Imote2 MCU (PXA271 XScale) has a very wide range of power applications through its use of Dynamic Voltage Scaling. It can be set to as low as 13 MHz and can operate as low as 0.85 V, which enables very low power operation.

The Imote2 on-chip DSP coprocessor can be used for wireless operations and multimedia operation acceleration. This co-processor improves the parallelism of the node, especially for storage and compression operations.

The nodes have large on-board RAM and Flash memories. Imote2 provides an interface to support a variety of additional or alternate radios. Further, Imote2 has a variety of targeted high-speed standard interface modules, such as I2S and AC97 for audio, a camera chip interface, and a fast infrared port, in addition to the usual MCU interfaces, such as UART and SPI.

The latest Imote2 board is quite compact, measuring 36 mm × 48 mm × 9 mm, enabling its inclusion in many sensor node applications. Further, the support for many RTOSs, especially Linux, makes it a good choice.

CITRIC's modular separation of the image processing unit from the networking unit makes it more adaptable to applications than Imote2. CITRIC's 16 MB external Flash is a NOR type memory with faster access times than the typical NAND based memories. It also is capable of the latest Linux supported eXecution-In-Place (XIP), which provides the capability to boot-up and execute code directly from non-volatile memory. The USB-UART bridge provides useful external mass storage options for CITRIC.

The very low standby current consumption of 20 μA makes CITRIC a good candidate for power conservation with duty cycling. Further, the choice of low-power memory is significant as memory typically consumes about the same power as the processor, that is, approximately 20 % of the node's power. The CITRIC cluster of boards can be powered with four AA batteries, a USB cable, or a 5 V DC power adapter.

3) Disadvantages: Though Imote2's PXA271 provides many peripheral interfaces suitable for multimedia acquisition and processing it depends heavily on external boards for basic operations. These external boards include daughter boards for battery power supply as well as JTAG and USB interfaces. The many attachments required for core functionalities make the platform eventually expensive. Also, the hierarchy of hardware PCBs required for core functionalities introduces latency and power drawbacks similar to those arising with Stargate.

Any high-throughput wireless transmission of multimedia will also need an external board attachment. The surface
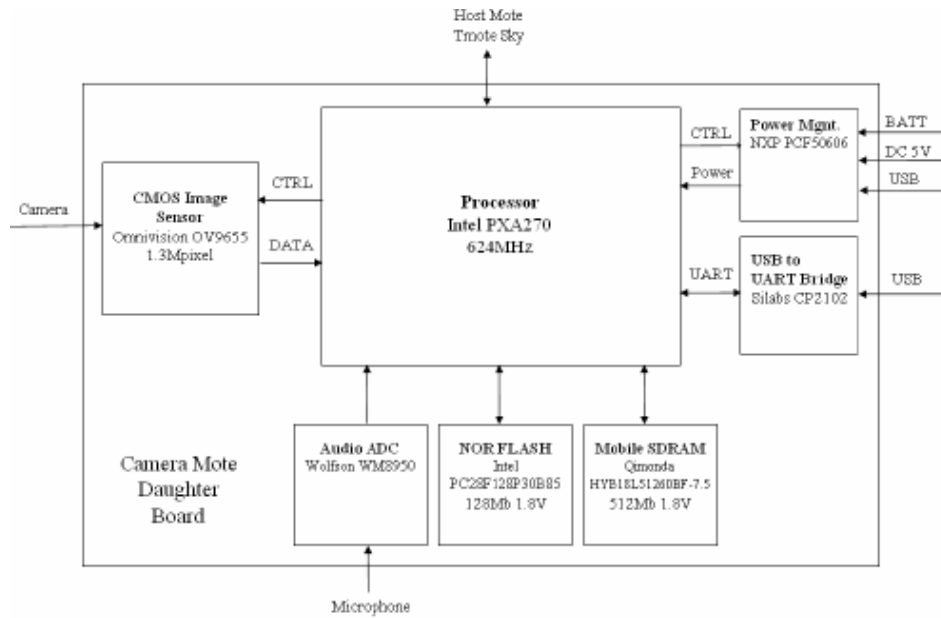
Fig. 12.   CITRIC block architecture [95].

mount antenna for the on-board Zigbee radio has only a range of 30 m line of sight, requiring an external antenna. Moreover, CITRIC depends on an external Tmote Sky mote with a Zigbee-ready radio for low data rate wireless transmissions. Additionally, CITRIC's camera is attached to a separate camera daughter card. Both the main processor board and the camera daughter board depend on the Tmote Sky mote for battery operated power. This introduces power inefficiencies due to the high number of passive components on each board.

The PXA270 CITRIC core does not support NAND type memories, which limits the designer's choices. Although CITRIC has a power management IC, it is located on the camera daughter board which means the main processor board is dependent on the camera to manage its power. Similar to Imote2 this architecture is heavily externally dependent and despite its higher computational power, it does not have the radio hardware resources for faster video streaming.

D. Freie Universität ScatterWeb's ESB430-, ECR430-COMedia C328x modules [96]

1) Overview: This is a platform designed for research and education. To accommodate diverse research and educational needs it consists of a mosaic of function-specific PCB modules that can be assembled for a desired application area. A sensor node built with these function-specific PCB modules may form an ad-hoc network with other nodes. Some nodes can act as data sources, some as relays, and some as data collectors. A node can simultaneously perform all three functionalities. There are many translator gateway boards to interface ScatterWeb-type boards with standard interfaces, such as RS485, Bluetooth, Ethernet, and USB.

A camera node can be assembled from the ScatterWeb boards by combining an embedded sensor board (ESB, i.e., ESB430), an ECR430 board, and a COMedia C328-7640 VGA (640×480 16-bit pixel) camera module. The camera's resolution can be configured to 80×64, 160×128, 320×240,

and 640×480 pixels. The ESB430 can be programmed via UART or USB. The ESB typically has a TI MSP430 MCU, a transceiver, a luminosity sensor, a noise detector, a vibration sensor, an IR movement detector and IR transceiver, a microphone/speaker, and a timer.

The radios are usually 868 MHz RFM TR1001 transceivers and lately the longer range 434 MHz CC1021 transceiver from Chipcon. For energy harvesting, the nodes store solar cell energy in gold-cap capacitors. Piezo crystals and other thermo-elements are also used.

The camera modules have a VGA camera chip and a JPEG compression block. They draw 50 mA while operating at 3.3 V. They are about 2×3 cm^2 in area. The camera module takes commands via the serial interface, processes/compresses the image, and feeds back the resulting image through the same serial port. The VGA frames can be compressed to 20–30 KB sizes. Images are first transferred from the camera module to the built-in 64 KB EEPROM and then transmitted over the air.

2) Advantages: The PCB module based architecture provides flexibility of reconfiguring the platform for different uses. A cascade of an embedded sensor board (ESB) with compatible GSM/GPRS modules and embedded web server modules (EWS) provides a gateway to receive configuration commands and send node data from/to the Internet and cellular networks.

