

An Ultralow-Power Wireless Camera Node: Development and Performance Analysis

Leonardo Gasparini, *Student Member, IEEE*, Roberto Manduchi, Massimo Gottardi, *Member, IEEE*, and Dario Petri, *Fellow, IEEE*

Abstract—This paper presents the design principles underlying the video nodes of long-lifetime wireless networks. The hardware and firmware architectures of the system are described in detail, along with the system-power-consumption model. A prototype is introduced to validate the proposed approach. The system mounts a Flash-based field-programmable gate array and a high-dynamic-range complementary metal–oxide–semiconductor custom vision sensor. Accurate power measurements show that the overall consumption is 4.2 mW at 3.3 V in the worst case, thus achieving an improvement of two orders of magnitude with respect to video nodes for similar applications recently proposed in the literature. Powered with a 2200-mAh 3.3-V battery, the system will exhibit a typical lifetime of about three months.

Index Terms—Complementary metal–oxide–semiconductor (CMOS) vision sensor, field-programmable gate array (FPGA), image processing, people counter, surveillance, ultralow-power node, wireless camera network (WCN).

I. INTRODUCTION

WIRELESS networks of battery-operated video nodes may find widespread application in fields such as security, assisted living, road-traffic monitoring, and natural sciences. Wireless-camera-network (WCN) nodes offer a number of advantages with respect to standard wired camera networks. In particular, they are much easier to deploy and remove, thus reducing installation time and cost. This is important for applications that require high density of placement, i.e., either to obtain many different views of the scene or for increased robustness. Impromptu surveillance installations (e.g., to monitor a building during a special event or animals in their natural habitat) also require fast installation, repositioning, and removal of a possibly large number of cameras and thus would benefit from this technology. In-home elder care and home security may be facilitated by camera monitoring, and the use

of wireless nodes would allow for discreet and unobtrusive installations at no cost since the intervention of an operator is not needed [1], [2]. WCNs represent also a convenient solution in those situations in which an external power source is unavailable or very expensive to provide, such as in borderlands and dangerous zones. For example, obtaining a low-voltage power supply from high-voltage power lines may increase the price of the wired surveillance system by a factor of 10. Finally, we note that WCNs are largely immune to failure of the power distribution system and thus may support the backup of wired systems in the case of natural or man-made disasters.

Unfortunately, the use of batteries as the source of power introduces severe restrictions in the capabilities of nodes. Batteries have a very limited energy budget, resulting in a tradeoff between the lifetime of the system (which is typically dictated by the application) and the desired computational power to process the video data. Thus, video nodes must be designed by carefully selecting the hardware components and producing high-efficiency embedded software. In this context, the imager plays a major role. A video sensor is much more complex and power demanding than simpler sensors such as temperature, pressure, or humidity sensors. Not only the sensing process itself is more expensive (as the imager is composed of an array of thousands or millions of sensors) but also data buffering, processing, and transmission require much more power.

It is possible to achieve better energy efficiency by decreasing the sensor resolution, as long as the resolution is enough for the task at hand. This, however, is only a partial solution. In general, the transmission of a full video stream via IEEE 802.11 or Bluetooth requires too much power to be viable. Hence, a certain amount of on-board processing is necessary. However, even data processing can be expensive in terms of power, and it introduces latency that may reduce the effective frame rate.

Several WCN nodes have been proposed in the literature in the past decade [3]–[7], but they typically have either high power consumption or limited performance. These systems mostly employ standard imaging sensors, which are typically designed “for humans,” providing high-resolution images with several bits per pixel. Standard imagers are not necessarily the optimal solution for WCN nodes. In fact, in most cases, only particular portions of an image (such as moving areas or objects in the foreground) or features (e.g., edges, corners, histograms of brightness, or color) are of interest. This suggests the use of imagers that only acquire the data that one is interested in, thus saving energy in the acquisition and processing phases.

In recent years, a new approach to the design of video sensors has been proposed by the researchers [8], [9], i.e., one

Manuscript received June 24, 2010; revised January 24, 2011; accepted March 28, 2011. This work was supported in part by the Italian Ministry of University and Research through the Programmi di ricerca di Rilevante Interesse Nazionale (PRIN) 2008 Project entitled “Methodologies and measurement techniques for spatio-temporal localization in wireless sensor networks.” The Associate Editor coordinating the review process for this paper was Dr. Jesús Ureña.

L. Gasparini and D. Petri are with the Department of Information Engineering and Computer Science, University of Trent, 38123 Trent, Italy (e-mail: gasparini@disi.unitn.it; petri@disi.unitn.it).

R. Manduchi is with the Department of Computer Engineering, University of California Santa Cruz, Santa Cruz, CA 95064 USA (e-mail: manduchi@soe.ucsc.edu).

M. Gottardi is with Smart Optical Sensors and Interfaces, Bruno Kessler Foundation, 38123 Trent, Italy (e-mail: gottardi@fbk.eu).

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

Digital Object Identifier 10.1109/TIM.2011.2147630

that attempts to limit power requirements by implementing the processing (or some parts of it) directly on-chip.

This paper proposes a design methodology for very low-power WCN nodes. Our strategy exploits the on-chip sensor processing capabilities and the low-power modes of the other hardware elements on the board. A prototype WCN node is also described, and its power consumption is analyzed in detail. The prototype node is based on an ultralow-power binary contrast-based imaging sensor that also features on-chip frame differencing. The node core is represented by a Flash-based field-programmable gate array (FPGA), which manages the system and processes the data generated by the sensor. We designed the node to operate as a people counter, which we used as a case study to verify the effectiveness of our approach in the design of a WCN node.

This paper is organized as follows. In Section II, we provide some background about other existing WCN nodes, as reported in the literature. In Section III, we describe the proposed general design principles for a low-power WCN node. The detailed power model is provided in Section IV. In Section V, we describe our developed hardware architecture and its application as a people counter. The power-requirement analysis and the lifetime estimate of the node in this application are presented in Section VI. Section VII has the conclusions.

