Smart Camera Networks: advanced applications for next generation Sensor Networks

Claudio Salvadori

ReTiS Lab.

Scuola Superiore Sant'Anna

A thesis submitted for the degree of

Doctor of Philosophy

Tutor: Prof. Marco Di Natale
Supervisor: Dr. Paolo Pagano

October the 29^{th}, 2013

Contents

Introduction

SMART CAMERA NETWORK s (or *SCN*s) represent the natural evolution of centralized computer vision applications towards full distributed systems. Indeed, in SCNs the application logic is not centralized, but spread among network nodes (also called *Smart Camera*, or *SC*): every SCN node has the capability to (i) pre-process images to extract significant features, and (ii) aggregate data to understand the surrounding environment. In such a scenario a strong cooperation among nodes is necessary, as well as the possibility of having pervasive and redundant SCN devices. These main requirements are nowadays addressed by the research community by proposing SCN nodes based on low-complexity, low-power and low-cost devices (see Chap. 1) able to exchange information through wireless communications. In this environment, in Chap. 2, we propose a possible architecture of future SCNs. More in detail, we describe a SCN leveraging *Service Oriented Architecture* (*SOA*) in the the *Internet of Things* (*IoT*) scenario capable to activate, reconfigure and compose a set of computer-vision based services by the control of a set of heterogeneous wireless camera nodes.

In this environment, we propose the realisation of several computer vision algorithm deployed on Smart Camera based on low complexity microcontrollers. In fact, one of the biggest efforts in designing pervasive SCNs is the porting (or the redefinition) of complex PC-based computer vision algorithms to low-complexity and low-memory embedded devices. "Embedded computer vision" usually means processing low resolution images (i.e., Q-VGA or QQ-VGA) or parts of them (i.e., Region of Interests, a.k.a. RoIs). In this direction in Chap. 3, after describing the architecture of a large scale *Intelligent Transportation System* (*ITS*), we take into account two embedded computer vision applications able to understand the scene processing RoIs by the exploitation of the surrounding conditions. Moreover, in Chap. 4 the wireless network noise impact is evaluated on low resolution raw images and on the output of a background subtraction based compressor.

However, these directions do not permit to port more complex but more efficient techniques: in Chap. 5 we describe how we have approximated the Gaussian Mixture Model algorithm for low-complexity and low-memory micro-controllers with the lack of Floating Point Unit. More in detail, we have defined a new computer arithmetic able to map floating point numbers into a sub-integer representation and achieve comparable qualitative results with a heavy reduction of processing time and memory footprint.

Even though this work achieves real-time performance over standard microcontroller-based SC, such a systems are not capable to handle more complex algorithms (e.g., machine learning techniques). Consequently in Chap. 6 we detail the architecture of the next generation SC based

on a micro-controller (in charge of managing the network communications an the high-level remote configuration requests) and a re-configurable FPGA controlled by a SoftCore, capable to perform the computational intensive tasks. More in detail the FPGA internal architecture is designed to fit with the above described IoT constraints, in which each node of a global network shares a set of resources among nodes. In this direction, a given hardware block written in a generic Hardware Description Language (HDL) can be abstracted as network resource and made available to other network nodes. This approach permits to see each hardware block (e.g., computer vision algorithm) as an abstract functional module, following the model-based design reasoning. Any computer vision pipeline can be easily created by connecting these blocks, thus reaching a higher level of abstraction. Although such abstract vision is similar to a model-based design approach, the proposed architecture is not affected by additional computation overheads, because each block is generated independently and the processing time strictly depends on its own optimisation level.

CHAPTER 1

From sensor network toward camera network

THE application domains where *Wireless Sensor Network*s (*WSN*s) are being deployed are becoming more and more complex, including monitoring and actuation in industrial, commercial, and ordinary social environments. Wireless devices are expected to be generic enough to run different software, designed for a target application, eventually released after network deployment. In this respect multimedia technologies and notably computer vision is felt as promising to handle versatile applications making use of on-board detection and classification capabilities for transmitting reports in one case or compressing, encoding and streaming images in the other case.

*Smart Camera Network*s (*SCN*s, a.k.a. Wireless Multimedia Sensor Networks or *WMSN*s) applications have been proposed and prototypes deployed in simple scenarios where a single node is expected either to analyze and report about a scene, or to compress and stream out videos to a remote control station. Thus, *Smart Camera*s (or simply *SC*s, see in [**RW08**]) combine video sensing, processing, and communication on a single embedded platform. Their on-board computation and communication infrastructure advances a shift in the processing paradigm of novel computer vision systems. The close collocation of sensing and processing in a SC transforms the traditional camera into a smart sensor. For multi-camera applications, image processing migrates from central workstations to the distributed embedded sensors. This distributed computing approach helps to reduce the communication load within the network of cameras and to increase the reliability and scalability of the multi-camera application.

As an example we report about the IPERMOB project [**ipe09**] where a SCN has been installed at the Pisa International Airport to monitor and control urban mobility in real-time. In the data collection layer of the system a set of embedded camera nodes are used to detect the status of parking spaces and the instantaneous flows by means of low-complexity computer vision techniques [**MMN+11**]. Some examples are shown in Fig. 1.

The whole camera network is managed by a custom middleware developed as ad-hoc solution, Scantraffic [**AAP+12**] exploiting the real-time features of the GTS mechanism in IEEE802.15.4 networks. The middleware permits to configure the monitoring Region Of Interest (RoI) of a camera view (see Fig. 2), as well as the period of a time-driven monitoring service in which only reports about occupancy are transmitted.

From the hardware side, several research projects produced prototypes of embedded vision platforms which may be deployed to build a SCN. Among the first experiences, Panoptes project [**FKFB05**] aimed at developing a scalable architecture for video sensor networking applications. The key features of Panoptes sensor are a relatively low-power and high-quality video capturing

FIGURE 1. Examples of the object detection techniques used to monitor the parking space occupancy (top) and the instantaneous traffic flow in a public road (bottom).

device, a prioritizing buffer management algorithm to save power and a bit-mapping algorithm for the efficient querying and retrieval of video data. Nevertheless the size of the sensor, its power consumption, its relatively high computational power and storage capabilities makes Panoptes sensor more akin to smart high-level cameras than to untethered low-power low-fidelity sensors.

The Cyclops project [**RBI$^+$05**] provided another representative SCN node. The camera nodes is equipped with a low-performance ATmega128 8-bit RISC micro-controller. From the storage memory point of view the system is very constrained, with 128 KB of FLASH program memory and only 4 KB of SRAM data memory. The CMOS sensor supports three image formats of 8-bit monochrome, 24-bit RGB color, and 16-bit YCbCr color at CIF resolution (352 x 288). In the Cyclops board, the camera module contains a complete image processing pipeline for

FIGURE 2. A snapshot of the remote service for defining the RoI for the SCN devices.

performing demosaicing, image size scaling, color correction, tone correction and color space conversion.

In the MeshEye project [**HPFA07**], an energy-efficient SC architecture was designed, mainly with intelligent surveillance as target application. MeshEye mote has an interesting special vision system based on a stereo configuration of two low-resolution low-power cameras, coupled with a high resolution color camera. In particular, the stereo-vision system continuously determines position, range, and size of moving objects entering its fields of view. This information triggers the color camera to acquire the high-resolution image sub-window containing the object of interest, which can then be efficiently processed.

Another interesting example of low-complexity and low-memory embedded vision system is represented by the CMUcam4 [**Agy**], developed at the Carnegie Mellon University. More precisely, the CMUcam4 is the fourth generation of the CMUcam series, which has been specially-designed to provide an open-source, flexible and easy development platform with robotics and surveillance as target applications. The hardware platform is more powerful with respect to its predecessors and may be used to equip low-complexity and low-memory embedded system with simple vision capabilities, so as to obtain smart sensors.

Moreover, the CITRIC platform [**CAB$^+$08**] integrates in one device a camera sensor, a CPU (with frequency scalable up to 624 MHz), a 16 MB FLASH memory and a 64 MB RAM. Such a device, once equipped with a standard RF transceiver, is suitable for the development of SCN. The design of the CITRIC system allows to perform moderate image processing task in network, that is along the nodes of the network. In this way, there are less stringent issues

regarding transmission bandwidth than with respect to centralized solutions. Such results have been illustrated by 3 sample applications, namely (i) image compression, (ii) object tracking by means of back ground subtraction and (iii) self localization of the camera nodes in the network.

In recent years programmable hardware such as FPGAs has been frequently used as processing platform for SCs, permitting to solve the bottlenecks of low-level and pixel-wise algorithms, exploiting their parallelisation capabilities. For example, in [**DBSM07**] a generic FPGA-based SC is described. The FPGA is used to implement several standard modules (e.g., interface to the image sensor, memory interface, Firewire interface) along with a programmable control module and a flexible number of processing elements. The processing elements can be interconnected arbitrarily according to the algorithms data-flow.

1.1. Selected boards

In this thesis, the selected Smart Camera devices to evaluate the performance are essentially three: the *SeedEye* [**Scu11**] board, *Raspberry Pi* [**Ras11**] and the *Terasic DE0-nano* [**Ter12**] board.

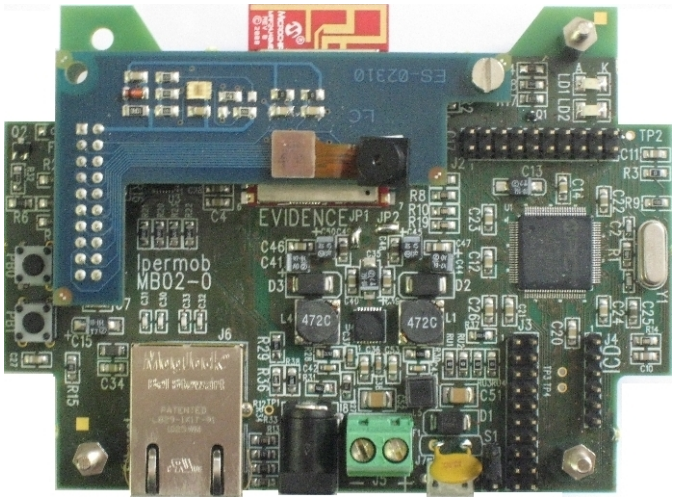


FIGURE 3. The SEED-EYE board.

1.1.1. The SeedEye board. The *SeedEye* board is an advanced multimedia Smart Camera Networks node based on PIC32MX795F512L by Microchip (see also [**Mic08**]) that embeds: a wireless transceiver compliant with the IEEE802.15.4 standard, an IEEE802.3 (Ethernet) interface, a USB interface and a CMOS camera. The HV7131GP camera (see [**SN09**]) can be programmed to acquire images at various resolutions (from QQ-VGA up to VGA) and supports the *windowing* functionality, that permits to acquire also a line. This camera can handle a frame rate of 30 fps at VGA resolution, but this value depends directly on the number of transmitted bytes: acquiring a line of 640 bytes it is possible to reach a rate of more than 200fps. Low power operation is an essential requirement for such mobile sensors and the specific hardware specifications allow the board operating at 0.45W with only 0.30W consumed by the micro-controller.

TABLE 1. SeedEYE board: standard operation times using integer and floating point representation

Precision	Sum $[s]$	Subtraction $[s]$	Multiplication $[s]$	Division $[s]$
uint32_t	3.31	3.43	4.11	9.80
double	41.98	50.08	44.97	130.58
Ratio T_{double}/T_{uint32_t}	∼13	∼15	∼11	∼13

The PIC32 micro-controller is characterized by a computational capability of 80MHz, and an internal RAM of 128KBytes. It lacks of FPU and consequently any floating point data are handled by software libraries. In Tab. 1 the processing time for executing a series of 32 millions arithmetic operations (+, -, *, /) on both floating point and integer data is shown: for every arithmetic operation, the ratio between the processing times of on floating point data (namely *float*) and on integer data T_{double}/T_{uint32_t} is at least 11. This ratio hints at the potential gain of our integer-based implementation.

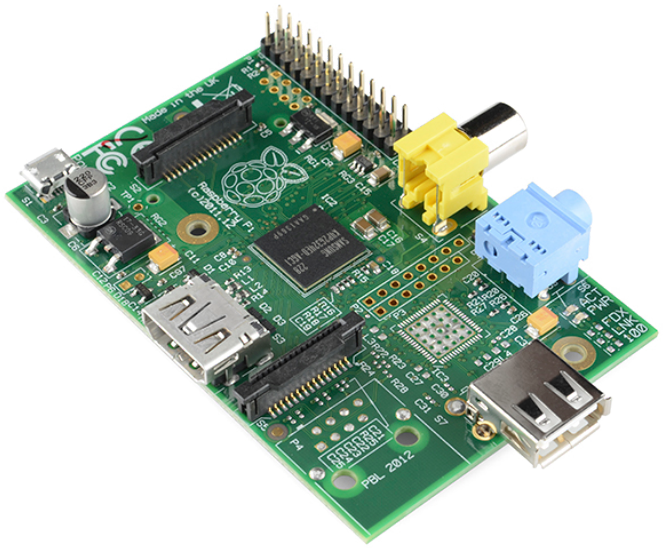


FIGURE 4. The Raspberry Pi board.

1.1.2. The Raspberry Pi board. The *Raspberry Pi* board is based on a Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S characterized by a computational capability of 700 MHz processor and 256 MBytes of RAM. An SD card is used for booting and long-term storage. Although this system lacks FPU, floating point arithmetics are supported

TABLE 2. Raspberry Pi board: standard operation times using integer and floating point representation

Precision	Sum $[s]$	Subtraction $[s]$	Multiplication $[s]$	Division $[s]$
uint32_t	0.289404	0.298648	0.474425	0.761673
double	0.859734	0.854660	0.906426	2.012707
Ratio T_{double}/T_{uint32_t}	∼3	∼3	∼2	∼2.5

by an optimised library which is part of the operation system *Raspbian* ([**Deb12**]): integer operations are about 2-3 times faster than floating point operations (see Tab. 2), therefore an integer-based implementation may still be valuable.

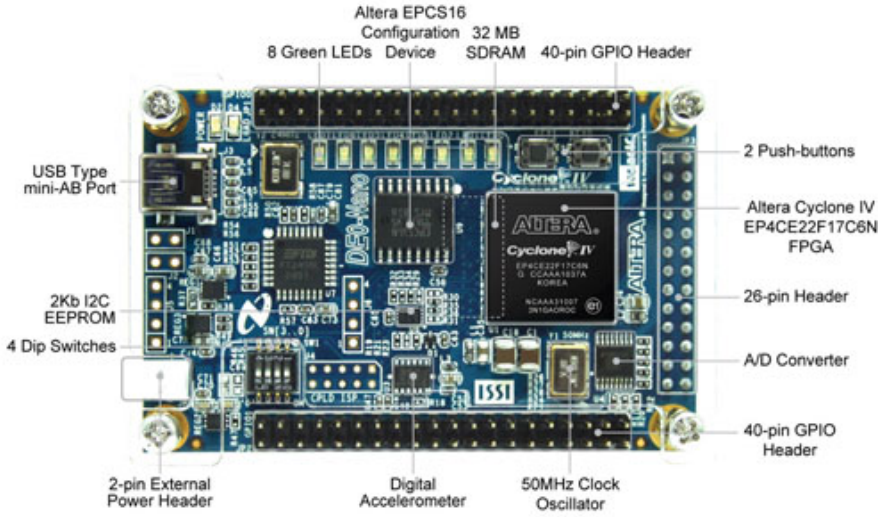


FIGURE 5. The Terasic DE0-nano board.

1.1.3. Terasic DE0-nano. On the FPGA side we have considered the *Terasic DE0-nano* development board, which embeds an Altera Cyclone IV FPGA, with 22000 Logic Elements (LE), 600 Kbits of on-board memory and 144 9x9 bits DSP modules. In such FPGA we included an Altera NIOS-II SoftCore as data flow controller embedded into the elaboration logic by using an internal bus, called Avalon Memory Mapped (or Avalon-MM). This board will be considered in Chap. 6 as the selected hardware to design the next-generation SC.

CHAPTER 2

Architectures of Smart Camera Networks

ALL the works proposed in pervasive SCNs literature (e.g., [**NC11**]) describe system able to solve specific monitoring problems (i.e., tracking, video-streaming, etc.) without defining the network architecture of a SCN. In this chapter we detail which is our point of view in this topic. Thus, we propose an *Internet of Things* (or IoT) based architecture where every single computational element (e.g., computer vision algorithm) installed inside the SC (e.g., inside an optimised computer vision library) can be abstracted as network resource and made available to other network nodes (see also [**PSM$^+$12**]). This approach permits to see each computer vision algorithm as an abstract functional module, following the model-based design reasoning. Then, any computer vision pipeline can be easily created by connecting these blocks, thus reaching a higher level of abstraction. In this environment, the necessity of an optimised processing library deployed into each SC has to be realised, and each function has to be abstracted as a functional block into the network. Then, each SC is capable to extract the features needed by the chosen algorithm and share their values with the other nodes.

Thus, our target is that of designing a distributed visual application where all these issues are addressed profiting (and complementing) all existing efforts in developing new hardware, networking, and middleware solutions suited or applicable to the vision case study. More specifically we will design a distributed visual application *"as a service"* following the approaches of the IoT and *Service Oriented Architectures* (SOA); the camera nodes (together with gateways and high-end PCs) will be part of a *Middleware of Things* (MoT), set in between of the collection layer and the final users (or third party systems). To better detail our proposal, in Fig. 1 the system is represented in a layered structure consisting of "Collection and pre-processing", "Network", "Middleware" and "Applications" functional blocks.

Final user applications making use of middleware APIs can offer 2d or 3d-rendering, locate the center of mass of the moving target within a map, propagate an alarm to other systems, etc. This diversified set of applications is not expected to be deployed together with the system but after the collection layer is set up.

Suppose that we want to compose camera views of a moving object keeping the target always in front; the middleware will be in charge of selecting the appropriate views at a certain rate, aggregating this information and eventually rendering it in a visual shape: we call this application "Virtual Camera". The latter has certain degrees of freedom, implemented as configuration parameters, like the frame rate and the required quality level of visualization. Virtual computation and storage space can eventually be provided through third party clouds seamlessly and transparently interoperated at middleware layer. Similar considerations are applicable to

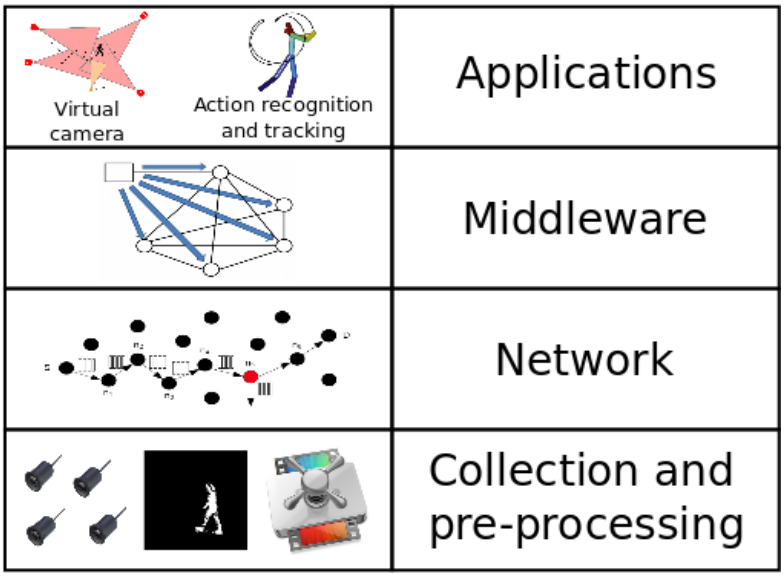


FIGURE 1. A layered architecture for an information system based on SCN.

action and behavioral recognition as different applications making use of the same middleware functional blocks (exported through APIs).

Decoupling the application from the middleware services permits to improve the capabilities of back-end components (e.g. in terms of MAC layer performances and error resiliency capabilities) without the need of redesigning the application logic.

2.1. Smart Cameras

A major issue in deploying SCN is related to the manufacture of the device node. Following the traditional approach in WSN, a camera is nothing else than a peripheral feeding the main core with vector-like data; the main core is therefore in charge of processing the raw data either for compressing and streaming them out to a remote station or to locally extract high-level information (e.g. classifiers) and produce a report.

2.1.1. Hardware examples. To perform complex on-board imaging operations on low power micro-controllers it is necessary to relieve the main processor of the image acquisition and pre-processing burden. The approaches presented in the following are based either on the design of a new camera with specific enhanced on-chip capabilities or on the use of COTS cameras, coupled with more powerful computing platforms.

The authors of [**FBCGCG11**] manufactured a new camera chip-set (the FLIP-Q device), performing (i) scale space and pyramid generation, (ii) multi-resolution imaging, (iii) energy-based representation; they coupled it with a popular wireless device, the Imote2 board,in order to set up a low-power, vision enabled WSN node (the Wi-FLIP device [**FBCGLC$^+$11**]), tested on the field for the early detection of fire in forests.

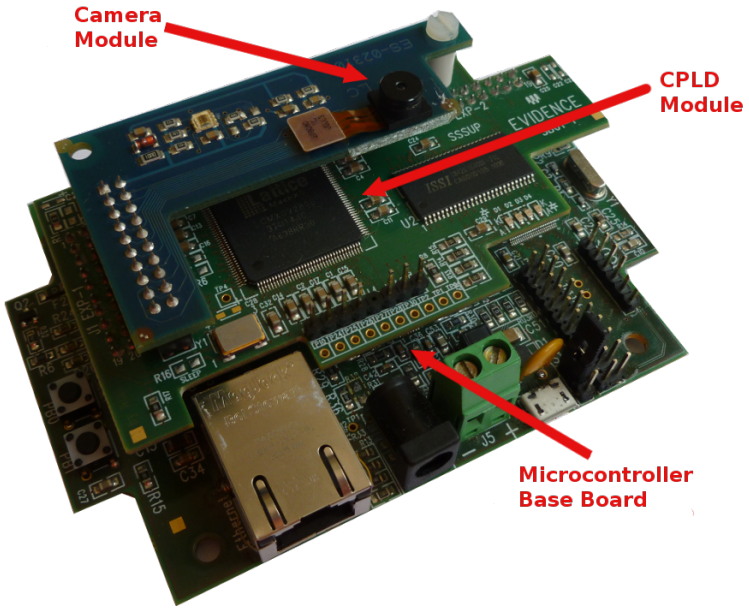


FIGURE 2. The SeedEye board (with CPLD).

Another approach, resulting from the IPERMOB project, is that of decoupling [**ipe**] raw data acquisition, storage, and pre-processing from vision algorithms: the first set of tasks can be implemented on dedicated devices like CPLD or FPGA directly connected to cameras whereas the main core will be used for high-level image analysis (see Fig. 2). A reduced version of the board is also available with no CPLD, suited for more simple applications and reducing a lot energy consumption.

In the first case, a strong effort is dedicated to the CMOS chip development whereas in the other one, the camera is an ordinary consumer electronics component, more and more including side functionality (like that of sensitiveness to Near Infra-Red) previously featuring products of higher quality only.

2.1.2. Software for computer vision in embedded systems. A good approach for designing a general purpose WMSN is that of decoupling back-end functionality (i.e. drivers, operating system, and networking primitives) from front-end running jobs which can be downloaded (together with the needed libraries) from remote upon the occurrence of certain events.

In this direction we are developing an optimized computer vision "C" library (called μCV) in order to port or to adapt computer vision algorithms to simple microcontrollers lacking memory, computational power, and specific resources like FPU (Floating Point Unit). For example in [**SMP$^+$12**] we have optimised/approximated the Gaussian Mixture Modeling (GMM) to match microcontroller-based boards specifications. In this direction, considering a QQ-VGA

image (i.e. 160x120 pixels), we have reduced by a factor of 12 the memory footprint of the algorithm (considering the case of a mixture of two Gaussians), maintaining the performance level compatible with the original realization in terms of frame-rate (in the order of 25 fps minimum).

Moreover we have investigated on image compression protocols in order to permit video streaming over WSN keeping: (i) simple and low-complexity and low-memory SC for data preprocessing; (ii) more complex concentrator nodes for the computational intensive operations; (iii) low bandwidth and noisy networks. In [**PGS+11**] we simulated a realistic noisy scenario to quote the streaming performances of QQ-VGA images on IEEE802.15.4 networks. The state-of-the-art compression algorithms (JPEG and JPEG2000), though ensuring a small compression ratio, are based on computational intensive transformations (i.e. DCT or wavelet transforms) and produce high-correlated outputs not resilient to bit-flips in datagrams. We specifically evaluated the effects of the bit-flips introduced by the network noise on the raw pixels and prospected techniques for both data protection (i.e. BCH codes) and data concealments.

Having demonstrated that streaming uncorrelated raw pixels has to be preferred to holistic compression techniques because of the noisy nature of the wireless communication channel, we developed a specific compressor [**SPP+12**,**SPM+13**] able to extract and transmit the regions of highest information content in each frame by discarding the still pixels and consequently reducing the datagram size. This solution is featured by low complexity (the proposed algorithm is based mostly on a simple image segmentation) and error resiliency (the innate characteristics of the raw images are kept) although it underperforms state-of-the-art compressors in terms of compression ratio (and network occupancy).

Having developed these techniques through a set of libraries, it is possible to implement a full customed computer vision application (e.g. streaming, detecting, or classifying mobile objects) depending on the specific WSN context.

2.2. Networks

Considering SCN architectures proposed in recent works, nobody has attempted any integration with the IoT. In [**MA11**] an interesting reference architecture for SCNs is presented while mapping usual SC functionality into multimedia capabilities: low-end sensor nodes have data acquisition and processing capabilities, while the network sink, tagged as *multimedia processing hub*, has the task of aggregating visual information.