One of ScatterWeb's PCB modules, the so-called Scatter-Flasher, can be attached to a PC for over-the-air programming (flashing) of all sensors, debugging, and remote sensor data collection. Other boards, such as embedded web server (EWS) use power over Ethernet (PoE) to power the host MCU and other PCB components. This is a good way to reduce cost. The EWS can be used to setup ad-hoc Ethernet networks.

The MCU requires about 2 μA in deep-sleep mode, which is power efficient for duty cycle applications. The entire camera module uses about 100 μA in power down mode.

The ESB can switch off the camera module's power supply for additional energy savings. The energy scavenging options provided by the nodes make them candidates for long-term outdoor deployment. The employed 1 F capacitors last for about ten hours for typical monitoring, which is enough energy for over 420 sensing and sending cycles.

3) Disadvantages: While the PCB module based architecture of ScatterWeb provides flexibility, this design strategy suffers from extensive component repetition and underutilization since the modules are expected to be stand alone. Also, the ESB lacks the interfaces and power management infrastructure to control power modes of the individual components on the attached boards.

The serial interface has a maximum data rate of 115 kbps, which is low for image transfer. The module can only wirelessly stream a 160×128 8-bit preview video at 0.75–6 fps. Downloading a compressed image from the camera module to the ESB takes about 2 s. Transmitting an image can draw 7 mA and take about 9.6 s. Overall, this consumes about 0.058 mAh per transmitted image. This translates into about 27,000 images for a rechargeable AA battery with a 2000 mAh capacity and a usable capacity of 80%. As evaluated in [96], a 20–30 kB image takes 12 to 17 s to send, which allows capturing and transmitting only 3 to 5 compressed images per minute.

E. CSIRO ICT Centre's FleckTM-3 [24], [97]

1) Overview: Fleck-3 is made up of an 8 MHz Atmega128 MCU running a TinyOS RTOS. The platform consists of a 76.8 kbps Nordic NRF905 radio transceiver and two daughter boards: one for the camera and one for all image processing operations, as illustrated in Figure 13. The daughter boards interface and communicate with Fleck-3 via SPI and GPIO interfaces and relevant interrupts.

The DSP daughter board consists of the TI TMS320F2812, a 32-bit, 150 MHz DSP with 128 KB of on-chip program FLASH and 1 MB of external SRAM. The camera board is made up of an Omnivision OV7640 VGA (640×480) or QVGA (320×240) color CMOS sensor with Bayer pattern filter [103]. The progressive scan sensor supports windowed and sub-sampled images. The DSP on the daughter board can control and set camera parameters via an I2C bus. Frames are moved from the sensors into external SRAM using the circuitry implemented in an FPGA on the DSP daughter board. Reference frames are also stored on the DSP board's external memory.

2) Advantages: The choice of a 32-bit DSP chip satisfies the 32-bit energy advantage over 16- or lower-bit MCUs, see Section II. The 32-bit DSP achieves 0.9 MIPS/mA compared to 2.1 MIPS8/mA for the 8-bit Atmega 128L. Also, the acquire, compress, and transmit strategy has been shown to be eight times more energy efficient than the acquire, store, and transmit strategy [24], [97], justifying the compression stage in the architecture.

The daughter cards can be turned on and off by the Fleck baseboard. This board-to-board power mode flexibility and the use of interrupts for communication with the Fleck-3 can be used by power management algorithms. The separation
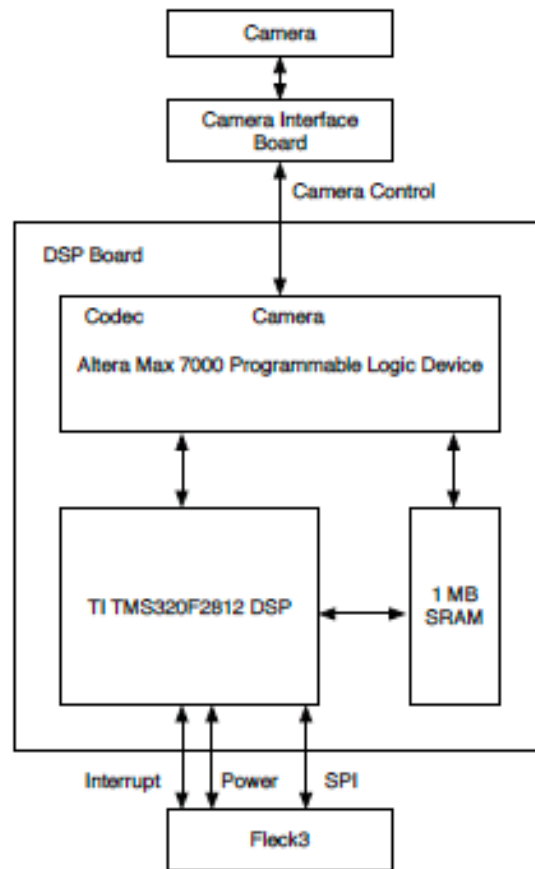

Fig. 13. Hardware architecture of a camera node formed by a Fleck sensor node, a DSP board, and an image sensor [97].

of functionality into daughter boards also provides flexibility and expandability for the Fleck-3 platform. The DSP chip is programmable in C which is widely supported.

Fleck-3 has a large 1 MB flash memory, which is sufficient for a real time operating system (RTOS). The combination of a real time clock (RTC), an integrated solar battery charging circuit, and regulator facilitates intelligent power management schemes. For example, RTC interrupts can be used based on the time of day to choose energy sources and even schedule storage or recharging. The circuit can also monitor battery and solar current and voltage. This makes the platform ideal for long-term outdoor use.

The camera board (in addition to the image sensor chip and lens holder) has two ultra-bright LEDs for capture illumination. The camera by itself is capable of acquiring a maximum of 30 VGA fps or 60 QVGA fps.

3) Disadvantages: Fleck-3 camera functionality requires both the camera and the DSP daughter boards. The DSP board alone draws a current of more than 290 mA when active. Taken together, this platform architecture is relatively power inefficient and has a costly component count.

Support and reusable software for the TinyOS are not as readily available as for other open source RTOSs, such as embedded Linux, uCos, and FreeRTOS. While the image sensor is capable of acquiring up to 60 QVGA fps, the camera can only stream compressed QVGA images at up to 2 fps [97], limiting its usefulness for a WVSNP.

Another Fleck-3 limitation is its use of a serial interface to a gateway computer to perform as a base node for network management. This is single point of failure, a bandwidth bottleneck for the network, and limits flexibility of the Fleck-3 network. The radio is very low data-rate and uses custom network access and radio management protocols. Taking advantage of open radio standards would likely reduce cost and improve compatibility with other WVSNPs.

F. University of Franche-Comte's ACME Fox based node [98]

While this node is very similar in its design as well as advantages and shortcomings to the preceding externally dependent architectures (and is therefore not included in Table IV), we briefly note its distinguishing features. This sensor node relies exclusively on a Bluetooth radio. This radio choice is an interesting attempt to strike the balance between a high-power 802.11 (WiFi) radio and a limited data rate 802.15.4 (Zigbee ready) radio with very low energy consumption. The node has also an energy analyzer module that reports current consumption. The energy analyzer helps in revealing an application's power consumption characteristics and enables designers to fine-tune operational algorithms.

