Journal of Real-Time Image Processing (2020) 17:1869–1882
https://doi.org/10.1007/s11554-020-00943-6

SPECIAL ISSUE PAPER

Communication and computation inter-effects in people counting using intelligence partitioning

Irida Shallari[1] · Silvia Krug[1] · Mattias O'Nils[1]

Received: 5 April 2019 / Accepted: 7 January 2020 / Published online: 23 January 2020

Abstract

The rapid development of the Internet of Things is affecting the requirements towards wireless vision sensor networks (WVSN). Future smart camera architectures require battery-operated devices to facilitate deployment for scenarios such as industrial monitoring, environmental monitoring and smart city, consequently imposing constraints on the node energy consumption. This paper provides an analysis of the inter-effects between computation and communication energy for a smart camera node. Based on a people counting scenario, we evaluate the trade-off for the node energy consumption with different processing configurations of the image processing tasks, and several communication technologies. The results indicate that the optimal partition between the smart camera node and remote processing is with background modelling, segmentation, morphology and binary compression implemented in the smart camera, supported by Bluetooth Low Energy (BLE) version 5 technologies. The comparative assessment of these results with other implementation scenarios underlines the energy efficiency of this approach. This work changes pre-conceptions regarding design space exploration in WVSN, motivating further investigation regarding the inclusion of intermediate processing layers between the node and the cloud to interlace low-power configurations of communication and processing architectures.

Keywords Intelligence partitioning · Smart camera · WVSN · Energy-efficiency · IoT · In-sensor processing

1 Introduction

Scenarios such as industrial monitoring [1], environmental monitoring [2], and smart city [3] have to a wide extent changed the constraints towards wireless vision sensor networks (WVSN), requiring several cameras to cover large areas, while performing image-processing and communication tasks with real-time performance. Looking back at the evolution of camera-based networks, many changes have been introduced through the years in terms of design and architecture. Initially, such systems consisted mainly of CCTV cameras connected either to monitors for constant visual inspection, or stored to a memory device. As a result, the architecture came with high financial costs due to the constant need for staff to perform visual inspection, as well as extensive memory requirements. The necessity for the systems to be more autonomous resulted in the design of smart camera nodes, which embedded processing capabilities in the camera node itself. However, scenarios such as industrial or outdoor monitoring bring attention to the complexity of wired vision systems architectures due to camera deployment limitations and electromagnetic noise. From the communication perspective, this has been overcome through the implementation of wireless communication technologies, i.e. WVSN. To address the problem as a whole, we need to investigate energy-efficient designs of smart cameras, enabling their implementation as battery-operated devices with a satisfactory lifetime.

Energy efficiency in the smart camera node is a twofold problem, highly influenced by the allocation of image-processing tasks. On the one hand, there are tasks such as background modelling, segmentation, morphology, detection and tracking that are computationally demanding. On the other hand, there are different communication technologies alternating in terms of energy efficiency and delay for

✉ Irida Shallari
irida.shallari@miun.se

Silvia Krug
silvia.krug@miun.se

Mattias O'Nils
mattias.onils@miun.se

[1] Mittuniversitetet, Holmgatan 10, Sundsvall, Sweden

different data rate requirements. An overview of state of the art research on WVSNs indicates that smart camera optimisation (either image-processing performance or energy consumption), and communication networks optimisation (energy efficient protocols) are analysed as separate entities. The energy efficiency of the smart camera node through processing focuses on energy-efficient implementation of image-processing algorithms [4], or distribution of tasks between the node and a server/cloud, connected by a communication technology chosen a priori [5]. The presented architectures consist of three main configurations:

- In-sensor processing: the processing is done in the sensor node, while the configuration between hardware and software implementation can vary. In [4, 6] the aim is to optimise such architectures for smart cameras.
- Sensor and remote processing: the processing tasks can be partitioned between the sensor and the remote processor, which can be a server or the cloud. In [7] they provide an example of outdoor monitoring with the allocation of image-processing tasks chosen a priori.
- Remote processing: the sensor node has no processing capabilities, and all the data are transferred to the cloud for processing and analysis.

However, in articles on the analysis of communication energy efficiency, they implement a predefined image-processing design on the node, while the emphasis is on the optimisation of the energy consumption for the communication model. As a result, all previously mentioned approaches provide partial understanding of the variations in the smart camera node energy consumption, as they omit the inter-effects of processing and communication configurations.

We use the term intelligence partitioning to refer to a design exploration approach that analyses the inter-effects of the processing and communication component in the WVSN, optimising the node energy consumption and the real-time performance of the system. This approach relies on the analysis of several partitioning configurations of the image-processing tasks between the computational entities. The schematic representation in Figs. 1 and 2 show that for each configuration, we select a partition point; the tasks on the left side are allocated in the smart camera, while the tasks on the right side are executed in the cloud. Furthermore, for the tasks allocated in the smart camera we also consider the partitioning configurations generated by hardware/software partitioning of the tasks. The overall energy consumption of the smart camera node, for each of the task partitioning configurations, is the sum of the processing energy consumption and the energy consumption of each of the communication technologies included in the analysis.

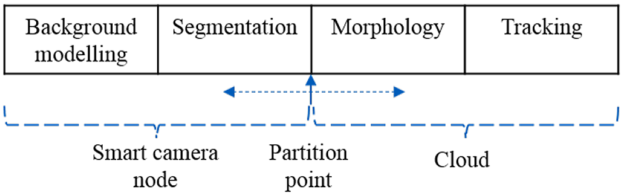


Fig. 1 Representation of the image-processing tasks partitioning

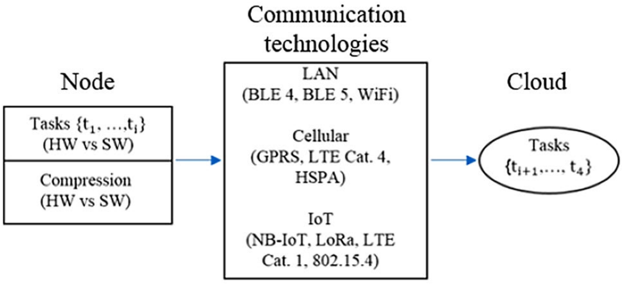

Node
Tasks $\{t_1, \ldots, t_i\}$ (HW vs SW)
Compression (HW vs SW)
Communication technologies
LAN (BLE 4, BLE 5, WiFi)
Cellular (GPRS, LTE Cat. 4, HSPA)
IoT (NB-IoT, LoRa, LTE Cat. 1, 802.15.4)
Cloud
Tasks $\{t_{i+1}, \ldots, t_4\}$

Fig. 2 Schematic representation of intelligence partitioning between the smart camera node and the cloud, including the communication technologies

In this paper, we introduce an analysis of the inter-effects between computation and communication energy consumption in a smart camera node, for different task partitioning configurations of image processing. We investigate how the allocation of image-processing tasks on embedded hardware, software or cloud affects the overall node energy consumption, for a people counting scenario. Furthermore, we implement this approach to the scenarios presented by [8] and [9] to compare the energy efficiency of this method with their results. The aim of this work is not only to provide an energy-efficient design for a people counting scenario, but mainly to explore the trade-off between computation and communication energy for data-intensive IoT scenarios.

The remainder of this paper is organised as follows. In Sect. 2 we review WVSN architectures focusing on approaches for energy-efficient nodes, and their decision-making process in the design space exploration. Section 3 introduces the models for intelligence partitioning and communication, while Sect. 4 provides a detailed analysis of the selected communication technologies. In Sect. 5 we present the design cases used for the intelligence partitioning analysis and measurements results for aspects such as communication delay and the node energy consumption, which are further discussed in Sect. 6. Section 6 summarises the results and key findings of this paper.