II. RELATED WORK

The first significant example of a WCN node is Panoptes [3]. Its most recent version is based on Crossbow's Stargate platform, which consists of a 400-MHz Intel XScale processor, with a 64-MB synchronous dynamic random access memory (SDRAM) device and a 64-MB Flash memory device. The imager is a universal-serial-bus (USB) camera, offering different resolutions, i.e., from 160×120 up to 640×480 pixels. The board runs in the Linux operating system. The main application considered was the compression and the transmission of frames only when a significant scene change takes place. The power consumption of the entire node is in the order of a few watts. In order to save power, the system switches components off or sets them to low-power modes whenever possible. This approach partially solves the problem, as ON/OFF or wake/sleep transitions may require a substantial amount of energy.

Another important example of a WCN node is Cyclops [4]. The node features a low-power microcontroller unit (MCU) to control the entire system and run the applications. It is supported by 64-KB static RAM (SRAM) and 640-KB Flash memory devices. The MCU is clocked at approximately 8 MHz and interfaces with a 352×288 complementary metal-oxide-semiconductor (CMOS) imager through a complex programmable logic device, which is also responsible for simple but fast image processing while receiving data. Sample applications include a simple object-detection algorithm, which is based on background subtraction, and hand-posture recognition for human-computer interactions. The processing capabilities of the system are limited, but the joint use of a low-power device and a fast device achieves a significant (one order of magnitude) improvement in terms of energy requirements with respect to Panoptes.

Kleihorst *et al.* proposed a different solution in [5]. In order to minimize the time spent for pixel-level processing on the images provided by up to two video-graphics-array (VGA) color image sensors, the node exploits a massively parallel single-instruction multiple-data (SIMD) processor. The rationale behind this approach is that, often, when processing an image, the same calculations (e.g., convolution) are repeated on relatively small blocks of neighboring pixels by sliding a "kernel" throughout the image. The SIMD architecture employs an array of processing cores managing all the fetched data in parallel with the same instruction, thus providing a very high throughput even with high-resolution images. This comes at the cost of high power consumption, which is on the order of several hundreds of milliwatts.

Sensor diversity is the main characteristics of MeshEye [6]. This node hosts up to eight low-resolution low-color-depth imagers and one VGA camera. A high-performance ARM7-based MCU, featuring 64-KB SRAM and 64-KB Flash memory devices, represents the node core and is responsible for both control and data processing. A number of functionalities have been implemented on this system, including motion detection, 2-D blob localization and tracking, and the acquisition of high-resolution images of the detected object. The authors claim that, when the frequency of events is low, the lifetime of the node, which is powered with two AA batteries (2850 mAh), ranges between 10 and 60 days. However, the frame rate of the system is low [i.e., less than 2 frames per second (fps)].

All of the aforementioned systems acquire and process images at full resolution and full pixel depth. This approach requires substantial memory to store the images and powerful processing units.

A different approach was taken in [7]. The work proposes a nonstandard imager in which the concept of frames is replaced by an address-event representation. In practice, every pixel is uniquely identified by an address and senses a precise property of the scene, such as temporal and spatial differences. Every time the measured "amount" of such a property exceeds a predefined threshold, the pixel address is "fired." The higher the frequency of firings associated to a pixel is, the greater the intensity of the phenomenon for that pixel. This architecture moves a significant amount of processing inside the sensor, therefore reducing the power requirements. The authors were able to develop applications for assisted living, including the recognition of the behaviors of people in a house. Unfortunately, the sensor node is a standard high-performance iMote2, featuring an Intel XScale processor operating at more than 100 MHz, which is supported by 32-MB SDRAM and 32-MB Flash memory devices. Such high processing capabilities have a critical impact on the power requirements, which are in the order of hundreds of milliwatts.

III. DESIGNING THE ULTRALOW-POWER WCN NODE

A. Hardware

A WCN node contains four main logic components:

- 1) A control unit (CU), which manages the whole node;
- 2) An imager;

- 3) A memory device, for image buffering;
- 4) A processing unit (PU), which processes the acquired images;
- 5) A transceiver (TRX), for wireless communication.

1) *CU*: The job of the CU consists of several simple tasks such as providing timing signals to the sensor or activating/disabling the PU. Since such tasks do not typically require high timing resolution nor high processing capabilities, the CU can be implemented on an ultralow-power device, which is clocked at a very low frequency [10].

2) *Imager*: Although standard imagers are mere light sensors, it may be convenient to use a device that senses specific visual properties of a scene, such as local contrast, image texture, or motion. This approach can reduce the power consumption of the overall node. First of all, the chip generates less data, thus limiting signal activity and required memory size. Performing operations on-chip rather than on an external component is also power efficient. Processing is inherently parallel, with the same local operator replicated at each pixel. By performing the initial (and often most computationally intense) operations at the sensor level, subsequent (and more power hungry) processors only need to deal with selected frames and image areas, with a positive impact in terms of latency, device occupancy, and system complexity.

For instance, suppose that the application requires to enhance the edges of the acquired image. Assuming a resolution of 128×64 with 8 bits per pixel (bpp), we have to transfer and buffer 64 Kb of data. Then, edges can be extracted sliding a 3×3 mask throughout the image. This operation requires nine multiplication, eight addition, and one division operations for each pixel. Thresholding may be used also to reduce the amount of data down to 8 Kb. If we assume one single clock cycle per operation, the execution of such operations on a processor running at 10 MHz introduces a latency period of 15 ms. Clearly, such memory and processing requirements are not suitable for an ultralow-power system. On the other side, a custom imager can achieve the same goal by performing the processing task in the analog domain or in the in-pixel digital circuitry, during the acquisition. The amount of data to be transferred and buffered is reduced by a factor of 8, no latency is introduced, and the requirements in terms of processing capabilities can be relaxed. This comes at the expense of a more complex sensor-design process.

