

Chapter 2

Visual Sensor Nodes

Abstract Recent developments in image and video technologies enabled easy access to a new type of sensor-based networks, Visual Sensor Networks (VSN). VSNs are gaining a lot of attention lately. They are used in several applications including surveillance and telepresence. They consist of several low-cost, low-power visual nodes with sensing, data processing, and communication capabilities. These tiny nodes are able to collect large volumes of images, process them, and send extracted data to each other and to the base station for further analysis. Unfortunately, the huge amount of data captured and processed is faced with the limited resources of such platforms. There are several challenges involved with the design and implementation of VSNs. This chapter presents an overview of visual nodes, architectures, and challenges. It also reviews available VSN platforms and compares their processing capabilities, highlighting the need for new lightweight but efficient image processing algorithms and architectures.

2.1 Introduction

With the advances in wireless communications and low-power sensor nodes designs, Wireless Sensor Networks (WSN) became a focal and active research area worldwide. WSNs consist of spatially distributed sensors that collaborate to monitor and capture data in a given environment [1]. The network has sensing, data processing, and communication capabilities.

Developments in image technology lead to the emergence of a new class of distributed sensor-based networks, Visual Sensor Networks (VSN). VSNs consist of tiny, battery-operated, visual sensor nodes that integrate the image sensor, embedded processor, and wireless transceiver. VSNs are able to capture, process, and transmit visual data collectively over the network to a central station for further processing as shown in Fig. 2.1 [2]. The difference between VSNs and WSNs lies in the type of data collected and processed. Sensors in WSNs capture scalar measurements such as temperature readings, sounds, vibration, or pressure;

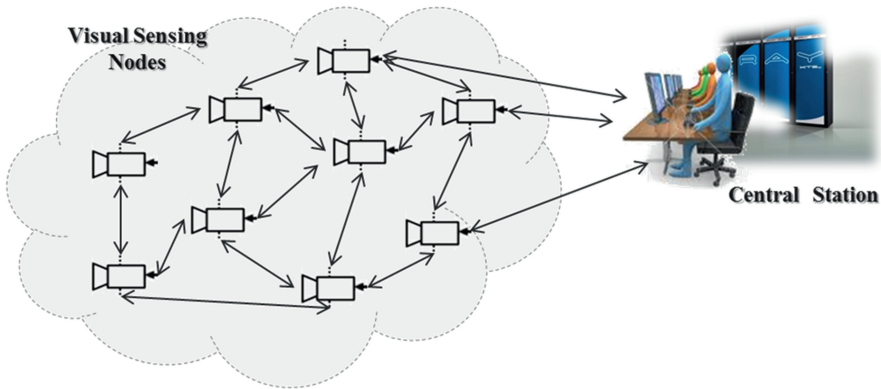


Fig. 2.1 Visual sensor network scenario

all of which are somehow limited compared to VSN data [3]. Imaging sensors in VSNs comprise a wide number of photosensitive cells that capture 2-D sets of data points or images [2]. VSNs thus produce much richer description of a situation of interest. Local processing extracts important information about the scene so the node transmits only intelligent data to the central station.

These capturing, processing, and communication capabilities enable a wide range of vision-based applications. VSNs are useful for surveillance, smart rooms, and many others. Some of these applications are summarized below:

- **Public and commercial surveillance:** VSNs may be used for monitoring public places such as parks, department stores, transport systems, and production sites for infrastructure malfunctioning, accident detection, and crime prevention [4, 5].
- **Environmental and building monitoring:** VSNs are perfect solutions for early detection of landslides, fires, or damages in mountain coasts, historical and archaeological sites and hence preservation of these areas [6, 7].
- **Military surveillance:** Such networks can be employed in patrolling national borders, measuring flow of refugees, and assisting battlefield command and control [8, 9].
- **Smart homes and meeting rooms:** VSNs can provide continuous monitoring of kindergarten, patients, or elderly requiring special care. This helps measure the effectiveness of medical treatments as well as detect early alarms [10]. VSNs are also used for teleconferencing and remote meetings.
- **Telepresence systems:** In VSN-based telepresence systems, the user can view any point from the remote location as if he or she was actually physically present at that location [11].

It is important to note that larger amounts of data require more processing and communications, increase analysis complexity, and exhaust more resources. This is particularly challenging for constrained platforms like VSNs. VSNs are faced with several challenges ranging from camera coverage to low-power data processing, and reliable delay-aware communication protocols [3].

This chapter reviews visual nodes architectures, VSN challenges, and current platforms. The rest of the chapter is organized as follows. Section 2.2 introduces the architecture design for visual sensor nodes. Section 2.3 discusses the challenges faced in VSNs, mainly in terms of visual data processing, sensor control, and communication protocols. Section 2.4 reviews and compares state of the art VSN platforms with the main focus on processing capabilities. The section underlines the need for a new class of image processing algorithms and architectures that suits the limited resources of VSNs, which is the focus on the remaining chapters.

2.2 Visual Sensor Nodes Architecture

Visual sensor nodes are defined as small, battery-operated nodes with image sensor, embedded processor, and wireless transceiver. This allows them to perform the following operations: capture images or videos from the scene, process them locally to extract relevant information, and transmit this data rather than raw images to a base station for activity analysis.

The main blocks in a typical node are shown in Fig. 2.2. They include: a sensing module, processing module, storage module, communication module, and power or energy module. Underlying blocks and the way they are connected differs from one platform to another. Some researchers use completely off-the-shelf components to form a node [12]. Other developers integrate separately designed modules into one node. For instance, the CITRIC node consists of an imager module with a Tmote Sky wireless module [13]. Others implement all building components on one single board [14]. This has the advantage of reducing energy dissipation due to inter-chip communication. A description of each module is provided next. More details about the specific components used in actual VSNs are provided in Sect. 2.4.