2 Related work

The architecture of people counting systems can be divided into two categories: image based and non-image based. Several non-image-based methods rely on the use of passive infrared sensors (PIR) in the region of interest, but with a major drawback due to occlusion handling. Alternative methods such as analysis of WiFi channel use [10], or a combination of infrared lasers alongside an IR camera to detect the IR rays displacement [11] have been implemented, providing an approximate count, but without distinguishing between people and other moving elements that can be present in the region of interest. Instead, image-based people counting models introduced in [12–15] rely on depth imagery to overcome occlusion limitations, while using GPU as processing element. The emphasis of their approach is on the accuracy of detection; hence, they propose models that are computationally intensive in devices that cannot be battery operated. Furthermore, cloud processing for such scenarios would require high bandwidth and inflict significant delays, which would affect the real-time performance.

In the last decade, many WVSN architectures have been developed and further optimised in terms of energy consumption. Several strategies have been introduced through the years, one of which has been an energy-efficient design through hardware/software optimisation in the smart camera node. Birem et al. [4] introduced DreamCam, a smart camera based on low-power Altera-Cyclone III FPGA. They optimised the design by implementing their own hardware modules for feature extraction. Another smart camera architecture is the SENTIOF-CAM designed by [5], where they implemented a low-complexity background modelling algorithm and duty cycling to optimise the energy consumption of the smart camera node. In addition, they partitioned the image-processing tasks between the node and a server. However, in their evaluation of a partitioning point they only considered the reduction of the data size to be transferred, not including the trade-off between processing and communication energy.

In several WVSNs, the smart camera node's duty cycling and sleep mode have been used to reduce energy consumption. Hengstler et al. [16] implemented a sleep mode in the smart camera MeshEye based on a predefined time interval. This resulted in a reduced performance of the smart camera, as the active intervals are not event driven. Instead, [7, 17] control the sleep/wake-up mode by using a passive infrared motion detector for scenarios of surveillance of restricted areas. However, the performance of the proposed system would be affected by two elements: wake-up delay and non-accurate pre-processing. The system wake-up delay can be significant in the chosen scenario, while the background modelling algorithm they implemented requires several history frames before providing a robust model to support an accurate detection.

An alternative approach towards the energy optimisation of WVSNs is introduced in the HuSIMS [18] project. They propose the use of semantic conversion to reduce the frame size before transferring the information to the server, without providing an analysis of the energy consumption changes between the proposed method and in-node processing. In addition, the model uses mobile communication networks to transfer full frames to the end user when the alarm is activated, without considering the resulting high data rate. Furthermore, Berni et al. [19] designed the WVSN node Wi-FLIP, based on analogue pixel-level processing. Their system provides parallelism and energy efficiency, while penalising detection performance and adaptability of the design towards alternative vision sensors. Cao et al. [20] have instead designed a self-optimising IoT WVSN, carrying out real-time processing and experimental communication configurations depending on the current energy levels. From the node architecture perspective, the configuration varies only in algorithms selection, while the processing allocation remains constant. Regarding communication, their prototype is promising for ideal environments, but the results might differ if implemented with current technology. In [21, 22, 23], they consider optimisation in WSN, and focus on the distribution of the computational load to optimise resource utilisation. The framework introduced in [21] optimises the distribution of computational load and communication from the server to the processing nodes; the framework in [22] is implemented on the node itself to provide reliable processing, communication and data, while the framework in [23] focuses on task scheduling for cloud architectures, to reduce the execution latency for intensive tasks. The three above-mentioned frameworks provide a limited view of the problem for battery-operated nodes, as they present no data regarding the energy consumption and delay for the sensor to server/cloud communication, or regarding the effects of implementing the framework in the node itself.

The architectures reviewed present a variety of algorithmic and architectural methods to optimise the energy consumption of the edge node represented by the smart camera. The main limitation of the models is the exclusion, or restriction of the communication component to a single protocol chosen a priori. Instead, the growth in size and complexity of WVSNs requires a more inclusive analysis of the energy consumption in the node, in the IoT context. In this paper, we provide an analysis of the energy efficiency of the smart camera node evaluating the trade-off in energy consumption for different allocations of image-processing tasks between the smart camera node and the cloud. In addition, the model includes the energy consumption estimation for three categories of communication technologies (LAN, Cellular, and IoT) related to IoT scenarios. The aim is to provide an insight into the inter-effects of processing allocation and communication technologies in the overall energy consumption of the WVSN.

One important aspect of distributed smart camera systems is the interconnection of the different nodes by means of communication networks. Several networking technology options have been suggested for the various systems described above. However, the focus of designing smart camera systems is typically on the image-processing features and their implementation rather than communication technologies, leading to the selection of communication interfaces based on modules availability and data rate requirements. As a result, such systems tend to employ energy hungry communication technologies, such as cellular mobile communication or WiFi (e.g. [17, 18]) that provide high data rate. In terms of overall node energy consumption, this is likely to result in sub-optimal technology selections.

The process of selecting the communication technology requires application-specific traffic pattern and technology-specific protocol operation. Previous approaches that do consider communication costs apply simplistic models of the interfaces, especially the technology-specific protocol operation that controls the timing of the communication, and thus the energy consumed is often omitted (e.g. [20, 24]). Hence, we developed a model [25] reviewing the available technology options, including the differences in energy consumption, based on different protocol behaviour. In contrast to other communication-specific articles with IoT focus, we target quite opposite assumptions regarding the data rate and sending intervals. The focus of other publications is on handling many devices in parallel with low data rates and typically long duty cycles. Based on these assumptions, several low-rate technologies were developed for both short-range (e.g. IEEE 802.15.4, Bluetooth) and long-range applications (e.g. LoRa, NB-IoT, SigFox) [26]. Due to the low data rates provided by these technologies, they are often not considered for WVSNs, even if they are designed to provide more energy-efficient communication and require less power during operation. In [26] , Morin et al. present node lifetime estimations for different ad hoc style IoT communication candidates based on the behaviour of the protocols. However, in their analysis they omit scenarios with frequent and large data transfers, such as WVSN scenarios. In [25], we provided models allowing this comparison and apply them in the current study.

3 Methods

3.1 Intelligence partitioning

The evolution of requirements and constraints in WVSN in terms of energy consumption and delay requires a paradigm shift in design space exploration. Intelligence partitioning, providing an insight into the inter-effects of processing and communication, provides the support needed for the design of energy-efficient WVSN. A WVSN consists of a set of tasks, where a specific task t_i is not bound to a specific geographical location; as such, it can be mapped to either the smart camera node or the cloud. The distribution function F in Eq. (1) has its functionality distributed between the camera node and the cloud, where the subsets f_{Node} and f_{Cloud} are the different clusters of tasks mapped to different computational elements.

$$F = \{t_1, t_2, t_3, t_4\}, F = f_{\text{Node}} \cup f_{\text{Cloud}} \quad \textit{and} \quad \emptyset = f_{\text{Node}} \cap f_{\text{Cloud}}. \tag{1}$$

The mapping function of the computational load to the computational elements is referred to as intelligence partitioning $\mathfrak{I}(F)$.

$$\mathfrak{I}(F) = \left\{ \begin{array}{l} \{f_{\text{Node}}, f_{\text{Cloud}}\} \\ D_{Node \to Cloud} \end{array} \right. . \tag{2}$$

Computational latency is defined in Eq. (3) as the time for processing each set of tasks $\{f_{\text{Node}}, f_{\text{Cloud}}\}$ on the respective computational platform at each computational layer $\{P_{\text{Node}}, P_{\text{Cloud}}\}$.

$$L = \sum_{p \in \{\text{Node,Cloud}\}} L_{\text{p}}(f_{\text{p}}, P_p) + \sum_{c \in \{\text{Node} \to \text{Cloud}\}} L_{\text{c}}(f_{\text{c}}, P_{\text{c}}). \tag{3}$$

Communication latency is defined as the need to communicate data between the different computational levels $D_{\text{Node} \to \text{Cloud}}$ on the communication link $C_{\text{Node} \to \text{Cloud}}$, between the layers. From these definitions, the latency for the intelligence partitioning function $\mathfrak{I}(F)$ is derived as in Eq. (3), where L_{P} and L_{C} are the measurements or estimation function for the processing and communication latency, respectively.