If most of the initial processing is performed at the sensor level, the sensor itself can communicate whether a frame contains relevant image data to be further processed. For example, the sensor may produce the number of “active” (interesting) pixels in a frame; then, the CU may decide whether the image data should be further analyzed by the PU or not. In the first case, we will say that the sensor is in active mode, which implies data transfer to the PU. In the second case, the sensor is in idle mode, and no data needs to be transferred. Note that, even when in idle mode, the sensor still acquires and processes the image, although data is not transferred to the PU.

3) *Memory*: The main characteristics of the memory component are speed and power consumption. Other important parameters include the method of access, latency, and interface modality.

4) *PU*: An ideal PU would support high-speed clocks, embedded memory devices, hardware multipliers, floating-point arithmetics, and multitasking. Support for standard communication protocols (e.g., for data transfer from/to an external memory device) is also desirable. In order to minimize the overall average power consumption, it is critical that the PU supports low-power modes. The energy consumed during mode transitions is normally nonnegligible and needs to be considered in the overall power budget.

5) *TRX*: Two main options for low-power TRX standards are currently available: Bluetooth and ZigBee. WiFi technology is usually very power demanding for these low-rate applications.

Such logic components are not necessarily identified with a hardware embodiment, but multiple elements may be contained on the same component, thus simplifying the system architecture. In order to maintain flexibility, such a component needs to support multiple clock domains or be able to change the clock frequency dynamically and to turn off different parts of the internal circuitry separately.

B. Firmware

By firmware, we refer to the implementation of both the CU and the PU. The main task of the CU is to ensure correct functionality of the node while maximizing its lifetime. We can consider two main operating conditions of the node. In the first one, (idle) the sensor reports no “activity” of interest in the scene, meaning that the frame needs no further processing. In the second case (active), the frame needs to be analyzed by the PU or transmitted by the TRX.

While the node is in idle mode, the CU only needs to manage the operations of the imager (which is in idle mode) and evaluate the reported number of “active” pixels at each frame. Moreover, the CU disables the memory device and the PU, whereas the communication activity is limited to receiving information from other nodes. If the number of active pixels is above the fixed threshold $n_{\text{pix}}^{\text{th}}$, the imager and the node are set to active mode. When in this mode, the CU wakes up the memory device and asks the sensor to readout the whole data. Once the readout has been completed, the CU activates the PU, which reads the data from the memory device and processes it. A high-frequency (HF) clock is usually required for this latter task, introducing an important source of power consumption.

The PU typically performs a two-stage process: image feature extraction (FE), which is followed by higher level decision making (DM). For example, the DM stage may consist in a classifier based on the features computed in the FE stage. The image features can be local, or a single feature may summarize the whole image content. In the latter case, we will say that the feature represents the state of the frame.

Here is an example of a power-efficient structure for the PU. At each time instant, i.e., at each frame, the FE algorithm generates state $s \in S$ from the images, where $S = \{s^1, s^2, \dots, s^N\}$ and N is the cardinality of the state space. The DM algorithm keeps track of how the state evolves with time to generate its output y , which is expressed as $y = f(s_0, s_1, \dots, s_t)$, where s_t represents a sequence of frames all with the same state.

Notice that the output of the DM stage may change only when a state transition occurs. Thus, at each frame, the PU needs to be activated for the amount of time necessary to accomplish the only FE stage if the state of the system does not change or both the FE and DM stages when the state changes.

After processing, several transmission policies can be considered. For example, one may transmit either full images or simply the low-rate data produced by the image-analysis algorithm.

Switching between idle and active modes needs careful planning in order to reduce the risk of frequent switches, resulting in possibly missed detections of events and high energy wasted during transitions. A typical policy is to switch the active node back to idle after the waiting period T_W in which the images are characterized by low activity (i.e., the number of active pixels in each frame is below the threshold).

IV. POWER-CONSUMPTION MODEL

According to the structure described in Section III-B, there are three main configurations in which the system can run:

- 1) idle mode;
- 2) active mode, when no system state transition occurs;
- 3) active mode, when the system state changes, thus requiring the execution of the DM algorithm.

We will refer to the average power P consumed and the average time T spent by the system in these configurations with P_I , $P_{A,FE}$, and $P_{A,DM}$ and T_I , $T_{A,FE}$, $T_{A,DM}$, respectively.

Their values depend on T_W (defined in the previous section), on the average time between the two following events T_E and on the average duration of events T_D . In addition, the probability of state transition Pr_{trans} has to be taken into account when in active mode.

Typically, $T_E > T_D + T_W$. When this condition occurs, in the interval between the two following events of duration T_E , the node works in active mode for time T_D . In this amount of time, the PU executes the DM algorithm with probability Pr_{trans} . After the event has terminated, the system remains in active mode and keeps monitoring the area for a T_W -long time window, in which the system state does not change since the scene is static. Then, the CU activates the system idle mode, which holds until a new event rises, i.e., after an interval of length T_I . Conversely, if $T_D < T_E \leq T_D + T_W$, the idle mode of the sensor is never enabled, whereas the amount of time in which the scene is static is less than T_W . Finally, if $T_E \leq T_D$, a state transition can happen at any time. Thus, we have

$$T_I = \max\{0, T_E - T_D - T_W\} \quad (1)$$

$$T_{A,FE} = \min\{T_E, T_D\} \cdot (1 - \text{Pr}_{\text{trans}}) + \min\{\max\{0, T_E - T_D\}, T_W\} \quad (2)$$

$$T_{A,DM} = \min\{T_E, T_D\} \cdot \text{Pr}_{\text{trans}} \quad (3)$$

Accordingly, the overall average power consumption turns out to be

$$P = \frac{T_I \cdot P_I + T_{A,FE} \cdot P_{A,FE} + T_{A,DM} \cdot P_{A,DM}}{T_E} \quad (4)$$