Next generation SCNs must be considered as part of the Internet world, thus making possible to address each single camera node worldwide in order to have access to its data and services. To this end, the IPv6 standard targeted to *Low power Wireless Personal Area Networks* (*6LoWPAN*) [**KMS07**,**KKGB12**] represents a solution to be considered jointly with the *Constrained Application Protocol* (*CoAP*) [**She10**], an HTTP-like protocol especially designed for constrained devices, presently in draft status along its standardization process at IETF ([**SHBB13**]).

As 6LoWPAN permits to have camera nodes worldwide addressable in large-scale distributed systems, CoAP permits to create embedded web services running on 6LoWPAN camera nodes, thus extending the transaction style of the web to the IoT.

The major power of the Web is to allow the flowing of information through different devices. This result is achieved using a series of common components: a language (the hypertext transfer protocol, HTTP), a scheme for resource identification (the *Uniform Resource Identification*, *URI*) and some content descriptions (the *Internet Media Type*, *MIME*). The Web is generally accepted as a mean for applications running on a plethora of devices to interoperate, sharing data and resources; in such respect CoAP is responding to the primary objective of supporting a "web of things", by providing a set of services useful for a variety of final user applications. As HTTP, CoAP is a working solution for supporting *machine-to-machine* (*M2M*) communication, in the context of *Web Services* and *Semantic Web Services* [**Fen11**].

6LoWPAN and CoAP let ad-hoc networks be interoperable with full-fledged devices by means of Web services. As the high-end ones, these low-end devices are expected to be fully virtualized in respect of their resources, the running application, and their capability of building up high level information from raw data via cooperation patterns.

In the case of SCN, each camera node can expose its own resources, such as a single pixel, a whole image, or a certain feature extracted after on-board processing, that can be used by other camera nodes (or high-end devices) involved in the distributed vision algorithms.

Considering the mapping in [**MA11**] and taking into account the usual topology of 6LoWPAN, it is possible to define a new reference architecture (see Fig. 3) for distributed vision in the IoT. The architecture is therefore composed by the following types of nodes:

- *Camera Sensor* (CS): it is a low-end node able to acquire and process video frames. CS is a node with reduced computational capabilities and strong energy constraints. In the 6LoWPAN namespace, CS is a *Host* node;
- *Multimedia Processing Node* (MPN): it is the equivalent of the multimedia processing hub proposed. The MPN is a node with higher computational capabilities with respect

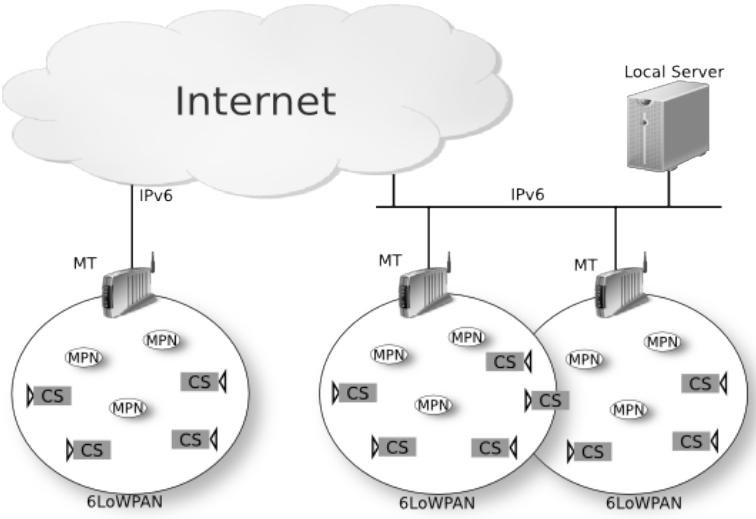


FIGURE 3. SCN architecture based on 6LoWPAN standards.

to CS, and able to aggregate visual information. While originally the MPN node was just considered as a sink in charge of forwarding data to network gateways, in our vision MPN is a *6LoWPAN Router*, thus in charge of organizing and managing peer-to-peer communications;

- *Multimedia Translator* (MT): it is a high-end device able to reconstruct complex visual events coming from the camera sensors before forwarding them to the IPv6 network. For example, a video stream service exposed by a CS featured by a low-complexity compression standard can be converted into a much more common video compression format. The 6LoWPAN network requirements for the MT are those of a *Border Router* node.

2.3. Middleware of Things for Distributed Computer Vision

In the previous sections we already discussed the capabilities belonging to the embedded devices, network architectures and communication protocols.

The main issue is that heterogeneous camera nodes should be enabled to participate to distributed vision applications although defined and dynamically assigned to sensing peripherals when the system is on-line. Sample applications are tracking, streaming, face recognition, fire detection, human action recognition, etc...

In other words another capability of a modular multi-node system is that of activating, reconfiguring, composing a set of services. In a SOA these capabilities, usually demanded to a middleware, can accomplish final user applications. The "middleware services" can be classified in three families depending whether they are oriented to the spreading of information (data-oriented services), to the generation of alarms (event-oriented services), or to the installation, versioning, or re-configuration of firmware and software (code-centered services).

In our distributed system, sensing peripherals, data sets, CPU time in high-end devices, available network bit-rate, firmware and software of embedded system, etc., are virtualized as "middleware resources". These resources are owned by the "things" which are intended to interoperate through the middleware (as pictorially depicted in Fig. 4). The latter can also rely on third party systems like "clouds" which can offer storage and computing resources if required by the running application.

The middleware is therefore required to:

- manage the data sets produced by camera nodes and aggregate them following selected algorithms (for instance composing or complementing views);
- manage the events produced by camera nodes;
- distribute (or activate) the code to those nodes appointed to most effectively handle the event;
- reconfigure the running application in order to retrieve rougher or more detailed information about the event (e.g. zooming in and out);
- flexibly reconfigure the service: e.g., switching from streaming to tracking, adding tracking as another service while keeping the streaming on, etc.;

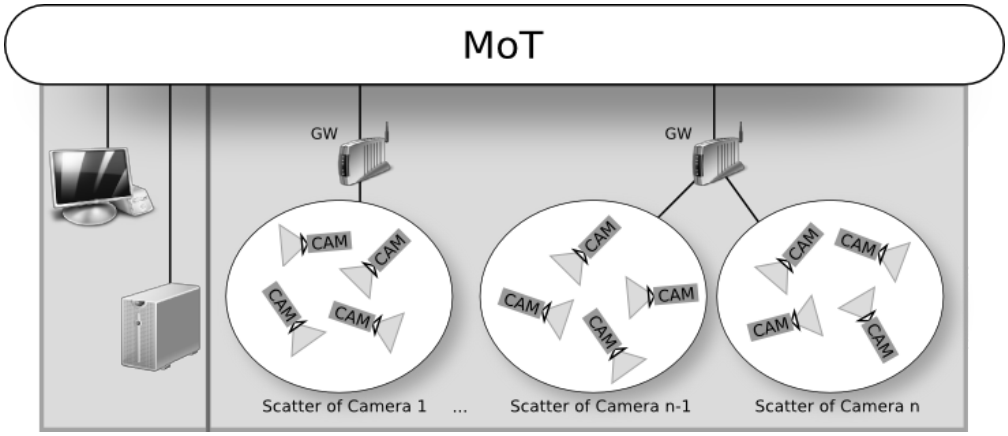


FIGURE 4. SCN architecture including a Middleware of Things.

- allowing for system fragmentation (keeping different services active in different subgroups of "things").

We already developed a prototype middleware handling these services for ordinary applications. We do not describe the implementation details in this context since they are extensively discussed in [**A. 12**]. The real challenge is to port this prototype to distributed computer vision where algorithms for extracting high level information and generate events accordingly is not trivial. We will discuss in the following sections a case study deeply motivating such a development effort followed by some high-level considerations.

2.3.1. A case study for the smart cities. Suppose an Intelligent Transport System encompasses a set of camera nodes involved in accident detection over a large area (see Fig. 5). Whenever an accident occurs, the middleware will be in charge of filtering out all data streams not related to the occurred accident; the software in the cameras will be refreshed or reconfigured in order to provide to final users the needed detail (for instance a 3d snapshot of the event, the covered area, the number of vehicles involved in the event, etc.). Other services will be interested in complementary information instead: what is happening far from the event, how the event is biasing the ordinary traffic flow, and which contingency plan (like reversing the direction of a one-way road or switching the light of a semaphore) can be implemented to reduce the effects at a larger scope.

The middleware will be in charge of scheduling both applications at the same time allocating the needed resources at hardware and network levels.

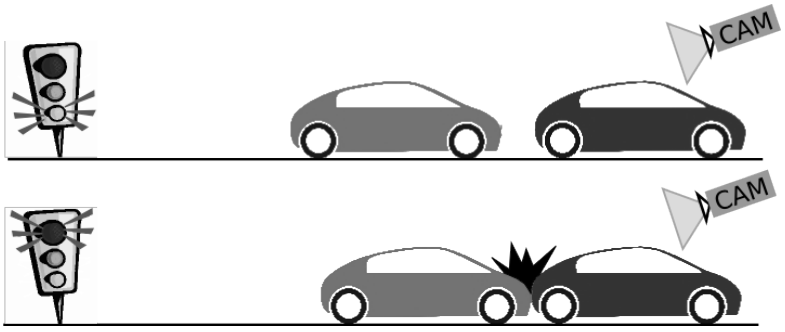


FIGURE 5. A case study for Intelligent Transport System.

CHAPTER 3

ITS advanced applications

INTELLIGENT TRANSPORT SYSTEM s (*ITS*s) apply Information and Communications Technology (ICT) to transport and offer an answer to the question of how to reduce the negative impact of transport on the environment and society: by integrating computers, electronics, satellites and sensors in the transport systems they can make every transport mode (road, rail, air and water) more efficient, safe and energy saving. ITS can play an increasingly important role in solving problems affecting transport systems; a large number of ITS applications are running today in many domains:

- Smart spaces for information points (e.g. based on public wireless access zones);
- Info-mobility (e.g. parking-availability monitor, variable-message signs, instantaneous or off-line traffic surveys and congestion control);
- Restricted access zones (e.g. London and Stockholm congestion charge systems);
- Freight logistics and tracking;
- Public transportation (including integrated ticketing, multi-modal and inter-modal options);
- Green vehicles (e.g. availability of recharging spots, allowance of free circulation disposing off congestion avoidance restrictions etc.);
- Transport offer/demand modelling.

In this scenario the main idea of an ITS system is that of providing drivers security and smart route selection by means of a set of collected traffic related data (possibly in real-time), fast data processing and a useful end-user communication. In few words, an ITS system is effective when the collection layer is wide enough (pervasiveness) and high quality information reaches the potentially interested users shortly enough to allow them other options. Although fast technical development contributes to the appearance of a high number of mature applications, uncoordinated deployment fails in 'providing geographical continuity of ITS services throughout the Union and at its external borders'; in this new directive [**Par10**] for the deployment of ITS, four priority areas have been defined as follows:

- Optimal use of road, traffic and travel data;
- Continuity of traffic and freight management ITS services;
- ITS road safety and security applications;
- Linking the vehicle with the transport infrastructure.

In this situation in Sec. 3.1 we propose our idea on the definition of an ITS architecture (see also [**PPAS13**]) eligible to scale to large surfaces, permitting full access to mobility data in raw

format, easily upgradable to accommodate new final user applications and services. In order to fulfil all the above mentioned requirements, we propose the design and implementation of a layered modular architecture permits to keep functionally separated sensors and vehicles from the TLC and calculus infrastructures, and final user applications.

From the proposed architecture in this chapter we focus on the sensor layer and consequently on the aspects of local processing, considering a Smart Camera Networks for monitoring applications [**MMN$^+$11**, **MMP$^+$10**]. Techniques to extract compact information from the acquired images are studied with a twofold objective: reduce the amount of data that has to be transferred by the camera node, thus limiting the network communication delay; have a light-weight efficient multi-tiers implementation, thus limiting the computation delay. In this direction we propose two different techniques to monitor the parking lot occupancy (Sec. 3.2) and car flows (Sec. 3.3) as use cases of the above mentioned sensor layer. More in detail in Sec. 3.2.1 a hierarchical organization is used to combine two different techniques at different levels: local data mining produces partial decisions, while the final decision is taken at an aggregation point. Two main approaches are considered: change detection and machine learning. The first, based on simple background subtraction, is generally lighter and faster, though less accurate. The second, based on a cascade of classifiers, is more effective, but requires more resources. The two can be combined in a hierarchical manner in a two tier architecture. In the lower tier, nodes with limited power execute the change detection algorithm and produce a waking signal for the upper layer. This signal triggers the elaboration of more powerful nodes running machine learning one. Moreover in Sec. 3.2 a more complex background subtraction based algorithm is proposed in order to be deployed into low processing capability but able fulfill the accuracy requirements of the above mentioned multi-tiers approach (see also [**SPGP12**]).

Finally in Sec. 3.3 a vehicles counter application (as described in [**CLZ$^+$09**, **CSP$^+$10**, **CSP$^+$12**]) is detailed with the aim of creating an effective low-cost solution to be used in next generation ITS. In such systems the lack of resources problem is addressed by processing a smaller amount of data, while taking benefit from surrounding conditions: we first define a generic theory targeted to low-end devices, called *Linesensor theory*, which aims at reducing the amount of data to be processed by leveraging on the temporal redundancy of the movement in the acquired images. Then, we present a dedicated lightweight background modeling algorithm to enhance the robustness of the algorithm on both illumination and stable image geometry changes.

3.1. Reference architecture for an EU-wide ITS.

In this section we propose the future architecture for ITS, eligible to scale to large surfaces, permitting full access to mobility data in raw format, easily upgradable to accommodate new final user applications and services. Ergonomic arguments require to consider as ITS actors the following set of basic profiles: public user (e.g. municipal authority, road managers), private user (e.g. drivers, web visitors), and system administrator. The design and implementation of a layered modular architecture permits to keep functionally separated sensors and vehicles from the TLC and calculus infrastructures, and final user applications.

Profiting of the technical results let available by collaborative research projects, following the standardization of ITS architecture, adopting the rational of the Future Internet we must design an ITS responding to abstract virtues, notably those related to openness and interoperability issues.

All in all, to avoid the pitfalls of application specific solutions, which do not respond to the requirements of an EU-wide infrastructure, a reference architecture must acquire a general consensus from academia, SDO, industries, and commercial stakeholders.

Generally speaking this consensus is related to road-side and in-vehicle sensors, networking, data exchange, storage and sharing, and distributed analytical processing (on-line and offline) on traffic related information.

IPv6 must be appointed as enabling technology for supporting large-scale ITS because it provides specific solutions for connecting the infrastructure with the smart vehicles.

Through the standardization of a common layered structure for ITS, the EU-directive for data management, surface coverage and continuity, etc. will be fulfilled by a plurality of actors / developers each focusing on a subset of layers while resorting to standard inter-layer protocols for interoperating with other, complementary ITS systems.

The best candidate for representing the state-of-the-art of ITS architecture is given by *Communications Access for Land Mobiles* (*CALM*) [**ISO10**]. In the CALM architecture each node is expected to host an ITS station (i.e. Vehicle Station, see in [**PPP$^+$12**], Roadside Station, Central Station and Personal Station) and implement the network protocol stack [**ETS10**] as shown in Fig. 1.

The standard considers a relevant set of technologies at Access Layer ranging from IR to GPS, Cellular, cabled, etc. in order to fit the communication requirements for the vehicle, roadside, central, and personal instances of the ITS Station.

The IPv6 technology plays a role at the Networking layer for non-safety critical applications where message latency is not considered an issue; the IPv6 compliance and the mandatory set of services (which an ITS station must implement) are considered in the working groups of ETSI TC ITS and ISO TC 204 WG16 and will be regulated in [**ISO12**].

Although revolutionary in itself the layered architecture promoted by SDOs is not explicitly considering two main non-functional issues of the target ITS deployment:

(1) the ITS infrastructure is expected to take opportunity of ad-hoc networks in order to span large (possibly EU-wide) surfaces; in the Internet of Things approach, sensors are expected to be directly connected to Information Systems and host simple applications previously envisioned to full-fledged devices only;
(2) the ITS infrastructure does not explicitly consider distributed calculus; the calculus is confined at the application layer in control rooms to respond to the requirements of (few) computational intensive services (like those extracting behavioral traffic models from large data sets [**BDVZ10**]).

Having in mind the requirement of large scale infrastructures, the optimum architecture should permit to connect heterogeneous devices, through a diversified set of access technologies

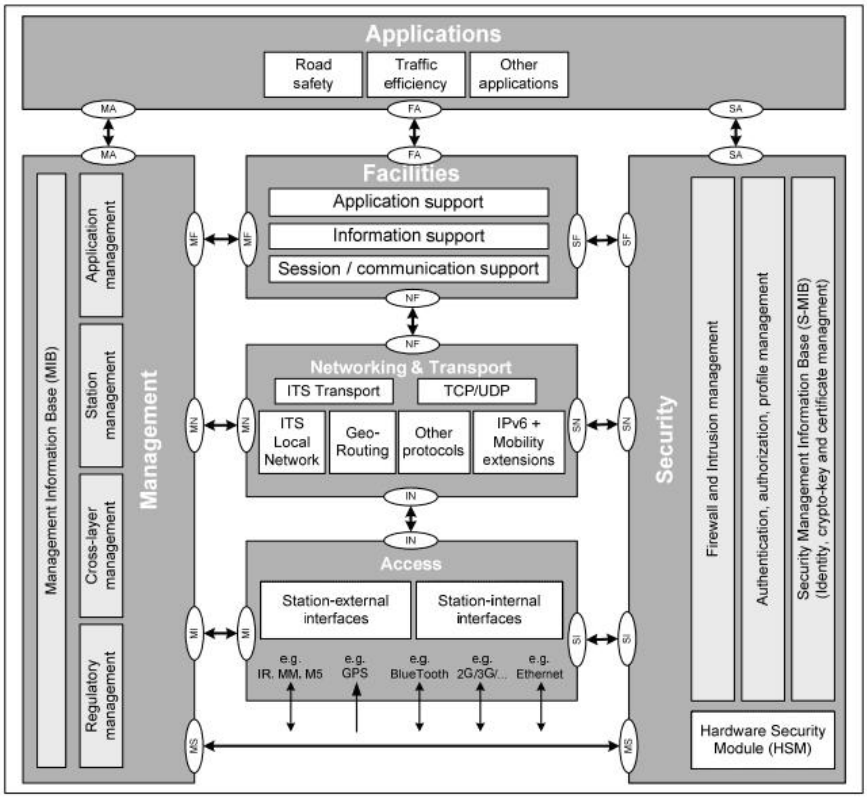


FIGURE 1. The ITS station layered architecture.

complying with the IPv6 suite, eventually making use of distributed calculus to respond to the computational and storage demand associated with the ITS application.

In our proposal we want to leverage the standardization efforts promoted by ITU for the *Next Generation Networks* (*NGN*) aiming at providing Telecommunication Services to users in a reference scenario where "service-related functions are independent of the underlying transport-related technologies".

NGN distinguishes the functionality related to the "Data Collection Function" from those related to the "Transport" and "Service" strata; the management function spans a vertical module and the final user function is connected via dedicated interfaces. A joint activity for designing next generation ITS according to NGN rational [**ITU04**, **ITU06**, **ITU10**] would be beneficial; a one-to-one mapping of a modular ITS design into an NGN-like architecture is shown in Fig. 2.

A key task is that of abstracting and generalizing the data collection tier including the network stack of the sensor nodes and maintaining transparent IP connectivity for data exchange and sharing. This ITS architecture permits that:

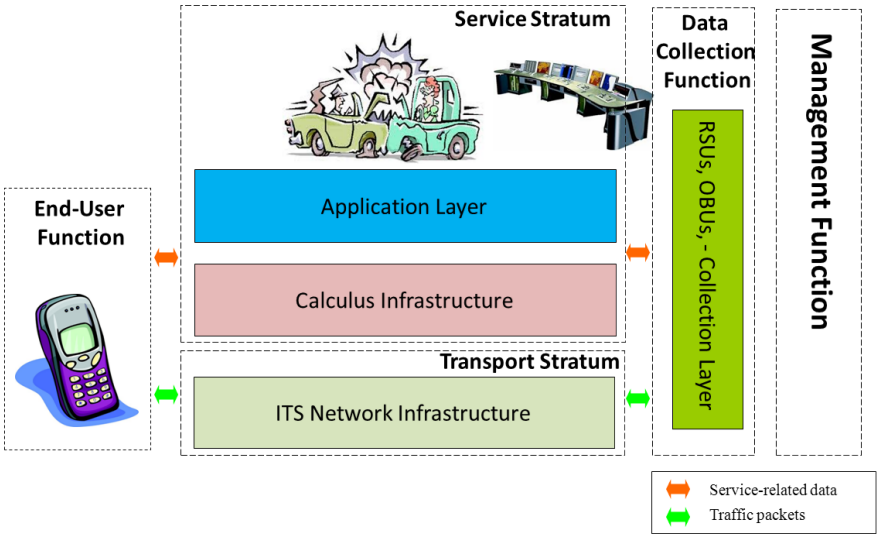


FIGURE 2. ITS functionality into NGN. In dashed boxes the NGN strata.

- new components can be easily integrated without the need of a full coverage specification at the time of pilot set-up's; thus the system can be extended, upgraded, modified during its lifetime without the need of major patching activities and legacy interventions;
- an open competition among the stakeholders can be undertaken considering their contribution to novel products and services for advanced vehicle safety, infotainment, access control, and navigation assistance.
- there is no application specificity: a variety of final users interact with the system following their profiles;
- data are shared in raw format and manipulation to extract mobility models and traffic surveys can be obtained by simulators and mining tools (also non-commercial packages like [**sum**] [**mon**]). The NGN-like architecture will embed the calculus infrastructure into the service stratum;
- as it happened for mobile communications (e.g. look at the development of the 2G and 3G standard families), NGN-like architectures naturally fits open, unrestricted, scalable, and toll-free road networks like those accounting to almost 80% of the national road networks;
- independent "stand-alone ITS" applications migrate to an integrated "system of systems" capable of offering data-oriented services to the external world. "Interoperability" is prominently important in public utility scenarios like the so-called "smart cities": the adoption of popular and standardized communication protocols permits to seamlessly inter-operate ITS with other systems.

In what follows, a detailed description of the enabling technologies for the service and transport strata will be given; moreover the frontier solutions for deploying a large scale sensing stratum will be specifically discussed.

3.1.1. The service stratum.

A web-based application layer. Some excellent papers have shown up in recent years promoting the idea that ITS must evolve to cooperative systems; thereby the stakeholders, profiled by means of Virtual Organizations, establish an "agile" and "easy-to-be-integrated process space" exploiting the ITS potential.

In [**ACm05**] Service Orientation is appointed as the strategic approach to tackle with the complexity growth in innovative business services and in [**OACM10**] such a technology is applied to the ITS domain, enabling the so-called "Collaboration-Oriented Road Mobility Infrastructure (CORMI)". We keep this solution as the reference architecture for the service layer: indeed, it permits to extend and scale the ITS offer in relation with Travel Information, Traffic Management, and Freight & Logistic services.

The novelty would stay in establishing a seamless process (i.e. a transaction) linking the user (by means of his portable and vehicular equipment) to the service provider reducing the need of protocol-to-protocol adaptation stages. SOA would therefore enable the ITS galaxy as a part of the Internet.

Data repositories, thin clients, and remote processes. Although there is an evident need for coherent integration (data formats and communication protocols), sometimes it happens that the off-line management of mobility data is left to the initiative of local (regional) authorities and third-party consultants.

For instance, the selection of the input data sample and the analysis process to extract the target mobility indicators is generally not regulated, resulting in the impossibility to increase knowledge by combining information coming from different sources or independent studies.

SDO and public authorities should be promoting an open and effective way to deploy transportation-specific off-line applications; a special attention must be devoted to privacy policies, so that any appointed technology should enable for user profiling, secure access, data and process hosting.

A nodal issue is related to the way the information is built up and consumed by clients; high-end information on traffic events and cumulated statistics can either make use of fresh data (made available by detectors on the fly) or being fed by indexed information stored in remote repositories. Yet, the processing tasks can require (distributed) computational power eventually served by a (distributed) batch system [**BDVZ10**].

Hereby we propose to use the grid technology to establish a data repository ("Data Grid") and to host the corresponding processes ("Grid Services"); we will show how this technology responds to the functional and non-functional requirements of building up a large scale ITS.

Grid computing is concerned with the sharing and coordinated use of diverse resources in distributed "virtual organizations". The dynamic and multi-institutional nature of these environments introduces challenging security issues that demand new technical approaches. The Open Grid Services Architecture [**WGK+03**] has addressed the security issues raised from resource sharing techniques and actual implementations exist in popular open source tool-kits like Globus.

The proposal of abstracting a networked set of ITSs as a semantic grid is pictorially depicted in Fig. 3 and theoretically discussed in [**WC08**].