Another limiting resource is the node energy consumption, which can also be expressed as battery life or energy harvesting resource. The node energy E_{Node} can be formulated as:

$$E_{\text{Node}} = E_{\text{p}}(f_{\text{Node}}, P_{\text{Node}}) + E_{\text{c}}(D_{\text{Node} \to \text{Cloud}}, C_{\text{Node} \to \text{Cloud}}). \tag{4}$$

E_{P} and E_{C} refer to estimation functions of the energy consumption for processing and communication, respectively. The objective is to find the lowest energy per sample E_{Node} under the constraint of a minimum latency L_{min}.

$$\underset{f(F)}{\overset{E_{\text{Node}}}{\min}} \; s.t. \; L < L_{\max}. \tag{5}$$

The constraints can also be formulated as in Eq. (6) where the aim is to find the lowest latency L under the constraint that the node energy consumption E_{Node} is below the node energy per sample available E_{a}.

$$\overset{L}{\underset{f(F)}{\min}}\ s.t.\ E_{\mathrm{Node}} < E_{\mathrm{a}}. \tag{6}$$

In Sects. 5 and 6, we provide the results and analysis of the effects of intelligence partitioning in three deployment scenarios, with consideration regarding latency in Eq. (5) and energy consumption constraints in Eq. (6). Table 1 presents the resulting configurations of image-processing tasks allocation based on Eq. (2).

3.2 Communication energy

The focus of this paper is on partitioning image-processing tasks between the smart camera node and the cloud, with regard to the inter-effects in processing and communication energy consumption. Hence, the communication has to reflect the impact on the sending node only for a complete picture of the trade-off between different configurations of task partitioning. We simplified the communication model by analysing a two-node system in Fig. 3 with point-to-point communication between the peer nodes. Even though we consider only the sending node for the energy consumption analysis, two nodes are required to describe the correct timing constraints of the communication, as the node will also receive control information from the peer. Furthermore, we assume that the nodes are already connected to each other and the sensor node can send its data, as soon as the image processing requires it. The model is based on ideal communication conditions with no transmission errors. In addition, the interference by other nodes/technologies has been omitted, assuming ideal resource assignment for the transmitting sensor node. To transfer larger amounts of data, subsequent transmissions take place until the required data amount is transferred, depending on the technology-specific aspects.

To model the energy consumed by the communication module or interface of the node, we calculate the required transmission time for the given data amount for each technology based on its physical and medium access layer operation. Higher layers of the communication stack are not considered. This assumes that fragmentation at the network layer is available, if the data to be transferred is larger than a single packet. To ensure reliable data transmission, acknowledgements are used as feedback from the receiver, according to the protocol specification of the given technology. The overhead considered includes all training and synchronisation sequences on the physical layer as well as any overhead due to header information at the MAC layer. The model uses a typical communication cycle according to Fig. 4, which is repeated multiple times, if more than one packet is needed to transfer the given data amount. This model follows the approach in [27] for subsequent packet transmissions. Based on this, we are able to describe how long each transmission stage (e.g. sending tx, receiving rx, or waiting idle) takes for a single transmission of the given data amount. We then derive the overall energy consumption of the communication module $P_{C(\mathrm{tech})}$ of the node based on the time it spends in the different states as well as current consumption taken from the data sheets of corresponding transceiver chips for each technology and the data amount d according to this equation:

$$\begin{aligned} P_{C(\mathrm{tech})}(d) = {} & t_{tx}(d_{rx})P_{tx} + t_{rx}(d_{tx})P_{rx} \\ & + t_{\mathrm{idle}}P_{\mathrm{idle}} + t_{\mathrm{sleep}}P_{\mathrm{sleep}}. \end{aligned} \tag{7}$$

P_{tx}, P_{rx}, P_{idle} and P_{sleep} are the different power consumption levels per state of the given transceiver. t_{tx}, t_{rx}, t_{idle} and t_{sleep} are the protocol-specific durations spent in each state to transfer the data amount d. It should be noted that not all generic states will be used, depending on the actual operational specification of the given technology. The proposed communication model provides an estimation of communication specific energy consumption, required for the task partitioning analysis. Energy consumption and delay estimation results are optimistic compared to real-world implementation, as the systems are considered isolated, and any re-transmission of packets is not included. However, this model provides the fundamentals upon which to analyse the

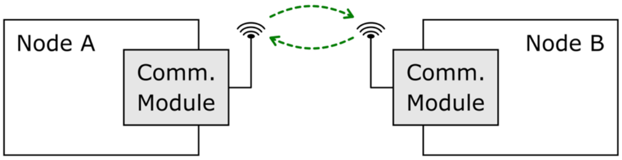


Fig. 3 Two-node representation of the node-cloud communication model

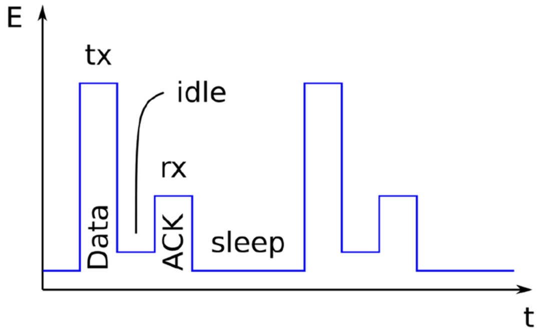


Fig. 4 Activity cycle of data transmission

Table 1 Configurations resulting from intelligence partitioning for the people counting scenario

Config.	t_1	t_2	t_3	t_4	Data size (B)	After data reduction (B)	Reduction type
c_1	N_{HW}	N_{HW}	N_{HW}	N_{SW}	1	0.05	A
c_2	N_{HW}	N_{HW}	N_{HW}	C	1117.5	91	CCITT
c_3	N_{HW}	N_{HW}	N_{SW}	N_{SW}	1	0.05	A
c_4	N_{HW}	N_{HW}	N_{SW}	C	1117.5	91	CCITT
c_5	N_{HW}	N_{HW}	C	C	1117.5	91	CCITT
c_6	N_{HW}	N_{SW}	N_{SW}	N_{SW}	1	0.05	A
c_7	N_{HW}	N_{SW}	N_{SW}	C	1117.5	91	CCITT
c_8	N_{HW}	N_{SW}	C	C	1117.5	91	CCITT
c_9	N_{HW}	C	C	C	8940	3179	PNG
c_{10}	N_{SW}	N_{SW}	N_{SW}	N_{SW}	1	0.05	A
c_{11}	N_{SW}	N_{SW}	N_{SW}	C	1117.5	91	CCITT
c_{12}	N_{SW}	N_{SW}	C	C	1117.5	91	CCITT
c_{13}	N_{SW}	C	C	C	8940	3179	PNG
c_{14}	C	C	C	C	8940	3179	PNG

inter-effects of communication and processing energy consumption regarding the allocation of image processing tasks between the camera node and the cloud.