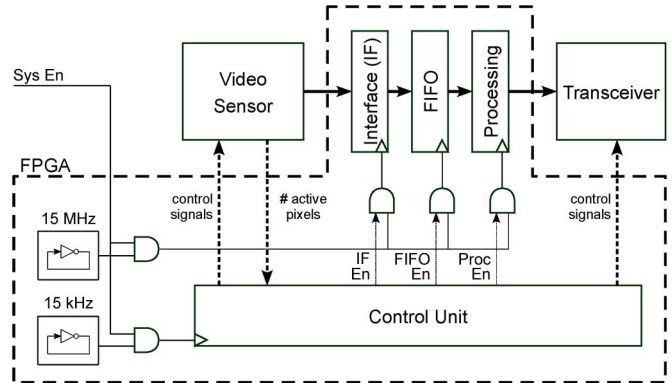


Fig. 1. Block scheme of the proposed video node.

The impact of T_W on the lifetime depends on ratio T_W/T_E , thus having its maximum for small values of T_E and decreasing accordingly. The minimum value for T_W is given by the minimum number of frames required to detect motion in most cases. This is usually limited to a very few frames. A safer approach consists on choosing T_W equal to T_D .

Using (4) and assuming to power the node with a battery of capacity C and voltage V , we can easily estimate the node's lifetime $T_{Lt} = C \cdot V/P$ as a function of T_E . In particular, (4) shows that the node lifetime increases as T_E increases, as expected.

V. IMPLEMENTED NODE ARCHITECTURE

A. Hardware

As shown in Fig. 1, the proposed video node is composed of three main hardware elements: the imager, the FPGA, and the TRX. A detailed analysis of these elements is presented in the following.

1) *Sensing*: The imager employed in our system is a prototype developed by researchers at the Bruno Kessler Foundation, Trent, Italy [9]. This is a 128×64 -pixel binary contrast-based sensor, which provides address-based output data asynchronously.

Imaging is achieved by a two-stage process: an analog phase, aiming at the acquisition of the current frame, and a digital phase, in which the output frame is generated. During the acquisition process, each pixel receives a binary value obtained by comparing the incoming irradiance at three locations: the pixel itself, the pixel above, and the pixel at its right. We refer to this L-shaped structure as the kernel. The pixel is said to be active if the difference of incoming light between the most and the least irradiated pixels in the kernel exceeds a predefined amount. Thus, at the end of the acquisition process, the active pixels represent the parts of the image characterized by high contrast. In the second part of the process, the sensor computes the pixel-by-pixel (bit-by-bit) difference of the current image with a previously acquired (binary) frame, which is stored in an internal memory device. For instance, by subtracting each frame with the previous one, we can detect the high contrast points in the scene undergoing motion. Fig. 2 shows some examples of the images that can be produced by this sensor. The images exhibit some isolated active pixels corresponding

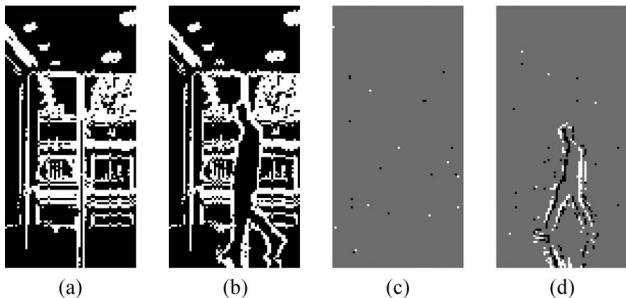


Fig. 2. Images taken by our camera from the same location: (a) and (c) represent a static background, whereas (b) and (d) represent a person walking by. The result of the acquisition process is given by (a) and (b). The sensor generates (c) and (d) by subtracting the last acquired frame to the current one. Note that, in this case, each pixel can take three values since it is the difference between two binary values.

TABLE I
COMPRESSION RATIOS ACHIEVED WITH THE USED SENSOR AND AN RLE WITH RESPECT TO A BITMAP CODING SCHEME WITH 2-bpp COLOR DEPTH. THE IMAGES IN Fig. 2 ARE CONSIDERED

Fig.	2(a)	2(b)	2(c)	2(d)
Active pixels	29.1%	28.9%	0.3%	4.9%
Sensor read-out	130.8%	129.9%	1.3%	22.1%
RLE	68.7%	68.3%	5.7%	21.4%

to points in the scene characterized by a measured contrast close to the threshold. The noise present in the analog section of the system causes the difference of irradiance in the kernel of these pixels to oscillate above and below the threshold, thus generating a blinking effect. Nevertheless, the method allows to effectively detect the foreground elements, as shown in Fig. 2(d).

The sensor features the two output modes introduced in Section III-A and behaves as follows. When the active mode is selected, the sensor waits for an external command to start the readout process, in which the column address and the sign of the nonzero pixels of the difference image are provided at the output pins (while the row address is derived from an end-of-row signal). This process is asynchronous, meaning that the difference image is raster scanned, and an output enabling signal running at 80 MHz is raised every time the data at the output pins of the chip represent the address of a nonzero pixel. This process is executed in less than 200 μ s. Notice that this kind of output data allows to achieve image compression when the active pixels are not too many. Typically, this is true when the sensor is set to detect motion, as shown in Table I. Conversely, when the scene is full of details, this method for compressing data is counterproductive.

In idle mode, the sensor scans the image, and at the end of the process, it provides at the output pins only the number of nonzero pixels present in the difference image.

The frame rate can be quite high since it is only limited by the duration of the acquisition process. Under normal light conditions, the integration time required by the sensor can be as low as 5 ms, enabling a frame rate as high as 200 fps.