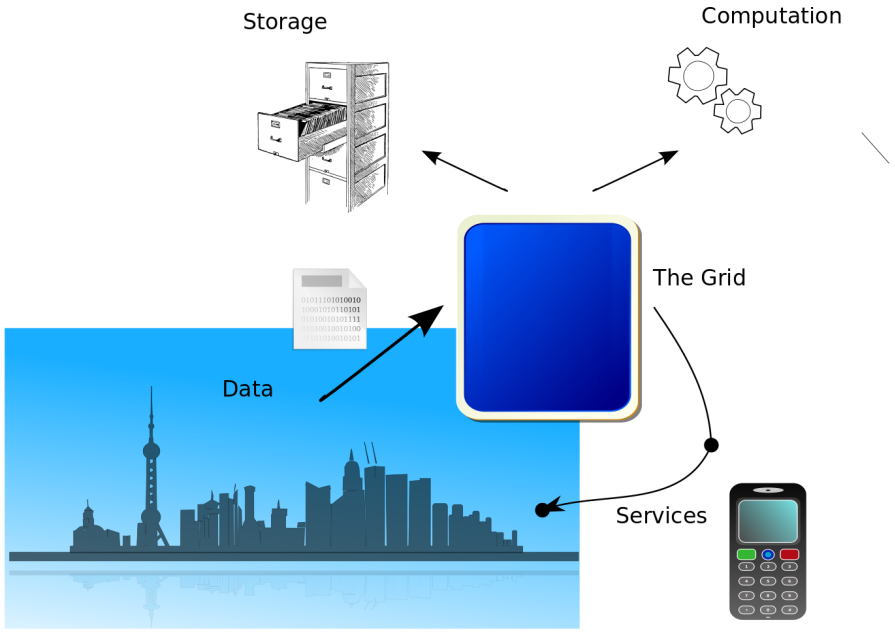


FIGURE 3. Grid-based data and information flow in ITS.

Processing data sets via the grid will permit a certain level of reliability in extracting information:

- only authorized users (grouped in virtual organizations) will access the grid;
- only certified data sets will populate the repositories;
- only certified processes can access the data.

Thin clients running on the web, on-board units, and wireless devices transparently access the grid without showing the process details; on-line services will be delivered making use of fresh (almost local) information in a timely fashion whereas off-line (planning) processes will require additional resources (e.g. disk space, computing time, etc.).

The fact that grid services inherit from Web services permits to fully interoperate the Computational Layer with the Application Layer designed following the Service Oriented Architecture paradigm. In Fig. 4, two Use-Cases will detail about grid-based services.

(a): A user belongs to the Virtual Organization of ¡¡traffic modelists¿¿;
- He decides to run a set of batch processes to correlate vehicle density (i.e. congestion) in zone A with the global pollution in the whole metropolis;
- He wants to make use of certified data sets and software logic;
- He does not want to deal with the physical location of storage pools and computing power got from physical machines.

(b): A user belongs to the Virtual Organization of ¡¡road operators¿¿;

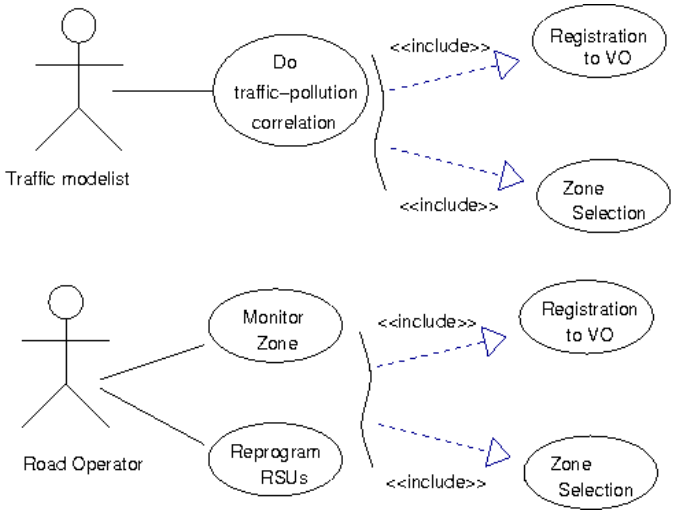


FIGURE 4. Use Cases of a grid-based application for ITS.

- He decides to adapt the ¡¡monitor¿¿ services in zone-1 at run time by defining which outputs are to be retrieved, where to store the data, the data collection radius, and the target transport modes;
- After authentication, he dynamically reconfigures the remote nodes and look into selected metrics.

Research and Development in Wireless Sensor Networks already looked at the semantics of a "Sensor Grid"; moreover some technical realization of a grid-based software stack has been published [**TkTB05**]. In [**BCD+11**] the use-case of an instrumentation grid in the context of the ITS is presented.

3.1.2. The transport stratum: connectivity through backhauling and mobile bridging. The pervasiveness of an EU-wide-scale ITS is met whenever a set of IP-enabled compliant sensor nodes establishes a fully meshed topology; of course Metropolitan Area Networks (MAN) are necessary to tunnel data between disconnected clusters of device nodes; wireless broadband protocols (i.e. 3G, LTE, WiMAX, etc.) are considered to provide location transparent access to the Internet whenever the wired medium (i.e. copper or fiber) is not deployed at the Physical layer (see Fig. 5).

According to this approach, gateways will be adopted for backhauling a set of independent data streams coming from the collection layer; Gateways offer Service Access Points with the Internet through the data transport segment (realized through the MAN).

Today Smart Phones offer seamless IP-based connectivity using different wireless wide band technologies. Future Mobile Nodes (MNs) will be especially designed to communicate with both road-side networks serving as data publishers, and the data transport segment serving as gateways; with respect to the latter, MNs will bridge with eventual isolated "Ad-hoc" segments maximizing the network coverage. Wired large scale infrastructures together with wireless wide

FIGURE 5. A network topology for a distributed ITS.

band technologies and a new generation of road-side devices realize an (easily scalable) instance of the Future Internet, permitting to inter-operate different, heterogeneous, physically distributed ITS at the European-scale.

This infrastructure will complement VANETs to optimize the ITS networking function. In fact messages originating from vehicles and targeted to other vehicles will follow the path going through the infrastructure if a gateway is available, otherwise they will be routed within the VANET until either the recipient or another gateway is met.

3.1.3. The data collection function. In the IPv6 approach, sensor devices and vehicular on-board units are considered IP nodes so that they participate in building and maintaining routes at the same level; the nomadic nature and the fast dynamics let the routes (involving vehicles) be valid for a short time interval so that pervasive infrastructures will play an essential role for robustness and reliability in point-to-point communication.

As it will be discussed in the remainder of this section, low-cost technologies mitigate the infrastructure cost per meter-squared so that road operators can afford the expenses for setting up large-scale ITS.

A new design for embedded systems. As previously discussed, it is rather difficult to scale ITS to a mid-sized city footprint making use of full-fledged expensive monitoring road-side devices; for instance accurate vehicle classification turns to be of little interest if a network of such devices is not sustainable for the public agencies managing mobility.

The target must be that of developing "scalable and pervasive" road-side and on-board units capable to cover a large ratio of the sensitive environment on one hand, and to reach a high market penetration rate in vehicle equipments on the other hand; in both cases effective technologies (pointing to low-cost consumer H/W components and public licensed software) have proven their effectiveness in pilot set-up's (see for instance IPERMOB [**ipe09**] and SMARTSANTANDER [**HMM13**]).

Sensing peripherals, embedded in smart phones (like the GPS receiver and the accelerometers) or grouped in Wireless Sensor Networks (WSNs), are fundamental building-blocks of the ITS collection layer; these nodes must be seamlessly and transparently integrated into the ITS despite of their physical and logical heterogeneity.

On one hand WSN software is still carried out in a rather primitive fashion, by building software directly atop the operating system and by relying on the individual, hard-earned programming skills; on the other hand much work has been carried on by the Internet community (e.g. the Internet Engineering Task Force) to bring abstract programming and the Internet Protocol Suite (including web services) to embedded and other constrained devices [**PKGZ08**, **YD09**, **LGA+05**, **OEL+06**].

In the Internet of Things approach the device nodes are addressable, and their "resources" may represent sensors, actuators, combinations of values or other information; devices send messages to change and query resources on other devices using GET and SET semantics imported from high-end network protocols; devices can send notifications about changed resource values to devices that have subscribed to receive notification about changes; a device can also publish or be queried about its resources.

State-of-the-art implementations for the IPv6 network stack on WSN exist thus permitting to operate the collection layer as a pure mesh of "discoverable" resources whose actual readings can be retrieved on-demand, aggregated with others to build up compound "events", and migrated to remote storage.

HTTP is a widely used protocol for Web transactions and it is oriented to virtual resource management through URL mapping. The concept of resource in HTTP perfectly matches event notion in embedded devices, but HTTP can exhaust memory and processing in embedded systems.

The usual implementation of IP services on constrained devices (see for example [**KMS07**]) is done on top of the IEEE802.15.4 Physical and MAC layers, allowing for ad-hoc routing, and permitting Point-to-Point communication in complex topologies; the IP enabled WSN, the vehicular Network and the smart mobile units establish the ITS sensing stratum.

The WSN, following the directives by the Internet Engineering Task Force (IETF) about the OSI layers 3 and above, therefore enables the so-called Constrained RESTful Environment (CoRE), a framework for the manipulation of simple resources on constrained networks [**cor**].

CoAP [**SHBB13**] is an HTTP adaptation for resource constraint devices, it follows REST architecture and allows event communication and device control (see Fig. 6) [**GMBA+11**].

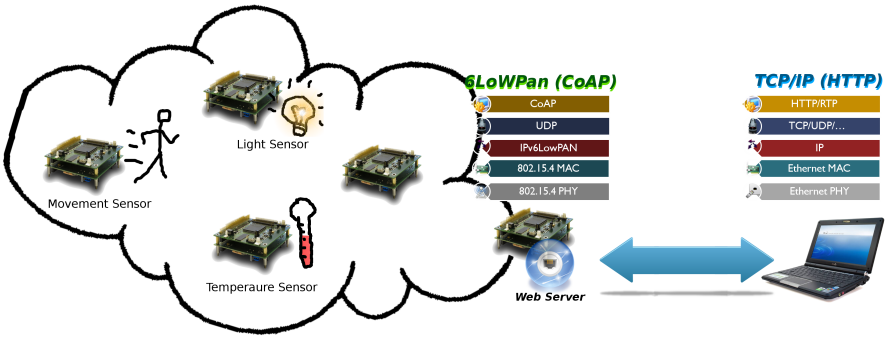


FIGURE 6. A realization of a Web-based transaction on WSN.

It is desirable to operate the ad-hoc segment as a distributed dynamic (smart) environment where new devices can join or leave the network in a plug 'n' play fashion; while on-line, the devices collaborate to build information up, following the semantics of their resource set.

Smart objects for the ITS infrastructure. A general interest in "Internet Connected Objects" (as part of the Future Internet) is considered in the R&D work programmes carried on by the European Commission both at the mid-term (within FP7) and in the long term (within Horizon 2020). Recently approved projects (see CALIPSO as an example) are tackling with the development of 6LoWPAN-based "novel methods to attain very low power consumption, thereby providing both interoperability and long lifetimes" in resource constrained devices.

Nonetheless nobody has tried a specific realization of 6LoWPAN in the ITS context: leveraging the standardization efforts carried on by ETSI and ISO to deploy generic ITS stations we promote our idea of implementing some ITS facilities and applications in 6LoWPAN devices.

This approach will be reflected in specializing the ETSI architecture to deal with subset of non safety-critical applications delivered by WSN as part of the infrastructure. The ITS station architecture is extendable profiting of the open-ness in the "Networking & Transport" layer where a specific module for supporting "Other Protocols" is explicitly considered. All in all (as depicted in Fig. 7), the protocol stack for a 6LowPAN-enabled ITS station will consist in (i) a specialization of the medium access and physical layer technologies (compliant to IEEE802.15.4); in (ii) the porting 6LowPAN protocol suite to fit the "Other Protocols" Layer-3 requirements; and in (iii) a selection of non safety-critical services to be implemented at Facilities and Application layers.

3.2. Data mining for the parking lots monitoring.

Data mining of visual information involve particular problems in computer vision, such as change detection in image sequences, object detection, object recognition, tracking, and image fusion for multi view analysis.

For each of these problems, a vast literature exists with several approaches and solutions, see e.g. [**RAAKR05**] for a survey of change detection algorithms. However, most of them are not suitable for SCNs, due to their high complexity demanding large memory and heavy

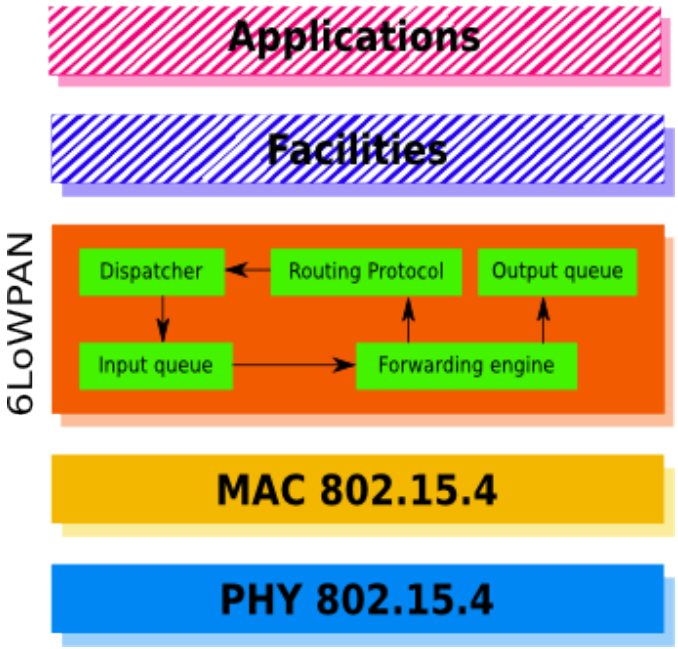


FIGURE 7. Realization of a 6LoWPAN-enabled ITS station.

computations. Nevertheless, some attempts to employ non-trivial image analysis methods to SCNs have been done. For example, [**YGM08**] presents a SCN able to support the query of a set of images in order to search for a specific object in the scene. To achieve this goal, the system uses a representation of the object given by the Scale Invariant Feature Transform (SIFT) descriptors [**Low04**]. SIFT descriptors are known to support robust identification of objects even among cluttered background and under partial occlusion situations, since the descriptors are invariant to scale, orientation, affine distortion and partially invariant to illumination changes. In other words, using SIFT descriptors allows retrieving the object of interest from the scene, no matter at which scale it is imaged.

Interesting computer vision algorithms are also provided on the CMUcam3 vision system (see in [**RGGN07**]). Besides basic image processing filters (such as convolutions), methods for real-time tracking of blobs on the base either of color homogeneity or frame differencing are available. A customizable face detector is also included. Such detector is based on a simplified implementation of Viola Jones detector [**VJ04**], enhanced with some heuristics to further reduce the computational burden. For example, the detector does not search for faces in the regions of the image exhibiting low variance.

SCs have limited FOVs and can only perceive a portion of a scene from a single view point. In addition, monocular vision totally lacks 3D information. To mine the entire scene and to deal with occlusions, it is natural to consider the multi-view capabilities of SCNs.

Due to bandwidth and efficiency considerations, however, images cannot be routinely shared on the network, so that no dense computation of 3D properties (like disparity maps and depth)

can be made. Nevertheless, the static geometrical entities observed in the scene may be suitably codified during the setup of the acquisition system. In addition, specially designed references may be introduced in the scene for obtaining an initial calibration of the views acquired by each camera, thus permitting to find geometrical correspondences among regions or points of interest seen by different nodes. To this end, a coordinator node, aware of the results of such calibration step, may be considered, so as to translate events from the image coordinates to physical world coordinates. Such approach may produce more robust results as well as a richer description of the scene.

3.2.1. A multi-tiers image mining approach. The first approach proposed is based on a hierarchical processing architecture. Essentially, three types of processing can be identified

- change detection: fast low-level algorithm that produces a preliminary information, i.e. occurrence of a change in the parking slot, which is used to trigger the processing of higher levels;
- object detection: slow and complex mid-level algorithm that produces a refined information, i.e. partial decision, about the occupancy status of the parking slot;
- decision making: highest level algorithm that aggregates the information of the lower layers, both the change detection and the object detection; such information are provided by more than one node, i.e. multiple views.

A common hypothesis for the multiple views aggregation process works on the basis of a given knowledge base of information, relative to the approximate possible positions, or Regions of Interest, where the objects can be detected. Under this assumption, the decision making algorithm can be "partially" implemented on the single node. For example, the same node performing the low- and mid-level processing can produce a partial decision on the level of occupancy of the parking slot. The decision-making node can use its knowledge about the network deployment to aggregate the nodes' partial decisions: e.g. weighted according to the nodes' view of the detected objects. In the following sections, some details about the low-level processing algorithms are given.

Change Detection. The task of identifying changes, and in particular regions in the image under analysis, taken at different times is a widely diffused topic encompassing very different disciplines [**RAAKR05**]. In this context, the goal is the identification of one or more regions of pixels, relative to a change in the scene, and occurring within a sensible and a priori known area. In order to perform fast, efficient, and enough reliable methods for this change detection, the generally adopted low-level methods are based on both frame differencing, and statistical methods. This assumption can be either the quick and immediate answer to a simple problems, or a preliminary synthesis for a deeper and more effective higher level analysis. The general model used is mainly based on background subtraction, where a single frame, acquired in a controlled situation, is stored (BG-image) and thereafter used as the zero-level, for different types of comparison in the actual processing.

The first methods are very low-level ones and are thought in such a way that it can be possible and feasible to implement them also at firmware level. Examples of these are be:

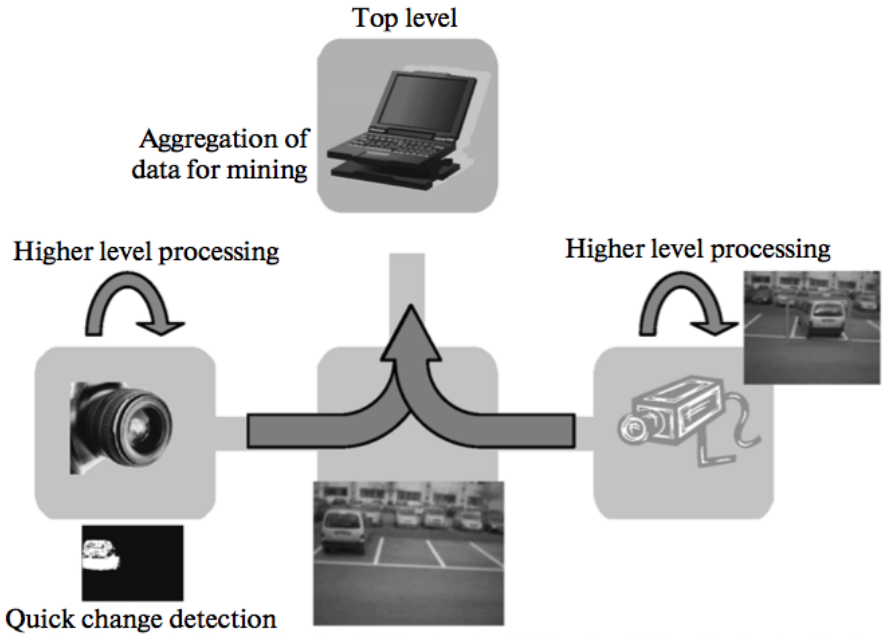


FIGURE 8. A multi-tier approach for parking slot monitoring.

- Thresholding of the background-subtracted images;
- Basic edge detection on binary image obtained by thresholding.

Another class of methods is based on statistical histogram analysis. Template histograms of the regions of interests in the BG-image are computed and stored. Then a standard distance, such as the Kullback-Leibler divergence is computed between the region of BG-image, and the region of the actual image. If such distance overcomes a fixed threshold, then a change event is detected.

Object Detection. As it is well-known, the problem of detecting classes of objects in cluttered images is challenging. Supervised learning strategies have demonstrated to provide a solution in a wide variety of real cases, but there is still a strong research activity in this field. In the context of SCN, the preliminary learning phase may be accomplished off-site, while only the already trained detectors needs to be ported to the network nodes.

Among machine learning methods, a common and efficient one is based on the sliding windows approach; namely rectangular sub-windows of the image are tested sequentially, by applying a binary classifier able to distinguish whether they contain an instance of the object class or not. A priori knowledge about the scene or information already gathered by other nodes in the network may be employed to reduce the search space either by (a) disregarding some region in the image and (b) looking for rectangular regions within a certain scale range (e.g. rectangular regions covering less than 30% of the whole image area). In our case, both information can be obtained by the change detection algorithm. Essentially, only those regions are processed by the classifier where a change has occurred and where the change has involved an area with the minimum scale requirements.

Concerning the binary classifiers itself, among various possibilities, the Viola-Jones method is particularly appealing. Indeed, such classifier is based on the use of the so-called Haar-like features, a class of features with limited flexibility but which is known to support effective learning. A Haar-like feature is essentially given by the difference between sum of pixels in rectangular blocks; such Haar-like feature is computable in constant time, once the integral image has been computed (see [**VJ04**] for details). Thus, having enough memory to store the integral image, the feature extraction process requires low computational power. Classification is then performed by applying a cascade of classifiers, as shown in Fig. 9.

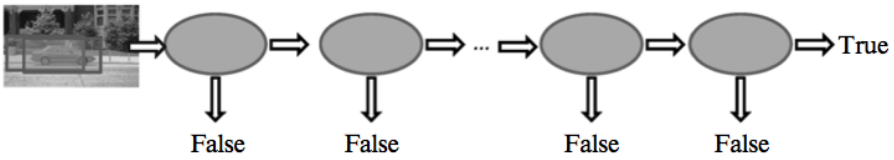


FIGURE 9. Example of a cascade of classifiers for object detection in case of parking slot monitoring.

A candidate sub-window which fails to meet the acceptance criterion in some stage of the cascade is immediately rejected and no further processed. In this way, only detection should go through the entire cascade. In addition, the cascade may be trained, for example using gentle AdaBoost, in such a way that most of the candidate sub-windows not corresponding to instances of the object class are rejected in the first stages of the cascade, with great computational advantages.

The use of a cascade of classifiers permits also to adapt the response of the detector to the particular use of its output in the network, also in a dynamical fashion, in order to properly react to changes to the internal and external conditions. First of all, the trade-off between reliability of detections and needed computational time may be controlled by adaptive real-time requirements of the overall network. Indeed, the detector may be interrupted at an earlier stage in the cascade, thus producing a quick, even though less, reliable output which may be sufficient for the current decision making problem. In the same way, by controlling the threshold in the last stage of the cascade, the SCN may dynamically select the optimal trade-off between false alarm rate and detection rate needed in a particular context.

Performance. For what regards object detection, several cascade of classifiers have been trained, each one specialized in detecting particular view of cars (e.g. front, rear and side views). A large set of labeled acquisition has been made of the real case study and used for training purposes. We notice that first stages in the cascade have low computational complexity but reasonable performance (that is almost 100% detection rate but even 50% false alarm rate). Composition of the stages entails then high performance of the entire cascade, since both overall detection rate and overall false alarm rate are just the product of the detection rate and false alarm rate of the single stages. For example, using $N = 10$ stages in the cascade, each one with detection rate $\nu = 99.5\%$ and false alarm rate $\mu = 50\%$, one gets an overall detection rate $\nu_{global} = \nu_N \approx 95\%$ and false alarm rate $\mu_{global} = \mu_N \approx 0.1\%$.

3.2.2. Synchronous background subtraction based algorithm for parking lots monitoring. The previous parking lots monitoring application is based on a two layers hierarchy of sensors characterized by different processing capabilities. In this situation the simplest nodes are in charge to trigger the most complex node recognise objects and to filter events. The idea we propose in the following considers a low-complexity algorithm able to understand the parking spaces occupancy state to be deployed in a low-complexity and low-memory SC, avoiding the duplication of the detection task. In this direction we have investigated a computer vision application based on background subtraction, capable to recognize the parking spaces occupancy modeling their status through a Markov chain. Moreover, in order to reduce the amount of data to be processed, only the Regions of Interest (RoI) related with the parking spaces are considered by the application (see Fig. 19).

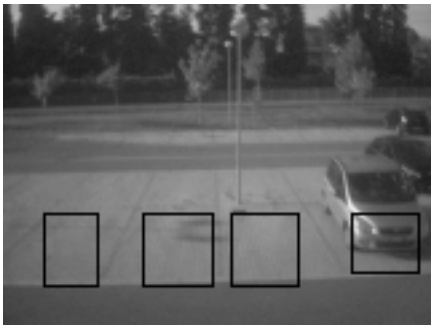


FIGURE 10. Parking spaces identified by their own RoIs.

Methodology. Considering a parking space, three main possible states are possible: *full*, *partially full/empty*, and *empty*. While the full and empty states (also called *stable states*) do not require further investigations, the third one must be better detailed. The car who is parking or leaving the monitored space usually requires a finite time to complete all the manoeuvrings, thus giving the possibility to slowly move from full to empty or viceversa. In a better explicative way it is possible to call the partially full/empty state as *transition* state. Moreover, because of the considered raw data are the projection of the 3D environment over a 2D image, it can be also used to filter spurious states as car or people passing and to avoid false empty to full or full to empty transitions.

All the above mentioned parking process can be modeled by means of the three states Markov chain, because the probability of being in a certain state at time $i+1$ depends only by the state at time i:

$$P\{s_{i+1} = x_{i+1} | s_0 = x_0, s_1 = x_1, ..., s_i = x_i\} = P\{s_{i+1} = x_{i+1} | s_i = x_i\}, \tag{1}$$

where s_i is the status at time i and $x_{i+1}, x_i, ..., x_1, x_0 \in \{empty, transition, full\}$.

Regarding the transition probabilities of the Markov chain, these represent the usage trends of the parking space and can be experimentally evaluated in time windows. Finally the diagram

shown in Fig. 11 describes the three state Markov chain and the related transition probabilities of a generic parking space.