VII. SURVEY SUMMARY

Of all the sensor node platforms reviewed in Sections IV through VI, only few node platforms approach the architectural requirements required for WVSNP functionality. For instance, Imote2 and CITRIC approach WVSNP functionality provided the daughter boards are judiciously selected and the HW/SW is efficiently integrated. Unfortunately, Imote2 and CITRIC still suffer from the limitations of the externally dependent architecture category. The externally dependent platforms have architectures that are extensible and general enough as WVSNP candidates. However, they lack critical features, such as compression modules, high bandwidth wireless transmission, power mode flexibility, memory resources, and RTOS capability.

As noted in Tables II through IV, the wireless video capture and transmission capabilities of many implemented platforms have not been quantitatively evaluated and reported. None of the existing sensor node platforms has demonstrated the wireless transmission of more than 4 fps of CIF video.

The prevailing shortcoming of the existing platforms is that they have some image acquisition capability but lack the necessary HW/SW integration to achieve commensurate processing and wireless transmission speeds. In other words, the HW/SW integration and performance considerations have not been consistently examined across all major stages of the video acquisition, processing, and delivery path. Further, consistent attention to power management has been lacking.

VIII. FLEXI-WVSNP DESIGN

A. Overview

The preceding survey of the state of the art in video/image capable node platforms for wireless sensor networks revealed the need for a platform that is designed to incorporate acquisition, processing, and wireless transmission of multimedia

signals. The sensor node should operate in practical application scenarios and with practically useful image resolution while satisfying the cost and resource constraints of a sensor node. In this section we outline a novel Flexi-WVSNP design to achieve these goals. We first provide the rationale for our major system design choices and then describe the hardware and software architecture.

B. Overall Flexi-WVSNP Design Concept and Architecture

We design *Flexi-WVSNP* as a video sensor node capable of wireless video streaming via both Zigbee and WiFi. Such a dual-radio system (i) integrates well with other Zigbee sensors, and (ii) provides gateway access for the sensors to the Internet via WiFi.

As analyzed in the preceding survey sections, most existing designs have the shortcoming of either attempting to incorporate too many components to cover an overly wide application range resulting in general purpose architectures, or attempting to be too specialized for a very narrow specific application resulting in heavily coupled architectures. In contrast, our design strives for high cohesion by meshing hardware and software architecture, while at the same time avoiding the tight binding (coupling) of components to each other as in the heavily coupled and externally dependent architectures. Our design strives to be highly adaptable and cost flexible; such that in its barest form, it may consist of only a processor. We believe that a WVSNP design needs to be application-targetable within a few days if it is to cover a wide array of cost-sensitive applications ranging from low-cost surveillance to remote instrument monitoring and conventional web camera.

Our generic WVSNP architecture follows a design concept that (i) eliminates the hard choices of anticipating a specific application scenario, and (ii) initially bypasses the tedious process of designing a comprehensive WVSNP. Our design concept is motivated by the basic fact that hardware and semiconductor processes will continue to improve and hence power savings will depend on the main components added for the specific application. This means that the major initial decision is the processor selection. The processor should be a powerful yet efficient System on a Chip (SoC) that satisfies essentially all requirements for a WVSNP in Section II. Almost each module within the SoC should be able to be independently controlled from active power state all the way to off. The SoC needs direct hardware supported co-processor module capability and accelerators useful for video capture, encoding, and streaming. The remaining functionalities can be achieved by flexible connectors, e.g., the open general purpose input output (GPIO) ports of the main processor, to the video sensor and wireless modules.

Another major motivating factor for our design concept is that software is improving continuously and open source software, in particular, is evolving at an astonishing rate. This means that tying the design to existing software and hardware limits the system and violates the requirements for application adaptability, as well as low power and cost. A real time operating system (RTOS) is definitely necessary and it should only serve the purpose of booting up the initial

master controller and allow loading modules as needed by the configuration. Depending on the configuration, modules should be able to control themselves if the master module is unable to (e.g., if the master crashes) or if application design prescribes that they should bypass the master under certain conditions, e.g. independent real time operation.

C. Middleware and the Dual Radio Approach

The Flexi-WVSNP design strives to achieve cost effectiveness and flexibility through a robust middleware that delivers two major capabilities. First, the middleware introduces an operating system (OS) independent abstraction layer for inter-chip communication. This provides a semi-high level application programming interface (API) that enables the processing modules to communicate with each other regardless of the OS or underlying hardware interconnect.

Second, we employ the middleware for seamless and dynamic interchange of radios as required by data rates or data type. For example, if a small volume of temperature data is sent by a temperature sensor, the small data amount should automatically go out via the Zigbee radio and not the WiFi radio. If a large volume of video data is sent to a remote site via the Internet through the home router, the data should automatically go out through WiFi. If the video data is requested by some low resolution display that is Zigbee capable and is within the limit of the Zigbee data rates the video data should go out via the Zigbee radio. More generally, the middleware should switch between the two radios and control their rates such that the dual-radio appears as a single radio to the application. This transparent parallel WLAN-Zigbee radio software design enables a seamless operation and handover between small sensors and WLAN, Wi-Max, and/or Cellular devices. Thus, the dual-radio design enables the Flexi-WVSNP to function as a primary sensor, a relay within a Zigbee or WiFi network, or a gateway between Zigbee and WiFi networks.

Our design avoids a software-based solution of the radio control, which would demand memory space, execution time, and power from the host MCU. Instead, we exploit the increasing processing power and decreasing cost of radio SoCs. These radio SoCs can operate as separate and stand-alone wireless components. Our design only requires a simple middleware that allows the MCU to interface with and transparently use both radios. Each radio SoC operates its own network stack, QoS, and power saving mechanisms. This approach offloads the radio communication tasks (such as channel monitoring) and networking tasks (such as routing) from the host MCU and RTOS. Furthermore, the radios within the wireless modules can operate statemachines for channel monitoring without using their built-in firmware, which saves more power. The host MCU would still control the power modes of the attached radios.

Instead of a customized optimal radio, we prefer to employ proven standardized radio technologies that take advantage of multi-channel and spread-spectrum technologies. We intend to rely heavily on the Zigbee half of the dual radio for the main power and network management and coordination functions. This choice is motivated by the energy efficiency of Zigbee
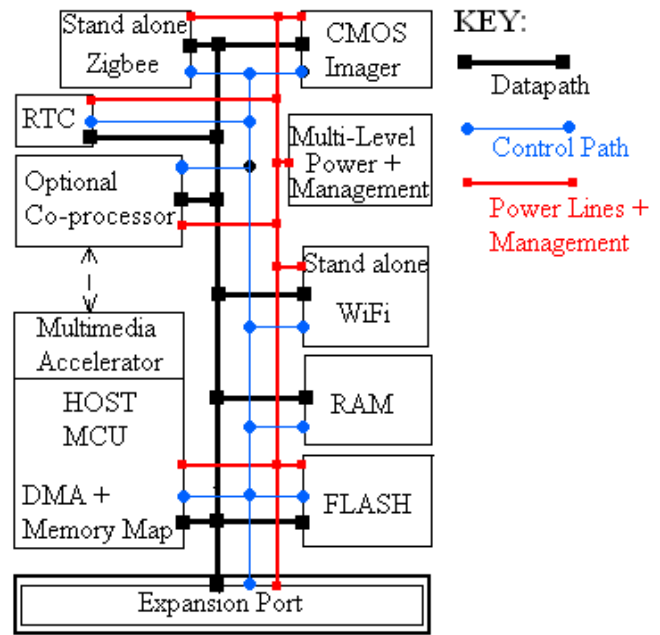

Fig. 14. Flexi-WVSNP hardware architecture block diagram.

which can last years on an AA battery [104], [105], as well as a wide range of useful Zigbee mechanisms, including built-in scanning and reporting, which automatically selects the least-interference channel at initial network formation, as well as Zigbee mesh networking and path diversity features. In this context, we also envision to exploit recent miniature antenna techniques, e.g., [106], for efficient video sensor platform design.