The sensing module consists of one or more imaging sensors to capture image and video sequences. Most embedded platforms have an integrated Complementary Metal–Oxide–Semiconductor (CMOS) imaging sensor. Although Charge-Coupled Device (CCD) components were the premier image capture technology, by 2004, CMOS sensors officially surpassed them as the overall image capture technology of choice especially for constrained environments [15]. This is mainly due to the low power consumption of CMOS (one tenth of CCD), low cost, and easy integration of all camera functions on a single chip, significantly reducing chip count and board space [15]. Moreover, CMOS sensors offer the same or better sensitivity compared to CCDs. They are able to eliminate smearing and other effects that can affect image quality and the integrity of the security system. Some CMOS imagers also have dual mode night vision capability, near infrared ones (NIR). This is essential for providing continuous 24-h surveillance. In fact, it is preferable to have different sensing modalities to get a better view of the scene. These include visible, thermal, infrared, and stereo.

Processing and storage modules are responsible for handling all visual data operations including image processing tasks and buffering data during processing.

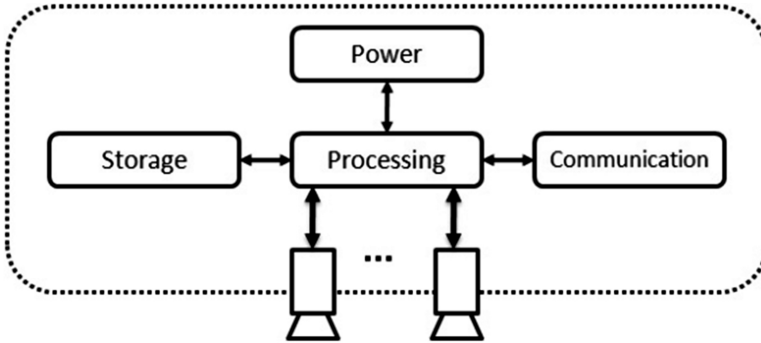


Fig. 2.2 Visual node block diagram

Although VSNs are a class of sensor-based systems, the large volume of data captured and processed requires considerably powerful processors and memory resources. There are three directions followed for processor and memory selection. Some nodes, like Panoptes [12] and Meerkats [16], are designed with relatively powerful general purpose processors and large storage components (internal and additional external memory cards) to achieve high performance. The drawback is they result in high energy dissipation (several Watts). Others use lightweight microprocessor or microcontroller (MCU) and specialized reconfigurable hardware components to address critical parts. For example, background subtraction in Cyclops is performed by a Complex Programmable Logic Device (CPLD) and all other operations are performed by the lightweight processor [17]. MicrelEye uses a Field Programmable Gate Array (FPGA) to implement computational tasks [18]. The last type of nodes uses a medium weight processor with moderate energy dissipation (less than 1 W). VSNs in this category, such as FireFly Mosaic [19] and Vision Mote [20], use 32-bit processors with relatively higher operating frequencies and capabilities.

The communication module or wireless transceiver is responsible for communication with the central station and collaborating with others nodes in the network. Different types of radio are found in current platforms: CC1000 [21], CC2420 [22], Bluetooth [23], and IEEE 802.11 [24]. Chipcon CC1000 transceiver was designed for very low power and very low voltage wireless application. It supports up to 38.4 Kbps raw channel rate [17]. CC2420 is another single-chip 2.4 GHz IEEE 802.15.4 compliant RF transceiver designed for low power and low voltage wireless applications. It supports up to 250 Kbps data rate, which is still not enough to transmit reasonable quality images in real-time [13, 14]. Bluetooth gets up to 230.4 Kbps although the maximum is theoretically 704 Kbps [18]. IEEE 802.11 supports real-time video streaming at the expense of more power dissipation [12, 16].

The energy module is responsible for powering the node. Visual nodes are usually and mostly battery-operated. Renewable and solar energy was recently researched as a potential alternative to prolong the lifetime of these networks [25]. The idea is to use energy harvesting and convert different forms of ambient energy (e.g. solar power, thermal energy, wind energy, salinity gradients, and kinetic energy) into

electricity to power the nodes. However, harvesting solutions proposed so far can produce limited amounts of energy and power nodes interruptedly [26]. Energy saving approaches (in terms of sensing, communication, and data processing) are mostly needed to reduce the nodes energy dissipation.

In terms of Operating Systems (OS), nodes may run a general purpose OS, application specific OS, or no OS at all [27]. General purpose OS, like Linux, provides flexibility to modify parts to fit the application need [12, 13, 16]. Programming and prototyping are easy but incur more overhead and energy consumption compared to application specific OS. OS specifically designed for sensor networks, like TinyOS and Nano-RK, require minimal hardware [17, 18]. Alternatively, some nodes including MeshEye do not have an OS; a finite state machine is responsible for resource management instead [14, 18]. This is faster and more energy efficient but requires longer time to program in hardware [27].

To sum up, this section described the generic model of visual nodes. Different kinds of nodes exist based on the type of sensors, processing hardware, transmission capabilities, level of integration, and OS [27]. These design choices actually determine the capabilities of the nodes, performance, energy consumption, and potential applications. The next section discusses the main challenges faced when incorporating these nodes in VSNs.

2.3 VSN Challenges

VSNs are capable of collecting large volumes of data about monitored scenes but are constrained with the available node resources and network bandwidth. Designing and implementing VSNs is thus faced with several challenges.

First, robust visual data processing are needed on-board to produce useful data and reduce the amount sent over the network but are typically restricted with the node resources (memory and power). Second, camera locations and modes of operation (active/sleep) should be carefully chosen to enforce continuous monitoring at the least energy cost. Third, reliable and delay-aware communication protocols are necessary to meet QoS requirements without exhausting the network. Other challenges include security, authentication, and privacy issues. Data confidentiality and authentication are raising great concerns. This is very critical in surveillance applications where sensitive personal data collected may be misused or distributed.

The next subsections elaborate on some of these issues: visual data processing, sensor management, and communication. Relevant work in each area is summarized.