The overall power consumption of the sensor is extremely small. At a frame rate of 50 fps, with 25% active pixels, the sensor draws approximately 100 μ W when in active mode.

Notice that, since active pixels represent high contrast areas (typically edges), it is unlikely that more than 25% of the pixels are active. If the system is set to idle mode, the power consumption reduces to 30 μ W. This value is over two orders of magnitude less than virtually any other image sensor available on the market.

2) *Control, Storage, and Processing*: Our video-node controls the imager and the TRX using an FPGA-based board. FPGA processors have well-known advantages in terms of speed and processing power with respect to general-purpose microcontrollers such as those used in standard low-power wireless network nodes. Furthermore, low-power FPGAs dissipate much less power than high-performance embedded processors such as the Intel XScale family [11], which is employed in several camera-based nodes [7]. FPGAs also allow for parallel task execution, which can be very advantageous in image processing, where the same mask can be replicated multiple times throughout the image [12], and in system control, where individual components may need to be managed independently. Another advantage of FPGA processors is that they allow for the implementation of several custom components (such as memory devices and central-processing-unit cores) within the same device. They also can support multiple clock domains, thus enabling separate clocks for different tasks. In particular, Flash-based FPGAs have two main advantages with respect to standard SRAM technology: ultralow static power consumption and nonvolatility. Unfortunately, since Flash-based FPGA technology is a few generations behind SRAM technology, it still offers a lower density of logic gates and a lower maximum operating frequency, which are due to longer signal-propagation delays. For our video node, we selected the Flash-based Actel IGLOO M1-AGL600, which is characterized by 600-K system gates and a static power consumption on the order of tens of microwatts [13].

Within the FPGA, we implemented a ring oscillator, a first-input–first-output (FIFO) memory device with its interface, a PU, and a controller. Two clock domains are implemented, as suggested in Section III-A: one running at a low frequency (\sim 15 kHz) and one running at a higher frequency (\sim 15 MHz), which are both derived from the output of the ring oscillator, which is divided through a frequency-divider chain. The CU controls this latter clock through a set of AND gates. Since the components implemented in the FPGA core need to be enabled in different time intervals, a dedicated gate is provided for each controlled component. The CU also generates the timing signals for the imager, asks it to perform detection of motion, and selects its output mode, i.e., either active or idle.

When the active mode of the imager is enabled, the data produced by the imager flow at a maximum rate of 80 MHz, which cannot be supported by big soft memory devices implemented in the FPGA, whereas adding an external FIFO memory device would increase the complexity and the power consumption of the node. Therefore, the data go through a serial-to-parallel interface to slow down the access to the internal soft memory device [14]. The soft FIFO memory device is an 8-KB memory device, in which the whole acquired image is stored. The data are currently packed in bytes, although the system can be configured for different word sizes. Once the readout process

is over, the unit processes the image to generate the desired output, which is then sent to the TRX for transmission.

In our prototype, the transitions between active and idle camera modes are triggered using threshold $n_{\text{pix}}^{\text{th}}$ on the number of active pixels in a frame and waiting time T_W , as aforementioned. The value for $n_{\text{pix}}^{\text{th}}$ has been chosen to be a little greater than the number of active pixels in a static scene due to analog noise. Moreover, we chose the value of T_W to be equal to the expected event duration T_D , i.e., 2 s, which is a reasonable choice since it prevents the system to be switched to idle mode while an event is still occurring.

3) *Communication*: In the current implementation, data are transmitted to a host computer through an RS-232 interface, but we are planning to install a wireless module such as ChipCon CC2420 IEEE 802.15.4-compliant 250-KB/s radio. This device draws a current of less than 20 mA when transmitting or receiving. Power consumption reduces to less than 0.5 mA when in idle mode and around 20 μA in power-down mode [15].

Bandwidth and power issues preclude video streaming and limit image transmission to very few situations. According to the data provided by the manufacturer and the power model presented in [16], the transmission of an entire frame (16 Kb) requires about 65 ms and consumes 18 mW. Therefore, the transmission of the whole set of images acquired by one single node at 15 fps would fill the available bandwidth, with a power consumption of about 17 mW. We can achieve better performance by compressing the images, e.g., using a run-length encoder (RLE), which well suits to the employed sensor.

The encoding process, which is implemented in the FPGA, is performed at the same time as the readout process (thus requiring less than 200 μs) by comparing the addresses of nonzero pixels present at the output pin of the sensor. The impact of the RLE in terms of device occupancy is almost negligible, whereas the reduction of transmitted bytes may be consistent, as shown in Table I. When the scene is full of details and the sensor generates a high number of active pixels, it reduces the amount of data by one third, with respect to the standard 2-bpp bitmap coding, and halves the size achieved with the address-based representation used by the sensor. If the image is sparse, the RLE cuts the data size to less than 10% of the bitmap image, but in this case, the sensor-compression method may achieve better rates.

B. Firmware: A People Counter

In order to demonstrate the capabilities of the proposed system in a realistic scenario, we configured it to function as a “people counter” [17]. The camera is placed on the ceiling of a passageway, facing downwards, and registers the passage of people in both directions. The algorithm exploits the ability of the sensor to subtract the previous frame to the current one, so that images provide information about high contrast points undergoing motion. Currently, the algorithm is designed so as to detect the passage of a single person; future work will address the possibility of detecting multiple persons at the same time.

The detection of persons transiting in a direction or the other is performed by exploiting the state-based mechanism de-

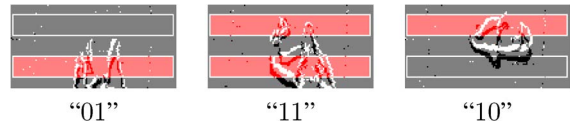


Fig. 3. Image sequence acquired by the sensor with two VILs drawn and the associated system state. The sequence represents a person walking upward. Each VIL activates when the number of nonzero pixels within it exceeds a given threshold. The system state is represented by a 2-bit variable, describing the status of the VILs.