A number of studies, e.g., [107]–[111], have examined Zigbee-WiFi co-existence issues and concluded that typical WiFi usage patterns do not severely disrupt Zigbee. Even under very severe interference conditions, such as overlapping frequency channels and real-time video traffic, the transmission of Zigbee packets is not crippled, but may experience increased latency. The studies [107]–[109], [111] have shown improved coexistence properties in later generations of WiFi, such as the 802.11g and 802.11n. This is explained by short on-air packet durations in and hence fewer interference and collision opportunities.

D. Powering the Flexi-WVSNP

Our architecture is designed for low power usage as well as to allow power management algorithms for a variety of applications. We therefore use the cheapest and readily available off-the-shelf battery technology, such as lithium-ion batteries. We propose to employ a multiple-output voltage regulator to disperse different voltage levels, as needed, to the various PCB components. As the WVSNP can also be used as a gateway node or continuous surveillance camera, we include an 110 V mains power supply.

E. Flexi-WVSNP Hardware Block Design

As shown in Figure 14, every component of the Flexi-WVSNP is connected to all other components. All functional

pins of all components are exposed to the outside of the PCB via a flexible "mega" expansion port. This design is inspired by CPU integrated circuits (IC) architectures and VLSI design concepts, which until now were only used within an IC, not outside it.

The Flexi-WVSNP requires a SoC with multimedia capabilities. Multimedia capabilities can be an MCU specific extended multimedia instruction set, or hardware compression and processing engines. Additionally, the SoC requires a built-in co-processor interface for high throughput cooperation with an attached module, such as a DSP or other dedicated video processing module. A Flexi-WVSNP SoC requires a direct memory access (DMA) sub-module and/or a memory management unit (MMU) sub-module to enable transparent data accesses and exchanges between the PCB modules without continuous MCU coordination.

Importantly the Flexi-WVSNP SoC needs to be a modern power managed chip with almost total control of the power modes of its sub-modules. In addition, dynamic voltage scaling can be employed to save power. The overall clock speed of the MCU should be tunable over a wide range of sub speeds via software control. The focus on power variability is critical as large power savings in an application is achieved through power aware algorithms [40].

The visibility of the control path signals to all major modules of the PCB enable all modules to participate in power management control, which can be software initiated or externally managed by a power management module. The individual modules should be selected based on their ability to support a wide range of the power modes supported by the SoC.

Unlike most existing designs, the Flexi-WVSNP wireless modules are stand-alone modules that incorporate the necessary protocol stacks and power management options. The SoC and other modules on the PCB view the wireless modules as available data channels. The control path still exposes the wireless modules to further control or interventions by the SoC if needed, especially by power management algorithms or configuration commands.

CMOS imagers have become very popular due to their low cost and increased in-chip processing of the acquired image [112], [113]. The Flexi-WVSNP imager module is intended to be mainly stand alone in capturing a VGA image with possible resolution, zoom, sampling, and color space selection commands issued by the SoC. Although an advanced module may have a built-in compression module, we believe that compression is more efficient on computationally advanced modules, such as the SoC and/or coprocessor. This is because high level inference algorithms can be used to decide when it is necessary to compress frames. Compression on the SoC or coprocessor also provides for flexibility of compression schemes as better algorithms are developed.

The RTC maintains temporal accuracy of the system and in conjunction with the power module (see Section VIII-D) can be used to implement time triggered duty cycles or power states for any module in the control path.

The direct access of the modules to memory resources through the use of DMA allows the imager to operate at camera frame speeds. As observed in Sections IV through VI
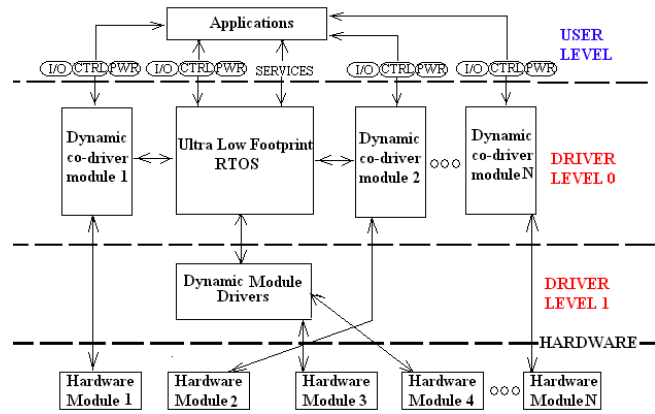

Fig. 15.   Flexi-WVSNP software architecture.

most existing platforms have high frame rate imagers but are limited by a "capture to pre processing-storage" bottleneck.

F. Flexi-WVSNP Software Architecture

Figure 15, shows the Flexi-WVSNP software architecture. The modular design of the architecture enhances power efficiency by enabling each sub-component of the platform to be powered and controlled individually as well as allowing applications direct and fine-grained access and control of the hardware. The architecture satisfies the WVSNP expectation of a decoupled but highly cohesive platform. This is an advantage over traditionally stacked or layered architectures whose components suffer from layer dependencies and power inefficiency.

As shown in Figure 15, applications are aware of the hardware modules and treat them as I/O data channels with control parameters (CTRL) and power states (PWR). This ensures that the relationship between the hardware modules and applications is data centric and enables application algorithms to be power aware. The real time operating system (RTOS) is the main scheduler for applications and driver modules.

An important feature introduced in Flexi-WVSNP is the ability for some trusted drivers to be at the same priority and capability level as the RTOS. As shown in Figure 15, the dynamic co-driver modules (DyCoMs) are those special drivers that at initialization load as normal drivers but then acquire full control of hardware and can run independently of the RTOS. Middleware, such as the transparent dual radio (see Section VIII-C), is implemented as DyCoMs. Thus, when the RTOS crashes, some critical applications continue to function for a graceful exit or intervention.

Each hardware driver provides a three part bidirectional generic interface to an application, that is, I/O, control, and power states. This enables uniform use of the hardware architecture. Traditional RTOS drivers can still be used as shown for example for Driver Level 1 controlling hardware modules 3 and 4 in Figure 15. This facilitates the low cost reusability advantage of popular RTOSs, such as the Linux based RTOSs.