2.3.1 *Visual Data Processing*

Object vision is one of the key features of VSNs marking it from other sensor-based systems. Visual sensors are capable of capturing large amounts of images of the monitored scene. Raw images are processed locally so only partial useful data is sent

to the central station for further analysis or other nodes for collaborative processing. Performing all this on-board is very challenging given the limited resources of these nodes (power and memory). Conventional processing algorithms intended for massive computers cannot be applied here. They must be modified and adapted to fit on constrained-resource platforms. Efficient and lightweight image processing algorithms and architectures need to be developed. This includes algorithms for preprocessing the image, object processing, and preparing for transmission.

Image registration and fusion are used as preprocessing steps to enhance the image quality by gathering information from multiple sources. Registration aligns two or more images acquired from different viewpoints, times, and sensors. Image fusion follows and combines complementary information from different sources of optical sensors into one composite image. Despite the importance of these two steps, there has not been any significant work concerning fusion and registration on visual nodes. Image fusion and registration are especially needed in heterogeneous VSNs or homogeneous ones with multimodal image sensors. In fact there are two types of VSNs, homogeneous and heterogeneous [3]. All nodes in homogeneous networks contain the same type of camera sensors and perform the same functionality. This reduces the complexity of the network, which makes it suitable for large-scale VSNs with lots of self-organized nodes. Camera nodes in heterogeneous networks have different capabilities to provide better performance. Heterogeneous VSNs consume overall less energy consumption than homogeneous ones, but at the expense of added complexity. Heterogeneous networks are usually divided into multiple stages or tiers. Different sensors and different functions, such as object detection and tracking, are performed at different tiers. SensEye is a heterogeneous multi-tier camera sensor network that follows this paradigm [28]. The first tier uses Mote nodes and CMUCam3 sensors to detect the location of objects. Once objects are detected, web cams in the second tier are woken up by to perform object recognition. Then, high-resolution pan-tilt-zoom cameras in the last tier perform object tracking. This brings us to the next type of processing, object processing.

Object detection and tracking identify interesting objects in the scene, their distinguishing features, and trajectories. This data is then sent to the central station for scene analysis. Considering detection, there are three main approaches: frame differencing, background subtraction, and optical flow [29, 30]. The first type consists of subtracting consecutive image followed by thresholding the difference. Frame differencing is simple and suitable for VSNs, but is not very reliable, and fails to detect all interior object pixels. Background subtraction schemes provide better detection results but are sensitive to dynamic changes in the scene. The current frame is compared to a background model or image rather than the previous image. There are different background modeling techniques ranging from simple ones to more complex and accurate ones that can handle variations in the scene. Optical flow schemes rely on flow vectors of moving objects over time to identify foreground in an image. They are computational and not suitable for VSNs. Tracking on the other hand follows either a top-down approach known as filtering and data association and bottom-up known as target representation and localization.

The former requires lots of matrix manipulation and is usually avoided despite its robustness to occlusions. Bottom-up approaches perform object detection followed by some kind of similarity or correlation-based matching. In fact, frame differencing was used in several platforms such as FireFly [19] and CITRIC [13]. Most platforms implemented some type of background subtraction. For example, Cyclops used a running average filter to estimate the background [17]. MicrelEye even used background subtraction assuming a fixed background [18]. To reduce the computations, MeshEye performed background subtraction and stereo matching on low resolution images (30×30) first. Once object are detected and matched, high-resolution images are triggered to take snapshots of regions of interest. When matching objects, simple features such as position, velocity, and bounding box were extracted [13, 16]. Most of the schemes are chosen for their simplicity but cannot reliably detect and track objects in complex cluttered outdoor scenes. This will be further explained and addressed in the next chapters.

Pre-transmission steps include video coding and compression, which help further reduce the size of data sent to the central station. There are two major types of coding: intra-coding and inter-coding. The former, also known as transform coding such as discrete cosine and wavelet transforms, is used in lossy compression. The latter, also known as motion compensated prediction, achieves higher compression rates for videos but requires more computations. Conventional approaches, especially inter-coding, are not suitable for VSNs. They provide high-resolution images at the cost of high complexity, memory resources, and power consumption. Most existing VSN platforms, such as Cyclops [17], Mosaic [19], Panoptes [12], use quality-scalable JPEG or platform-specific optimized JPEG compression for intra-coding. Redondi et al. proposed a hybrid coding that combines intra and inter-coding to exploit the benefits of both approaches [31]. CITRIC uses a different concept, Compressed Sensing (CS), which efficiently senses image signals and compresses them on the fly [13]. The idea is to benefit from the signal's sparseness allowing the entire signal to be determined from relatively few measurements [32, 33]. CS seems a promising alternative for coding and compressing on VSNs that needs further research. Distributed Source Coding (DSC) is a different approach based on Slepian-Wolf and Wyner-Ziv's theorem [34, 35]. Instead of the classical predictive encoding that jointly encodes and decodes, DSC exploits separate encoding at each node and joint decoding at the central station. This shifts the complexity to the decoder side, lowers power consumption, reduces in transmission rate, and improves error resilience [27]. Di et al. proposed an improved DSC scheme that classifies video data into several sub-sources to exploit additional video statistics and generate more reliable motion vectors [36]. Although DSC has been widely studied, it has not been implemented in real VSNs yet. According to [37], combining distributed source coding and network coding can make significant progress toward performance and lower energy consumption.

To sum up, current lightweight image processing schemes are not sufficient to provide the accuracy needed in video surveillance. It is important to choose efficient visual processing techniques on-board as their output is the basis for further analysis

in the central station. On the other hand, accurate techniques require lots of computations and storage, which exhausts the node quickly (if it can be implemented on the node in the first place). Low-power image operations, collaborative visual processing and coding are all active and promising research areas in VSNs. There is a need to develop novel lightweight but reliable image processing algorithms and architectures for VSNs. This will be addressed in the next chapters.