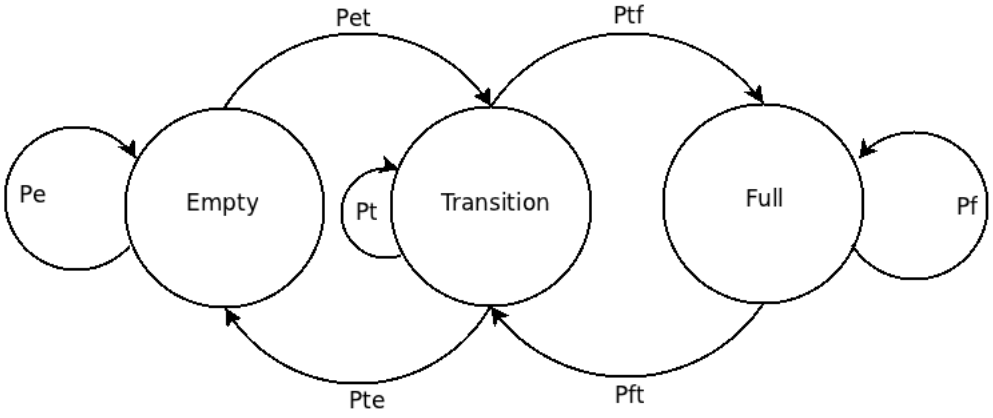


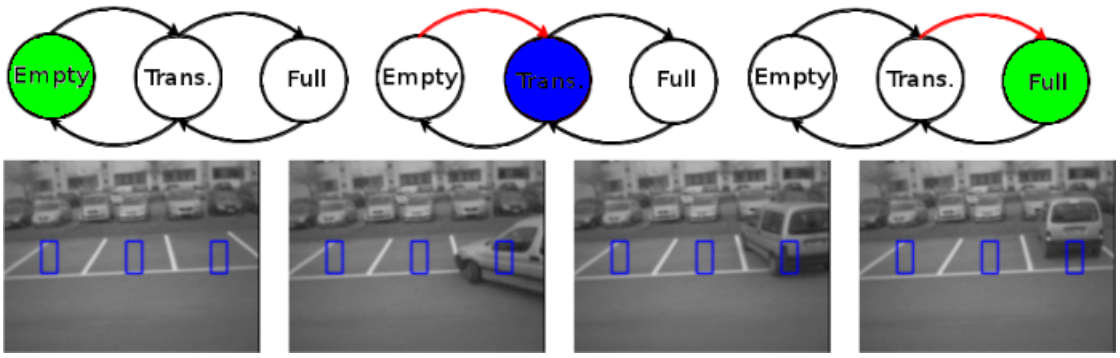
FIGURE 11. Markov chain based parking space model.

In the three states based Markov chain adopted to describe the parking space behavior a transition from one state to the other is achieved when a change in the scene is detected. In the proposed work the change detection is based on a joint *background subtraction* (BS) and *frame differencing* (FD) techniques. Particularly, the outputs of both the above mentioned algorithms (also called *difference images*) are evaluated through thresholding operations (controlled by the TH_D constant value, defined by the user) to obtain the number of foreground pixels inside the RoI (n_{BS} and n_{FD} respectively). The above mentioned variables n_{BS} and n_{FD} are in charge to enable the transition inside the Markov chain of Fig. 11: in fact if $n_{BS} > TH_{BS}$ and $n_{FD} > TH_{FD}$ (where TH_{BS} and TH_{FD} are two constants thresholds on the number of foreground pixels) a change in the system status is detected and the state of the system moves from a stable to the transition state. Once the system is in the transition state, the following conditions rule the chain:

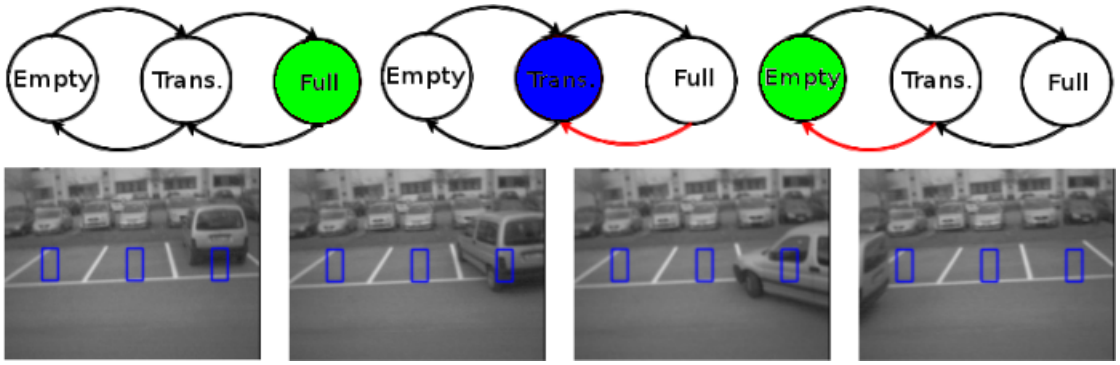
- If $n_{BS} > TH_{BS}$ and $n_{FD} > TH_{FD}$, the scene is not stabilized (e.g., car manoeuvrings, people going down from the car, etc.) and no possible transitions to stable states is possible.
- If $n_{BS} < TH_{BS}$ and $n_{BS} < TH_{BS}$ the new stable state is the same held before the transition.
- If $n_{BS} < TH_{BS}$ for a number of frame bigger than a given threshold TH_N but $n_{BS} > TH_{BS}$ the new stable state is the opposite with respect to the one held before the transition.

In Fig. 12 all the possible transition between the states are show following the previously listed rules.

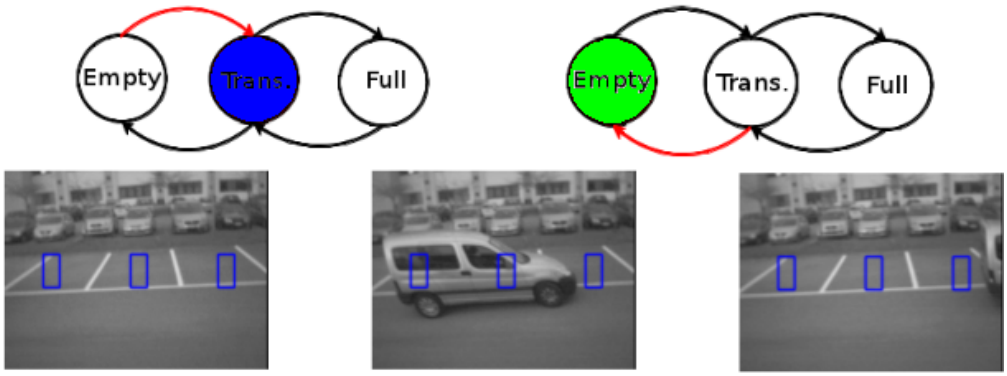
Since a background subtraction logic is used, then an on-line adaptation model for the reference image has to be defined, to permit the algorithm to be robust to illumination and stable geometrical changes. In this direction we propose a specifically designed algorithm, able



(A) From empty to full transition



(B) From full to empty transition



(C) Spurious event transitions

FIGURE 12. Transitions due to car parking activities and spurious events.

to guarantee both good performance and low computational complexity. Thus, the idea is to resolve the reaction to illumination changes using the *running average* algorithm [**RBI$^+$05**] is proposed. This configuration updates each background pixel according to the following equation:

$$B_{i,j}(t_n) = (1-\alpha)B_{i,j}(t_{n-1}) + \alpha I_{i,j}(t_n), \tag{2}$$

where $B(t_{n-1})$ is the old background, $I(t_n)$ is the last acquired frame and α is the *learning rate* ($\alpha \in (0,1)$). However, because the change detection paradigm enable the Markov chain transition, the stable geometrical changes have to be handled manually and the running average algorithm is disabled during the stay into the transition state. Finally we propose to synchronise the geometrical changes updating with respect to the Markov chain: when a transition into one of stable state occurs the last acquired image is set as background and the running average is enabled again.

The confidence index. In this section we describe the logic for deciding whether the parking space is empty or full. To implement this logic, a *confidence index* (CI) is considered: this index describes the probability of a parking space to be full. The CI is evaluated as a function of the time and it is retrieved by the parking space occupancy algorithm. The index is evaluated at the end of each change detection evaluation and then quantized on a 8 bit value in order to reduce the packet payload in a wireless communication: we represent with 0 an empty space, with 255 a full space. In an application scenario in which several parking spaces are monitored, several CI must be sent together with other possible acquired data (e.g., temperature, light, CO2 level), the use of a tiny amount of bytes for each status notification allows to reduce the transceiver usage, thus saving energy.

Due to the Markov chain, the confidence index values range has been divided in three main parts (see Fig. 13) in order to associate each of them to one of the possible system state. In this situation the range associated to the empty state goes from 0 to T_e, and the full from T_f to 255, where T_e and T_f are constant value fixed by the user close to 0 and 255 respectively.

Moving from empty to full the CI increases as a broken line from 0 to 255 (as sown in Fig. 14), following two different behaviors in the transition state. The change detection procedure splits the transition zone in two parts: a *transition unstable zone* and a *transition stable zone*. The transition unstable zone is close to the previous stable state and represents the period of time dedicated to enter or leave the parking space. The transition stable zone, instead, represents the period of time between the end of the car manoeuvrings and the transition to the stable state.

Metrics and data-sets. The effectiveness of the proposed algorithm can be seen as its ability in reflecting the real behavioral trend of monitored parking spaces. In terms of performance,

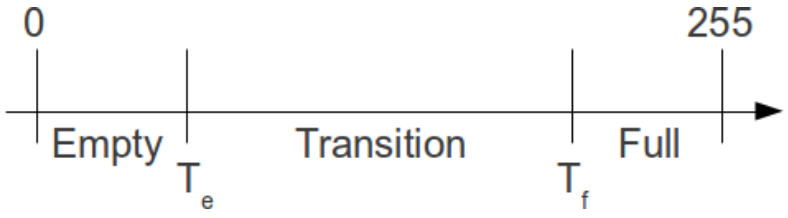
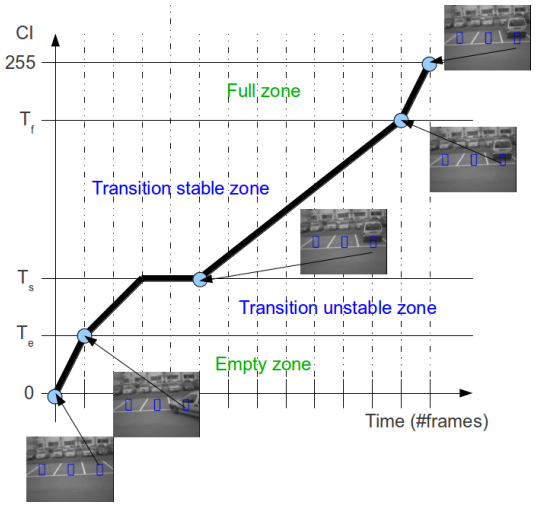
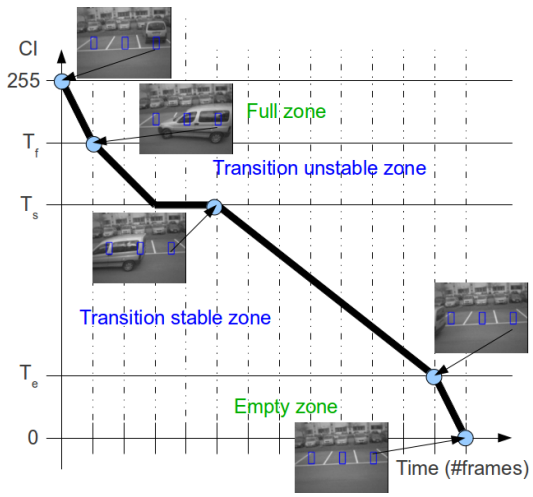


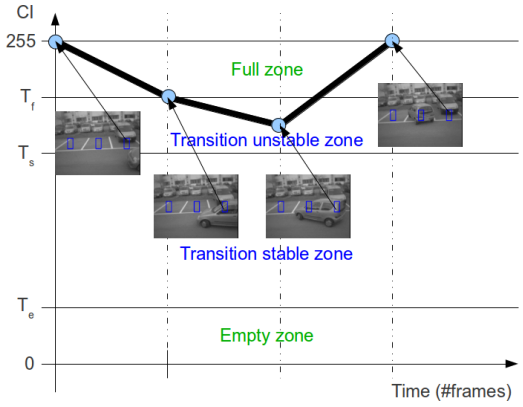
FIGURE 13. Confidence index values range and system states



(A) From empty to full transition



(B) From full to empty transition



(C) Spurious event transitions

FIGURE 14. Effect of different types of transitions on CI values

the algorithm detection capabilities can be measured with respect to real ground-truth values evaluated by means of a human-based analysis, making use of the following three metrics:

$$Sensitivity = \frac{TP}{TP + FN} \tag{3}$$

$$Specificity = \frac{TN}{TN + FP} \tag{4}$$

$$Accuracy = \frac{TP + TN}{P + N} \tag{5}$$

where TP, TN and FN are the number of *true positives*, *true negatives* and *false negatives* respectively, and P and N the number of *positives* and *negatives*. More in detail, if in one side

both the *sensitivity* (Eq. 3) and the *specificity* (Eq. 4) describe the ability of the algorithm on correctly detecting full and empty state respectively, the *accuracy* (Eq. 5) measures the algorithm capability on detecting events.

In order to evaluate the above mentioned metrics we select three datasets: the first two collected within the IPERMOB [**ipe09**] project (taken from IPERDS [**IPE11**]) and composed by a set of gray scale QQ-VGA images acquired at 1 fps that cover an acquisition of 15 minutes, and the third acquired using a NIR camera (@1fps) with a duration of 24 hours in order to cover both the night/day and day/night transitions. Among all the traces we tag each image with the true state of each parking space (*ground truth*) using the same CI range: the empty status has been notified with the value 0, the transition with 127, and the full with 255.

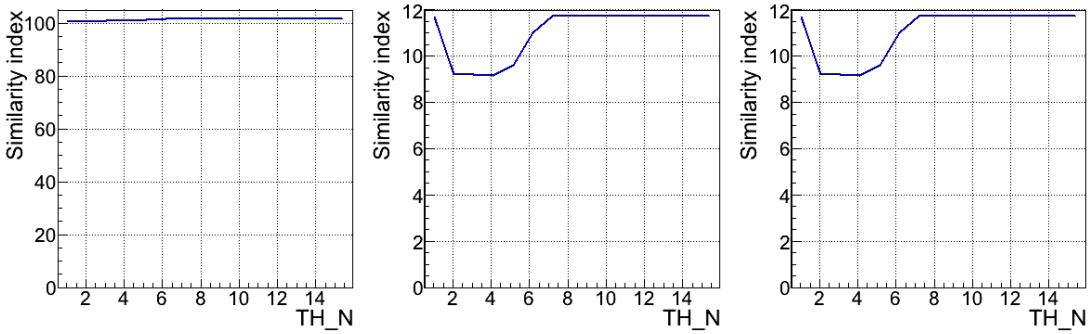
Algorithm parameters setting. All the algorithm thresholds need to be tuned by means of a comparison between the algorithm output and the ground-truth, using a dataset from IPERDS. Although the used thresholds are four, only two of them must be properly tuned: TH_D and TH_N, namely the threshold to filter the difference images, and the number of frame to pass to a stable state. The two remaining thresholds, TH_{BS} and TH_{FD} can be fixed equal to a portion of the RoI. More in detail, the TH_{BS} threshold has been imposed equal to 1/4 of the RoI area due to its requirements to trigger the state transitions, while TH_{FD} has been imposed equal to 1/8 of the RoI area in order to guarantee event stabilisation. The TH_D and TH_N have been jointly varied in the range from 50 to 60 and from 1 to 15 respectively, while evaluating the ground-truth similarity trend. In mathematical terms, the tuning procedure consists in finding from the set $\Im$ of all the possible combination between both the thresholds, the pair $\hat{TH} = (\hat{TH}_D, \hat{TH}_N) \in \Im$ which minimises the Euclidean distance between the algorithm output and the ground-truth normalized by the number of frames M:

$$S = \frac{\sqrt{\sum_{k=1}^{N}(G_{gt}[k] - G[k])^2}}{M} \tag{6}$$

where G_{gt} is the ground-truth and G is the algorithm output derived from the pair $TH = (TH_D, TH_N) \in \Im$. In Fig. 15 the trend of S is shown as a function of TH_N, under different values of TH_D, and the minimum value of S is met with the pair $\hat{TH} = (50, 5)$.

Performance evaluation. The detection performance of the developed algorithm have been evaluated by means of simulations using the same implementation suitable to run in real embedded devices. By adopting the selected thresholds pair $\hat{TH}$, we evaluate the system over the remaining two datasets. Particularly, in Fig. 17 the output of the system for each parking space (see Fig. 16) for the first dataset is shown, and it can be compared with the ground truth of Fig. 18: the graphical comparison between the ground-truth and the output confirms the algorithm capabilities in detecting the effective parking spaces occupancy status.

Using the same approach for the second dataset (see Fig. 19), in Fig 20 the graphical trend of the algorithm output are shown as a function of the time. Also in this case we can state that the algorithm is strong enough in the detection of the parking spaces real occupancy, even using a NIR-camera under different conditions of illumination.



(A) $TH_D = 50$ (B) $TH_D = 55$ (C) $TH_D = 60$

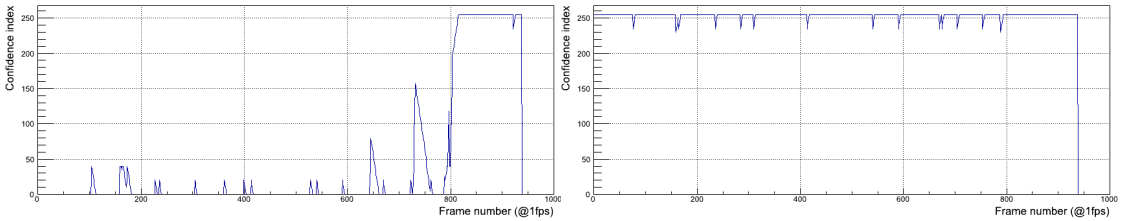
FIGURE 15. Similarity trend analysis.

FIGURE 16. Parking spaces considered in a testing trace.

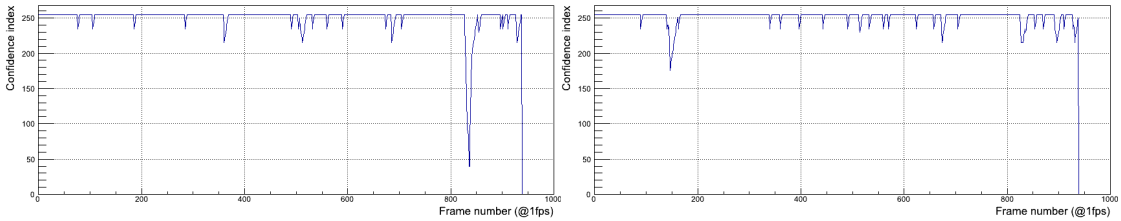
In Tab. 1 the evaluation of the metrics from Eq. 3, 4 and 5 are shown for both the datasets. The values shown in the above mentioned table confirm the results discussed in the graphical comparison of Fig.17 and 20: the application is able to detect the more than the 90% of the full state events full and more than 95% of the empty state in both the cases. Finally the accuracy of the measurements is bigger than 95%.

TABLE 1. Aggregated results for both the datasets.

	First dataset	Second dataset
Sensitivity	99.93%	91.85%
Specificity	95.60%	99.05%
Accuracy	98.97%	96.67%

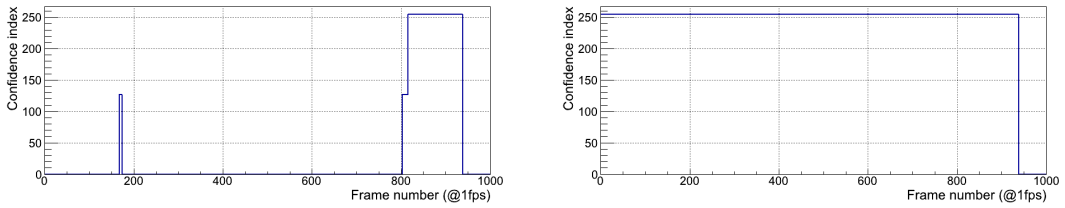


(A) P21 (B) P22

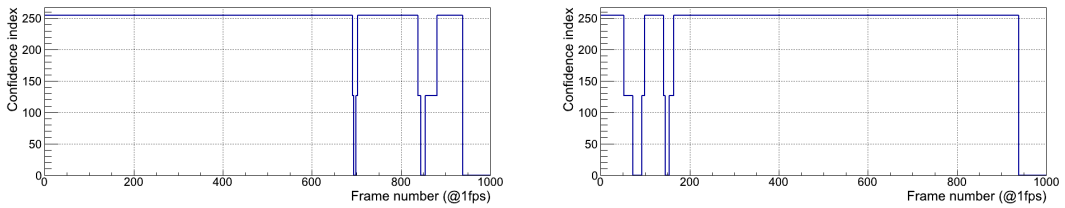


(C) P23 (D) P24

FIGURE 17. Algorithm confidence index trend for the first dataset (15 minutes).



(A) P21 (B) P22



(C) P23 (D) P24

FIGURE 18. Ground-truth confidence index trend for a testing trace.

3.3. Counting vehicles: the *Linesensor* application.

This section describes a vehicles counter application to be used in low-complexity and low-memory devices to be deployed in next generation pervasive ITS. The first part of the section introduces the Linesensor theory, which exploits the temporal redundancy of the movement to

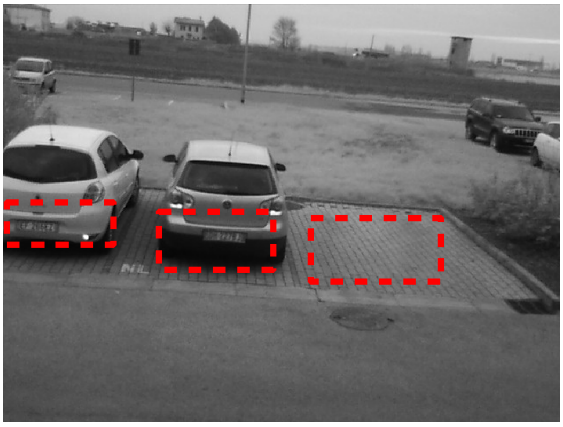


FIGURE 19. Parking spaces considered in a testing trace.

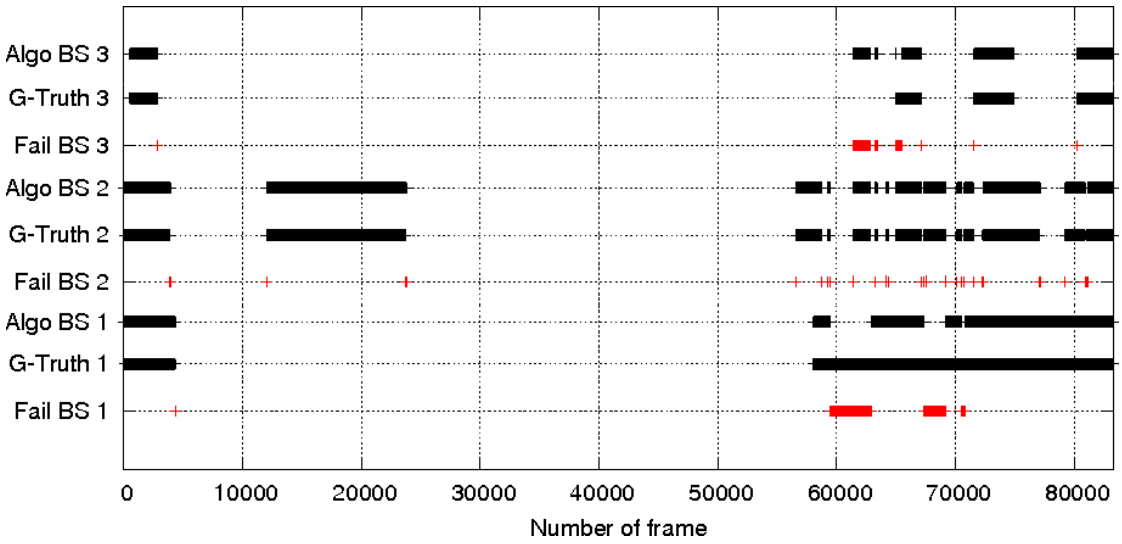


FIGURE 20. Algorithm confidence index trend for the second dataset (24 hours).

enable the processing of a 1D images (i.e., lines), thus reducing the complexity for extracting features and understanding the environment. Because of the high speed of the objects to be detected, the proposed application requires a very high frame rate and consequently an optimised design for the whole computer vision pipeline. For these reasons, in the second part, we propose a low-complexity background modeling algorithm permitting to extract information related to the whole image from a single metric. Our arguments demonstrate that our algorithm has comparable performance in the segmentation operation as other state-of-the-art techniques, but reducing significantly the computational cost.

FIGURE 21. Reconstructed image.

3.3.1. The vehicles counter application. In this section we describe how the vehicles counter application works by defining a general theory, called *Linesensor theory* and consequently detailing the specific computer vision pipeline.

The Linesensor theory. The Linesensor is a camera-based sensor capable to acquire and process 1D vectors of pixels. In this condition, the acquired image can be seen as a line developed along a single dimension: every scene is totally represented along one dimension (*principal dimension*), while only a negligible part in the other (*secondary dimension*). Thus, the use of state-of-the-art computer vision techniques permits only to extract blobs, without any semantic content. However, this configuration permits to exploit the temporal redundancy of moving objects increasing the information content along the secondary dimension. More in detail, by considering a *1D pixel-wise operator* ($\hat{\mathcal{D}}$) (i.e., an operator performing background subtraction, edge detection, etc.), and a *reconstruction filter* ($\mathcal{R}$), their composition permits to have the same result of the $\mathcal{D}$ operator applied to the 2D image:

$$\mathcal{D} = \mathcal{R} \circ \hat{\mathcal{D}} \tag{7}$$

To better explain the above concepts it is possible to consider Fig. 21, which is a set of lines acquired in subsequent time instants, and positioned one close to the other respecting the temporal order (we call such a figure *reconstructed image*). Eq. 7 states that the reconstruction

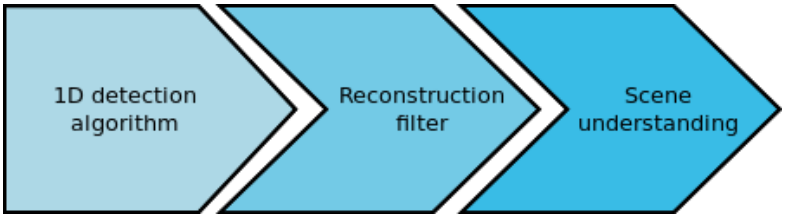


FIGURE 22. Computer vision pipeline for Linesensor.