TABLE II
CLASSIFICATION PERFORMANCE: ERRORS CORRESPOND TO PERSONS CLASSIFIED AS MOVING IN THE WRONG DIRECTION; MISSED DETECTIONS REPRESENT NOT CLASSIFIED PERSONS; FALSE ALARMS REPRESENT DETECTIONS OF PERSONS WHEN NOBODY IS CROSSING THE AREA

test events	156	163
errors	0.0%	0.1%
missed detections	3.2%	3.1%
false alarms	0.0%	13.7%

scribed in Section III-B. Each state gives an extremely concise representation of the presence and the location of a person in the field of view of the camera. We use the idea of virtual inductive loops (VILs) [18], which are nonoverlapping windows defined in the image, as shown in Fig. 3. If enough active pixels are found within a VIL, the VIL is said to be “active;” otherwise, it is “off.” For each image frame, the state of the system is represented by the corresponding values of the VILs.

The state evolution is modeled as a second-order Markov process. The transition probabilities for each “event” (such as a person transiting in a certain direction) are learnt during a training phase based on manually labeled sequences. At run time, sequences of states are identified and then grouped into “intervals” corresponding to different events according to a maximum-likelihood criterion. This is accomplished by a modified version of Dijkstra’s algorithm, with a complexity value of $\mathcal{O}(n_{\text{st}})$, where n_{st} represents the number of state transitions that occurred since the last time the sensor has been activated [19]. We analyzed the complexity of the algorithm and its implications on the overall power consumption in [14].

We measured the classification performance of our algorithm by recording two long videos and alternating them in the role of training and test sets. Results are shown in Table II. In one case, almost 97% of the detections were correct, with a negligible number of errors, i.e., persons classified as moving in the opposite direction with respect to the real one, a limited number of missed detections, and almost no false alarms, i.e., detections of events which did not take place. The results achieved in the other experiment are similar, except for the amount of false alarms, which is very high. This is due to the presence of long shadows on the ground, which the sensor is unable to filter out. Since the shadows are present in the test set but not in the training set, the system sees a person and his/her shadow as two persons walking in a close sequence.

VI. POWER MEASUREMENTS AND LIFETIME ESTIMATION

In the following, the power consumed by the system when executing the people-counting algorithm described in

TABLE III
SUPPLY VOLTAGE AND CURRENT DRAIN FOR OUR PROTOTYPE NODE.
(a) IDLE MODE. (b) ACTIVE MODE: ONLY FEATURE EXTRACTION.
(c) ACTIVE MODE: FEATURE EXTRACTION AND DESIGN MAKING

	$V \pm \Delta V$ [V]	$I \pm \Delta I$ [mA]
Video-sensor	3.28 ± 0.02	0.26 ± 0.01
FPGA core	1.22 ± 0.01	0.44 ± 0.02
FPGA I/O banks	3.28 ± 0.02	0.34 ± 0.02
Overall power P_I	2.50 ± 0.14 mW	

(a)

	$V \pm \Delta V$ [V]	$I \pm \Delta I$ [mA]
Video-sensor	3.28 ± 0.02	0.45 ± 0.02
FPGA core	1.22 ± 0.01	1.09 ± 0.06
FPGA I/O banks	3.28 ± 0.02	0.39 ± 0.02
Overall power $P_{A,FE}$	4.08 ± 0.23 mW	

(b)

	$V \pm \Delta V$ [V]	$I \pm \Delta I$ [mA]
Video-sensor	3.28 ± 0.02	0.45 ± 0.02
FPGA core	1.22 ± 0.01	1.15 ± 0.06
FPGA I/O banks	3.28 ± 0.02	0.41 ± 0.02
Overall power $P_{A,DM}$	4.22 ± 0.23 mW	

(c)

Section V-B is analyzed. We implemented the prototype on a commercial development board mounting several components in addition to the Actel M1-AGL600 FPGA: a 1-MB SRAM device, a 16-MB Flash memory device, a crystal oscillator, a programmer, a USB-to-RS-232 converter chip, expansion connectors, light-emitting diodes, and switches. We connected the camera to the board through one of the available connectors. The node communicates to a host computer through a USB connection that implements the RS-232 interface. The board allows for measurement of the current flowing through the FPGA core, its input/output banks, and the camera separately. We omitted the contribution due to the wireless module partly because the energy required for transmission and reception varies significantly according to the node positioning and the amount of data to be transmitted.

Measurements were carried out by forcing the system to run in one of the configurations described in Section IV.

When the node switches from the idle mode to the active mode, the system state needs to be continuously monitored. In our implementation, the execution of this task is performed during the readout process; thus, it would not require a soft memory device. Nevertheless, in order to be able to monitor the behavior of the node, we programmed the FPGA to transmit the whole acquired image to a host personal computer at every frame; thus, the soft memory device has been finally implemented. We set the transmission rate through the RS-232 interface to 256 Kb/s. At this speed, the system could not support acquisition rates higher than 10 fps; therefore, the frame rate had to be reduced. We do not expect a significant power-consumption increment at 30 fps, i.e., the frame rate that the algorithm has been designed for. This is because the

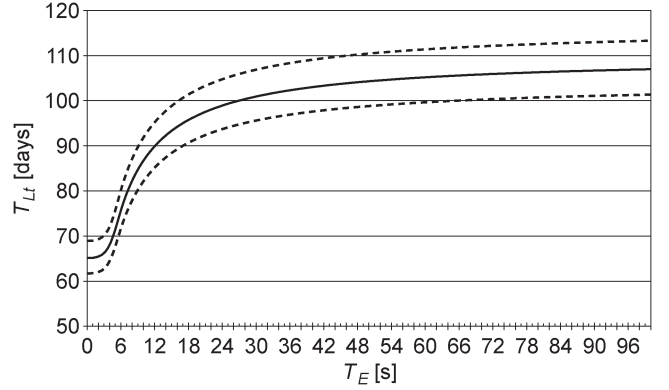


Fig. 4. Node lifetime versus average event period.