3.2.1 Compression and data aggregation

In current literature, the addition of image compression techniques is considered a priori as a method to reduce the smart camera node energy consumption, because of the reduction in communication workload. Data rate requirements vary significantly amongst them due to the different configurations of tasks partitioning, resulting from the application of intelligence partitioning. Considering this data rate variation, and the inclusion in the analysis of several communication technologies, we decided to investigate how the inter-effects of processing and communication energy consumption would be affected by image compression techniques. Initially, we implemented lossless greyscale image compression on embedded software. The results motivated us to investigate further, hence we implemented CCITT group 4 binary image compression in both embedded hardware and software. The software implementations are based on the OpenCV library, while the hardware implementation is from the work of Imran et al. [28]. Another method to reduce communication workload is data aggregation. For partitioning configurations where all the processing is done locally in the node and only the counting result is sent to the cloud, data transfers for each frame would be redundant. Instead, we transfer the data in predefined time intervals, without affecting the statistics of people counting, while optimising the communication workload.

4 Communication technologies

There is a wide range of communication technologies that could be suitable for IoT systems. However, the suitability of a technology mainly depends on the constraints of the specific user case [29–32]. In the case of battery-operated smart camera systems, these constraints are energy consumption; relatively high data amount compared to traditional IoT systems with low duty cycle; and real-time performance. The latter two result in high data rate requirements, if nodes exchange raw image data. Despite the need for high data rates, previous approaches on smart camera systems also considered traditional communication for wireless sensor networks such as Bluetooth Low Energy (BLE) or, 802.15.4 with rather low data rates [29, 33]. Therefore, we introduce the following categories for communication technologies.

4.1 Local area network communication

This category contains Bluetooth and WiFi as two technologies that are able to provide higher data rates and are available in various devices other than traditional sensors. The achievable communication range is approximately 100 m, and thus rather short. Out of these, we selected two versions of Bluetooth Low Energy (BLE): version 4.2 with a data rate of 1 Mbps and version 5 with 2 Mbps as well as WiFi according to the IEEE 802.11n standard.

4.2 Cellular communication

This category contains traditional public land mobile networking technologies that were developed for mobile phone communication, which are mainly used for server to user communication, rather than communication between the sensor nodes and the server (e.g. in [18]). In this category, different technologies support a wide range of data rates, which are asymmetric between up- and downlink. Hence, the sensor node has different rates for receiving (downlink) and sending (uplink) data. The exact rate of a user will depend on the current state of the network and the resources that are assigned to the device in question. All technologies have long-range capabilities with ranges over 1 km and mainly require more energy than technologies designed for low-power IoT applications. However, for long-range IoT applications that require a high data rate, employing cellular communication technologies can be the optimal choice. In this study, we consider GPRS, HSPA and LTE Cat4 in this category.

4.3 IoT-specific communication

This category covers both traditional communication for low data rate wireless sensor networks and long-range technologies designed for the IoT context. The IEEE 802.15.4 standard represents the traditional low-rate communication in its conventional specification. BLE can also be a candidate in this category, but we chose to add it to the LAN category as it is also used there and supports higher data rates. Among the recently developed IoT-specific technologies, we consider NB-IoT, an LTE extension for machine-to-machine communication [34], and LoRa, but we omit SigFox from our analysis due to its significantly low data rate for WVSN applications. Furthermore, we added LTE Cat. 1 devices to this group, which were designed as intermediate technology between LTE devices of the previous category and NB-IoT, in terms of low power consumption requirements. Both NB-IoT and LTE Cat1 belong to the group of cellular technologies.

With this selection, we are able to cover the complete data rate of currently available technologies for WVSN communication. Table 2 gives an overview of the considered technologies. It also indicates which transceiver hardware was used to evaluate the energy consumption of the technologies. We based this analysis on the capabilities of suitable embedded transceiver chips that might result in lower data rates than expected for each technology, especially in the case of WiFi.

5 Results

5.1 Design examples

5.1.1 People counting

Applications such as surveillance systems, or environmental monitoring, consist of outdoor deployment of the camera node, with public spaces captured within the field of view. The main complexity of such systems is to provide a robust application despite the abrupt illumination changes throughout the day/night cycle. In addition, in many countries, including Sweden, there are law restrictions prohibiting installation of cameras in public spaces; hence, privacy concerns become a major constraint in the design of the system.

To overcome these complexities, we used a low-resolution thermal sensor that gives a generic temperature profile of the region/object of interest while providing a low-weight system compared to RGB cameras [45]. For the people counting scenario, we created our own dataset from a setup installed in Härnösand, Sweden. The camera sensor used is the FLIR Lepton 3, a long-wave infrared sensor with wavelength 8–14 μm. The analysis is based on the processing of a video of 1:45 hours recorded from a stationary camera, with a frame rate of 9 fps, and frame size 60×149 pixels. The processing platform used for the smart camera is the TE0726-03M Raspberry Pi with a System on Chip module that includes the Xilinx Zynq-7010 FPGA. The

Table 2 Configurations of communication technologies

Category	Technology	Max. data rate		Range	Chip			Ploting order
		Downlink	Uplink		Name	Energy Rx (mA)	Energy Tx (mA)	
LAN	BLE 4.2	1 Mbps		~100m	[35]	6.4	5.3	(1)
	BLE 5	2 Mbps		~100m	[36]	5.4	5.3	(2)
	802.11n	72.2 Mbps		~100m	[37]	61	287	(3)
Cellular	GPRS	80 kbps	40 kbps	> 1 km	[38]	300		(4)
	HSPA	7.2 Mbps	5.76 Mbps	>1 km	[39]	425		(5)
	LTE Cat. 4	150 Mbps	50 Mbps	> 1 km	[40]	610		(6)
IoT	802.15.4	250 kbps		~100m	[41]	5.9	6.1	(7)
	NB-IoT	27.2 kbps	62.5 kbps	> 1 km	[42]	46	220	(8)
	LoRa	37.5 kbps		> 1 km	[43]	13.8	28	(9)
	LTE Cat. 1	10.3 Mbps	5.2 Mbps	> 1 km	[44]	540		(10)

image-processing tasks used in the people counting scenario are listed below:

- Background modelling and subtraction use low-pass IIR filter due to energy efficiency and accuracy in the background model [45].
- Segmentation is based on a global threshold defined experimentally.
- Morphology relies on erosion and dilation operations with a 3×3 mask.
- Detection uses the bounding box method for each foreground element, while tracking relies on Kalman filter.
- Image compression uses CCITT group 4 and PNG compression for the binary and greyscale images, respectively, with more implementation details in Sect. 3.2.1.

Tasks such as background modelling, segmentation, morphology and image compression were implemented in both the programmable logic and the ARM Cortex A9 processor; the remaining were only implemented on embedded software [46]. Software implementation of the image-processing tasks is based on OpenCV libraries. To estimate the energy consumption of the different configurations of task partitioning included in our analysis, we used the Xilinx Power Estimation tool with an error margin of 20% [47] and the post-synthesis hardware description.