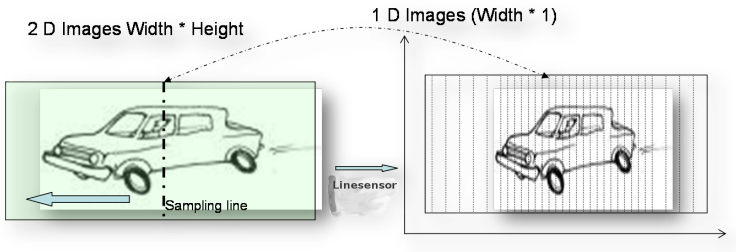


FIGURE 23. Linesensor working principle applied to the vehicles counter application.

filter $\mathcal{R}$ has to rule the output of $\hat{\mathcal{D}}$ applied to every vertical line in order to obtain the same output achievable by $\mathcal{D}$ applied to Fig. 21. Therefore, we can define the reconstruction filter as a function with memory, and capable to spatially and temporarily arrange the features extracted by $\hat{\mathcal{D}}$ in order to permit the detection and the recognition of real objects.

Starting from Eq. 7 it is possible to define the pipeline suitable to process lines instead of 2D images: in the first step the 1D detection pixel-wise algorithm is applied, then the results are arranged by the reconstruction filter, and, finally, the scene is understood using state-of-the-art computer vision pipelines. The described pipeline is graphically reported in Fig. 22.

Linesensors for vehicle counting. We propose an interesting application derived from the Linesensor theory and able to count moving vehicles by using low-end devices. The application scenario of this work is that of next generation ITS to be developed as a building block of the future Smart Cities. In such a scenario the possibility of pervasively counting the number of cars passing in a road segment enables to create a whole set of advanced services, as real-time congestion avoidance, traffic prediction services, etc..

From a computer vision point of view this application performs an object detection task of moving vehicles based on 1D images (i.e., a single line) acquired at a high frame rate. As working principle, the whole process aims at reconstructing a 2D image from a single line, as illustrated in Fig. 23. The use of a single line guarantees to reduce the amount of data to process, as well as the amount of memory to be used, thus resulting to be an effective solution for low-complexity and low-memory embedded devices. On the other hand the necessity of processing temporally subsequent lines introduces a limitation in the detection: to avoid missing detections of moving objects a high frame rate is necessary.

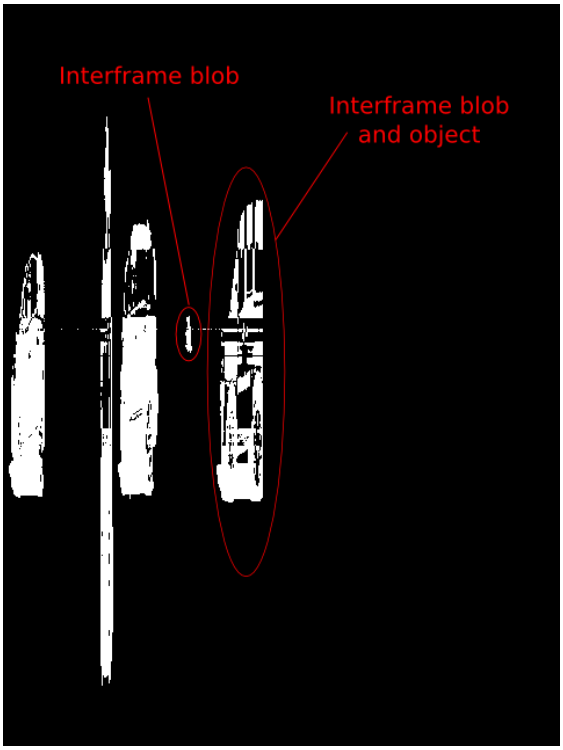


FIGURE 24. Blobs and objects.

To better explain the vehicles counter application, we define *intraframe blob* the interval of vertically connected foreground pixels that satisfy the *intraframe condition*, i.e. the number of connected foreground pixels is greater than the threshold λ_{intra}; we define *interframe blob* the set of horizontally connected lines where the intraframe condition is verified. Thus, deeply analyzing the vehicles counter pipeline, as described by [**CLZ$^+$09**], in respect of the Linesensor theory, the whole system performs a 2D object detection ($\mathcal{D}$) which is the result of a blob detection algorithm ($\hat{\mathcal{D}}$), based on the background subtraction, and a reconstruction filter ($\mathcal{R}$) able to detect the interframe blobs.

Following the pipeline reported in Fig.22, the last step is the understanding of the scene, which is based on a low-complexity decision-making algorithm. The reconstruction filter and the scene understanding task are deeply correlated. Indeed, we state that an *object* is detected if the interframe blob is composed by a number of consecutive lines bigger than the threshold λ_{inter} (*interframe condition*). In Fig. 24 a real example, in which the concept of interframe blobs and objects are highlighted using a binary B/W rendering.

From these considerations it follows that the whole system performance depends on the correct intraframe blob detection during the analysis of each line. Consequently, the crucial step to let the background subtraction algorithm working in real conditions is to define a background model able to react to the illumination changes, as well as to absorb stable geometrical variations. Moreover, both the following constraints have to be satisfied: (i) the computational power of

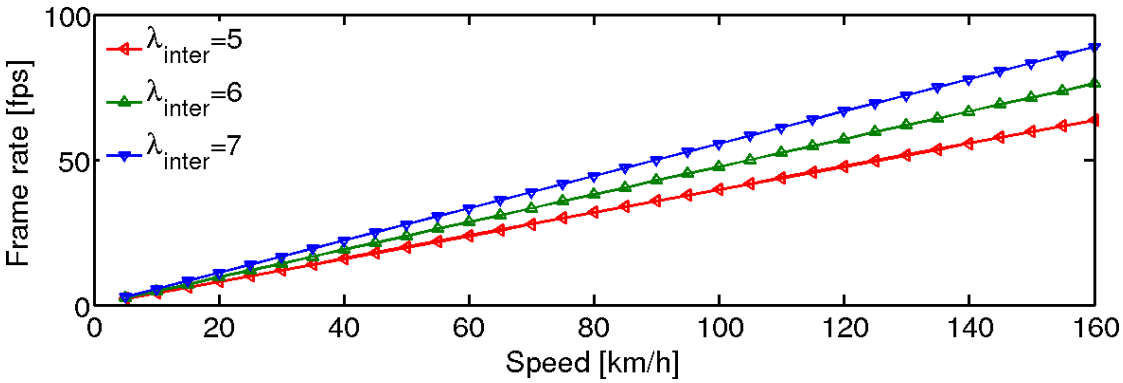


FIGURE 25. Frame rate versus speed ($l = 3.5m$).

the considered embedded systems is limited, and (ii) the algorithm used to understand the scene introduces a limitation in the maximum vehicle speed as a function of the frame rate F. In fact, considering l the length of the object passing in front of the camera, it is detected if its maximum speed is lower than a v_{max} value which is equal to:

$$v_{max} = l \cdot \frac{F}{\lambda_{inter}} \tag{8}$$

The necessary frame rate as a function of the maximum speed is reported in Fig. 25 for several λ_{inter} values.

In the following we present a low-complexity background modeling technique suited to satisfy all the above mentioned constraints, eligible as a general solution for applications based on both background subtraction and Linesensor theory.

3.3.2. Background modeling.

State-of-the-art. One popular approach for background modeling in embedded systems is the temporal median filter (see [**LV01**, **CGPP03**]) and its approximations. For example, in [**HPH10**], a *selection-based median filter* algorithm is proposed to reduce the computation and consequently to speed-up processing, and in [**MS95**] a very fast and low memory footprint *approximated median filter* is depicted. In [**RBI$^+$05**], another low computation and low memory footprint method is based on *running average*: each background pixel is updated by its difference with the corresponding value in the current image, using a value $\alpha \in [0, 1]$ called *learning rate*. Moreover, in [**ILRLB10**], a combination of the two previous methods is described to increase the robustness of the *median filter* to sudden changes.

The above mentioned techniques represent the variations of each image pixel using only one value to model the background, while foreground is considered as an outlier. Methods that explicitly represent foreground as part of *multimodal approaches* have demonstrated higher accuracy. For instance, in [**AVF$^+$09**] a *multimodal mean technique* is presented: every image pixel is described by a group of K mean values and a fixed threshold, to define a membership criterion, and a pixel is labeled as background if it matches a popular mean value. However, all

above assumptions lack a rigorous statistical base, as no deviation measurement is used to adapt the membership threshold.

Finally, in [**SMP$^+$12**] (see also Chap. 5) an integer based Gaussian Mixture Model realisation is described to allow deployment on processors with no FPU. Mean value and variance updating process uses a rounding operator, while weight updating is based on a counter-based representation. This technique demonstrated both low computation cost and a highly optimised memory footprint (1/12 of the original GMM memory footprint) for a two-Gaussian mixture. For example, they achieved real-time performance for a QQ-VGA resolution using a low-complexity and low-memory micro-controller (PIC32-80MHz). However its extension to models with more than two Gaussians is not trivial and it introduces restrictive limitations on the learning rate domain. We propose the basis to overcome those limitations and deal with any given number of Gaussians.

In all these described techniques, the whole background is modeled with a per pixel evaluation (*pixel-based* modeling), thus introducing an overall computation load which strongly reduce the system frame rate. To achieve a low processing latency able to increase the frame rate, we propose a technique able to model the whole background with a reduced set of metrics modeled in a statistical manner (*picture-based* modeling), as described in [**KJC07**]. Such a method is a suitable solution only when the following two conditions hold:

(1) only a single event occurs in the monitored region (holistic approach);
(2) the size of the object (expressed in number of pixels) is comparable with the monitored region size, in order to have a not negligible impact on the considered metrics.

Both the mentioned conditions are satisfied by the considered vehicles counter application, so that a background model based on a single metric is rather effective as it will be detailed in the next section.

Single metric model. As reported in Sec. 3.3.1, the vehicles counter application requires a high frame rate: this requirement makes completely unfeasible the use of pixel-based techniques in low-end devices.

On the other side, the usage of 1D images well fits with the picture-based approach, because it satisfies both the conditions given previously reported. In fact, since a single Linesensor based system can monitor several lanes of a road (with a single event per lane, see Fig. 26), the definition of several Region of Interests (RoIs) where to apply background modeling permits to detect a single event per RoI; moreover, because the lane width is comparable with the size of the vehicles, the second condition is satisfied. Consequently we can define a picture-based modeling, where a single metric is appointed to: (i) discriminate background from the foreground, and (ii) update the background in case of fast and slow changes. For this reason we call such a method *Single Metric Model* (SMM), and we define the SMM metric $d(i)$ as follows:

$$d(i) = \sum_{j=1}^{N} |p_{i,j} - b_j| \tag{9}$$

where p_i is the i-th line, b is the background, N the number of pixels per line and j is the pixel index. Thus, considering a situation with no events and constant illumination, the theoretical

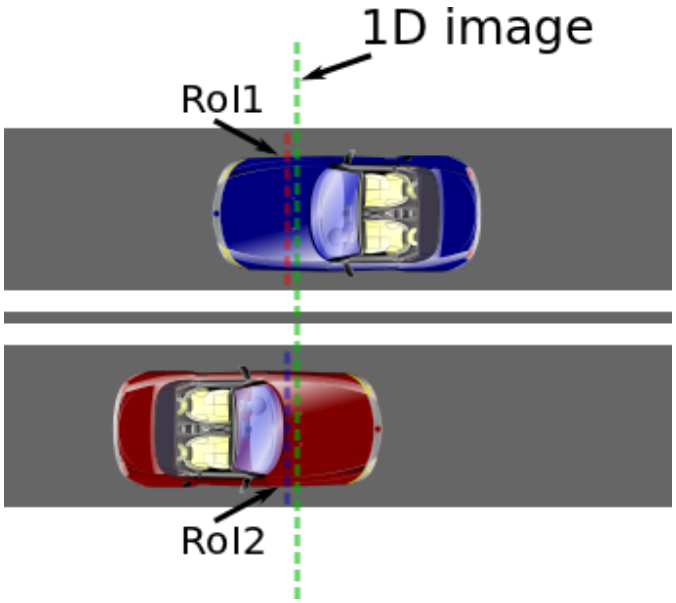


FIGURE 26. Monitoring several lanes of a road.

value of the metric is $d_i = 0$. However, considering a more realistic case, in the same conditions, it is possible to experience $d_i \neq 0$ values: d_i is affected by the noise generated by the camera circuitry. Considering a population of images with no events and constant illumination, we can compute the characteristic $d(i)$ distribution for the chosen camera, we call it *Reference probability density function* (*R-pdf*) and modeled as a Gaussian g_r. In a real scenario (i.e., with events and illumination changes) the $d(i)$ distribution (modeled as a Gaussian $g_c(i)$) is updated and compared with the R-pdf g_r; our decision making algorithm will follow this comparison on the basis of the occurrence of the three events defined below:

- **(i):** the *background event*, triggered when a sensitive variation in the mean value and a negligible one in the standard deviation occur. In this case the reference image b is overwritten by the current line (background update);
- **(ii):** the *blob event*, triggered when a sensitive variation in both mean and standard deviation values occur. In this case the presence of an intraframe blob is validated using the algorithm in Sec. 3.3.1;
- **(iii):** *no event*, when no meaningful variation occurs and no action is taken to save energy.

From these consideration we claim that our algorithm can keep the mean and the standard deviation values of the Gaussian $g_c(i)$ comparable with (*R-pdf*) so that the background subtraction operation is robust to illumination changes as it will be proven in the experimental sections (see Sec. 3.3.3).

Finally, to better understand the presented decision-making algorithm, in Alg. 1 we detail it as a pseudo-code.

Algorithm 1 Decision making algorithm pseudo-code

```
create_gaussian g_r(m_0, sd_0);
create_gaussian g_c(m_0, sd_0);
background=acquire_image_from_camera();
while 1 do
  image=acquire_image_from_camera();
  d_i = evaluate_metric(image, background);
  g_c = update_gaussian(d_i, g_c);
  if (g_c.m > l_m + g_r.m) and (g_c.sd > l_sd · g_r.sd) then
    Linesensor(image, background);
  end if
  if (g_c.m > l_m + g_r.m) and (g_c.sd < l_sd · g_r.sd) then
    update_background(image, background);
  end if
end while
```

The mean value and standard deviation updating rules. As previously described, the decision-making algorithm is based on the comparison between the current Gaussian $g_c(i)$ and the (*R-pdf*) g_r: in this section we describe the rules to keep updated the $g_c(i)$ mean and standard deviation values as a function of the current value of d_i. Thus, a Finite Impulse Response (FIR) filter is defined for both the parameters as follows:

$$g_c(i) : \begin{cases} \mu_c(i) = (1-\delta)\cdot\mu_c(i-1) + \delta\cdot d(i) \\ \sigma_c(i) = (1-\delta)\cdot\sigma_c(i-1) + \delta\cdot(d(i)-\mu_c(i)) \end{cases} \tag{10}$$

where $\mu_c(i)$, $\sigma_c(i)$ and $\mu_c(i-1)$ and $\sigma_c(i-1)$ are the mean value and the standard deviation of $g_c(i)$ at the i-th and $(i-1)$-th time instant respectively, and δ is a *learning rate* ($\delta \in [0,1]$). Moreover, the relations shown in Eq. 10 guarantee the absorption into the background of stable standing objects, making the considered algorithm robust to permanent changes of the reference image. In fact, considering a homogeneous vehicle, characterized by a constant d_i equal to D_0, that stands in front of the camera for an infinite time, the following relations hold:

$$\lim_{i\to\infty} \mu_i(d(i))_{|d(i)=D_0} = D_0 \tag{11}$$

$$\lim_{i\to\infty} \sigma_i(d(i))_{|d(i)=D_0} = 0. \tag{12}$$

Thus, considering these relations and the Gaussian parameters trend in Fig. 27, we can state that:

- in point A, the vehicle is detected as a deformation of g_i with respect to g_r;
- in point B, a *background event* is generated, and the vehicle is absorbed in the background overwriting the reference image.

Ghost avoidance. The robustness of the considered model against the stable geometrical changes on one hand guarantees to increase the detection performance, on the other hand it raises several problems in critical situations. In an ITS scenario, for instance, critical situations

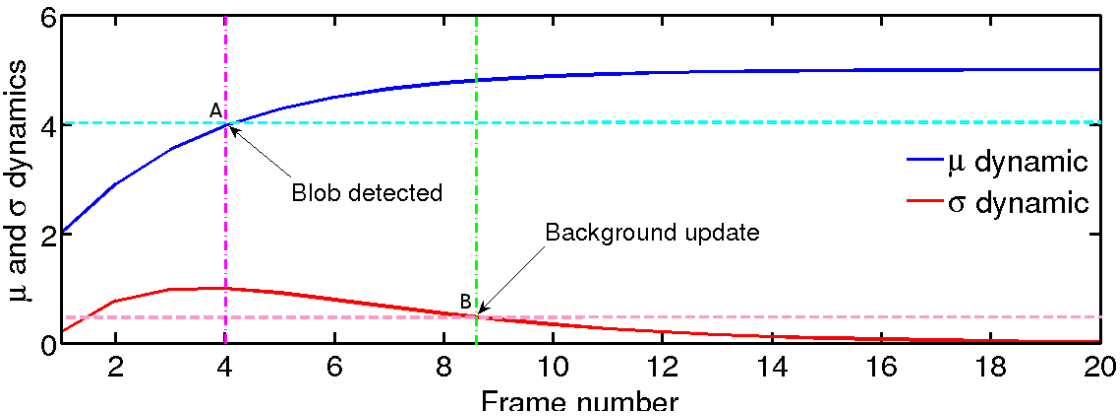


FIGURE 27. μ_i and σ_i dynamics.

to address are those related to vehicles running at reduced speed, as well as vehicles stopped in a traffic queue. In such situations it can happen that, after a certain amount of time in which the car stands in front of the camera, the object is absorbed into the background. Therefore, when the car starts to move again, there will be the detection of a false object called *ghost*. In other words, the ghost is an artifact caused by an inappropriate absorption of an object into the background. This absorption is due by the long permanence of a vehicle in front of the SC.

To better explain the ghost phenomenon in Fig. 28 the ideal case, in which there is no ghost artifact (on bottom of the figure), and the case in which a ghost appears (on top of the figure) are shown. In the ideal case this is what happens:

- in A the object is detected;
- in C the object exits from the line field of view and no artifact object is detected.

What happens in the reality is that the object is absorbed into the background and then an artifact is detected:

- in A, the object is detected;
- in B, the object is absorbed into the background because of a *background event*;
- in C, the the ghost artifact is detected, because the previously absorbed object exits the line view;
- in D, a new *background event* re-establishes the correct background.

A real example of the ghost phenomenon is reported in Fig. 29, where on the left the original reconstructed image has been reported, and on the right the binarized reconstructed image resulting from the background subtraction algorithm is shown.

In a real condition the ghost problem cannot be solved, but only delayed. In fact, it is not possible to understand whether a vehicle has been parked, and then it can be correctly absorbed into the background, or whether it stopped in a traffic queue. We propose a *ghost avoidance* algorithm based on the idea of delaying as much as possible the *background event* when an object is detected, considering two learning rates: (i) δ_h to update $g_c(i)$ in case of no object detected,

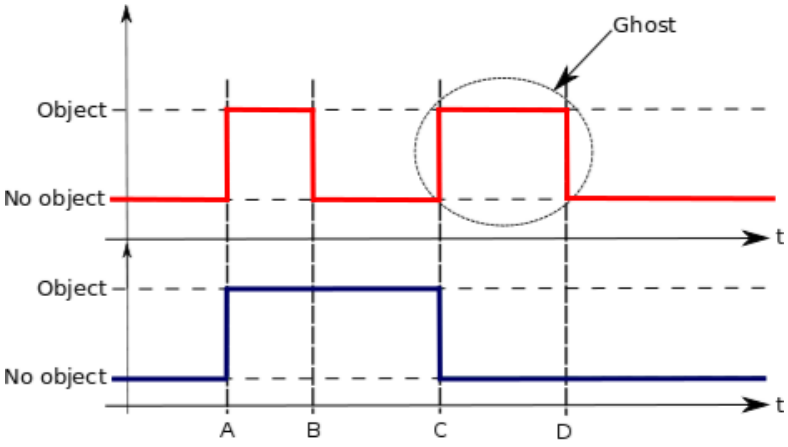


FIGURE 28. Detection: no ghost vs. ghost appearance.

(A) Original image. (B) Binarized image.

FIGURE 29. The ghost phenomenon: a real example.

and (ii) δ_l, such that $\delta_h > \delta_l$, in order to slowly update both the parameters $\mu_c(i)$ and $\sigma_c(i)$, when an object is detected.

3.3.3. Performance evaluation. In this section the performance of the vehicles counter application based on the Linesensor theory is evaluated in terms of efficiency on both background modeling and counted number of objects. As first, in Sec. 3.3.3 we evaluate the robustness of the presented background model against illumination changes, and in Sec. 3.3.3 the adopted SMM is compared qualitatively and quantitatively with a state-of-the-art background modeling technique (i.e., Gaussian Mixture Model). Moreover in Sec. 3.3.3 the effect of the ghost avoidance algorithm is described. Finally, in Sec. 3.3.3 the performance of the vehicles counter is evaluated in terms of number of detected objects and compared with the ground-truth.

To evaluate the above reported performance we used two different datasets acquired at 15 fps during the IPERMOB Italian project, such that: (i) the first dataset, that covers different situations of traffic and illumination, is composed by 4 tracks of 5000 raw lines (see Fig. 33a, 34a, 35a and 36a) and its segmentation ground-truth (see Fig. 33b, 34b, 35b and 36b); (ii) the second dataset is composed by 4 long period acquisition tracks and each line is labeled with "1" if a vehicle is present on it and with "0" otherwise. Particularly, the first dataset is used to evaluate both the segmentation performance and the optimal algorithm parameters configuration (i.e., as a training set), and the second is used as a testing set for the vehicles counter performance.

To understand the overall performance of the proposed algorithms, an aggregated analysis, based on *Precision-Recall* (or *P-R*) curves is presented. Particularly, the minimum distance of the P-R curves from the perfect classification point is computed for each value of learning rate so that the smaller is the distance, the better is the performance. Then, the configuration parameters that minimise in average this distance are used to evaluate the vehicles counter performance.

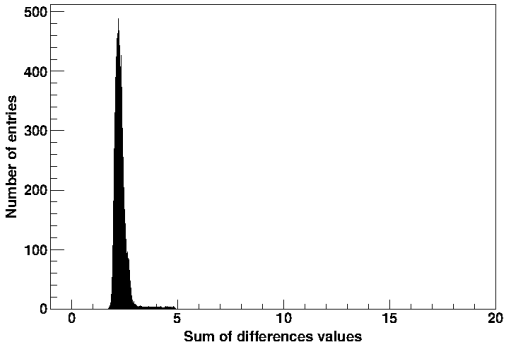
Finally, to evaluate the vehicles counter results in an aggregated manner, we define the *measurement error* ϵ as:

$$\epsilon = \frac{|N_{GT} - N_M|}{N_{GT}} \tag{13}$$

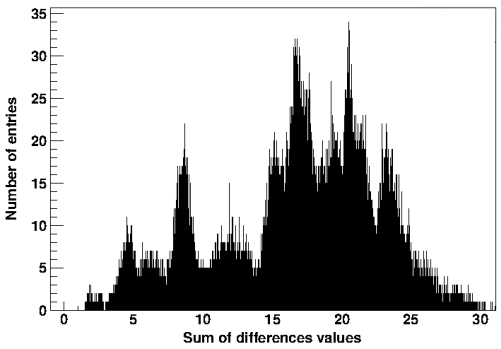
where N_{GT} and N_M are the number of vehicles passing in front of the sensor counted by a human operator and the considered application respectively.

Background model validation. In this section we demonstrate that the SMM technique satisfies the background adaptation requirements: on one side it makes the $g_c(i)$ parameters comparable with those of g_r (*R-pdf*), and on the other side it makes the background subtraction algorithm robust to fast and slow illumination changes. To this end, by using the HV7131GP camera, we acquired a large population of lines under the conditions of constant illumination and no events, thus characterizing its internal circuitry noise and retrieving the *R-pdf*. In Fig. 30a the *R-pdf* for the specific camera is shown: such a distribution has been used in all performed analyses.

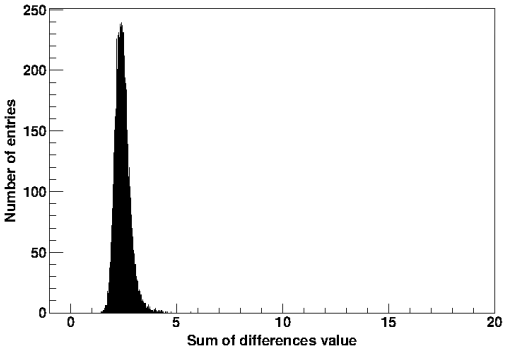
In Fig. 30 the $d(i)$ distribution of a variable illumination dataset (without events) is shown when the considered modeling technique is adopted and when it is not: as we can see in Fig. 30c, the proposed algorithm keeps the updated Gaussian $g_c(i)$ comparable with g_r in terms of mean and standard deviation values without the artifacts visible in Fig. 30b.