2.3.2 Sensor Coverage and Management

VSNs consist of several spatially distributed nodes to provide a wide and efficient coverage of the monitored scene. Sensor management is important to maintain the most coverage at the least energy cost even when certain visual sensors fail. To meet these requirements, several design issues must be considered. First, the location, orientation, and mode of operation of each camera node need to be carefully chosen to ensure a well-covered scene. This is even more challenging in the case of pan-tilt-zoom capabilities and multimodal lenses. Second, intelligent but lightweight management schemes should ensure the system is fault tolerant at the least energy dissipation. Redundant nodes are put to sleep to save energy and extend the network lifetime and are waken up when needed (especially when other nodes fail). To sum up, camera coverage and management involves selecting, scheduling, and optimizing nodes' activities to ensure the required coverage at the minimum resource cost. Efficient optimization algorithms were extensively developed for WSNs [38], but this task is more complicated and challenging for VSNs. This is due to the wider 3-D camera coverage, more control parameters, and energy constraints [3].

Huang et al. proposed a solution to the coverage problem by reducing 3-D space to 2-D and then 1-D space [39]. It is possible to simplify the coverage problem by assuming cameras have fixed focal length lens and are mounted on the same plane [3]. For instance, several surveillance cameras are mounted on the ceiling or top of a building facing downward. This results in rectangular fields of view, in which case, it is possible to use WSN solutions. Soro and Heinzelman implemented a WSN application-aware routing and coverage preservation protocol on VSNs [40]. The purpose was to control which nodes are active and which ones are sleeping to minimize energy dissipation. Unfortunately, the protocol did not behave as expected. There is a need for specific routing and coverage preservation protocols for VSNs [39]. Not much work is reported in this area. Yoshida et al. used a cooperative control model in which pan-tilt-zoom cameras dynamically adjust their coverage areas to maintain full coverage of the scene without any central control [41]. Cameras adjust their field of view based on the scene conditions and changes in the system structure by using a spatial pattern generator. They can reduce blind spots in the surveyed area by adjusting their field of view to overlap with that of neighboring nodes. Still, more research is needed to optimize sensor management to meet VSN energy constraints.

2.3.3 *Communication Protocols*

Supporting multimedia applications over wireless networks is challenging due to the resource constraints and stringent Quality of Service (QoS) requirements. These include the need for reliable data transmission and low data rate delays without sacrificing energy efficiency.

VSNs rely heavily on processing and transmitting large amounts of data from nodes to the central station as well as other nodes. Reliable data transmission is a major issue that must be addressed in VSNs. WSNs use data retransmission and link layer error correction to handle transmission problems [42]. However, retransmitting visual data introduces intolerable delays in VSNs, degrades the network quality, and reduces the available bandwidth [43]. Instead, image compression and data aggregation are used in single-path routing scenarios to meet QoS requirements and prolong the network lifetime. Lecuire et al. proposed a wavelet-based image transmission that decomposes an image into packets of different priorities [44]. High-priority packets are sent first and subsequent packets are only transmitted if the node battery level is above a given threshold. An alternative approach considers low-power error correction encoders to ensure efficient data transmission. For example, Reed Solomon encoder reduces the amount of data transmitted while tolerating up to 16 errors in the transmitted codeword [45]. The (255,223) Reed Solomon encoder tolerates even more errors, requires less retransmission and consequently lower power consumption than the original Reed Solomon one [46]. Moreover, multi-path routing may be used instead of single-path routing to reduce packet losses and balance energy consumption more efficiently [47]. Another way is to repartition loads on multiple source-sink paths to avoid network congestion [48]. This will in turn reduce losses in the network and improve reliability. However, multi-hop latency becomes a main concern here especially for video surveillance applications where real-time results are needed.

Several energy-efficient delay-aware Medium Access Control (MAC) protocols were proposed to reduce multi-hop latency [49]. The idea is to adapt nodes operations (sleep and active times) based on network traffic. For instance, Ye et al. developed a MAC protocol that uses adaptive listening to control nodes operations [50]. Alternatively, unified cross-layer approaches were explored to minimize delays in the network. In fact, delays may occur at different layers of the network protocol stack due to channel contention, packet retransmission, long packet queues, node failure, and network congestion [2]. Instead of designing each layer independently, cross-layer design considers close interactions between different layers of the protocol stack to optimize system performance as a whole. All layers are considered dependent. Information from one layer is visible to other layers and may be used to optimize at lower layers. Some cross-layer works are listed next. Andreopoulos et al. proposed an optimization algorithm that finds the optimal routing path, maximum number of retransmissions at the MAC layer, and best modulation scheme at the physical layer [51]. The purpose is to maximize the network's capacity-distortion utility function, given the delay-constraints. Li and Der Schaar

investigated several application layer and MAC strategies jointly to improve multimedia quality [52]. Der Schaar and Turaga adaptively enhanced the robustness and efficiency of scalable video transmission by optimizing MAC retransmission strategy, application layer forward error correction, bandwidth adaptive compression, and adaptive packetization strategies [53].

It is important to note that most research focuses on delivery of visual data over general wireless networks. Optimized VSN cross-layer solutions need to be developed. Another area that needs further investigation is collaborative image data routing [2]. Cross-layer optimization should include strategies for collaboration among different cameras with overlapped field of view to minimize the amount of data transmitted.

2.4 Current VSN Platforms

This section reviews ten currently available VSN platforms. As stated earlier, all these platforms share the same generic blocks. They are battery-operated and consist of some sort of imaging sensor, processor, memory, and communication card. The choice of underlying components and the way they are integrated differs from one platform to another to address different functionality and applications.