5.1.2 Pedestrian and particle detection

Besides the people counting scenario, we also evaluate the effects of intelligence partitioning in the node energy consumption for the smart camera nodes introduced by Imran et al. [8] and Maggiani et al. [9].

The energy consumption results regarding the computation are, respectively, based on the Xilinx Power Estimation tool and PowerPlay Early Power Estimator tool [48] with a 20% error margin. The flowcharts in Fig. 5 show the computational tasks considered for intelligence partitioning in each of the implementation cases analysed.

5.2 Measurement results

The analysis of the overall energy consumption in the smart camera node for the three scenarios is based on the combined results of image-processing task partitioning and communication technologies. The computational tasks in Fig. 5 are allocated among the processing elements in the node and cloud, resulting in several configurations for the people counting scenario as shown in Table 1. In the following analysis, the communication technologies will be referred to according to the plotting order in Table 2, while all the calculations for delays and energy consumption in the smart camera node are frame based.

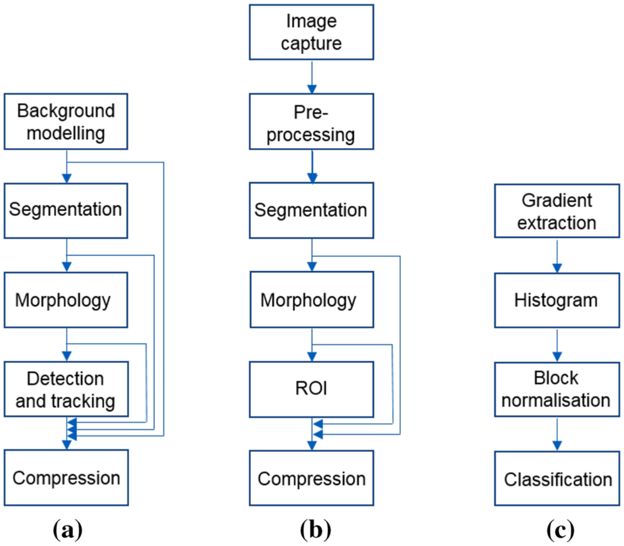


Fig. 5 Flowchart representation of computational tasks for: **a** people counting; **b** particle detection [8]; **c** pedestrian detection [9]

5.2.1 Delay and channel usage

Scenarios such as environmental monitoring, industrial monitoring, or surveillance system are highly dependent on the timing constraint to maintain the performance of the system. Task partitioning configurations result in data rate requirements ranging from 8940 down to 0.5 bytes per frame for the people counting scenario and 964, 608 down to 259 bytes per frame for the other two scenarios. Subsequently, this affects the performance of the communication technologies considered, especially IoT communication technologies, as they have been designed for long duty cycles and low data rate operations. To define the delay constraint for each scenario, we referred to the frame rate of the smart cameras and compared it to the communication delay. We omit from our consideration the processing delay in the smart camera node, as it is in the range of ns to µs. Table 3 summarises the effects of the delay constraints in the ten communication technologies considered, based on the data rate requirements resulting from intelligence partitioning in each scenario.

As expected, the IoT communication group is the most affected by the delay constraint. For the people counting scenario, LoRa supports only transfers of a few bytes per second resulting from full in-node implementation of the image-processing tasks, while NB-IoT and 802.15.4 support data rates up to binary image transfer (compressed and non-compressed, respectively). Instead, for the particle and pedestrian detection cases, the support of IoT technologies is highly restricted due to higher data rate requirements, and a three times shorter delay interval. The LTE Cat. 1 technology is an exception to the group

Table 3 Delay constraints in the communication technologies for data rates resulting from intelligence partitioning

Communication technologies	1	2	3	4	5	6	7	8	9	10	Monthly data rate
People counting											
0.5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	1.23 MB
1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	22.2 MB
75	✓	✓	✓	✓	✓	✓	✓	✓	X	✓	1.6 GB
91	✓	✓	✓	✓	✓	✓	✓	✓	X	✓	1.98 GB
1 117.5	✓	✓	✓	X	✓	✓	✓	X	X	✓	24.3 GB
3 179	✓	✓	✓	X	✓	✓	X	X	X	✓	69.1 GB
8940	✓	✓	✓	X	✓	✓	X	X	X	✓	194 GB
Particle detection [8]											
259	✓	✓	✓	X	✓	✓	✓	X	X	✓	18.8 GB
500	✓	✓	✓	X	✓	✓	✓	X	X	✓	36.2 GB
680	✓	✓	✓	X	✓	✓	✓	X	X	✓	49.24 GB
7 217	X	X	✓	X	✓	✓	X	X	X	✓	522.6 GB
32,000	X	X	✓	X	✓	✓	X	X	X	X	2.26 TB
256,000	X	X	X	X	X	X	X	X	X	X	18.1 TB
Pedestrian detection [9]											
11,264	X	X	✓	X	✓	✓	X	X	X	✓	815.7 GB
119,808	X	X	✓	X	X	✓	X	X	X	X	8.47 TB
964,608	X	X	X	X	X	X	X	X	X	X	68.2 TB

performance, as it meets the delay requirements for data rates up to 11, 264 bytes per frame.

In the cellular communication group, HSPA and LTE Cat. 4 technologies meet delay requirements for data rates up to 119, 808 bytes per frame, while GPRS only supports data rates up to 91 bytes per frame for the people counting scenario. The LAN communication group provides support for all the data rates resulting from intelligence partitioning in the people counting scenario. However, for the particle and pedestrian detection scenarios, Bluetooth Low Energy devices are restricted to only three configurations, with data rates up to 680 bytes per frame, unlike 802.11n technologies that support up to 119, 808 bytes per frame.

One of the three communication technology groups included in our analysis is cellular technologies. In this case, the communication channel is owned by a third party, and its utilisation results in additional monetary costs for the smart camera node. Hence, we need to take such costs into consideration in terms of subscription costs, which subsequently limits the amount of data to be transferred. As a result, cellular technologies would not be an optimal choice for the pedestrian detection scenario with monthly data rates in the range of terabytes. In the remaining two scenarios, the monthly data rate requirements are significantly reduced for task partitioning configurations with image compression, enabling the use of cellular technologies.

5.2.2 Data reduction

The constraints in terms of delay and channel utilisation showed a significant reduction in the number of communication technologies capable of meeting the requirements as the data rate increases. For the people counting and particle detection scenarios, the task partitioning configurations with the most restrictions were the ones without any data reduction method implemented. However, the focus is on exploring the energy efficiency of the smart camera node; hence, we analyse how the implementation of image compression affects the overall node energy consumption for the people counting scenario.

Table 4 shows the resulting energy consumption in the smart camera with and without greyscale image compression for configuration 9 of the task partitioning. The results from configuration 13 and 14 of task partitioning have been omitted; all three showed the same behaviour. Contrary to expectations, the overall energy consumption of the smart camera is higher for the case with image compression. From a comparison of the points with the lowest energy consumption, consisting of BLE 5 communication, the energy consumption is 24 times higher for the scenario with greyscale compression. This shows that processing the image compression algorithm on embedded software increases the processing energy consumption much more than the reduction in communication energy due to data rate reduction.