(A) Evaluation of *R-pdf*.



(B) Distribution of $d(i)$ in case of variable illumination with no adaptation algorithm.



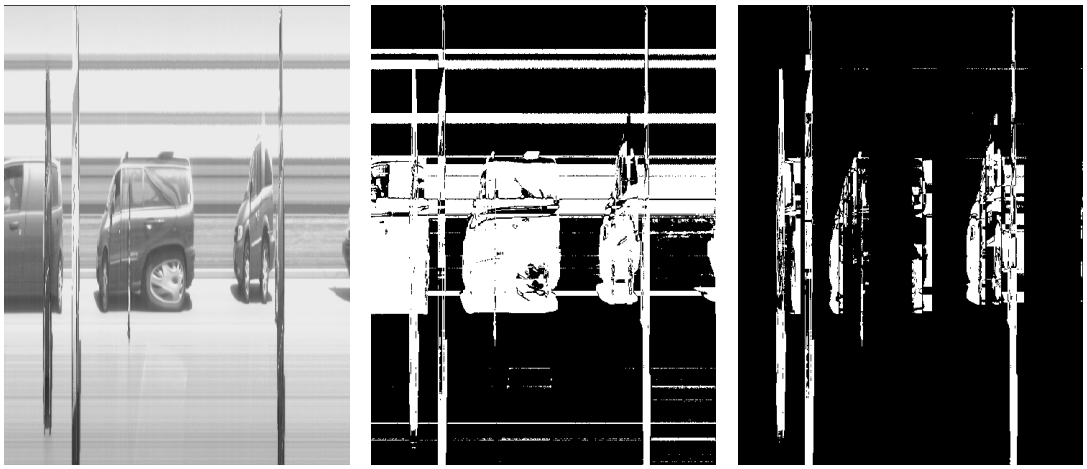
(C) Distribution of $d(i)$ in case of variable illumination with the considered adaptation algorithm.

FIGURE 30. Distribution of $d(i)$, with and without SMM

Finally, in Fig. 31 the effect of SMM is evaluated in terms of binarized images in output to the background subtraction algorithm: the SMM method enhances the algorithm robustness to the illumination changes, permitting an efficient foreground segmentation as it is depicted in Fig 31c.

SMM vs GMM. In this section the SMM and GMM techniques are compared, by means of a qualitative analysis, i.e., a visual comparison on binarized foreground images, and quantitative analysis, i.e., an analysis on aggregated metrics derived from P-R curves. Finally the processing time of both the techniques are evaluated using the SEED-EYE boards (introduced in Chap. 1).

Qualitative and quantitative comparison. In Fig. 32, the trends of the distance for P-R curves are shown as a function of the learning rate for SMM and GMM. Even though the two techniques have different trends, we are interested on their minimum value, that gives an idea on how good is the model. From this point of view, both the techniques have a comparable minimum distance for



(A) Original reconstructed image.

(B) Binarized reconstructed image without SMM.

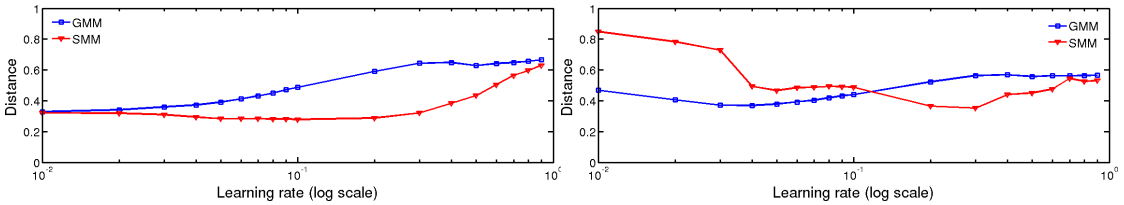
(C) Binarized reconstructed image with SMM.

FIGURE 31. The effect of SMM on the binarized reconstructed images.

P-R curves and consequently similar performance. This assumption is confirmed by Fig. 33, 34, 35 and 36, where the reconstructed binarized images, output of both the background subtractors are shown. It is worthy to note that SMM has comparable segmentation performance of GMM because it satisfies the two conditions in Sec. 3.3.2: in more complex situations (as a 2D image) the proposed technique loses its capability on describing the whole image, and its performance heavily decreases with respect to any pixel-based modeling algorithm (e.g., GMM).

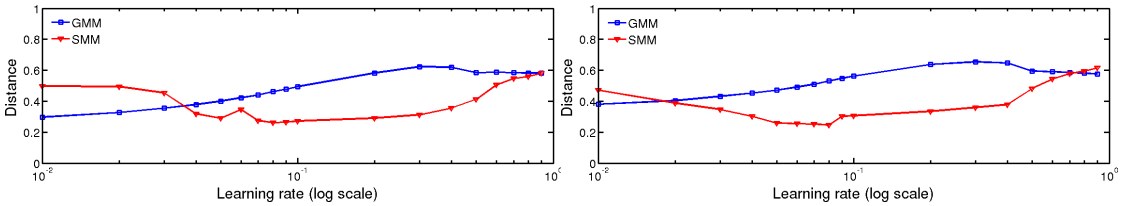
Focusing on SMM and considering *Test n. 2* (see Fig. 32b and 34), the minimum distance for P-R curves is met for higher learning rate than in other tests having comparable trends. These results are consistent with the peculiarity of the chosen dataset characterized by strong illumination variations. Thus, if on one hand the usage of larger learning rate means to "learn" quickly and then to filter the illumination changes, on the other hand it anticipates the objects absorptions, as it is shown in Fig. 34c (the red circle) and in Fig. 36c (the green circle). In Sec. 3.3.3 we demonstrate that the ghost avoidance algorithm fixes this problem, since a larger learning rate is used to update the background, and a smaller one to handle the objects.

Processing time comparison. To evaluate these results we select the SeedEye board: as described in Sec. 1.1.1 the PIC32, embedded on it, lacks of FPU, and implements the floating-point number using a software library. To enhance the processing capabilities and consequently satisfy the frame rate requirements, we have implemented the SMM technique following follow an integer-based implementation. In this direction all the parameters have been represented by integer variables: all the parameters of both g_r and g_c Gaussians are represented as *uint32_t* (i.e.,



(A) Test n. 1.

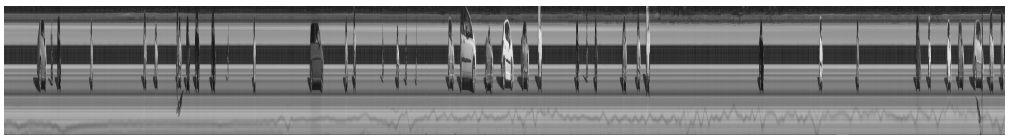
(B) Test n. 2.



(C) Test n. 3.

(D) Test n. 4.

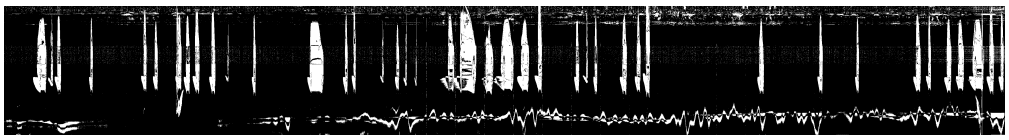
FIGURE 32. Performance comparison between SMM and GMM.



(A) Raw image.

(B) Ground truth.

(C) SMM (no ghost avoidance).

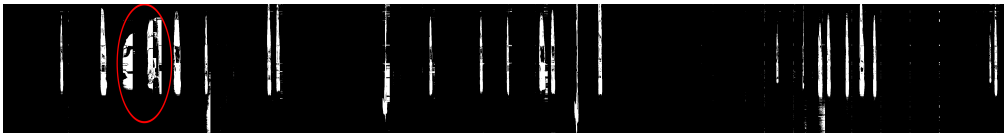


(D) GMM.

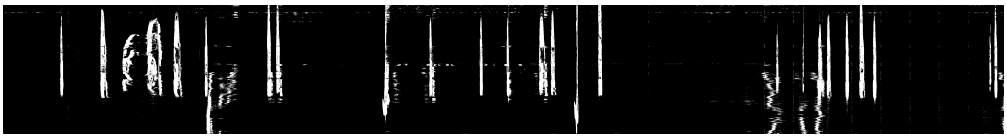
FIGURE 33. Test n. 1.

(A) Raw image.

(B) Ground truth.



(C) SMM (no ghost avoidance).



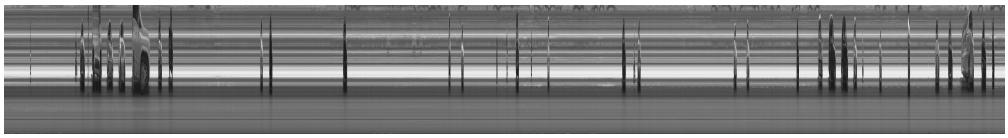
(D) GMM.

FIGURE 34. Test n. 2.

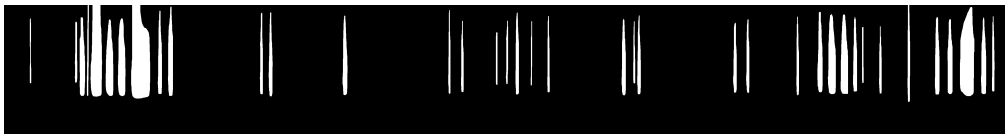
4 bytes) without any sensitive qualitative performance reduction with respect to a floating-point representation.

In Fig. 37 the SEED-EYE processing time distribution of both the considered techniques is shown. As we expected GMM is strongly slower than SMM because of its higher computational complexity and the usage of floating-point variables. Moreover in Tab. 2 the minimum, the average and the maximum processing times and the derived allowed maximum frame rates are shown for both the techniques. Considering the worst cases and using Eq. 8, it is possible to state that since the application based on GMM can monitor speed smaller or equal than $6km/h$, the SMM one can handle speed until $\sim 200km/h$: even the former has good segmentation capabilities, its processing time is to high and not acceptable compared with the normal vehicles speed; on the other side the latter has segmentation capabilities comparable with GMM, but very small processing time that permits use it on all types of road (in Italy the maximum speed allowed is $130km/h$).

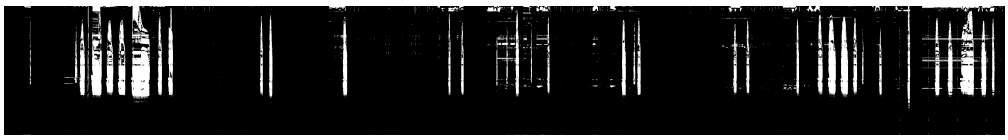
Ghost avoidance algorithm performance evaluation. In this section the effects of the ghost avoidance algorithm are detailed. To plot Fig. 38, we have computed the minimum P-R distance having fixed δ_h and varying δ_l: even the P-R distance trends for small values of δ_h (i.e., $\delta_h < 0.1$) are comparable with those obtained in Sec. 3.3.3, the ghost avoidance algorithm improves the



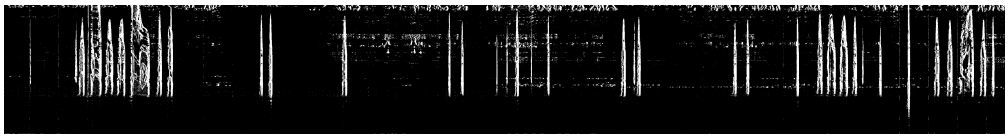
(A) Raw image.



(B) Ground truth.



(C) SMM (no ghost avoidance).

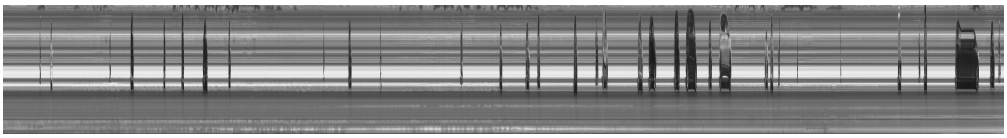


(D) GMM.

FIGURE 35. Test n. 3.

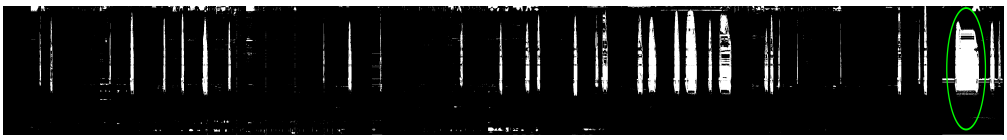
TABLE 2. Processing time and derived frame rates.

	Processing time [μs]			Frame rate [fps]		
	Min.	Avg.	Max.	Min.	Avg.	Max.
GMM	13927	18852	29809	∼72	∼53	∼34
SMM	235	262	1071	∼4255	∼3817	∼994

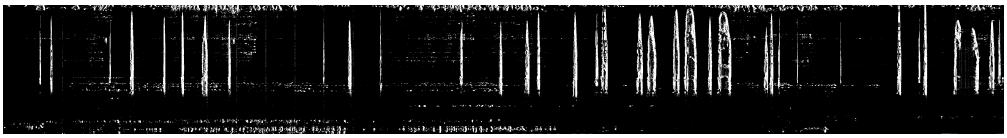


(A) Raw image.

(B) Ground truth.



(C) SMM (no ghost avoidance).



(D) GMM.

FIGURE 36. Test n. 4.

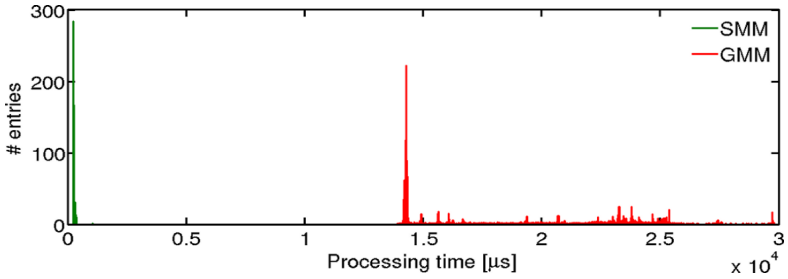
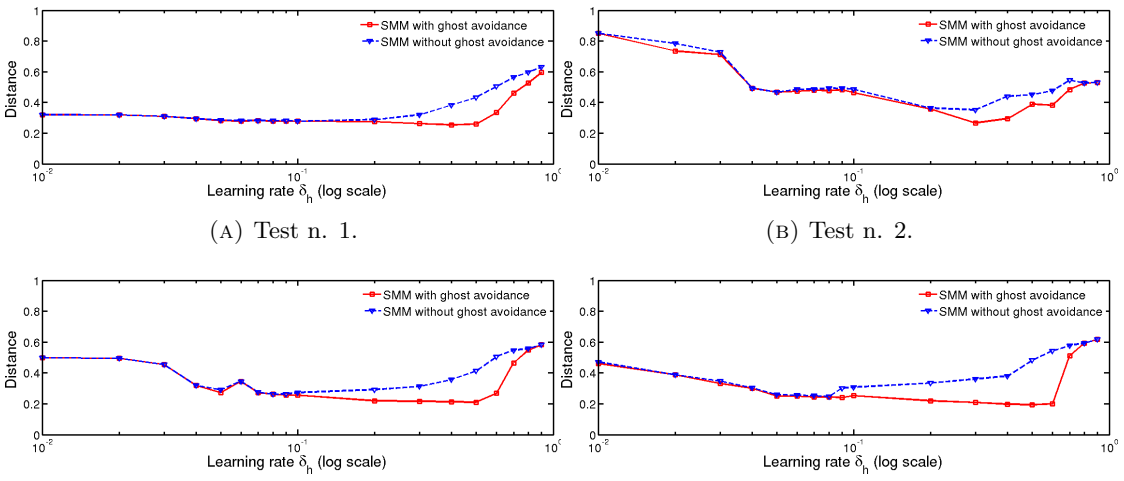


FIGURE 37. GMM and SMM processing time using the SEED-EYE board.



(A) Test n. 1. (B) Test n. 2.

(C) Test n. 3. (D) Test n. 4.

FIGURE 38. Performance evaluation comparison using SMM with and without ghost avoidance technique.

TABLE 3. Optimal learning rate values for the used datasets.

	Test n. 1	Test n. 2	Test n. 3	Test n. 4
δ_h	0.4	0.3	0.5	0.5
δ_h	0.1	0.05	0.01	0.07

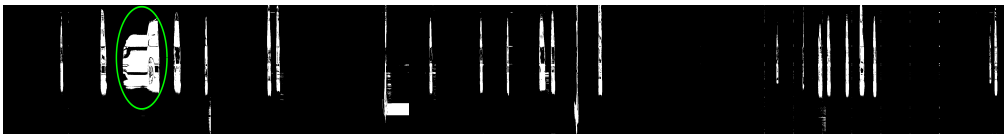
performance for higher values of δ_h (i.e., $\delta_h \geq 0.1$), reducing the minimum P-R distance and consequently enhancing the detection algorithm efficiency.

Moreover, considering Fig. 38 and Tab. 3, we confirm what we have discussed in Sec. 3.3.3; in fact the usage of two learning rates permits to: (i) make the background model robust to sudden illumination changes, because $d(i)$ is updated fastly (using δ_h) when no object is detected, and (ii) avoid/delay the ghosts by the use of the smaller learning rate δ_l.

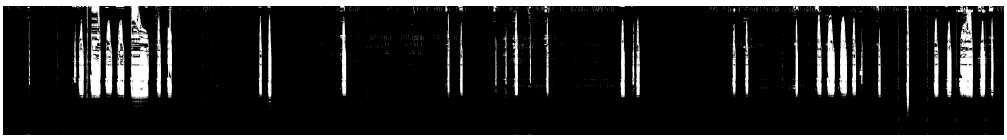
Finally in Fig. 39, the visual demonstration on what we have stated before is shown, by means of the reconstructed binarized images in output of the background subtraction algorithm. Particularly when no object is detected, the usage of higher learning rate than Sec. 3.3.3 heavily reduces the number of false positives pixels; otherwise when an object is detected, the ghost phenomena are avoided/delayed (compare the green circle in Fig. 39b with the red one in Fig. 34c).

Vehicles counter performance. In this section the performance of the vehicles counter application is evaluated in terms of number of detected objects compared with the number counted by a human operator (ground-truth). To permit this, the optimal value of δ_h and δ_l are retrieved by minimising the average of the results of Sec. 3.3.3: configuring the application with these parameters we obtain the results shown in Fig. 40.

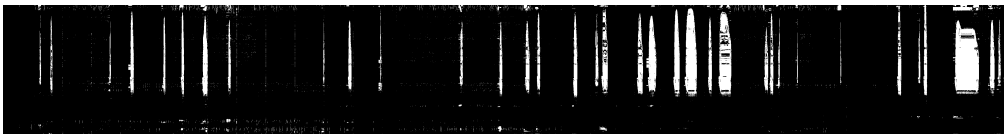
(A) Test n. 1.



(B) Test n. 2.



(C) Test n. 3.



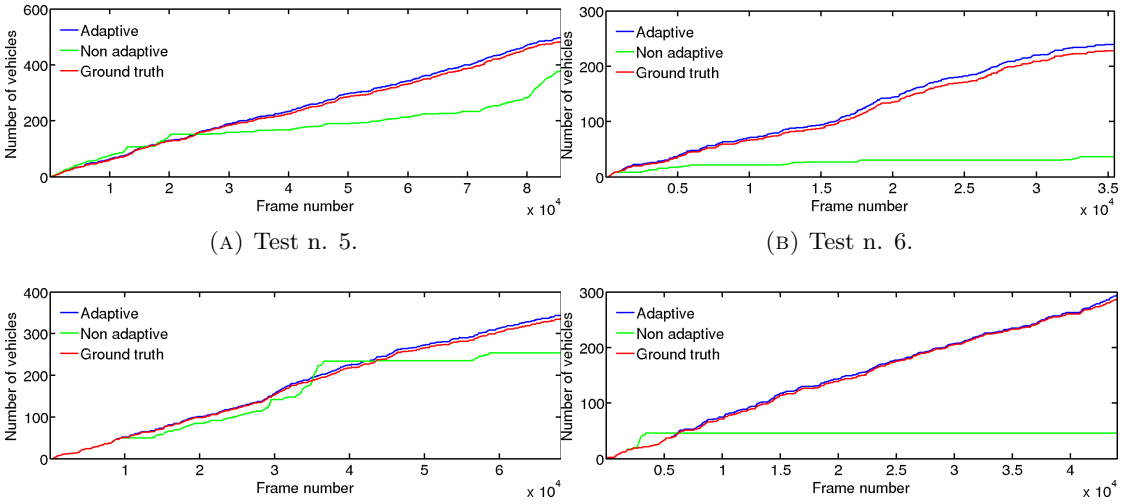
(D) Test n. 4.

FIGURE 39. Binarized images using SMM and the ghost avoidance technique.

TABLE 4. Measurement error ϵ.

	N_M	N_{GT}	ϵ [%]
Test n. 5	496	482	2.9
Test n. 6	239	228	4.8
Test n. 7	344	335	2.7
Test n. 8	294	287	2.4

In Tab. 4, the measurement error ϵ is computed for each dataset: the proposed application based on SMM and ghost avoidance algorithms introduces a very small error in counting vehicles with respect to the ground-truth (less than 5%) confirming the results obtained in the previous sections about the background-foreground segmentation.



(A) Test n. 5.

(B) Test n. 6.

(C) Test n. 7.

(D) Test n. 8.

FIGURE 40. Number of vehicles counted using Linesensor compared with the ground-truth.

CHAPTER 4

Video surveillance advanced applications

VIDEO streaming in SCN is a promising and challenging application to enable high-value services, especially if the low-power and low-bandwidth IEEE802.15.4 standard [**st103**] is considered. In such a context the reduced amount of available bandwidth, as well as the low-computational power available for acquiring and processing video frames on tiny devices, imposes the transmission of low resolution video frames at a low frame rate. Considering the aforementioned limitations, the amount of information carried by each video frame must be considered of utmost importance and preserved, as much as possible, against network losses that could introduce possible artifacts in the reconstructed dynamics of the scene.

As first, in Sec. 4.3, the effects of a noisy network are evaluated over raw images to understand the advantages and the drawbacks of such a solution ([**PGS$^+$11**]). In fact, if in one side the uncorrelated representation of the pixels in raw images provides to them an innate error resiliency, in the other the large amount of data to be sent congests the low bandwidth network reducing considerably the feasible frame-rate. Moreover, the impact of a low complexity Forward Error Correction (FEC) technique (i.e., BCH codes) and three low-complexity concealment algorithms are evaluated to enhance the quality of the received video stream.

Finally, in Sec. 4.4, a low complexity codec oriented to micro-controllers technology is presented (see [**SPP$^+$12**, **SPM$^+$13**]): this codec exploits all the advantages of using raw images representation, but reduces the amount of data to be sent considering only the zones of the images where a change is detected. The simulative performance of this codec will be evaluated supposing to use the 6LoWPAN communication protocol, as described in Chap. 2.

4.1. Background

Considering a SCN based on the IEEE802.15.4 standard, a very big issue is that of matching with a transmission bandwidth equal to 250 Kbps at physical (PHY) layer and even lower at the Medium Access Control (MAC) layer (e.g., 163 Kbps in case of unslotted based transmissions [**LDMM$^+$06**] and slightly bigger values in case of slotted based transmissions [**PKC$^+$05**])). This limitation implies the transmission of small size images (e.g., 320x240, 160x120) at very low frame rates to avoid network congestion [**CJAS07**]. For these reasons, one of the most important issue in image streaming is the definition and the realisation of video compressors: in traditional video streaming applications high-end devices are used to compress and send multimedia contents by means of of state-of-the-art video encoders, such as MPEG-2 (see [**ISO96**]) and H.264 (see [**IT03**]), so that the best trade-off between video quality and bandwidth reduction

is achieved. The aforementioned standards rely on time-consuming and on demanding processing algorithms which are not supported by the considered SCN devices, characterized by low processing power, reduced memory availability and limited energy budgets. In such a scenario, proposed video compression techniques are based on simple image compression schemes with possible optimizations based on a differential coding (see [**MRG08**]). In [**CM02**] the use of JPEG [**ISO92**] and JPEG2000 [**ISO04**] standards with a fixed point Discrete Cosine Transform (DCT) implementation, instead of the commonly used floating point, has been preliminarily investigated in respect to the image quality and node energy consumption. The paper shows as a fixed point DCT implementation permits, in the case of JPEG standard, to lower the energy consumption in the image compression process, thus showing as it is possible to adopt the JPEG compression in a SCNs context. An improvement of this work is presented in [**MMM03**] where a video compression solution based on a change detection approach specifically developed for JPEG compression is adopted. The use of JPEG and JPEG2000 standards in IEEE802.15.4 networks has been addressed in [**PSOB05**] where both compression schemes have been tested in a real scenario, while highlighting network limitations and evaluating transmission performance in terms of Peak Signal to Noise Ratio (PSNR) and byte error rate parameters. A JPEG-lossless coding in IEEE802.15.4 network is analyzed in [**ZLY+10**] where it is shown how it is able to reach better PSNR with respect to JPEG in case of an enhanced device node architecture for acquiring and processing images is adopted. The use of a JPEG-like compression technique targeted to video streaming applications in IEEE802.15.4 networks has been investigated in [**PBR+11**] where a new hybrid DPCM/DCT coding scheme is proposed and adopted to achieve an acceptable compression gain with a low computational complexity.