Ensuring that the DyCoMs and the rest of the drivers are dynamic and have a direct relationship with the hardware modules enables the Flexi-WVSNP software architecture to

closely match the hardware architecture. Adding or removing a hardware module is directly related to adding or removing a software module. For example, exchanging the imager only requires unloading the old imager driver module, loading the new driver module together with the new hardware. The "software module" to "hardware module" match in the Flexi-WVSNP design further enables design time PCB software simulation, which enables high flexibility in component choices and hence low system cost. Moreover, forcing modules to follow the three part (I/O, CTRL, and PWR) bidirectional generic interface with an application reduces the maintenance cost, improves upgradeability, and enables power sensitive operation.

We expect the Flexi-WVSNP to deliver real-time frame rates via WiFi of between 15 and 30 VGA fps. This assumes an average WiFi data rate of around 1.5 Mbps using H.264 SVC compression [114], [115]. We expect to deliver between 15 and 30 CIF fps via Zigbee transmission. This assumes an average Zigbee data rate of 250 kbps with H.264 SVC compression. Since the management of the Flexi-WVSNP network is done primarily with Zigbee, we expect Flexi-WVSNP to last months to a year with 4 AA batteries for an application with a low frequency of events requiring video streaming.

IX. Conclusions

We have conducted a comprehensive survey of wireless video sensor node platforms (WVSNPs), which we classified into general purpose platforms, heavily coupled platforms, and externally dependent platforms. We found that the existing platforms do not meet practical WVSNP power consumption limits, cost constraints, and video frame throughput requirements and uncovered the shortcomings of the existing platforms. Many existing platforms lack the cohesive design necessary to match image processing and transmission capabilities to the image capture capabilities. Based on the insights from our survey, we proposed a novel Flexi-WVSNP design that strives for a cohesive HW/SW co-design and provides a flexible, low-cost design framework for a wide range of sensor application and networking scenarios. Core components of the Flexi-WVSNP design are (i) a dual WiFi-Zigbee radio for flexible video streaming, (ii) a middleware layer that transparently controls the dual radio, (iii) an innovative hardware-software module match architecture that allows dynamic (software) co-driver modules full control of hardware modules. Our future work is focused on testing and refining the Flexi-WVSNP design, initially through simulations with models based on system description languages and comparisons with performance measurements of actual Flexi-WVSNP nodes.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, Mar. 2007.

[2] A. Bachir, M. Dohler, T. Watteyne, and K. Leung, "MAC essentials for wireless sensor networks," *IEEE Commun. Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, Second Quarter 2010.

[3] J. L. Hill, "System architecture for wireless sensor networks," Ph.D. dissertation, University of California at Berkeley, 2003.

[4] S. Misra, M. Reisslein, and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE Commun. Surveys & Tutorials*, vol. 10, no. 4, pp. 18–39, Fourth Quarter 2008.

[5] M. A. M. Vieira, C. N. Coelho Jr., D. C. da Silva Jr., and J. M. da Mata, "Survey on wireless sensor network devices," in *Proc. 9th IEEE International Conference on Emerging Technologies and Factory Automation*, Lisbon, Portugal, Sept. 2003, pp. 537–544.

[6] Q. Zhang, W. Zhu, and Y. Zhang, "End-to-end QoS for video delivery over wireless Internet," *Proc. IEEE*, vol. 93, no. 1, pp. 123–134, Jan. 2005.

[7] M. Bramberger, R. Pflugfelder, B. Rinner, H. Schwabach, and B. Strobl, "Intelligent traffic video sensor: Architecture and applications," in *Proc. Workshop on Telecommunications and Mobile Computing*, Mar. 2003.

[8] F. D. Real de Oliveira, P. Chalimbaud, F. Berry, J. Serot, and F. Marmoiton, "Embedded early vision systems: Implementation proposal and hardware architecture and hardware architecture," in *Proc. COGnitive systems with Interactive Sensors*, Mar. 2006.

[9] S. Bouaziz, M. Fan, A. Lambert, T. Maurin, and R. Reynaud, "PICAR: experimental platform for road tracking applications," in *Proc. IEEE Intelligent Vehicles Symposium*, June 2003, pp. 495–499.

[10] P. Chen, S. Oh, M. Manzo, B. Sinopoli, C. Sharp, K. Whitehouse, G. Tolle, J. Jeong, P. Dutta, J. Hui, S. Schaffert, S. Kim, J. Taneja, B. Zhu, T. Roosta, M. Howard, D. Culler, and S. Sastry, "Experiments in instrumenting wireless sensor networks for real-time surveillance," in *Proc. IEEE Int. Conference on Robotics and Automation*, May 2006.

[11] M.-H. Lu and T. Chen, "CMUseum: A location-aware wireless video streaming system," in *Proc. IEEE Int. Conference on Multimedia and Expo*, July 2006, pp. 2129–2132.

[12] C. H. Kuo, C. C. Chen, W. C. Wang, Y. C. Hung, E. C. Lin, K. M. Lee, and Y. M. Lin, "Remote control based hybrid-structure robot design for home security applications," in *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, Oct. 2006, pp. 4484–4489.

[13] H. Yin, C. Lin, B. Sebastien, and X. Chu, "A novel secure wireless video surveillance system based on Intel IXP425 network processor," in *Proc. 1st ACM Workshop on Wireless Multimedia Networking and Performance Modeling (WMuNeP)*, 2005, pp. 62–69.

[14] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan, "MeshEye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance," in *Proc. Int. Conference on Information Processing in Sensor Networks*, 2007, pp. 360–369.

[15] G. Kogut, M. Blackburn, and H. R. Everett, "Using video sensor networks to command and control unmanned ground vehicles," in *Proc. AUVSI Unmanned Systems in International Security*, Sept. 2003.

[16] J. Friedman, D. Lee, I. Tsigkogiannis, P. Aghera, A. Dixit, S. Wong, D. Levine, D. Chao, A. Kansal, W. Kaiser, and M. Srivastava, "RAGOBOT: A new platform for wireless mobile sensor networks," in *Proc. IEEE Int. Conference on Distributed Computing in Sensor Systems*, June–July 2005.

[17] A. M. Tabar, A. Keshavarz, and H. Aghajan, "Smart home care network using sensor fusion and distributed vision-based reasoning," in *Proc. 4th ACM International Workshop on Video Surveillance and Sensor Networks*, Oct. 2006, pp. 145–154.

[18] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J. A. Stankovic, "An assisted living oriented information system based on a residential wireless sensor network," in *Proc. 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference*, Apr. 2006, pp. 95–100.

[19] R. Pon, M. A. Batalin, J. Gordon, A. Kansal, D. Liu, M. Rahimi, L. Shirachi, Y. Yu, M. Hansen, W. J. Kaiser, M. Srivastava, G. Sukhatme, and D. Estrin, "Networked infomechanical systems: a mobile embedded networked sensor platform," in *Proc. Fourth International Symposium on Information Processing in Sensor Networks*, Apr. 2005, pp. 376–381.

[20] J. Campbell, P. B. Gibbons, S. Nath, P. Pillai, S. Seshan, and R. Sukthankar, "IrisNet: an internet-scale architecture for multimedia sensors," in *Proc. ACM Int. Conference on Multimedia*, 2005, pp. 81–88.