Table 4 Energy consumption per frame for greyscale compression

Communication technology	Uncompressed (mJ)	Compressed (mJ) Software	Energy variation (ratio)
1	1.6	20.8	13
2	0.8	20.5	25.6
3	1.3	20.7	16
4	–	–	–
5	22.6	29.9	1.3
6	4.6	22.5	4.9
7	–	–	–
8	–	–	–
9	–	–	–
10	31	32.5	1.05

Table 5 Energy consumption per frame for binary image compression

Communication technology	Uncompressed (mJ)	Compressed (mJ)		Energy variation (ratio)	
		Hardware	Software	Hardware	Software
1	0.2	0.04	20	0.2	100
2	0.1	0.03	20	0.3	200
3	0.3	0.2	20	0.67	66.7
4	–	9.4	29	–	–
5	3.2	3.2	23	1	7.2
6	2.3	2.3	22	1	9.6
7	5.1	0.9	21	0.18	4.1
8	–	7.4	27	–	–
9	–	–	–	–	–
10	4.1	2	22	0.49	5.4

Considering the high impact of the processing energy consumption for the compression algorithm, we investigated into hardware/software partitioning, this time for the lossless binary image compression with CCITT G4. In addition, we omitted the frame header from the compressed packets, considering an already acknowledged communication between the node and the cloud. The task partitioning configurations resulting in binary frames are 2, 4, 5, 7, 8, 11 and 12, all showing similar behaviour; hence, the results in Table 5 are only for configuration 2. Similarly to the previous case, the software implementation of the compression algorithm increased the overall energy consumption of the smart camera, with a difference of up to 200 times higher than the case with no compression. Instead, due to the fine-grained use of computational and memory requirements for the hardware implementation, the overall energy consumption is five times lower than for the non-compressed case.

Among the several task partitioning configurations considered in this analysis, configurations 1, 3, 6 and 10 rely on full in-node implementation of the image-processing tasks. For the people counting scenario, this would result in transferring only the counted number to the cloud for further statistical analysis of the data. Hence, we considered the use of data aggregation to avoid redundant information regarding the people count, while optimising the communication energy consumption. Figure 6 shows the energy consumption per frame for all the task partitioning configurations, including those with full in-node implementation with and without data aggregation. The effects of applying data aggregation vary for the different communication groups, with minimal effects in the LAN technologies, and a maximum reduction of 50%. The cellular group shows moderate results with a reduction of 14–29 times energy consumption, while the IoT group has the highest reduction for the cases with NB-IoT and LoRa communication, resulting in a reduction of 33 and 124 times, respectively, compared to the case without data aggregation. This is due to the design of such communication technologies, optimised for low-data rate transfers and long duty cycles.

5.2.3 Node energy consumption

The aim of this paper is to analyse how the overall energy consumption in the smart camera node is affected by processing allocation and communication technology choice. The results in Fig. 6 show the energy consumption per frame for the people counting scenario, based on the combination of task partitioning configurations, communication technologies and data reduction approaches introduced above. We begin the analysis of the results from left to right, with the two data rate groups related to full in-node implementation with and without data aggregation. The variation in energy consumption among the communication technologies is about one order of magnitude, while all the communication technologies support the resulting data rate. This is followed by the data rate groups resulting from binary image compression after and before morphology, respectively. The variation in the overall node energy consumption is of about two orders of magnitude, representing the variations due to hardware/software partitioning of the tasks within the smart camera, but most importantly representing the variation in energy consumption between the different communication technologies. The results show that LAN technologies provide a lower energy consumption compared to the remaining, while LoRa and NB-IoT have been omitted due to the delay constraint. The following data rate groups consist of binary frame transfer, compressed greyscale and non-compressed greyscale. The results show high energy consumption, while the number of communication technologies that support the data rates is reduced. The minimum energy consumption among all the configurations considered is achieved when we implement background modelling, segmentation,

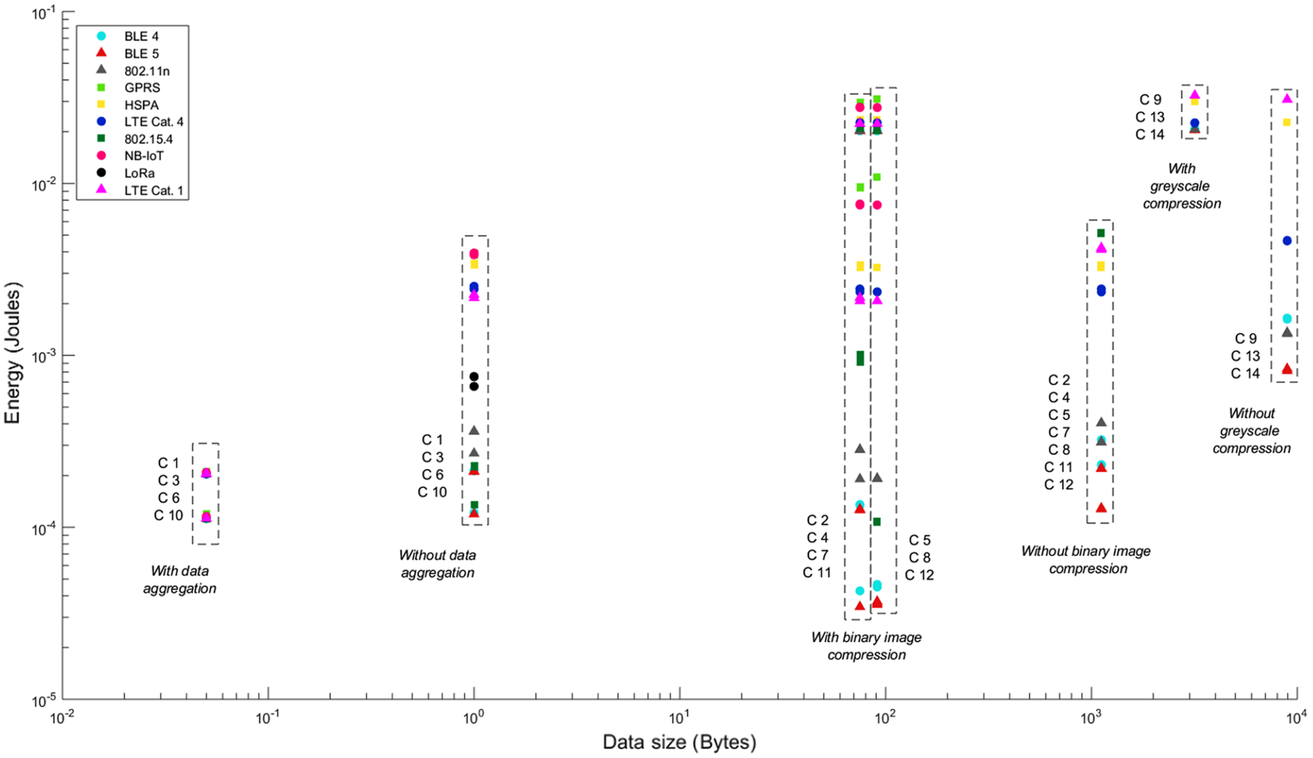
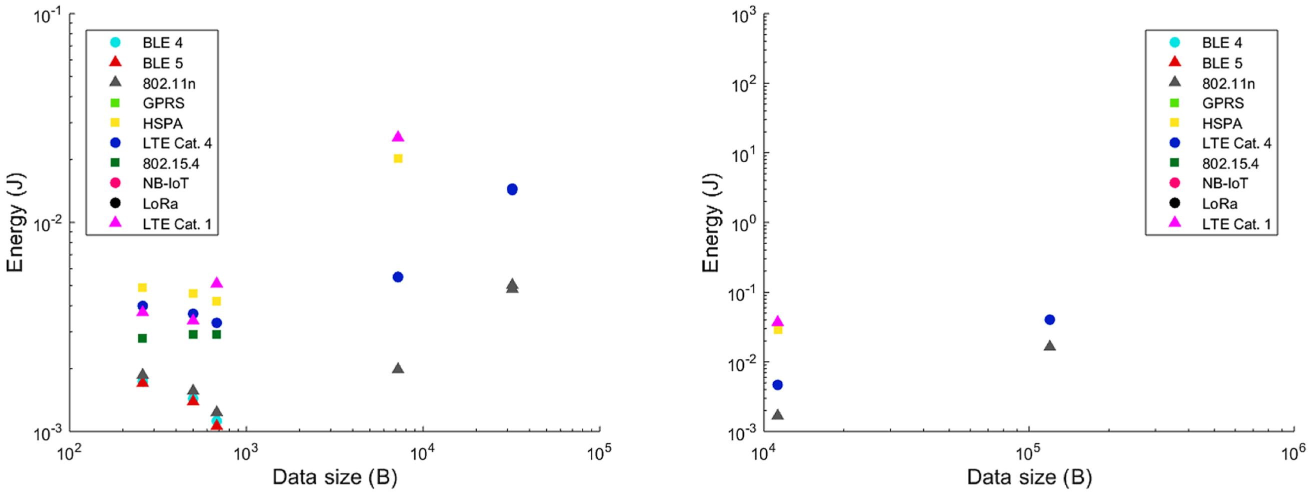