The use of the JPEG compression standard in tiny mote devices has been experimentally evaluated in [**MATB11**], where the JPEG implementation provided by the Independent JPEG Group (IJG) [**IJG**] has been implemented on the Imote2 [**NJS+08**] platform equipped with the Imote2 Multimedia Sensor Board. In case of a 320x240 gray scale images, and a working clock frequency of 13 MHz, the compression algorithm requires up to 500 ms to compress the image for any quality factor, thus allowing up to 2 fps in the video stream. Better compression performance has been reached in [**PBR+11**], where on the same platform up to 2.7 fps can be enabled. Moreover, considering the transmission bandwidth limitations of the IEEE802.15.4 standard the maximum allowable frame rate decreases in both cases to values lower than 1 fps. This last value has been obtained without considering the use of channel coding schemes, such as forward error correction (FEC) or erasure correction (EC), necessary to overcome data losses in DCT coefficients at the cost of further reducing the frame rate.

The biggest issue in using JPEG in low-end SCNs is the lack of error resilience properties [**ZEP+04**] that can be overcome by Forward Error Correction (FEC) [**LL81**] techniques, Erasure Correction (EC) codes [**MS04**], Interleaving scheme [**DFL08**] and Variable-Length Coding (VLC) [**DFLL11**]. The same issues can be experienced in other JPEG-based compression approaches [**HBY08**], as well as solutions in which a coding scheme is performed by using frequency transformation functions, such as in [**PBR+11**] where a new hybrid DPCM/DCT coding

scheme is proposed to achieve an acceptable compression gain with a low computational complexity, and [**WPW+08**] where a wavelet based coding technique is proposed and its performance jointly evaluated with Unequal Error Protection (UEP) [**MHPS12**] techniques.

4.2. Data collection scenario and datasets

The scenario selected to collect real communication traces between sensor network devices is the testing area of the IPERMOB Italian project [**ipe09**] consisting of an outdoor parking lot and its accesses in the Pisa International Airport land-side. The full data collection scenario with nodes positions and obstacles is depicted in Fig. 1, where the numbered dots represent the sender nodes and the receiver is marked with the C letter. The network is organized in a star topology. The application scenario is that of a typical video surveillance system in which a set of nodes, six in this case, are installed to monitor malicious events inside the parking. Each node of the system can in turn acquire and send an image to the network coordinator, the C node, which works as a point of service for the backhauling network. The data collection environment is heavily affected by reflections as well as permanent and temporary Non-Line-Of-Sight (NLOS) transmission conditions due to trees and moving cars.

FIGURE 1. The data collection scenario within the Pisa International Airport landside.

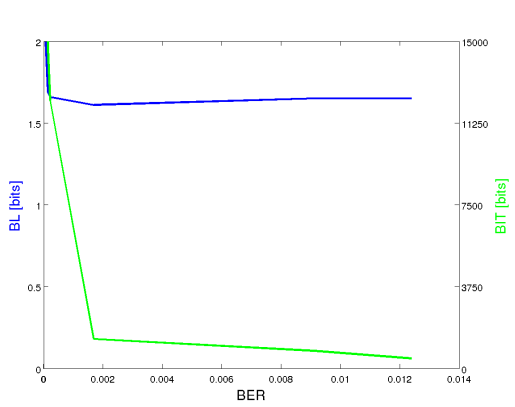
In all the performed experiments the hardware devices have been installed at 2.5m height and IEEE802.15.4 data packets of maximum dimension have been sent at the maximum bitrate collecting a dataset of three traces, each one consisting of nine thousands packets, for each position. The transmission power of sending devices has been set equal to +9 dBm to fulfill the ETSI requirements [**ETS06**]. The results of the experimental analysis are reported, for each position, in Tab. 1 in terms of BER, Burst Length (BL), Burst Inter-arrival Time (BIT), Link

Quality Indicator (LQI) and Received Signal Strength Indicator (RSSI). All results in the Tab. 1 have been averaged among the three collected traces in order to perform an overall comparison among data collected in different positions.

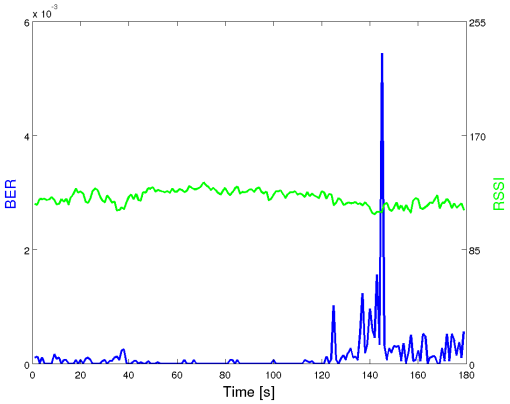
Position	Distance [m]	BER	BL [bits]	BIT [bits]	LQI	RSSI
1	64	$2.18 \cdot 10^{-4}$	1.66	12282.91	108.38	120.97
2	53	$1.69 \cdot 10^{-3}$	1.61	1347.91	99.86	122.09
3	43	$6.56 \cdot 10^{-7}$	2.23	1615032.00	116.38	147.88
4	68	$1.43 \cdot 10^{-4}$	1.69	14688.46	110.86	131.14
5	57	$1.24 \cdot 10^{-2}$	1.65	453.32	93.97	104.18
6	43	$9.12 \cdot 10^{-3}$	1.65	800.79	94.61	121.96

TABLE 1. Data collection analysis results for the selected positions in the airport scenario.

The complexity of the selected scenario is reflected by the high variability of the BER values spanning in a range from 10^{-2} to 10^{-7}. While higher BER values are expected for a larger distance between sender and receiver, when a Line-Of-Sight (LOS) transmission is performed (e.g., positions 1 and 3), the presence of moving obstacles causes unpredictable effects on the BER. Communications in NLOS conditions make impossible to correlate the BER with the transmission distance (e.g., positions 4 and 5). NLOS communications in the adopted scenario have been experienced to be permanent and temporary. While a permanent obstruction results in BER values dramatically high, as it happens for positions 5 and 6, which can be in any case mitigated with a careful sensor planning activity, temporary NLOS communications are totally unpredictable and associated to a sudden BER increment. As a matter of example, during the experimental data collection a trace collected in position 2, and afterwards discarded, has been acquired in condition of partial NLOS caused by a truck stopping for half of the data collection time in the middle of the transmission path, so that the associated BER sharply increased showing a final value ten times bigger than the one experienced by previous traces collected in the same position. Regarding the BL this is independent from the BER and close to 1.6 for each position. On the contrary, the averaged BIT is strictly correlated to BER, showing as bigger bit error values reduce the inter-arrival time among burst errors. The described behavior for both BL and BIT is shown graphically in Fig. 2a. Concerning the LQI and the RSSI, the BER is dependent on the LQI, when the LQI increases the BER decreases, while the RSSI is not simply related to the BER because its value is the sum of both effective signal and its reflections, higher values of RSSI do not coincide with lower values of BER (e.g., positions 1 and 2). The BER and RSSI behaviors as a function of the time are reported in Fig. 2b for a trace collected in position 1. In the picture it can be seen how peaks in the BER values can be experienced even if the RSSI remains close to its average value.



(A) BL and BIT as a function of the BER.



(B) BER and RSSI behaviors as a function of the time for a collected trace.

FIGURE 2. Network quality metrics comparisons.

On the other side the images adopted in the performed simulations belong to the IPERDS [**IPE11**] dataset created within the IPERMOB project. IPERDS is basically a collection of images related to traffic and parking lots conditions and hence characterized by slow and high motion. Each cataloged set of images is a video trace of more than 5 minutes with a frame rate equal to 1 fps. The dataset is composed of gray scale images with a size equal to 160x120 and 320x240 pixels. All the images composing the dataset have been collected by using real wireless sensor network devices equipped with a low-cost camera, hence they have all the necessary characteristics to prototype video streaming algorithms targeted to low-end devices.

4.3. Streaming of raw images

This section considers a contingent condition where image compression techniques cannot be applied due to their computational complexity, expensive computation time [**PBR+11**, **MATB11**] and device memory limitations (a typical low-complexity and low-memory SC has a RAM size of around 30KBytes): in these situations only raw images must be sent, thus reducing again the image size, the frame rate or both of them. Taking into account these limitations, it is important to stress that when reducing the frame rate and image size of a video stream, each video frame acquires a greater significance due to the lower redundancy in the information content it carries (e.g., a malicious behavior must be discovered using a smaller sequence of images). As a consequence, a reliable video streaming transmission must be ensured to overcome the effects of those variations of the wireless channel able to cause unexpected errors in the transmitted bits with consequently losses of video frame portions. In the literature two main classes of data protection mechanisms are traditionally employed to mitigate the unreliability of the wireless channel at the PHY and MAC layer, namely Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC). The former mechanisms use bandwidth efficiently, retransmitting data upon request, at the cost of additional latency, thus resulting unfeasible or strongly limiting for

applications requiring real-time delivery of multimedia content. In FEC techniques, instead, the data to be sent through the network are protected a priori, thus requiring only additional bandwidth, without additional delays, to reconstruct an image at the receiver side. FEC techniques in a wireless sensor network scenario have been started to explore in the last years, by proposing both Unequal Error Protection (UEP) [**WHP+10**] and cross-layer approaches [**YYY10**] targeted to compressed video frames and without considering transmission standard compatibility issues (e.g., MAC layer limitations).

In this section we propose and evaluate the performance of forward error correction techniques based on BCH codes [**Moo05**] for recovering bit errors in case of uncompressed images based multimedia streaming over wireless sensor networks compliant with the IEEE802.15.4 standard. In a real outdoor scenario we first measure the Bit Error Rate (BER) for one-hop transmissions, star topology based networks, and we show its impact on the video streaming quality in terms of Peak Signal-to-Noise Ratio (PSNR) [**SG07**]. Then we detail the use of the BCH codes as a solution fully compliant with the IEEE802.15.4 standard, presenting a simulated performance evaluation, based on the collected real loss traces, of its capabilities in recovering bit errors when codes with different error correction capabilities are adopted.

4.3.1. Impact of bit errors on video quality. In the simulations each image is fragmented according to the maximum payload allowed by the IEEE802.15.4 standard, and an image fragment is considered lost if at least one bit error occurs in the received data packet, thus being fully compliant with the transmission standard in which packets with wrong Frame Check Sequence (FCS) values are discarded at the receiver side. In the presented results an evaluation of the impact of three low-complexity concealment algorithms able to recover image data losses is shown. The considered concealment techniques will be identified in the following as *black insertion*, *white insertion* and *copy frame*. In case of black insertion concealment the lost fragment of a video frame is replaced with black pixels, while in case of white insertion concealment the color of the replaced pixels is white. The copy frame concealment is little more complicated with respect to the previous ones and it consists in replacing the lost fragments of a video frame with the ones of the last received frame. The BER impact on video quality has been evaluated in terms of PSNR for each one of the proposed concealment techniques, and results reported in Tab. 2 for a numeric comparison. In the table the PSNR in case of black concealment is labeled as PSNR-bc, the one for the white concealment as PSNR-wc, while PSNR-cc is the video quality when the copy frame concealment is applied.

From results reported in Tab. 2 it is possible to see how the PSNR is directly related to the BER experienced in the transmission link. Higher values of BER produce low values of PSNR for each concealment technique considered in the simulation process. In case of BER values higher than 10^{-3} the video quality reduction is bigger than 60% with respect to the PSNR value experienced with a BER equal to $6.56 \cdot 10^{-7}$, while when the BER is slightly higher than 10^{-4} the video quality reduction is bigger than 35%. Although a BER equal to 10^{-4} cannot be considered as a critical value for wireless communications based on the IEEE802.15.4 standard [**SAB+07**], it produces a substantial quality reduction of the received video stream. The selection of an

Position	BER	PSNR-bc [dB]	PSNR-wc [dB]	PSNR-cc [dB]
1	$2.18 \cdot 10^{-4}$	42.57	43.40	56.80
2	$1.69 \cdot 10^{-3}$	19.14	20.08	35.88
3	$6.56 \cdot 10^{-7}$	89.91	89.96	90.25
4	$1.43 \cdot 10^{-4}$	53.84	54.45	62.35
5	$1.24 \cdot 10^{-2}$	11.86	14.53	27.84
6	$9.12 \cdot 10^{-3}$	13.35	16.23	31.82

TABLE 2. PSNR as a function of the BER and concealment techniques.

appropriate concealment technique mitigates the effects of the packet losses, guaranteeing a gain in the quality of the received video. More in particular, the copy frame concealment outperforms all the other concealment techniques under test, reaching a substantial gain in video quality for each position affected by BER larger than $6.56 \cdot 10^{-7}$. A graphical output of the applied concealment solutions is depicted in Fig. 5, where the frame with copy frame concealment is almost completely reconstructed.

In a SCN scenario, in which tiny devices send image frames with low resolution and low frame rate, error recovery techniques must be applied in order to avoid poor video quality and possible artifacts in the reconstructed dynamics of the scene at the receiver side, as it could happen in video surveillance systems.

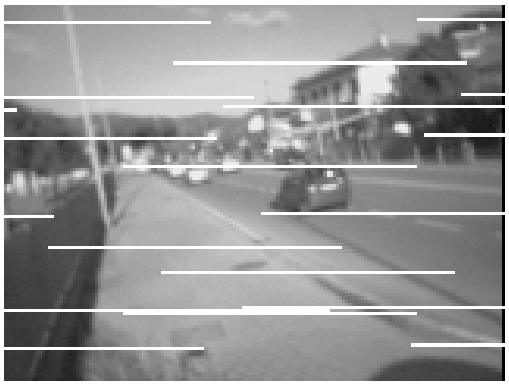
4.3.2. BCH codes based error recovery strategy. In this section the use of BCH codes in wireless multimedia sensor networks is considered as a possible FEC strategy for recovering bit errors. According to the coding theory, the BCH codes is a class of error-correcting block codes in which the coding and decoding procedures are characterized by low complexity and low power consumption [**BYJK07**], thus making these codes very suitable for a real implementation in low-end devices. In a very simplistic statement the aim of the block coding process is to add redundancy bits to the initial block of bits which constitute the information to be transmitted, thus providing the capability of correcting a certain number of errors caused by channel variability. Each BCH code is characterized by three main parameters:

- n: the total number of bits after the coding procedure. Its value is given by the number of information bits plus the number of redundancy bits;
- k: is the number of the information bits which must be protected ($k < n$);
- t: is the error correction capacity of the code. Each BCH code can correct up to t errors in each block on n bits, while adding $n - k$ bits of redundancy.

In literature a BCH code is identified by the above introduced parameters with the labeling $BCH(n, k, t)$. The value $Rc = k/n$ is the code rate and is related to the redundancy level and overhead introduced by the code. Lower Rc values mean higher protection levels and higher additional overhead in terms of redundancy bits to transmit.

(A) Reference image.

(B) Black concealment.

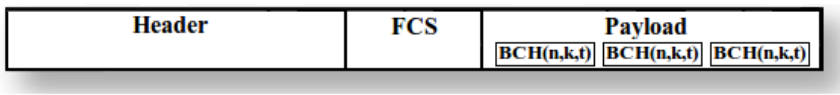


(C) White concealment.

(D) Copy concealment.

FIGURE 3. Visual impact of the selected concealment algorithms for a selected loss trace.

(A) Standard IEEE802.15.4 data message.



(B) Proposed MAC data message with FEC protection.

FIGURE 4. Standard MAC data message and proposed solution for error recovery.

The use of a FEC strategies based on BCH codes in sensor networks compliant with the IEEE802.15.4 standard requires to define new policies in accepting corrupted packets at the

receiver. According to the standard, a transmitted data packet is composed, at the MAC layer, by three main fields: header, payload and a frame check sequence, which is evaluated on the two previous fields (Fig. 4a). Once a packet is received from a network node, a new FCS is evaluated and compared with the one encapsulated inside the packet by the sender. If the two FCS are identical the packet is accepted and transferred to the upper layers of the network stack, otherwise it is discarded and the carried information is lost. This approach cannot be pursued when FEC strategies based on error correction codes are applied, because it would result in neglecting the effects of the protection strategy itself. The approach proposed to effectively apply BCH codes based FEC strategies within IEEE802.15.4 networks is depicted in Fig. 4b. The FCS field is evaluated only on the packet header, thus avoiding to deliver packets to wrong nodes, while in the payload no check on the correctness of the data is applied. In the payload, instead, the n bits of each BCH code word are considered and the k bits of the original information are extracted. The error correction capacity of the code guarantees the correction of up to t errors for each block on n bits. If a higher number of errors is experienced these cannot be revealed and corrected. The proposed protection strategy is especially indicated for the transmission of multimedia data in which the residual errors do not affect the validity of the full packet, but slightly affect the quality of the received media stream.

The proposed FEC strategy has been specially designed to avoid to change the MAC layer of the IEEE802.15.4 standard for a specific application. The new data messages, in fact, can be defined using the bits reserved from the standard for specifying new possible messages. The basic messages of the IEEE802.15.4 standard, as well as the reserved bits combinations are reported in Tab. 3. The use of the reserved bits to extend the types of data messages does not impose to change the network firmware of the sensor devices, although only the nodes with the new features will be able to use those messages.

Frame type value (b2, b1, b0)	Description
000	Beacon
001	Data
010	Acknowledgment
011	MAC command
100-111	Reserved

TABLE 3. Standard IEEE802.15.4 messages at MAC layer.

4.3.3. Performance evaluation. The performance of the proposed error recovery strategy has been evaluated with a simulative study as a function of the error correction capacity of the code (t) and its code rate (Rc). In the performed simulations we have considered the noise trace collected in position 1 (see Sec. 4.2) and IPERDS datasets. Moreover we consider only the copy frame concealment due to its better performance in terms of PSNR with respect to black and

white insertion concealments, as it is shown in Sec. 4.3.1. All results of the performed analysis are reported in Tab. 4, where for each code, together with the code rate, the additional overhead requested by the protection technique, the percentage of packets discarded due to wrong headers, the percentage of recovered errors and the obtained PSNR value is reported. As it is possible to see in Tab. 4, four main classes of BCH codes have been considered, with n equal to 255, 127, 63 and 31 bits respectively, and for each one of them several correction capacities have been tested. The overhead introduced by the code, and reported in the third column of the table has been evaluated as the additional number of bits required to send the video frames, including the headers of additional necessary packets. In fact, this parameter is affected not only by the additional redundancy bits imposed by the adopted code but also by the IEEE802.15.4 packet dimension limitation that imposes to fragment each video frame in a bigger number of packets with respect to the case in which no protection is applied. The first row entry of the table, labeled as No protection, reports the video quality performance when the standard IEEE802.15.4 transmission mechanism, depicted in Fig. 3a, is applied, and it is the same value for the position 1 in Tab. 1.

A first main effect of the proposed error recovery mechanism is the reduction of the number of discarded packets, from 4.06% to values lower than 1.26%, at the cost of accepting possible bit errors which cannot be correct with the selected BCH code. Analyzing the video quality for a single class of codes, it is possible to see how it mainly depends on the error correction capacity of the code. Increasing the error correction capacity a higher percentage of errors is recovered, thus increasing the PSNR. This behavior is depicted by means of graphs in Fig. 5 and graphically in Fig. 6, and it is common to all the four classes of codes analyzed. Low percentage of errors are recovered with $t = 1$, despite the high value of the additional overhead, this is because the code cannot recover from all the burst error, which is equal to 1.66 on average. With $t \geq 2$ the improvement in percentage of recovered errors, and in PSNR, becomes more significant.

To analyze the benefit of a class of codes with respect to the other ones we considered the code rate parameter. Considering a fixed value of error correction capacity the general trend in the PSNR values is that of reaching higher quality as much as the code rate decreases. This behavior is evident for t equal to 1 and 5, while in case of t equal to 2, 3 and 4 it is not directly observable due to lower PSNR values reached by the class of $BCH(63, k, t)$ codes in which a higher percentage of discarded packets with respect to the other class of codes has been experienced. This behavior can be seen graphically in Fig. 5 where the curve related to the $BCH(63, k, t)$ codes is positioned below the one related to the $BCH(127, k, t)$ codes.

The use of the proposed protection technique for video streaming over IEEE802.15.4 network guarantees significant performance improvements in terms of PSNR at the cost of an increased overhead. Considering BCH codes with error correction capacity larger than the average burst length and minimum overhead, $BCH(127, 113, 2)$, the performance improvement in terms of PSNR is equal to 4.16dB at the cost of an increased overhead of 22.51%. When larger transmission bandwidth overhead can be tolerated, higher values of PSNR can be reached by using BCH codes with higher values of error correction capacity and lower values of code rate. In the presented results, considering the maximum error correction capacity and lowest code rate,

Protection	Rc	Overhead [%]	Packet discarded [%]	Error recovered in the payload [%]	PSNR-cc [dB]
No protection	1	0	4.06	0	56.80
$BCH(255, 247, 1)$	0.97	19.31	1.21	2.04	57.96
$BCH(255, 239, 2)$	0.94	23.14	1.21	15.66	59.22
$BCH(255, 231, 3)$	0.91	27.36	1.21	27.95	60.97
$BCH(255, 223, 4)$	0.87	32.09	1.21	31.27	61.46
$BCH(255, 215, 5)$	0.84	37.08	1.21	35.54	61.50
$BCH(127, 120, 1)$	0.94	15.30	1.22	2.06	58.26
$BCH(127, 113, 2)$	0.89	22.51	1.22	16.69	60.96
$BCH(127, 106, 3)$	0.83	30.61	1.22	33.96	65.19
$BCH(127, 99, 4)$	0.78	39.74	1.22	38.92	65.76
$BCH(127, 92, 5)$	0.72	50.08	1.22	41.59	65.93
$BCH(63, 57, 1)$	0.90	17.94	1.26	2.76	58.42
$BCH(63, 51, 2)$	0.81	31.90	1.26	19.62	60.02
$BCH(63, 45, 3)$	0.71	40.74	1.26	40.44	63.25
$BCH(63, 39, 4)$	0.62	71.10	1.26	45.63	63.73
$BCH(63, 36, 5)$	0.57	85.44	1.26	47.83	67.47
$BCH(31, 26, 1)$	0.84	24.10	1.15	3.65	58.43
$BCH(31, 21, 2)$	0.68	53.25	1.15	20.34	62.39
$BCH(31, 16, 3)$	0.52	100.98	1.15	39.26	66.68
$BCH(31, 11, 5)$	0.35	192.42	1.15	46.99	68.05

TABLE 4. Video quality performance results for the four classes of BCH codes selected.

$BCH(31, 11, 5)$, the gain in PSNR is equal to 11.25dB with an exaggerated additional overhead of 192.42% in number of transmitted bits. Although the protection level must be chosen according to the desired improvement in terms of video quality, the minimum suggested error capacity correction to be used is equal to 2, due to the capability of such codes to recover burst errors bigger than the ones experienced in the collected traces and equal to 1.66 in average.

4.4. Streaming of raw image RoIs

State-of-the-art video-surveillance applications are based on a streaming service between a static camera and a remote control point; the service is required to fulfill the temporal constraints usually defined in terms of frame rate. The minimum frame rate required by such applications is

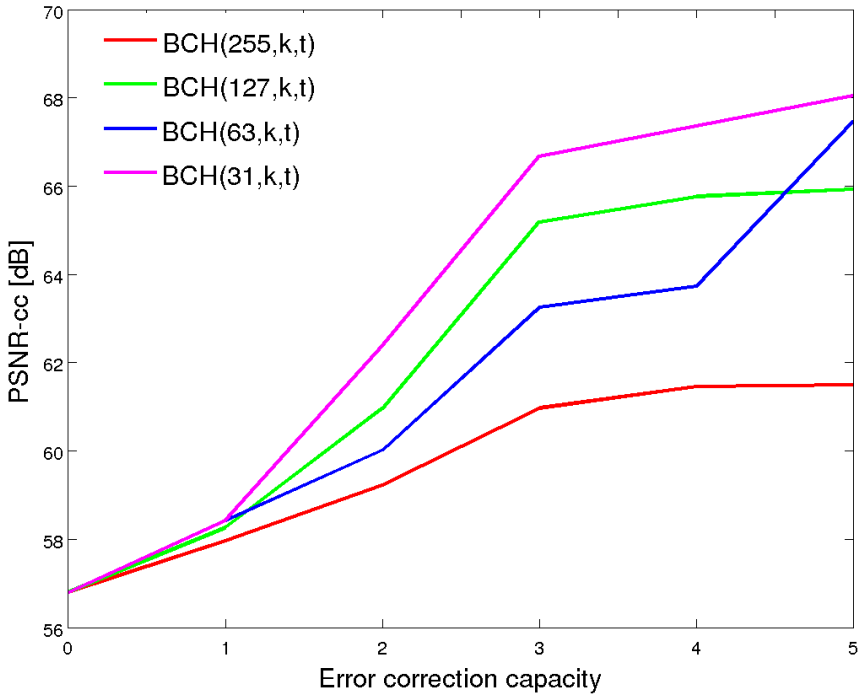


FIGURE 5. PSNR as a function of the error correction capacity of the code.

about 1 fps. The diagram shown in Fig. 7 describes the main functional blocks, and their relations, in a standard video-surveillance application based on a streaming service. In this situation, on the transmitter side, the images acquired by a camera network devices are compressed using a *video-encoder* and fragmented in packets to be transmitted through the wireless network. At the receiver side, the received packets are aggregated in order to decompress the images (using a *video-decoder*) to be shown on the user screen.

As it is discussed in [**PGS$^+$11**], a single stream, at a frame rate of 1 fps and QQ-VGA resolution (i.e., 160×120 pixels), fully saturates the available bandwidth in IEEE802.15.4 networks. In the same conditions, considering the 6LoWPAN overhead, a single stream would not fit the temporal constraints as imposed by the minimum required frame rate.