The next subsections summarize the hardware components and OS used in each node, typical applications, and local image processing operations. Platforms are presented starting with lightweight ones to intermediate, and then relatively powerful ones capable of handling more local processing. Platforms covered are: Cyclops [17], MeshEye [14], XYZ-Aloha [54], FireFly Mosaic [19], MicrelEye [18], Vision Mote [20], CITRIC [13], WiCa [55], Panoptes [12], and Meerkats [16]. A summary table follows comparing the processing capabilities of different platforms and focusing mainly on image processing tasks (which is the focus of this book). It highlights the need for more accurate but lightweight image processing algorithms and architectures for fusion, registration, detection, and tracking.

2.4.1 *Cyclops*

Cyclops is one of the lightest smart energy-efficient visual sensors [17]. It has a multi-board level architecture that integrates the TinyOS-based Cyclops board with the well-known IEEE 802.15.4/ZigBee-based Mica2 wireless sensor mote [56]. Cyclops board consists of the following components [17]: ultra-compact CIF resolution ADCM-1700 CMOS imager, 8-bit ATMEL ATmega12 RISC MCU, Xilinx XC2C256 CoolRunner CPLD, 64 KB external SRAM (TC55VCM208A from TOSHIBA), and 512 KB external CMOS Flash programmable and erasable read only memory (AT29BV040A from ATMEL). SRAM is used for image buffering and during processing. Flash is used for permanent template storage.

MCU is responsible for controlling Cyclops and communicating with Mica2. Critical parts such as background subtraction and image capture are performed on CPLD. Any idle block is put to sleep to save energy. The maximum power consumption of Cyclops is about 110.1 mW.

Cyclops was used in object detection and hand posture recognition applications. A simple and lightweight background subtraction is used for objects detection. This involves running average filter for background modeling and single thresholding for foreground detection. As for hand posture recognition, Cyclops extracts feature vectors using an orientation histogram transformation, which is robust to illumination changes and translation invariant.

2.4.2 *MeshEye*

MeshEye is a single board that fully integrates imager, processor, memory, and radio blocks [14]. Two types of imagers are used: two low resolution ADNS-3060 high optical mouse (30×30 pixels) and a VGA resolution ADCM-2700 landscape CMOS camera. The rest of the components include an Atmel AT91SAM7 processor, 64 KB SRAM, 256 KB FLASH, 256 MB MMC/SD memory card, and CC2420 2.4 GHz IEEE 802.15.4/ZigBee-based radio module [14]. MeshEye does not have an OS. It relies on finite state machines to handle resource management and operation scheduling. The maximum power consumption is 175.9 mW.

MeshEye was intended for low-power distributed surveillance applications. Typical vision tasks include object detection, stereo matching, and object recognition. Background subtraction is applied on the low resolution images first. This involves comparing current image to the background one, thresholding the difference, and blob filtering. Correlation-based stereo matching follows at the low resolution. Once an object is detected and matched, the high resolution camera is triggered to take a better snapshot of this region for further processing.

2.4.3 *XYZ-Aloha*

XYZ-Aloha integrates two boards [54]: XYZ node and Aloha imager. XYZ node includes a 32-bit OKI ML67Q500x ARM THUMB MCU, 32 KB internal RAM and 256 KB FLASH, 2 MB external RAM, and a Chipcon CC2420 radio. Other imagers were also tested with the XYZ node such as a VGA resolution OV 7649 camera module from OmniVision. XYZ uses a lightweight OS, called SOS that follows event driven design. The overall maximum power consumption is about 238.6 mW.

XYZ was used in pattern recognition problems like letter recognition and hand gesture recognition.

2.4.4 *Vision Mote*

Vision Mote is a fully integrated board with the following components [20]: CMOS imager, 32-bit Atmel 9261 ARM 9 CPU, 128 MB Flash (K9F1G08), 64 MB SDRAM (K4S561632), and CC2430 ZigBee-based module. Running on Linux OS, Vision Mote can benefit from OpenCV libraries to implement image capture, image compression, and other processing functions. The maximum power consumption measured is 489.6 mW.

Vision Mote was used for water conservatory engineering applications. Several motes aggregate into Vision Mesh, a network that collects and compresses images before sending them to the central station in a multi-hop route.

2.4.5 *MicrelEye*

MicrelEye is a fully integrated board that includes the following blocks [18]: a QVGA resolution OV7620 CMOS camera, a reconfigurable ATMEL FPSLIC SoC, 1 MB SRAM for frame storage, and LMX9820A Bluetooth transceiver. The SoC contains an 8-bit AT40K MCU, reconfigurable FPGA, and 36 KB onboard SRAM. Similar to Cyclops, critical image processing tasks like background subtraction are performed on FPGA and lightweight operations are performed on the MCU. Having both MCU and FPGA on the same chip accelerates processing and eliminates energy dissipation due to inter-chip connections (maximum 500 mW measured). MicrelEye has no operating system.

MicrelEye node was used for detecting people. Object detection involves pixel-based background subtraction assuming a fixed background frame. Detection is performed on FPGA and the remaining classification on the MCU. Object classification involves extracting the feature vector and feeding it to a State Vector Machine-like (SVM) learning structure to establish if this is a human being or not.

2.4.6 *FireFly Mosaic*

Mosaic is the first VSN with multiple collaborative visual nodes [19]. FireFly Mosaic platform consists of the FireFly WSN platform integrated with the CMUCam3 vision board [57]. The vision board consists of a CIF resolution OmniVision OV6620 camera, an Averlogic AL440b FIFO for frame buffering, a low-cost 32-bit LPC2106 ARM7TDMI MCU running at 60 MHz, 64 KB on-chip RAM, and 128 KB on-chip FLASH memory. The FireFly node contains a low-power 8-bit Atmel Atmega1281 processor, 8 KB RAM, 128 KB FLASH memory, and a Chipcon CC2420 802.15.4 radio. Each node runs on Nano-RK and includes an AM receiver for external time synchronization. The maximum power consumption measured is 572.3 mW.

FireFly Mosaic was mainly used in assisted living applications for elderly. It identifies regions with frequent particular activities. The end result of the application is a Markov model, which characterizes the transition probabilities of activity regions. Several image processing functions are supported including JPEG compression, frame differencing, color tracking, convolutions, histogramming, edge detection, connected component analysis, and face detection.