Fig. 6 Energy consumption per frame in the smart camera node resulting from intelligence partitioning in the people counting scenario

morphology and image compression on embedded hardware in the smart camera, supported by BLE 5 communication.

Similarly to the people counting scenario, the results in Fig. 7a for the particle detection scenario show reduction of the node energy consumption as we distribute the processing tasks between the node and the cloud. As the data size increases, the number of communication technologies supporting it decreases, with cases of full cloud processing becoming obsolete due to communication delay. The optimal configuration consists of capturing, pre-processing, segmenting and compressing the images before transferring the data to the cloud with BLE 5 communication.



(a) Particle detection

(b) Pedestrian detection

Fig. 7 Energy consumption per frame in the smart camera node resulting from intelligence partitioning

A contrast to the previously introduced results is the energy consumption of the smart camera node in the pedestrian detection scenario. The data in Fig. 7b shows that the optimal node energy consumption is achieved for full in-node processing, while supported by 802.11n communication. The high data rates resulting from the partitioning configurations of this scenario exclude most of the communication technologies due to the delay constraint.

6 Discussion on intelligence partitioning in WVSNs

In this paper, we have analysed the effects of intelligence partitioning in the energy efficiency of the smart camera node. The combination of three processing environments and ten communication technologies provided a broader perspective of the problem of design space exploration for smart camera architecture. Traditional architectures focus on either full in-node implementation of the processing tasks or remote processing of the captured frames. The results from intelligence partitioning challenge such views, proving that the distribution of the image-processing tasks between the node and the cloud optimises the node energy consumption due to inter-effects of processing and communication energy consumption. The processing energy consumption for the node is generated by power estimation tools with a 20% error margin. This uncertainty from the processing component leaves the results of intelligence partitioning introduced above unchanged, as the difference between intelligence partitioning groups is much higher than 20%.

Performing intelligence partitioning in the three design examples provided configurations with data rates varying from 0.5 to 964,608 bytes per frame, which has a major effect on constraints regarding delay and channel utilisation. The results showed that partitioning configurations with no data reduction methods implemented have a limited choice of communication technologies. Furthermore, the cases with fully remote processing of the data in the particle and pedestrian detection scenarios cannot be supported by any of the communication technologies considered due to the delay and channel utilisation costs. To summarise on the performance of communication technologies, BLE 4 and 5 are the two better performing technologies of the communication technologies analysed for all the configurations in the people counting scenario. Furthermore, for the particle and pedestrian detection scenarios, BLE provides the best energy performance for data below 1 kB, while for the remaining configurations, the 802.11n and LTE Cat. 4 provide better energy efficiency.

The use of data reduction techniques not only affects the choice of communication technologies, but also the overall node energy consumption. The results obtained from the analysis on greyscale and binary image compression for the people counting scenario disproved the general assumption that image compression a priori reduces the node energy consumption. Fine-grained use of computational and memory resources for the hardware implementation of the compression algorithm reduces the processing energy consumption compared to the software implementation. Therefore, hardware/software partitioning of the compression algorithm influences the outcome due to the trade-off between processing and communication energy consumption.

A comparative assessment of the energy consumption results obtained from intelligence partitioning in the three scenarios shows that intelligence partitioning can improve the overall node energy consumption, while satisfying the constraints of real-time performance. However, the results also show that the effects of intelligence partitioning are affected by the relationship between additional processing load and the resulting data rate reduction, which can be the product of image-processing tasks or data reduction techniques. For the people counting and particle detection scenarios, this enables the allocation of the partitioning point in between the image-processing tasks. However, for the pedestrian detection scenario, there is a negative trade-off between the processing and data reduction, resulting in an energy-efficient partitioning at the end of the image-processing pipeline.

7 Conclusion

The results presented show that to improve the energy efficiency of WSNs, we should review preconceptions regarding design space exploration. Energy-efficient distribution of image-processing tasks between the smart camera node and the cloud, as well as the selection of the communication technology, can improve the longevity of battery-operated nodes, compared to full in-node or remote processing scenarios. Therefore, these results can motivate future work that investigates in the introduction of intermediate processing layers between the camera node and the cloud for further node energy efficiency.

Acknowlegements Open access funding provided by Mid Sweden University.