CU in the FPGA enables the HF clock for all the time needed for image transmission over the RS-232 interface, which takes much longer than the execution of the algorithm. In this way, large portions of the FPGA are active for most of the time, thus reducing the benefits of employing a Flash-based FPGA.

We employed two Agilent 34411A digital multimeters to perform power measurements. We set one of them to operate as an ammeter with a range of 10 mA and an integration time equal to 100 power line cycles, i.e., 2 s. With these settings, the instrument measures the voltage drop on a shunt resistance of 2 Ω . We measured the voltage drop on the device with the other multimeter, which is operating on a range of 10 V with an integration time of 100 power line cycles and characterized by an input resistance greater than 10 G Ω . With this configuration, the normal-mode noise rejection of the instruments is maximized, and their loading effects are negligible. Results are shown in Table III. Note that the values related to the video sensor also comprise the consumption of other elements present on the same board, such as trimmers and other resistors required for debugging.

We estimated the node lifetime T_{Lt} assuming to power the node with two batteries with capacity $C = 2200$ mAh at voltage $V = 3.3$ V. Results are shown in Fig. 4, where the dotted lines represent the uncertainty interval in which the lifetime may range. This follows from the uncertainty in the measured values of the current and the voltage drop, according to the law of uncertainty propagation [20].

The graph shows that the efficient use of the low-power capabilities of the components present in the system results in a lifetime of a few months. This is critical for video-surveillance applications, where the system should work for a long time without the need to replace its power source. It should be noted, however, that the lifetime depends on the rate of events to be monitored and is therefore application specific.

VII. CONCLUSION

We have presented a methodology for designing the node of a WCN featuring long lifetime. The designer needs to clearly identify each logic element of the node along with its requirements in terms of processing capabilities and power consumption. These elements are typically the vision sensor, the CU, the PU, and the TRX. We can achieve significant energy

savings if the sensing unit is able to perform some processing on-chip and, in particular, if it is able to detect the presence of the events of interest on its own, i.e., without the intervention of the PU. In this way, most of the node hardware can sleep when the observed scene is static.

In order to validate this methodology, we have presented a prototype node for WCNs mounting a CMOS custom vision sensor and a Flash-based FPGA. The imager is a binary contrast-based sensor capable of extracting certain visual features and detecting significant motion in the scene. Thus, significant power savings are achieved. The Flash-based FPGA is responsible for both system control and data processing. This architecture enables the execution of relatively complex high-level algorithms using a very limited energy budget.

We have measured the power consumed by the node when implementing a people counter, which is a nontrivial task requiring a considerable amount of on-board processing. Results show that the overall consumption has been reduced by two orders of magnitude with respect to other wireless camera-based nodes with similar capabilities recently proposed in the literature. For instance, by powering the system with two batteries providing 2200 mAh at 3.3 V, the expected lifetime of the system is between two and almost four months.

ACKNOWLEDGMENT

The research activities described in this paper have been co-funded by the Italian Ministry of University and Research within the Programmi di ricerca di Rilevante Interesse Nazionale (PRIN) 2008 project titled “Methodologies and measurement techniques for spatio-temporal localization in wireless sensor networks.”

REFERENCES

- [1] S. Soro and W. Heinzelman, “A survey of visual sensor networks,” *Adv. Multimedia*, vol. 2009, p. 640 386, 2009.
- [2] A. M. Tabar, A. Keshavarz, and H. Aghajan, “Smart home care network using sensor fusion and distributed vision-based reasoning,” in *Proc. 4th ACM Int. Workshop VSSN*, New York, 2006, pp. 145–154.
- [3] W.-C. Feng, E. Kaiser, W. C. Feng, and M. L. Baillif, “Panoptes: Scalable low-power video sensor networking technologies,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 1, no. 2, pp. 151–167, May 2005.
- [4] M. Rahimi, D. Estrin, R. Baer, H. Uyeno, and J. Warrior, “Cyclops, image sensing and interpretation in wireless networks,” in *Proc. 2nd Int. Conf. SenSys*, New York, 2004, p. 311.
- [5] 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 Conf. AVSS*, Sep. 5–7, 2007, pp. 69–74.
- [6] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan, “MeshEye: A hybrid-resolution smart camera mote for applications in distributed intelligent surveillance,” in *Proc. 6th Int. Symp. IPSN*, Apr. 25–27, 2007, pp. 360–369.
- [7] T. Teixeira, D. Lymberopoulos, E. Culurciello, Y. Aloimonos, and A. Savvides, “A lightweight camera sensor network operating on symbolic information,” in *Proc. 1st Workshop Distrib. Smart Cameras*, 2006, pp. 1–5.
- [8] T. Teixeira, E. Culurciello, J. Park, D. Lymberopoulos, A. Barton-Sweeney, and A. Savvides, “Address-event imagers for sensor networks: Evaluation and modeling,” in *Proc. 5th Int. Conf. IPSN*, 2006, pp. 458–466.
- [9] M. Gottardi, N. Massari, and S. Jawed, “A 100 μ W 128 \times 64 pixels contrast-based asynchronous binary vision sensor for sensor networks applications,” *IEEE J. Solid-State Circuits*, vol. 44, no. 5, pp. 1582–1592, May 2009.