2.4.7 CITRIC

CITRIC platform consists of two boards [13]: a CITRIC imaging board and Tmote Sky wireless module. Tmote Sky runs TinyOS/NesC and contains a 16-bit MSP430 MCU, 10 KB RAM, 48 KB FLASH, a Chipcon CC2420 I.E. 802.15.4 radio, and 1 MB external FLASH [58]. The imaging board includes a 1.3 megapixel SXGA resolution OV9655 CMOS imager, a frequency-scalable 32-bit PDA class CPU (Intel XScale PXA270 with a wireless coprocessor to accelerate multimedia tasks and 256 KB internal SRAM), 16 MB FLASH, 64 MB RAM, and Wolfson WM8950 mono audio ADC. Total power consumption is about 970 mW.

CITRIC was used for target tracking and camera localization. Single target tracking proceeds by segmenting the image into background and foreground. To do that, simple frame difference, single thresholding, and median filter are performed. Bounding boxes are then computed for each detected object and sent to the central station. Camera localization uses tracking results from multiple cameras to estimate the position and orientation of the camera and its field of view [13].

2.4.8 WiCa

WiCa platform includes the following components [55]: VGA resolution OM6802 image sensor, Xetal-II Single Instruction Multiple Data (SIMD) processor with 320 processing elements, ATMEL 8051 processor, 10 MB SRAM, and Chipcon CC2420 Zigbee-based radio module. High-level operations are performed on the ATMEL CPU. Low-level image tasks suitable for parallel processing are implemented and accelerated on the SIMD processor. The idea is to benefit from regularity in image processing operations to parallelize the work and speed it up. Moreover, both processors can access data from the RAM at the same time which enables each one to operate on its own pace. Processing one image line in WiCa takes a single clock cycle.

WiCa platform is used in several applications including distributed face detection, Canny Edge detection, gesture recognition, and many others. WiCa is a promising

VSN platform as it exploits parallelism using SIMD processor rather than FPGA. However, energy dissipation is a main concern for such platforms.

2.4.9 *Panoptes*

Panoptes is one of the first reported powerful Linux-based VSNs capable of processing VGA images. It is designed using several off-the-shelf components. These include: an Intel StrongARM 206 MHz embedded platform, a Logitech 3000 USB-based video camera, 64 MB of on-board memory, and IEEE 802.11-based networking card. The powerful processor allows relatively more computational vision algorithms than other lighter weight node types at the expense of more energy consumption (maximum of 5.3 W measured).

Panoptes was used for environmental observation and surveillance applications. It offers several functionalities: video capture, spatial compression, filtering, buffering, adaptation, streaming, storage and retrieval of video data from sensor.

2.4.10 *Meerkats*

Meerkats is a powerful node designed with several off-the-shelf components [16]. It is built on top of a Crossbow Stargate platform [56] and contains the following parts: a USB-based VGA resolution Logitech QuickCam Pro 4000, XScale PXA255 400 MHz CPU, 32 MB FLASH memory, 64 MB SDRAM, and an Orinoco Gold IEEE 802.11 wireless card. Similar to the powerful Panoptes nodes, the energy consumption of Meerkats is 1 order higher than others (maximum of 3.5 W).

Meerkats was used for outdoor and indoor monitoring applications. The main image processing steps performed onboard include object detection, object tracking, followed by compression to reduce the amount of data sent to the central computer. Objects are detected using a motion analysis scheme. It is assumed that objects are constantly moving; objects that stop are no longer detected [16]. Once foreground regions are detected, their positions and velocities are computed. Clustering is used to identify multiple objects in the scene. This data is then JPEG compressed and transmitted.

2.4.11 *Observations*

Table 2.1 compares the VSN platforms described above in terms of processing capabilities, maximum power consumption, vision algorithms, and applications. Lightweight platforms such as Cyclops, which is the only VSN with an 8-bit

Table 2.1 Comparison of current visual nodes in terms of processing capabilities

Reference	Processor(s)	Energy (mW)	Applications/image processing capabilities
[17]	8-bit ATMEL ATmega12 MCU, XC2C256 CoolRunner CPLD	110.1	Object detection: background subtraction Hand posture recognition: edge detection
[14]	32-bit ATMEL AT91SAM7 processor	175.9	Distributed surveillance: background subtraction, stereo matching
[54]	32-bit OKI ML67Q500x ARM MCU	238.6	Pattern recognition: histogram reconstruction, motion detection, edge detection
[20]	32-bit Atmel 9261 ARM 9 CPU	489.6	Water conservation: JPEG compression
[18]	8-bit AT40K MCU and ATMEL FPSLIC SoC	500	People detection: background subtraction
[19]	32-bit LPC2106 ARM7TDMI MCU and 8-bit Atmel Atmega1281 processor	572.3	Assisted living: frame differencing, color tracking, convolution, edge detection
[13]	32-bit Intel XScale PXA270 CPU and 16-bit MSP430 MCU	970	Single target tracking: background subtraction Camera localization using multi-target tracking: frame differencing, object extraction, data association
[55]	Xetal-II SIMD	–	Face detection and gesture recognition: Canny edge detection
[16]	32-bit Xscale PXA255	3500	Tracking moving bodies: background subtraction, frame differencing
[12]	32-bit StrongARM	5.3	Video surveillance: motion detection

processor, can perform very limited image processing tasks on-board. These include background subtraction using running average filter, frame differencing, and single thresholding. Video surveillance typically requires more accurate schemes, which need more powerful processors and larger memory. Intermediate 32-bit processors provide a middle solution that trades off computations and energy. Examples include FireFly Mosaic, CITRIC, MeshEye, VisionMote, and MicrelEye. Most of these VSNs implement frame differencing or simple unimodal background subtraction. MicrelEye uses a reconfigurable hardware (FPGA) to handle computational parts such as background subtraction and image capture and MCU for the rest of the operations. To reduce the computations, MeshEye performs background subtraction and stereo matching on low resolution images (30×30) first. Once object are detected and matched, high-resolution images are triggered to take snapshots of regions of interest. Simple features are extracted for object matching such as position, velocity, and bounding box. These schemes are chosen for their simplicity but cannot reliably detect and track objects in complex outdoor scenes with clutter motion and multiple occlusions. Relatively powerful nodes such as Meerkats and Panoptes can