[21] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu, "SensEye: a multi-tier camera sensor network," in *Proc. ACM International Conference on Multimedia*, Nov. 2005, pp. 229–238.

[22] R. Blum and Z. Liu, *Multi-Sensor Image Fusion and Its Applications*. CRC Press, 2005.

[23] E. Nakamura, A. Loureiro, and A. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Computing Surveys*, vol. 39, no. 3, pp. 9/1–9/55, Aug. 2007.

[24] J. Karlsson, T. Wark, P. Valencia, M. Ung, and P. Corke, "Demonstration of image compression in a low-bandwidth wireless camera

network," in *Proc. Int. Conference on Information Processing in Sensor Networks (IPSN)*, 2007, pp. 557–558.

[25] S. Rein and M. Reisslein, "Low-memory wavelet transforms for wireless sensor networks: A tutorial," *IEEE Commun. Surveys & Tutorials, in print*, DOI 10.1109/SURV.2011.100110.00059.

[26] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica, "A unifying link abstraction for wireless sensor networks," in *Proc. 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2005.

[27] H. Liu, T. Roeder, K. Walsh, R. Barr, and E. G. Sirer, "Design and implementation of a single system image operating system for ad hoc networks," in *Proc. Int. Conference on Mobile Systems, Applications and Services*, 2005, pp. 149–162.

[28] W. Dong, C. Chen, X. Liu, and J. Bu, "Providing OS support for wireless sensor networks: Challenges and approaches," *IEEE Commun. Surveys & Tutorials, in print*, DOI: 10.1109/SURV.2010.032610.00045, 2010.

[29] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 5, pp. 645–658, May 2005.

[30] Z. He and D. Wu, "Resource allocation and performance analysis of wireless video sensors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 590–599, May 2006.

[31] S. Pudlewski and T. Melodia, "A distortion-minimizing rate controller for wireless multimedia sensor networks," *Computer Communications*, vol. 33, no. 12, pp. 1380–1390, July 2010.

[32] C. Buschmann, D. Pfisterer, S. Fischer, P. F. Sándor, and A. Kröller, "Spyglass: a wireless sensor network visualizer," *ACM SIGBED Review*, vol. 2, no. 1, pp. 1–6, Jan. 2005.

[33] S. Hengstler and H. Aghajan, "WiSNAP: A wireless image sensor network application platform," in *Proc. Int. IEEE/Create-Net Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, Mar. 2006.

[34] P. Sitbon, N. Bulusu, and W.-C. Feng, "Cascades: An extensible heterogeneous sensor networking framework," in *Proc. 4th International Conference on Embedded Networked Sensor Systems*, Oct. 2006, pp. 339–340.

[35] P. Sitbon, W.-C. Feng, N. Bulusu, and T. Dang, "SenseTK: a multimodal, multimedia sensor networking toolkit," in *Multimedia Computing and Networking*, R. Zimmermann and C. Griwodz, Eds., Jan. 2007.

[36] J. Hill, M. Horton, R. Kling, and L. Krishnamurthy, "The platforms enabling wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 41–46, June 2004.

[37] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, May 2002.

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

[39] A. Seema, "Blueprint for a low-cost piggyback wireless video sensor node/platform architecture, M.S. thesis, Arizona State University, Electrical Engineering," pp. 1–201, Apr. 2009.

[40] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, May 2009.

[41] I. Buchmann, *Batteries in a Portable World*, 2nd ed. Cadex Electronics Inc, 2001.

[42] A. Chandrakasan, W. Bowhill, and F. Fox, *Design of High-Performance Microprocessor Circuits*. Wiley, 2000.

[43] K. Parhi, *VlSI Digital Signal Processing Systems: Design And Implementation*. Wiley, 2007.

[44] A. Raghunathan, N. K. Jha, and S. Dey, *High-Level Power Analysis and Optimization*. Kluwer Academic Publishers, 1998.

[45] H. Singh, K. Agarwal, D. Sylvester, and K. Nowka, "Enhanced leakage reduction techniques using intermediate strength power gating," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 11, pp. 1215–1224, Nov. 2007.

[46] G. Yeap, *Practical low power digital VLSI design*. Kluwer Academic Publishers, 1998.

[47] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surveys & Tutorials, in print*, DOI: 10.1109/SURV.2011.060710.00094.

[48] N. Pantazis and D. Vergados, "A survey on power control issues in wireless sensor networks," *IEEE Commun. Surveys & Tutorials*, vol. 9, no. 4, pp. 86–107, Fourth Quarter 2007.

[49] F. Wang and J. Liu, "Networked wireless sensor data collection: Issues, challenges, and approaches," *IEEE Commun. Surveys & Tutorials, in print*, DOI: 10.1109/SURV.2011.060710.00066, 2010.

[50] M. Krause, M. V. Hartskamp, and E. Aarts, "A quality metric for use with frame-rate based bandwidth adaptation algorithms," in *Proc. SPIE Human Vision and Electronic Imaging*, Jan. 2008.

[51] J. Anderson and B. Anderson, "The myth of persistence of vision revisited," *Journal of Film and Video*, vol. 45, no. 1, pp. 3–12, 1993.

[52] P. Brun, G. Hauske, and T. Stockhammer, "Subjective assessment of H.264-AVC video for low-bitrate multimedia messaging services," in *Proc. of International Conference on Image Processing (ICIP)*, vol. 2, 2004, pp. 1145–1148.

[53] M.-C. Hwang, L. T. Ha, N.-H. Kim, C.-S. Park, and S.-J. Ko, "Person identification system for future digital TV with intelligence," *IEEE Trans. Consum. Electron.*, vol. 53, no. 1, pp. 218–226, 2007.

[54] A. J. Kun and Z. Vamossy, "Traffic monitoring with computer vision," in *Proc. International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2009, pp. 131–134.

[55] I. Downes, L. B. Rad, and H. Aghajan, "Development of a mote for wireless image sensor networks," in *Proc. COGnitive systems with Interactive Sensors (COGIS)*, Mar. 2006.

[56] A. B. Atitallah, P. Kadionik, F. Ghozzi, P. Nouel, N. Masmoudi, and P. Marchegay, "Hardware platform design for real-time video applications," in *Proc. 16th International Conference on Microelectronics*, Dec. 2004, pp. 722–725.

[57] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach, "Distributed embedded smart cameras for surveillance applications," *IEEE Computer*, vol. 39, no. 2, pp. 68–75, Feb. 2006.

[58] Z.-Y. Cao, Z.-Z. Ji, and M.-Z. Hu, "An image sensor node for wireless sensor networks," in *Proc. Int. Conference on Information Technology: Coding and Computing*, 2005, pp. 740–745.

[59] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: application driver for wireless communications technology," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 2, pp. 20–41, Apr. 2001.

[60] Y. M. Chi, R. Etienne-Cummings, G. Cauwenberghs, P. Carpenter, and K. Colling, "Video sensor node for low-power ad-hoc wireless networks," in *Proc. Annual Conf. on Information Sciences and Systems (CISS)*, 2007, pp. 244–247.