References

1. Martinez, B., Monton, M., Vilajsana, I., Vlajosana, X.: Early scavenger dimensioning in wireless industrial monitoring applications. IEEE Internet Things J. **3**, 170–178 (2016)
2. Lawal, N., O'Nils, M., Imran, M.: Design exploration of a multi-camera dome for sky monitoring. In: Proceedings of the 10th International Conference on Distributed Smart Cameras (2016)
3. Abate, F., Carratu, M., Liguori, C., Paciello, V.: A low cost smart power meter for iot. Meas. J. Int. Meas. Confeder. **136**, 59–66 (2019)
4. Birem, M., Berry, F.: DreamCam: a modular FPGA-based smart camera architecture. J. Syst. Architect. **60**, 519–527 (2014)
5. Imran, M., Shahzad, K., O'Nils, M.: Energy-efficient SRAM FPGA-based wireless vision sensor node: SENTIOF-CAM. IEEE Trans. Circuits Syst. Video Technol. **24**, 2132–2143 (2014)
6. Senouci, B., Charfi, I., Heyrman, B., Dubois, J., Miteran, J.: Fast prototyping of a SoC-based smart-camera: a real-time fall detection case study. J. Real-Time Image Proc. **12**, 649–662 (2016)
7. Abas, K., Porto, C., Obraczka, K.: Wireless smart camera networks for the surveillance of public spaces. Computer **47**, 37–44 (2014)
8. Imran, M., Khursheed, K., Ahmad, N.: Architecture of wireless visual sensor node with region of interest coding. In: IEEE 3rd International Conference on Networked Embedded Systems for Every Application (2012)
9. Maggiani, L., Bourrasset, C., Berry, F.: Bio inspired heterogeneous architecture for real time pedestrian detection applications. J. Real Time Image Process. **47**, 37–44 (2018)
10. Cheng, Y., Chang, R.: Device-free indoor people counting using Wi-Fi channel state information for internet of things. In: IEEE Global Communications Conference (2017)
11. Wu, H., Gao, C., Cui, Y.: Multipoint infrared laser-based detection and tracking for people. Neural Comput. Appl. **29**, 1405–1416 (2018)
12. Kristoffersen, M., Dueholm, J.V., Gade, R., Moeslund, T.: Pedestrian counting with occlusion handling using stereo thermal cameras. Sensors (2016)
13. Bondi, E., Seidenari, L., Bagdanov, A.: Real-time people counting from depth imagery of crowded environments. In: 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (2014)
14. Song, D., Qiao, Y., Corbetta, A.: Depth driven people counting using deep region proposal network (2017)
15. Kuo, J.Y., Fan, G.D., Lai, T.Y.: People counting base on head and shoulder information. In: IEEE International Conference on Knowledge Engineering and Applications (2016)
16. Hengstler, S., Prashanth, D., Fong, S., Aghajan, H.: MeshEye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance. In: Proceedings of the 6th International Conference on Information Processing in Sensor Networks, pp. 360–369 (2007)
17. Abas, K., Obraczka, K., Miller, L.: Solar-powered, wireless smart camera network: an IoT soultion for outdoor video monitoring. J. Comput. Commun. **118**, 217–233 (2018)
18. Fernandez, J., Calavia, L., Baladron, C.: An intelligent surveillance platform for large metropolitan areas with dense sensor deployment. Sensors **13**, 7414–7442 (2013)
19. Berni, J., Galan, R., Cembrano, G., Zarandy, A.: Wi-FLIP: a wireless smart camera based on a focal-plane low-power image processor (2011)
20. Cao, N., Nasir, S., Sen, S., Raychowdhury, A.: Self-optimising iot wireless video sensor node with in-situ data analytics and context-driven energy-aware real-time adaption. IEEE Trans. Circ. Syst. **64**, 2470–2480 (2017)
21. Ju, Q., Sun, G., Li, H., Zhang, Y.: Collaborative in-network processing for internet of battery-less things. IEEE Internet Things J. **6**, 5184–5195 (2019)
22. Kopestenski, I., Van Roy, P.: Achlys : Towards a framework for distributed storage and generic computing applications for wireless IoT edge networks with Lasp on GRiSP. In: 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 875–881 (2019)
23. Sun, J., Zhang, Y., Wu, Z., Zhu, Y., Yin, X., Ding, Z., Wei, Z., Plaza, J., Plaza, A.: An efficient and scalable framework for processing remotely sensed big data in cloud computing environments. IEEE Trans. Geosci. Remote Sens. **57**, 4294–4308 (2019)
24. Magno, M., Tombari, F., Brunelli, D., Di Stefano, L., Benini, L.: Multimodal video analysis on self-powered resource-limited wireless smart camera. IEEE J. Emerg. Select. Topics Circ. Syst. **3**, 223–235 (2013)
25. Krug, S., O'Nils, M.: Modeling and comparison of delay and energy cost of IoT data transfers. IEEE Access J. **7**, 58654–58675 (2019)
26. Morin, E., Maman, M., Guizzetti, R., Duda, A.: Comparison of the device lifetime in wireless networks for the internet of things. IEEE Access **5**, 7097–7114 (2017)
27. Martinez, B., Monton, M., Vilajosana, I., Prades, J.: The power of models: modeling power consumption for IoT devices. IEEE Sens. J. **15**, 5777–5789 (2015)
28. Imran, M., Ahmad, N., O'Nils, M.: Implementation of wireless vision sensor node with lightweight bi-level video coding. IEEE J. Emerg. Select. Topics Circ. Syst. **3**, 198–209 (2013)
29. SanMiguel, J., Cavallaro, A.: Energy consumption models for smart camera networks. IEEE Trans. Circuits Syst. Video Technol. **27**, 2661–2674 (2017)
30. Andreev, S., Galinina, O., Pyattaev, A., Gerasimenko, M., Tirronen, T.: Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap. IEEE Commun. Mag. **53**, 32–40 (2016)
31. Samie, F., Bauer, L., Henkel, J.: IoT technologies for embedded computing: a survey (2016)
32. Li, X., Li, D., Wan, J., Vasilakos, A., Lai, C., Wang, S.: A review of industrial wireless networks in the context of the Industry 4.0. Wireless Netw. **23**, 23–41 (2017)
33. Wang, Y., Velipasalar, S., Casares, M.: Cooperative object tracking and composite event detection with wireless embedded smart cameras. IEEE Trans. Image Process. **19**, 2614–2633 (2010)
34. Alvarino, A., Vajapeyam, M., Xu, H., Wang, X., Blankenship, Y.: An overview of 3GPP enhancements on machine to machine communications. IEEE Commun. Mag. **54**, 14–21 (2016)
35. Texas Instruments. CC256x dual-mode bluetooth controller (2016)
36. Nordic Semiconductor. nRF52832 Product Specification v1.4 (2017)
37. Microchip Technology. ATWINC15x0-MR210xB IEEE 802.11 b/g/n SmartConnect IoT Module (2017)
38. ublox. SARA-G3 series Dual and quad-band GSM/GPRS modules (2017)
39. ublox. SARA-U2 series HSPA modules with 2G fallback (2018)
40. ublox. TOBY-L2 series Multi-mode LTE Cat 4 modules with HSPA+ and/or 2G fallback (2017)
41. Nordic Semiconductor. nRF52840 Product Specification v1.0 (2017)
42. ublox. SARA-N2 Power-optimized NB-IoT (LTE Cat NB1) modules (2017)

43. Semtech. SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver (2015)
44. ubox. LARA-R2 series Size-optimized LTE Cat 1 modules in single and multimode configurations (2017)
45. Shallari, I., Imran, M., ONils, M.: Background modelling, analysis and implementation for thermographic images. In: International Conference on Image Processing Theory, Tools and Applications (IPTA) (2017)
46. Shallari, I., Krug, S., ONils, M.: Architectural evaluation of node—server partitioning for people counting. In: Proceedings of the 12th International Conference on Distributed Smart Cameras (2018)
47. Xilinx. Xilinx Power Estimator User Guide (2018)
48. Intel. Cyclone IV and Cyclone V PowerPlay Early Power Estimator

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Irida Shallari received her B.Sc. degree at the University of Tirana, Albania (2014), and her M.Sc. from the University of Southampton, United Kingdom and Norwegian University of Science and Technology, Norway (2016), in embedded computing systems. She has a Licentiate degree from Mid Sweden University (2019), Sweden, and is currently a PhD student at the same university. Her research focuses on studying the energy efficiency of smart cameras for WVSN applications, through processing and communication inter-effects.

Silvia Krug received the B.Sc. degree in computer science from the University of Applied Sciences Darmstadt, Germany, the M.Sc. degree in computer engineering and the Dr. -Ing. Degree with a focus on communication networks from Technische Universität Ilmenau, Germany in 2013 and 2017, respectively. She is currently working as a Adj. Assistant Professor in the embedded systems design research group at Mid Sweden University, Sweden. Her research interests include communication aspects for wireless sensor systems with a focus on energy-efficiency and design considerations for distributed IoT systems.

Mattias O'Nils received the B.Sc degree from Mid Sweden University, Sweden, and the Licentiate and Ph.D degrees from the Royal Institute of Technology, Sweden, in 1996 and 1999. He is currently a Professor at the Department of Electronics design and leads a research group in embedded systems design at Mid Sweden University. His current research interests include design methods and implementation of embedded systems, focusing on real-time video processing systems.