support more operations. But even these did not investigate accurate implementations for fusion, registration, detection and tracking in challenging outdoor scenes. This includes outdoor surveillance where scenes undergo large changes, clutter motion, and objects occlusions. Another note here is regarding power consumption. The more computations are, the more energy dissipation (about 5 W in Panoptes). This will quickly exhaust the sensor nodes. Actually, VSN power consumption is dominated by the large amount of data computations rather than communication [27]. WiCa is the first platform that relies on Xetal SIMD. It enables fast implementation of edge detection and face recognitions schemes. WiCa seems very promising in terms of processing capabilities. However, there are no results regarding the overall energy consumption, which is expected to be large on such platforms.

To sum up, one of the main challenges of VSNs is the tradeoff between the algorithms accuracy, memory/processing capabilities, and power consumption of the nodes. Investigating novel low-power but robust algorithms and architectures for video sensor nodes is an active research topic. Accurate and lightweight algorithms should be further investigated and fast hardware architectures should be designed to accelerate critical tasks.

References

1. W. Dargie and C. Poellabauer, *Fundamentals of wireless sensor networks: theory and practice*, John Wiley and Sons, 2010.
2. S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Advances in Multimedia*, vol. 2009, 2009.
3. Y. Charfi, B. Canada, N. Wakamiya and M. Murata, "Challenging issues in visual sensor networks," *IEEE Wireless Communications*, pp. 44-49, 2009.
4. D. M. Sheen, D. L. McMakin and T. E. Hall, "Three-dimensional millimeter-wave imaging for concealed weapon detection," *IEEE Transactions on Microwave Theory and Techniques*, vol. 49, no. 9, pp. 1581-1592, 2001.
5. J. Wang, C. Qimei, Z. De and B. Houjie, "Embedded wireless video surveillance system for vehicle," in *International Conference on Telecommunications*, Chengdu, China, 2006.
6. G. Barrenetxea, F. Ingelrest, G. Schaefer and M. Vetterli, "Wireless sensor networks for environmental monitoring: the SensorScope experience," in *IEEE International Zurich Seminar on Communications*, Zurich, 2008.
7. T. H. Chen, P. H. Wu and Y. C. Chiou, "An early fire-detection method based on image processing," in *IEEE International Conference on Image Processing*, Singapore, 2004.
8. L. Cutrona, W. Vivian, E. Leith and G. Hall, "A high-resolution radar combat-surveillance system," *IRE Transaction on Military Electronics*, Vols. MIL-5, no. 2, pp. 127-131, 2009.
9. M. Skolnik, G. Linde and K. Meads, "Senrad: an advanced wideband air-surveillance radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 4, pp. 1163-1175, 2001.
10. S. Fleck and W. Strasser, "Smart camera based monitoring system and its application to assisted living," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1698-1714, 2008.
11. O. Schreer, P. Kauff and T. Sikora, *3D Videocommunication*, Chichester, UK: John Wiley and Sons, 2005.
12. W. C. Feng, E. Kaiser, M. Shea and B. L., "Panoptes: scalable low-power video sensor networking technologies," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 1, no. 2, pp. 151-167, 2005.

13. P. Chen, P. Ahammed, C. Boyer, S. Huang, L. Lin, E. Lobaton, M. Meingast, S. Oh, S. Wang, P. Yan, A. Y. Yang, C. Yeo, L. C. Chang, D. Tygar and S. S. Sastry, "CITRIC: a low-bandwidth wireless camera network platform," in *Proc. International Conference on Distributed Smart Cameras*, 2008.
14. S. Hengstler, D. Prashanth, S. Fong and H. Aghajan, "MeshEye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance," in *6th International Symposium on Information Processing in Sensor Networks*, Cambridge, 2007.
15. "Security & surveillance: envisioning a safer world," [Online]. Available: <http://www.ovt.com/applications/application.php?id=10>.
16. J. Boice, X. Lu, C. Margi, G. Stanek, G. Zhang, R. Manduchi and K. Obraczka, "Meerkats: a power-aware, self-managing wireless camera network for wide area monitoring," in *Proceedings Workshop on Distributed Smart Cameras*, 2006.
17. M. Rahimi, R. Baer, O. I. Iroez, J. C. Garcia, J. Warrior, D. Estrin and M. Srivastava, "Cyclops: in situ image sensing and interpretation in wireless sensor networks," in *International Conference on Embedded Networked Sensor Systems*, New York, 2005.
18. A. Kerhet, M. Magno, F. Leonardi, A. Boni and L. Benini, "A low-power wireless video sensor node for distributed object detection," *Journal on Real-Time Image Processing*, vol. 2, pp. 331-342, 2007.
19. A. Rowe, D. Goal and R. Rajkumar, "FireFly Mosaic: a vision-enabled wireless sensor networking system," in *IEEE International Real-Time Systems Symposium*, 2007.
20. M. Zhang and W. Cai, "Vision mesh: a novel video sensor networks platform for water conservation engineering," in *IEEE International Conference on Computer Science and Information Technology*, 2010.
21. "CC1000: single chip very low power RF transceiver," [Online]. Available: <http://www.ti.com/lit/ds/symlink/cc1000.pdf>.
22. "CC2420: 2.4 GHz IEEE 802.15.4/ZigBee-ready RF transceiver," [Online]. Available: <http://inst.eecs.berkeley.edu/~cs150/Documents/CC2420.pdf>.
23. "A look at the basics of bluetooth wireless technology," [Online]. Available: <http://www.bluetooth.com/Pages/basics.aspx>.
24. B. P. Crow, I. Widjaja, J. G. Kim and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116-126, 2002.
25. "Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) - survey and challenges," in *International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, Aalborg, 2009.
26. D. Niyato, E. Hossain, M. M. Rashid and V. K. Bhargava, "Wireless sensor networks with energy harvesting technologies: a game-theoretic approach to optimal energy management," *IEEE Wireless Communications Magazine*, vol. 14, no. 4, pp. 90-96, 2007.
27. B. Tavli, K. Bicakci, R. Zilan and J. M. Barcelo-Ordinas, "A survey of visual sensor network platforms," *Multimedia Tools and Applications*, vol. 60, no. 3, pp. 689-726, 2011.
28. P. Kulkarni, D. Ganesan, P. Shenoy and Q. Lu, "SensEye: a multi tier camera sensor network," in *ACM International Conference on Multimedia*, 2005.
29. A. M. McIvor, "Background subtraction techniques," in *Image and Vision Computing New Zealand*, Hamilton, 2000.
30. L. Wang, W. Hu and T. Tan, "Recent developments in human motion analysis," *Pattern recognition*, vol. 36, no. 3, pp. 585-601, March 2003.
31. A. Redondi, M. Cesana and M. Tagliasacchi, "Low bitrate coding schemes for local image descriptors," in *IEEE International Workshop on Multimedia Signal Processing*, 2011.
32. D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289-1306, 2006.
33. M. Fornasier and H. Rauh, "Compressive sensing," in *Handbook of mathematical methods in imaging*, Springer, 2011, pp. 187-228.