[61] C. Decker, A. Krohn, M. Beigl, and T. Zimmer, "The particle computer system," in *Proc. Fourth Int. Symposium on Information Processing in Sensor Networks*, Apr. 2005, pp. 443–448.

[62] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler, "Trio: enabling sustainable and scalable outdoor wireless sensor network deployments," in *Proc. Int. Conference on Information Processing in Sensor Networks (IPSN)*, Apr. 2006, pp. 407–415.

[63] K. Fife, A. E. Gamal, and H.-S. P. Wong, "A multi-aperture image sensor with 0.7 μm pixels in 0.11 μm CMOS technology," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2990–3005, 2008.

[64] K. Hachicha, S. Bouton, D. Faura, O. Romain, A. Pinna, and P. Garda, "Integration of MPEG4 codec on the M.E.R.I.T.E platform for wireless sensor networks," in *Proc. IEEE Int. Conference on Industrial Technology*, Dec. 2004, pp. 1215–1220.

[65] R. Harris, "Review: Home spy camera hidden in clock," *Physorg.com*, Feb. 2007.

[66] H.-C. Huang, J.-W. Ding, and Y.-M. Huang, "An implementation of battery-aware wireless sensor network using ZigBee for multimedia service," in *Proc. Int. Conference on Consumer Electronics*, Jan. 2006, pp. 369–370.

[67] E. Ladis, I. Papaefstathiou, R. Marchesani, K. Tuinenbreijer, P. Langendorfer, T. Zahariadis, H. C. Leligou, L. Redondo, T. Riesgo, P. Kannegiesser, M. Berekovic, and C. J. M. van Rijn, "Secure, mobile visual sensor networks architecture," in *Proc. IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops (SECON Workshops)*, 2009, pp. 1–3.

[68] J. Lifton, M. Broxton, and J. A. Paradiso, "Experiences and directions in pushpin computing," in *Proc. Int. Symposium on Information Processing in Sensor Networks (IPSN)*, 2005, pp. 57–62.

[69] M. Martina, A. Molino, F. Quaglio, and F. Vacca, "A power-scalable motion estimation co-processor for energy constrained applications," in *Proc. Int. Symposium on Signal Processing and Its Applications*, July 2003, pp. 603–604.

[70] D. P. Panda, C. A. Anderson, and P. G. Michalopoulos, "A new integrated video sensor technology for traffic management applications," in *Proc. Eighth Annual Meeting of ITS America*, May 1998.

[71] C. Park and P. H. Chou, "Eco: Ultra-wearable and expandable wireless

sensor platform," in *Proc. International Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, Apr. 2006, pp. 162–165.

[72] P. Sambhoos, A. Hasan, R. Han, T. Lookabaugh, and J. Mulligan., "Weeblevideo: Wide angle field-of-view video sensor networks," in *Proc. 4th ACM Conference on Embedded Networked Sensor Systems: Workshop on Distributed Smart Cameras*, Oct.–Nov. 2006.

[73] A. Sharma, K. Yoon, D. Vanhorn, M. Dube, V. McKenna, and M. S. Bauer, "ScoutNode: A multimodal sensor node for wide area sensor networks," in *Proc. Int. Conference on Computer Communications and Networks (ICCCN)*, 2009, pp. 1–6.

[74] T. Teixeira, A. G. Andreou, and E. Culurciello, "Event-based imaging with active illumination in sensor networks," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, 2005, pp. 644–647.

[75] B. A. Warneke, M. D. Scott, B. S. Leibowitz, L. Zhou, C. L. Bellew, J. A. Chediak, J. M. Kahn, B. E. Boser, and K. S. J. Pister, "An autonomous $16mm^3$ solar-powered node for distributed wireless sensor networks," in *Proc. IEEE International Conference on Sensors*, June 2002, pp. 1510–1515.

[76] W. Wolf, B. Ozer, and T. Lv, "Smart cameras as embedded systems," *IEEE Computer*, vol. 35, no. 9, pp. 48–53, Sept. 2002.

[77] L. Yihan, S. S. Panwar, M. Shiwen, S. Burugupalli, and H. L. Jong, "A mobile ad hoc bio-sensor network," in *Proc. IEEE International Conference on Communications (ICC)*, May 2005, pp. 1241–1245.

[78] "BDTI Focus Report: FPGAs for DSP, Second Edition," 2008, available at http://www.bdti.com/products/reports_fpga2006.html.

[79] L. Edwards, "Choosing a microcontroller and other design decisions," in *Embedded Hardware: Know It All*. Newnes, 2008.

[80] W.-C. Feng, E. Kaiser, W. C. Feng, and M. LeBaillif, "Panoptes: scalable low-power video sensor networking technologies," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 1, no. 2, pp. 151–167, May 2005.

[81] T. Teixeira, E. Culurciello, J. H. Park, D. Lymberopoulos, A. Barton-Sweeney, and A. Savvides, "Address-event imagers for sensor networks: evaluation and modeling," in *Proc. International Conference on Information Processing in Sensor Networks*, Apr. 2006, pp. 458–466.

[82] E. Culurciello and A. G. Andree, "ALOHA CMOS imager," in *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS)*, 2004, pp. IV–956–9.

[83] D. Lymberopoulos and A. Savvides, "XYZ: a motion-enabled, power aware sensor node platform for distributed sensor network applications," in *Proc. Int. Symposium on Information Processing in Sensor Networks (IPSN)*, 2005, pp. 449–454.

[84] T. Fan, L. Xu, X. Zhang, and H. Wang, "Research and design of a node of wireless multimedia sensor network," in *Proc. Int. Conference on Wireless Communications, Networking and Mobile Computing (WiCom)*, 2009, pp. 1–5.

[85] C. Park and P. H. Chou, "eCAM: ultra compact, high data-rate wireless sensor node with a miniature camera," in *Proc. 4th International Conference on Embedded Networked Sensor Systems*, Oct.–Nov. 2006, pp. 359–360.

[86] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava, "Cyclops: in situ image sensing and interpretation in wireless sensor networks," in *Proc. 3rd International Conference on Embedded Networked Sensor Systems*, 2005, pp. 192–204.

[87] R. Kleihorst, B. Schueler, A. Danilin, and M. Heijligers, "Smart camera mote with high performance vision system," in *Proc. ACM Conference on Embedded Networked Sensor Systems: Workshop on Distributed Smart Cameras*, Oct.–Nov. 2006.

[88] R. Kleihorst, A. Abbo, B. Schueler, and A. Danilin, "Camera mote with a high-performance parallel processor for real-time frame-based video processing," in *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2007.

[89] A. Rowe, A. G. Goode, D. Goel, and I. Nourbakhsh, "CMUcam3: An open programmable embedded vision sensor," Robotics Institute, Carnegie Mellon University, Tech. Rep., May 2007.

[90] A. Kandhalu, A. Rowe, and R. Rajkumar, "DSPcam: A camera sensor system for surveillance networks," in *Proc. ACM/IEEE Int. Conference on Distributed Smart Cameras (ICDSC)*, 2009, pp. 1–7.