- [10] J. Polastre, R. Szewczyk, and D. Culler, “Telos: Enabling ultra-low power wireless research,” in *Proc. 4th Int. Symp. IPSN*, 2005, pp. 364–369.
- [11] Intel XScale Microarchitecture Technical Summary, Santa Clara, CA, 2000. [Online]. Available: www.intel.com/design/intelxscale/
- [12] B. Draper, J. Beveridge, A. Bohm, C. Ross, and M. Chawathe, “Accelerated image processing on FPGAs,” *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1543–1551, Dec. 2003.
- [13] Igloo Low-Power Flash FPGAS Datasheet, Mountain View, CA, 2009. [Online]. Available: <http://www.actel.com/>
- [14] L. Gasparini, R. Manduchi, M. Gottardi, and D. Petri, “Performance analysis of a wireless camera network node,” in *Proc. IEEE Instrum. Meas. Technol. Conf. I²MTC*, May 2010, pp. 1331–1336.
- [15] *2.4 GHz IEEE 802.15. 4/ZigBee-Ready RF Transceiver*, Dallas, TX: Texas Instrum., 2006.
- [16] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.
- [17] L. Gasparini, R. Manduchi, and M. Gottardi, “An ultra-low-power contrast-based integrated camera node and its application as a people counter,” in *Proc. 7th IEEE Int. Conf. AVSS*, Aug. 29–Sep. 1 2010, pp. 547–554.
- [18] E. Viarani, “Extraction of traffic information from images at DEIS,” in *Proc. Int. Conf. Image Anal. Process.*, 1999, pp. 1073–1076.
- [19] T. Cormen, *Introduction to Algorithms*. Cambridge, MA: MIT press, 2001.
- [20] I. BIPM, I. IFCC, and I. IUPAC, *OIML 1995 Guide to the Expression of Uncertainty in Measurement*, Geneva, Switzerland, 1995.
- [21] D. Huttenlocher, G. Klanderman, and W. Rucklidge, “Comparing images using the Hausdorff distance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993.
- [22] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun. 25, 2005, vol. 1, pp. 886–893.
- [23] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu, “SensEye: A multi-tier camera sensor network,” in *Proc. 13th Annu. ACM Int. Conf. MULTIMEDIA*, New York, 2005, pp. 229–238.
- [24] C.-F. Hsin and M. Liu, “A distributed monitoring mechanism for wireless sensor networks,” in *Proc. 1st ACM Workshop WiSE*, New York, 2002, pp. 57–66.



Leonardo Gasparini (S’10) received the B.Sc. degree and the M.Sc. degree in telecommunication engineering from the University of Trent, Trent, Italy, in 2004 and 2007, respectively. He is currently working toward the Ph.D. degree in information and communication technology with the Department of Information Engineering and Computer Science, University of Trent.

Since 2007, for his doctoral studies, he has been working on the development and the metrological characterization of an ultralow-power wireless system equipped with a camera. In 2010, he joined the Integrated Optical Sensors and Interfaces Group, Bruno Kessler Foundation, Trent, where he was involved in the design of integrated optical sensors fabricated in deep-submicrometer complementary metal–oxide–semiconductor technology. His research interests include the design, the development, and the metrological characterization of embedded systems, with particular emphasis on low-power applications.



Roberto Manduchi received the Ph.D. degree in electrical engineering from the University of Padova, Padova, Italy.

Then, he joined Apple Computer, Inc. and the Jet Propulsion Laboratory. Since 2001, he was with the University of California Santa Cruz, Santa Cruz, where he is currently an Associate Professor of computer engineering. His research interests include computer vision and sensor processing, with applications to assistive technology for persons with visual impairments.



Massimo Gottardi (M'97) received the Laurea degree in electronics engineering from the University of Bologna, Bologna, Italy, in 1987.

In the same year, he joined the Integrated Optical Sensors Group, Bruno Kessler Foundation, Trent, Italy, where he was initially involved in the design and characterization of charge-coupled-device (CCD) and CCD/complementary metal-oxide-semiconductor (CMOS) optical-sensor arrays with on-chip analog processing, in collaboration with Harvard University, Cambridge, MA, and Interuniversity Microelectronics Centre (IMEC) (B). Since 1993, he has been involved in the design of CMOS integrated optical sensors. He is the holder of three patents on vision sensors and microelectromechanical-system (MEMS) interfaces and is the author or coauthor of over 60 papers in international journals and conferences. His research interests include CMOS image sensor architectures, energy-aware vision sensors, high-speed CMOS optical position-sensitive detectors, low-power capacitive MEMS interfaces and applications in wireless sensor networks.

Mr. Gottardi has served as a Reviewer with the *Journal of Solid-State Circuits*, TRANSACTIONS ON CIRCUITS AND SYSTEMS, the *Journal of Electromechanical Systems*, TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENTS.



Dario Petri (F'91) received the M.Sc. degree (*summa cum laude*) and the Ph.D. degree in electronics engineering from the University of Padova, Padova, Italy, in 1986 and 1990, respectively.

From 1990 to 1992, he was an Assistant Professor with the Department of Electronics and Information Engineering, University of Padova. In 1992, he joined the University of Perugia, Perugia, Italy, as an Associate Professor. Then, in 1999, he has been elevated to full professorship of measurement and electronic instrumentation. Since 2002, he was with the Department of Information Engineering and Computer Science, University of Trent, Trent, Italy, where he is currently the Head. From 2004 to 2007, he was the Chair of the International Ph.D. School in Information and Communication Technology, University of Trent, and, from 2007 to 2010, the Chair of information engineering study programs. He is the author of over 200 papers published in international journals or in proceedings of peer-reviewed international conferences.

Dr. Petri was the Chair of the Italy Chapter of the IEEE Instrumentation and Measurement (I&M) Society from 2006 to 2010. He is currently the Vice Chair of the IEEE Italy Section. In addition, he has been a Cofounder and the General Chair of the Ph.D. School "International Measurement University" of the IEEE I&M Society from 2008. He is an Associate Editor of the IEEE TRANSACTIONS ON INSTRUMENT AND MEASUREMENT.