34. J. W. D. Slepian, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, pp. 471-480, 1973.
35. A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1-10, 1976.
36. J. Di, A. Men, B. Yang, F. Ye and X. Zhang, "An improved distributed video coding scheme for wireless video sensor network," in *IEEE Vehicular Vehicular*, 2011.
37. C. Li, J. Zou, H. Xiong and C. W. Chen, "Joint coding/routing optimization for distributed video sources in wireless visual sensor networks," *IEEE Transactions on Circuits, Systems and Video Technology*, vol. 21, no. 2, pp. 141-155, 2011.
38. X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *International Conference on Embedded Networked Sensor Systems*, 2003.
39. C.-F. Huang, Y.-C. Tseng and L.-C. Lo, "The coverage problem in three-dimensional wireless sensor networks," *Journal of Interconnection Networks*, vol. 8, no. 3, pp. 209-227, 2007.
40. S. Soro and W. B. Heinzelman, "On the coverage problem in video-based wireless sensor networks," in *IEEE Conference on Broadband Networks*, 2005.
41. A. Yoshida, K. Aoki and S. Araki, "Cooperative control based on reaction-diffusion equation for surveillance system," in *Knowledge-Based Intelligent Information and Engineering Systems*, 2005.
42. Y. Charfi, N. Wakamiya and M. Murata, "Adaptive and reliable multipath transmission in wireless sensor networks using forward error correction and feedback," in *IEEE Conference on Wireless Communications and Networking*, 2007.
43. K.-Y. Chow, K.-S. Lui and E. Y. Lam, "Efficient on-demand image transmission in visual sensor networks," *EURASIP Journal on Applied Signal Processing*, vol. 2007, pp. 1-11, 2007.
44. C. D.-F. V. Lecuire and N. Krommenacker, "Energy-efficient transmission of wavelet-based images in wireless sensor networks," *EURASIP Journal on Image Video Processing*, 2007.
45. S. B. Wicker and V. K. Bhargava, Reed-Solomon codes and their application, John Wiley and Sons, 1999.
46. J. J. Ong, L. Ang and K. Seng, "FPGA implementation reed solomon encoder for visual sensor networks," in *International Conference on Computer Communication and Management*, 2011.
47. H. Wu and A. A. Abouzeid, "Error resilient image transport in wireless sensor networks," *Computer Networks*, vol. 50, no. 15, pp. 2873-2887, 2006.
48. M. Maimour, C. Pham and J. Amelot, "Load repartition for congestion control in multimedia wireless sensor networks with multipath routing," in *International Symposium on Wireless Pervasive Computing*, 2008.
49. S. Misra, M. Reisslein and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 4, pp. 18-39, 2008.
50. W. Ye, J. Heidemann and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *International Annual Joint Conference of the IEEE Computer and Communication Societies*, 2002.
51. Y. Andreopoulos, N. Mastrorade and M. v. d. Schaar, "Cross-layer optimized video streaming over wireless multi-hop mesh networks," *IEEE Journal on Selected Areas in communications*, vol. 24, no. 11, pp. 2104-2115, 2006.
52. Q. Li and M. V. D. Schaar, "Providing adaptive qos to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 278-290, 2004.
53. M. v. d. Schaar and D. Turaga, "Content-based cross-layer packetization and retransmission strategies for wireless multimedia transmission," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 185-197, 2007.
54. D. Lymberopoulos and A. Savvides, "XYZ: a motion-enabled, power aware sensor node platform for distributed sensor network applications," in *International Conference on Information Processing in Sensor Networks*, 2005.

55. R. Kleihorst, A. Abbo, B. Schueler and A. Danillin, "Camera mote with a high-performance parallel processor for realtime frame-based video processing," in *International Conference on Distributed Smart Cameras*, 2008.
56. "Crossbow technology," [Online]. Available: <http://www.xbow.com>.
57. "CMUcam: open source programmable embedded color vision sensors," [Online]. Available: <http://www.cmucom.org/>.
58. "Tmote sky," [Online]. Available: <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>.

Video Surveillance for Sensor Platforms

Algorithms and Architectures

Al Najjar, M.; Ghantous, M.; Bayoumi, M.

2014, XV, 202 p. 89 illus., 47 illus. in color., Hardcover

ISBN: 978-1-4614-1856-6