[91] A. Kandhalu, A. Rowe, R. Rajkumar, C. Huang, and C.-C. Yeh, "Real-time video surveillance over IEEE 802.11 mesh networks," in *Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2009, pp. 205–214.

[92] C. B. Margi, V. Petkov, K. Obraczka, and R. Manduchi, "Characterizing energy consumption in a visual sensor network testbed," in *Proc. 2nd Int. Conference on Testbeds 'I&' Research Infrastructures for the*

[93] W. Cheng, "Embedded system design and power-rate-distortion optimization for video encoding under energy constraints," Master's thesis, Univ. of Missouri, Aug. 2007.

[94] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling, "IMOTE2: Serious computation at the edge," in *Proc. IEEE Int. Wireless Communications and Mobile Computing Conference (IWCMC)*, 2008, pp. 1118–1123.

[95] P. Chen, P. Ahammad, C. Boyer, S.-I. Huang, L. Lin, E. Lobaton, M. Meingast, S. Oh, S. Wang, P. Yan, A. Y. Yang, C. Yeo, L.-C. Chang, J. D. Tygar, and S. S. Sastry, "CITRIC: A low-bandwidth wireless camera network platform," in *Proc. ACM/IEEE Int. Conference on Distributed Smart Cameras (ICDSC)*, 2008, pp. 1–10.

[96] J. Schiller, A. Liers, and H. Ritter, "ScatterWeb: A wireless sensornet platform for research and teaching," *Computer Communications*, vol. 28, no. 13, pp. 1545–1551, Aug. 2005.

[97] T. Wark, P. Corke, J. Karlsson, P. Sikka, and P. Valencia, "Real-time image streaming over a low-bandwidth wireless camera network," in *Proc. International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP)*, 2007, pp. 113–118.

[98] E. P. Capo-Chichi and J. M. Friedt, "Design of embedded sensor platform for multimedia application," in *Proc. Int. Conference on Distributed Framework and Applications (DFmA)*, 2008, pp. 146–150.

[99] E. Culurciello, J. H. Park, and A. Savvides, "Address-event video streaming over wireless sensor networks," in *Proc. IEEE ISCAS*, Apr. 2007, pp. 849–852.

[100] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low-power wireless," *ACM Trans. Sen. Netw.*, vol. 6, no. 2, pp. 1–49, Feb. 2010.

[101] C. McKay and F. Masuda, "Empirical study of 802.11b wireless networks in the presence of bluetooth interference," *Institute for Telecommunication Sciences*, vol. 2003, no. 1, pp. 1–29, 2003.

[102] J. Dionne and M. Durrant, "uClinux—Embedded Linux Microcontroller Project," 2010, http://www.uclinux.org.

[103] B. E. Bayer, "Color imaging array, US Patent 3971065," July 1975.

[104] J. Adams, "Building low power into wireless sensor networks using ZigBee technology," in *Proc. Sensors Expo Chicago*, June 2005, pp. 1–33.

[105] C. Evans-Pughe, "Bzzzz zzz [ZigBee wireless standard]," *IEE Review*, vol. 49, no. 3, pp. 28 – 31, Mar. 2003.

[106] J.-T. Huang, J.-H. Shiao, and J.-M. Wu, "A miniaturized Hilbert inverted-F antenna for wireless sensor network applications," *IEEE Trans. Antennas and Propag.*, vol. 58, no. 9, pp. 3100–3103, Sept. 2010.

[107] S. Y. Shin, H. S. Park, and W. H. Kwon, "Mutual interference analysis of IEEE 802.15.4 and IEEE 802.11b," *Computer Networks*, vol. 51, no. 12, pp. 3338–3353, Aug. 2007.

[108] G. Thonet, P. Allard-Jacquin, and P. Colle, "Zigbee–wifi coexistence - white paper and test report," Schneider Electric, Tech. Rep., Apr. 2008.

[109] ZigBee Alliance, "Zigbee alliance – resources."

[110] P. Brown, "Zigbee and low-cost wireless applications," 2008, www.freescale.com/zigbee.

[111] ZigBee Alliance, "Zigbee and wireless radio frequency coexistence - zigbee white paper," ZigBee Alliance, Tech. Rep., June 2007.

[112] DALSA Corp., "Image sensor architectures for digital cinematography," white paper 03-70-00218-01.

[113] A. E. Gamal, D. X. D. Yang, and B. A. Fowler, "Pixel-level processing: why, what, and how?" in *SPIE Sensors, Cameras, and Applications for Digital Photography*, N. Sampat and T. Yeh, Eds., vol. 3650, no. 1, 1999, pp. 2–13.

[114] G. V. der Auwera, P. David, and M. Reisslein, "Traffic and quality characterization of single-layer video streams encoded with H.264/MPEG-4 advanced video coding standard and scalable video coding extension," *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 698–718, Sept. 2008.

[115] M. Wien, H. Schwarz, and T. Oelbaum, "Performance analysis of SVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1194–1203, Sept. 2007.

DEvelopment of NeTworks 'I&' COMmunities (TRIDENTCOM), Mar. 2006.
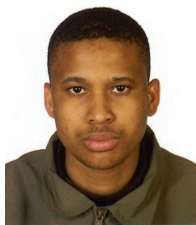
Adolph Seema has a wide range of experiences spanning more than eight years in the Semiconductor Industry, including VLSI chip verification at Apple Computer, Inc., Cupertino, CA; PCB driver development for satellite synchronized network time servers and displays at Spectracom Corp., Rochester, NY; and Zigbee and RF232 wireless PCB design at FSI Systems, Inc., Farmington, NY. Adolph is currently a Software Engineer for Advanced Semiconductor Materials, Phoenix, AZ, where he develops software that controls sensors, actuators, robots, and implements atomic layer deposition processes and other factory automation tools. He holds a B.S. in Computer Engineering from the Rochester Institute of Technology (RIT), Rochester, NY. and an M.S. in Electrical Engineering from Arizona State University (ASU), Tempe, AZ. He is currently pursuing his Ph.D. in Electrical Engineering at ASU's School of Electrical, Computer, and Energy Engineering. His research focuses on developing low-cost wireless video sensor platform architectures with emphasis on HW/SW integration.

Martin Reisslein is an Associate Professor in the School of Electrical, Computer, and Energy Engineering at Arizona State University (ASU), Tempe. He received the Dipl.-Ing. (FH) degree from the Fachhochschule Dieburg, Germany, in 1994, and the M.S.E. degree from the University of Pennsylvania, Philadelphia, in 1996. Both in electrical engineering. He received his Ph.D. in systems engineering from the University of Pennsylvania in 1998. During the academic year 1994–1995 he visited the University of Pennsylvania as a Fulbright scholar. From July 1998 through October 2000 he was a scientist with the German National Research Center for Information Technology (GMD FOKUS), Berlin and lecturer at the Technical University Berlin. He currently serves as Associate Editor for the *IEEE/ACM Transactions on Networking* and for *Computer Networks*. He maintains an extensive library of video traces for network performance evaluation, including frame size traces of MPEG-4 and H.264 encoded video, at `http://trace.eas.asu.edu`. His research interests are in the areas of video traffic characterization, wireless networking, optical networking, and engineering education.