As previously introduced, JPEG and Lossless JPEG are state-of-the-art solutions for compressing streams in SCNs. Although the above mentioned compressors offer optimal performance in terms of compression ratio, they are characterized by high computational complexity ($O(NlogN)$) and lack of error resilience [**MRG08**]. The latter descends from the strong correlation that the compressed blocks have with each other. Consequently the scientific objective we address deals with the realization of (i) a low-complexity video codec and (ii) a data protection schema showing a stronger error resilience than state-of-the-art solutions. As a further improvement, three types of concealment are applied and compared against an improved version of JPEG.

(A) Reference image.

(B) No FEC applied.

(C) FEC applied with $t = 1$.

(D) FEC applied with $t = 3$.

FIGURE 6. Visual impact of the selected concealment algorithms for a selected loss trace.

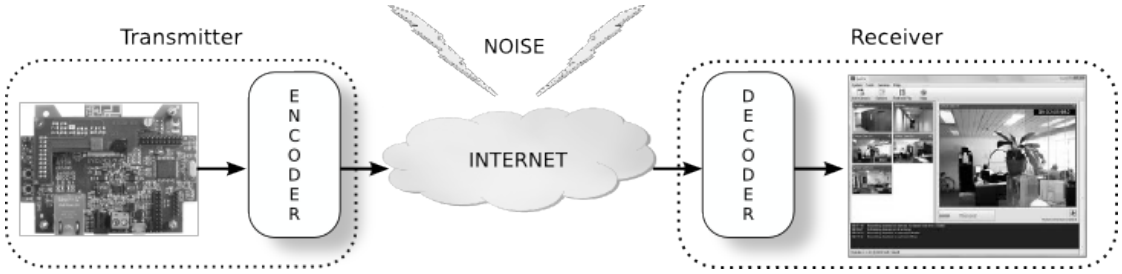


FIGURE 7. Video-surveillance functional block diagram.

4.4.1. The video codec. The considered compression algorithm is based on the classification of each pixel of an image in terms of its information content. More specifically it is possible to label as *background* the static component of a monitored scene, and as *foreground* its dynamic component. Thus, a compression algorithm based on background subtraction is eligible for low

rate networks (as SCNs) since it allows to transmit only the relevant parts (also called Regions Of Interest, RoIs) of the considered raw image. When a background subtraction approach is adopted, a background updating rule is needed to make robust the compression algorithm to luminosity variations and once-off changes. We deterministically apply the following logic: at a certain rate (called *Background Refresh Period*, BRP, and measured in number of frames) a raw frame is labeled as local background image and is entirely transmitted to the control point to keep it synchronized and updated.

From a high level point of view, the *video-encoder* is required to detect, for each frame, the occurred changes with respect to a background image, discarding the parts of the frame without information content (inter-frame compression). The *video-decoder*, instead, is in charge of reconstructing the frames starting from the received parts of the image (RoIs) and the last received background image. Along with the raw pixels content, a header descriptor is associated to each RoI introducing a certain overhead in the proposed approach. For instance, in order to properly position a rectangular RoI, the descriptor should contain the image identifier (e.g., 1 byte), x and y coordinates of the left corner (e.g., 2 bytes each), width and height dimensions (e.g., 2 bytes each). In this example the overhead amounts to 9 bytes per RoI.

The diagram in Fig. 8 shows the main functional blocks of the video-encoder. Accordingly, in the *blob segmentation phase*, an operation of pixel-wise subtraction of the current frame with respect to the background one is performed: the output of this operation is called *difference image*. The background component inside the difference image are characterized by pixel values close to zero (darkest pixels), while the foreground shows bigger pixel values (clearest pixels). To distinguish background regions from foreground, the difference image is thresholded and denoised using morphological operators (i.e., opening) in order to generate a *binarized image*. In this latter image the background component is represented using black pixels, and the foreground with white. Then, in the *blob extraction phase*, the foreground pixels of the *binarized image* are grouped in order to understand the presence of relevant objects (in terms of occupied area) and the obtained RoIs are represented using rectangular bounding-boxes. Finally, the RoIs are extracted from the *difference image* and sent through the network. Background image update follows the periodicity discussed above, in order to decorrelate the protocol to the light variation and once-off changes. The whole encoder algorithm is reported as pseudo-code in Algorithm 2.

On the other side, in the decoding phase, the *image reconstruction* process is performed by inserting the received RoIs of the i-th image frame on top of the last received background (by performing a pixel-wise sum operation), as it is shown in Fig. 9. The decoder algorithm is reported as pseudo-code in Algorithm 3. The decoding phase has been kept as low-complexity as possible in order to permits to reconstruct more video flows in the remote control point.

4.4.2. Codec performance assessment and comparison against JPEG. The codec performance has been evaluated against the metrics reported in Tab. 5 by considering four IPERDS traces at a QQ-VGA resolution and characterized by low and high motion. Tab. 6 shows the overall performance in terms of data rate, compression ratio and PSNR, as a function of the background refresh period expressed in number of frames, a graphical representation is reported

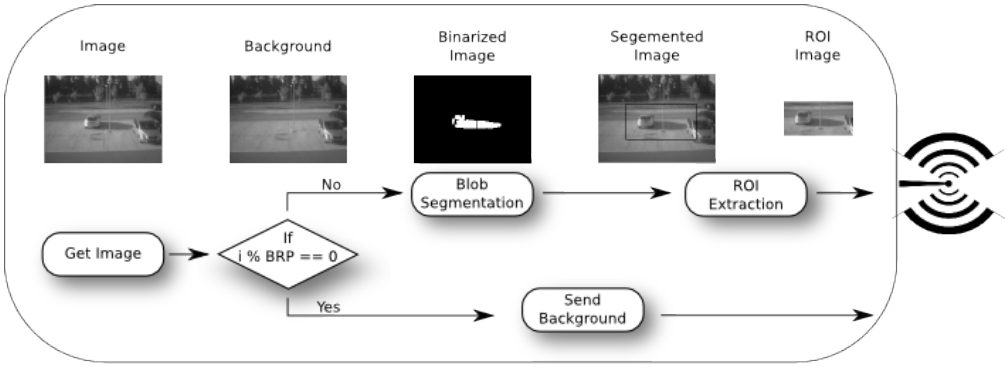


FIGURE 8. The video-encoder.

Algorithm 2 Encoder.

```
Frame_id = 0;
while 1 do
  get_image(image);
  if (Frame_id mod BRP) == 0 then
    send_image(image);
    bgnd = set_background(image);
  else
    diff = blob_segmentation(image, bgnd);
    roi = roi_extraction(diff);
    send_roi(roi, header);
  end if
  Frame_id = Frame_id+1;
end while
```

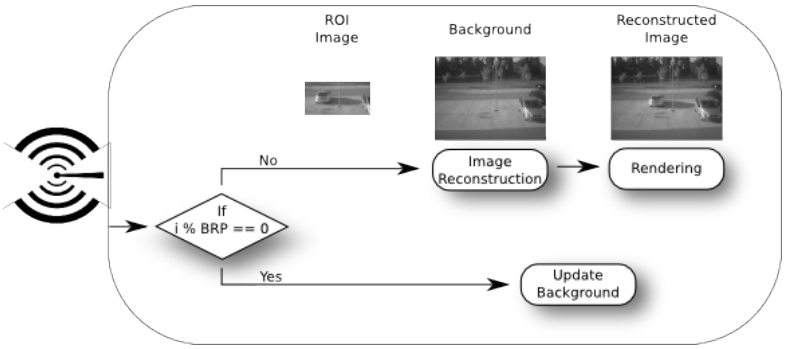


FIGURE 9. The video-decoder.

in Fig. 10. The first row of the table reports the results when all raw images are sent trough

Algorithm 3 Decoder.

```
Frame_id = 0;
while 1 do
  receive_data(buffer);
  if (Frame_id mod BRP) == 0 then
    bgnd = set_background(buffer);
  else
    image = reconstruct_image(bgnd, buff);
  end if
  Frame_id = Frame_id+1;
end while
```

the network (no compression applied), thus showing the maximum video data rate and quality ($PSNR = \infty$).

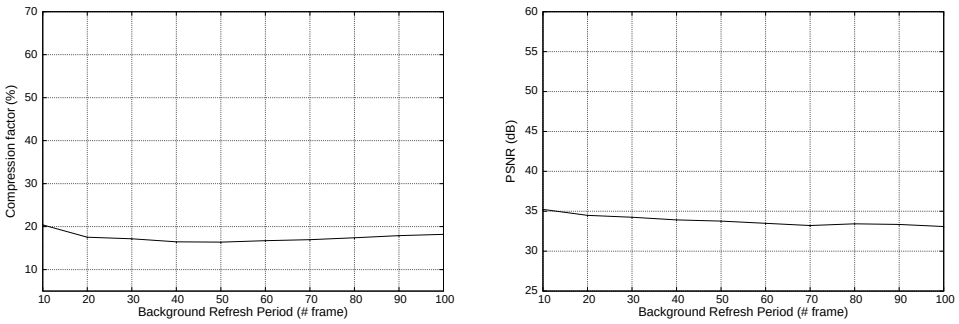
TABLE 5. Selected metrics.

Name	**Formula**
Mean Squared Error (MSE)	$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \big(A(i,j) - B(i,j)\big)^2$
Peak Signal-to-Noise Ratio ($PSNR$)	$PSNR = 10 \cdot \log_{10} \left(\frac{2^8 - 1}{MSE} \right)$
Compression ratio (ρ) [**Sal06**]	$\rho = \frac{compressed\_size}{uncompressed\_size}$

Considering the average results between the high and low motion scenario, reported in the right side of Tab. 6, it follows that the best trade-off between perceived video quality ($PSNR = 35.23dB$) and compression ratio ($\rho = 20.40\%$) is obtained when $BRP = 10$ frames is set (minimum considered value). In a preliminary analysis it could result intuitive to have an infinity PSNR for each BRP value due to the codec approach in "cutting and pasting" raw RoIs. Albeit this result is completely true in ideal conditions, the external environment, as well as the hardware limits, introduce luminosity variations which results in image differences and PSNR variations. Considering again results in Tab. 6, When $BRP \leq 50$ frames, ρ and PSNR both shows a negative derivative. These trends simply demonstrate that a reduction in the compression ratio is paid at the cost of a lower quality in video reconstruction: indeed the temporal light variations (due to weather changes or internal camera reconfiguration as for what concerns auto-exposure) degrade the quality of the reconstructed images because they invalidate the background image. Conversely, when $BRP > 50$ frames, ρ slightly increases and PSNR

TABLE 6. Performance results of the proposed codec.

	Low motion			**High motion**			**Overall results**		
BRP	Data rate [Kbps]	ρ [%]	PSNR [dB]	Data rate [Kbps]	ρ [%]	PSNR [dB]	Data rate [Kbps]	ρ [%]	PSNR [dB]
NC	153.60	100.00	∞	153.60	100.00	∞	153.60	100.00	∞
10	20.06	13.06	37.34	42.62	27.75	33.11	31.34	20.40	35.23
20	12.97	8.45	36.90	40.89	26.62	32.08	26.93	17.53	34.49
30	10.67	6.95	36.74	42.07	27.39	31.75	26.37	17.17	34.25
40	10.05	6.54	36.20	40.54	26.39	31.63	25.30	16.47	33.92
50	10.02	6.52	36.05	40.34	26.25	31.48	25.18	16.39	33.77
60	9.04	5.89	35.77	42.37	27.58	31.20	25.70	16.73	33.49
70	8.66	5.64	35.28	43.44	28.27	31.13	26.05	16.96	33.21
80	9.49	6.18	35.26	43.96	26.66	31.60	26.73	17.40	33.43
90	10.23	6.66	35.15	44.79	29.16	31.55	27.51	17.91	33.35
100	11.28	7.34	34.80	44.62	29.05	30.38	27.95	18.20	33.09



(A) ρ vs BRP (B) PSNR vs BRP

FIGURE 10. Proposed codec: ρ and PSNR as a function of the BRP.

remains stable around 33 dB. These trends descend from the deterministic nature of the background image update (occurring at every $N : N\%BRP = 0$) that generates the so-called *ghosts* phenomenon. Indeed, when the deterministically acquired background image contains a moving object G_N (see Fig. 11a), this is expected to leave the view after a certain number of frames (see Fig. 11b) causing a false convergence (i.e., an artifact) of the blob detection algorithm into the region where the moving object was wrongly appointed to be part of the background image (see Fig. 11c). Ghosts are marginally affecting the considered performance when $BRP \leq 50$

frames because of the more frequent background image update, while they largely downgrade the observed metrics when the dynamics of the moving objects start to be compatible with the refreshing period, as in the case of $BRP > 50$ frames.

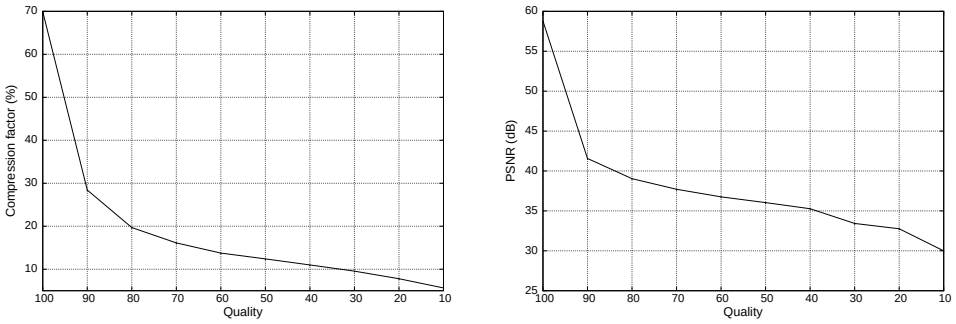
(A) Background figure.

(B) i-th image.

(C) Binarized i-th image.

FIGURE 11. The *ghost* phenomenon.

The performance of the proposed codec has been compared with those of the standard JPEG. Concerning computation complexity, the developed solution is lighter than JPEG scaling as $O(N)$ instead of $O(NlogN)$, thus resulting more suitable for a real implementation in low-end camera network devices. On the other hand other metrics are in favor of JPEG by considering only a video stream coming from a compression and decompression operation. The results for the JPEG, the compression ratio, data rate, and PSNR are reported in Tab. 7 by comparing results achieved by the proposed technique. Moreover, overall JPEG results are reported in a graphical form in Fig. 12, by showing the ρ and PSNR behaviors as a function of BRP and JPEG quality. The Quality parameter is appointed as counterpart of BRP in JPEG since the latter is undefined. As it can be seen, in the ideal case of noise-free transmission, JPEG outperforms the proposed compressor. In fact fair quality in the perceived stream (e.g., $PSNR \simeq 35$) corresponds to a compression ratio double in JPEG than in our proposal. If we set the compression ratio (i.e., $\rho \simeq 20$), the quality of reconstruction obtained by JPEG compressor is higher than in our case (35.23 dB for our compressor, 39.04 dB for JPEG).



(A) ρ vs Quality (B) PSNR vs Quality

FIGURE 12. JPEG: ρ and PSNR as a function of the Quality.

4.4.3. Error resilience and data concealment techniques. In this section, different techniques aiming at improving the error resilience of the proposed codec over error-prone networks are described. These techniques derive directly from the intrinsic uncorrelation of raw image data (the numerical representation of the luminance of each pixel is independent from the others). In this situation, if the decoder can reconstruct the image (applying error concealment techniques), the sporadic bit flips of certain pixels do not disrupt the general consistency in the reconstructed frame. The residual errors which cannot be removed (or concealed) rely on the metadata (i.e.,

TABLE 7. Performance results of the proposed codec versus JPEG.

Proposed codec				**JPEG**			
BRP	Data rate [Kbps]	ρ [%]	PSNR [dB]	Quality	Data rate [Kbps]	ρ [%]	PSNR [dB]
NC	153.60	100.00	∞	NC	153.60	100.00	∞
10	31.34	20.40	35.23	100	107.41	69.92	58.79
20	26.93	17.53	34.49	90	43.66	28.42	41.57
30	26.37	17.17	34.25	80	30.25	19.69	39.04
40	25.30	16.47	33.92	70	24.79	16.13	37.71
50	25.18	16.39	33.77	60	21.14	13.76	36.76
60	25.70	16.73	33.49	50	19.05	12.40	36.04
70	26.05	16.96	33.21	40	16.91	11.00	35.27
80	26.73	17.40	33.43	30	14.67	9.55	33.43
90	27.51	17.91	33.35	20	11.96	7.79	32.76
100	27.95	18.20	33.09	10	8.67	5.64	29.99

descriptor fields) inserted as headers at the application and communication layers. In the following we first describe (i) a proposed resilience schema for the intra-frame codec, then (ii) an improved version of JPEG aiming at increasing its robustness in wireless communications.

In the proposed codec a RoI must be transmitted through the network while adding additional information (e.g., x and y coordinates, width and height dimensions). Nonetheless because of the tiny size of the payload in 6LoWPAN data packets (equal to 96 bytes), a single RoI (or background image) is fragmented into many 6LoWPAN packets. Following the arguments introduced above, any error occurring in the application header (encoded in the first packet) will critically disrupt the image reconstruction or the background image update at the receiver side. We can better encode the RoI (or background image) at the network level coupling a descriptor with each fragment and enhance its degree of uncorrelation. In other words we can define a "*self-reconstructing*" packet where the header contains all the information required for reconstructing the bytes included in the local payload (see Fig. 13). The following fields are contained in the header format: ID is the image identifier (to keep transmitter and receiver synchronized); x and y are the absolute coordinates of the first pixel in the payload (in the reference frame of the image); w_{RoI} represents the RoI (or background image) width and w_r is the residual width representing the remaining bytes to be written in the y row. This encoding technique introduces a certain transmission overhead thus worsening the compression ratio and, consequently, the bandwidth used to send an image. On the other hand, it permits a stronger error-resilience because of the higher degree of uncorrelation and the granularity (self consistency) introduced at the fragment level. In order to avoid mixture of RoIs in fragments, in our simulations we padded by zeros the final fragment referred to a RoI.

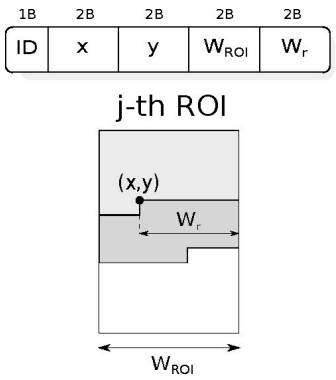


FIGURE 13. Self-reconstructing packet header fields description.

To further improve on error resilience we modified the basic functionality of the IEEE802.15.4 standard (which 6LoWPAN relies on) to avoid to completely discard packets with single bit errors. Indeed, according to the IEEE 802.15.4 standard only sent and received packets showing the same

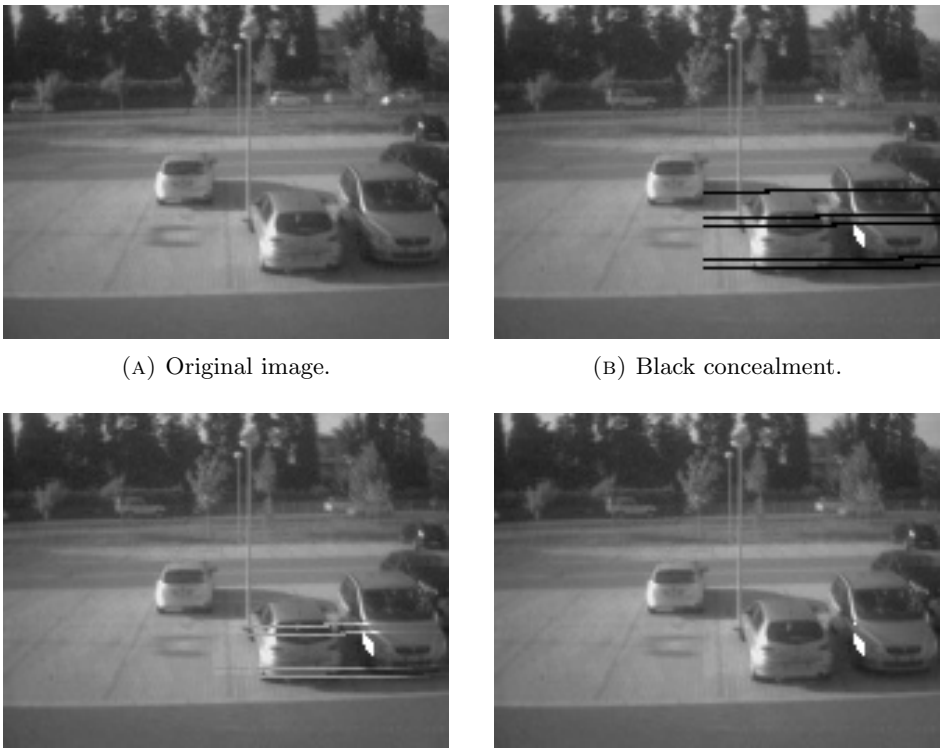
Frame Check Sequence (FCS) can be accepted. The FCS is evaluated on the whole data packet. We applied FCS on the application and 6LoWPAN headers only, in order to consider valid all those packets having sporadic bit flips in the payload. In Fig. 14b our proposal is sketched in comparison with the native IEEE802.15.4 format (Fig. 14a). In the standard the FCS field is positioned at the end of the packet in order to check the consistency of all the bytes in it; our proposal is that of swapping it with the payload to check upon header integrity only. Indeed the proposed solution has the advantage of reducing the number of discarded packets (those having corrupted headers only) while keeping unchanged the complexity of FCS calculation (marginally reducing the actual overhead because of the reduced number of bits, domain of FCS calculation).

(A) IEEE802.15.4 compliant packet. (B) Error-resilient packet.

FIGURE 14. Standard and proposed MAC data messages.

Even if the above proposed approach guarantees to receive a bigger amount of information at the cost of accepting isolated bit errors, it does not avoid to lose whole packets with errors in the packet header. To reconstruct parts of lost images three different types of low-complexity concealment algorithms are proposed and discussed in order to improve on the perceived reconstruction quality: (i) *black concealment*, (ii) *copy-background concealment*, and (iii) *copy-image concealment*. All the mentioned techniques consist on replacing lost fragments of a video frame with certain buffers. Consequently in case of black-concealment (see Fig. 15b) the lost fragment is replaced with black pixels; in case of copy-background concealment (see Fig. 15c) with the relative background image pixels, in case of copy-image concealment (see Fig. 15d) with the content of the previous reconstructed image. The "copy-image concealment" algorithm is the most memory consuming and computationally intensive since it requires extra-memory to store the previous image and computational overhead for copying data; from the performance evaluation it results to maximize PSNR (see next section). On the other hand, the "black concealment" algorithm has no memory overhead, it needs to set to zero a certain zone of the current reconstructed image, but its performance in terms of PSNR does not seem promising. Finally the "copy-background algorithm" is the most promising since it shows performance comparable with "copy-image" at no extra-cost in terms of memory and complexity.

Finally, in order to have a fair comparison between the proposed codec and JPEG in error-prone channels, some improvements for the JPEG are presented in order to boost its performance in presence of communication noise. JPEG format is not byte-oriented and intrinsically highly correlated, therefore there is marginal room to introduce both low complexity error resilience and concealment techniques. For this reason we applied FCS to the IPv6 header only: if a packet has the IPv6 header corrupted will be discarded, otherwise, even if the payload is corrupted, it will be decompressed at the application layer. In the first situation, an irrecoverable discontinuity inside the JPEG buffer occurs without any possibility of concealment. In this case



(A) Original image.

(B) Black concealment.

(C) Copy-background concealment.

(D) Copy-image concealment.

FIGURE 15. Concealment techniques.

the entire image is discarded and substituted with the previous retrieved image, using a sort of *global copy-image concealment*. When the bit flips occur in the IPv6 header, the decompressor attempts to reconstruct the image frame: if the errors occur within the JPEG header (where the metadata as width, height, the Huffman tables, etc. are contained) or a JPEG marker (e.g., the "End of Block (EOB)"), the decompressor cannot reconstruct the frame and the "global copy-image concealment" is applied. In all the other cases (errors occurred in the DCT blocks), the decompressor converges and reconstructs the image. All the performance results presented in the next section refer to this modified version of JPEG.

4.4.4. Performance evaluation. In this section we comment upon the effects introduced by environmental noise in point to point communication. For all performed simulations we used a loss trace characterized by a Bit Error Rate (BER) equal to $5 \cdot 10^{-5}$ as from the best fit of the data acquisition campaigns of the IPERMOB project [**ipe09**]. Must be stressed that all the results discussed in this section derives from simulative experiments in which the impact of bit errors is evaluated by using images with a QQ-VGA resolution coming from the dataset described in Section 4.2. Performance results are presented by using the metrics described in Tab. 5 and

8. The section is organized as follows: in the first part the proposed codec performances are evaluated against the error resilience algorithm improvements described in Section 4.4.3: the best techniques are selected in terms of quality of reconstructed images, complexity and memory footprint. In the second part a comparison between the improved version of both the JPEG compressor and our codec is described.

TABLE 8. Network metrics.

Name	**Formula**
Data rate	$Data\_Rate = \frac{CompressedImage\_Size}{Frame\_Period}$
Packet loss rate (PLR)	$PLR = \frac{\#Discarded\_Packets}{\#Transmitted\_Packets}$

The impact of the FCS based error resilience technique is evaluated using the three types of concealment described in Section 4.4.3. The actual values of PSNR, data rate, and PLR retrieved from simulation are shown in Tab. 9. As it is easy to expect a relevant reduction in PLR occurs in the case in which the FCS check is performed only on the packet header (header check): the length of the headers (40 bytes) is considerably shorter than the length of the entire packet (127 bytes) and consequently the probability of error within the headers is smaller than the probability of error occurring within the entire packet (packet check). Since the data rate is not affected by concealment policies (handled at the receiver), the best techniques are selected following arguments related to image reconstruction quality, computational complexity, and memory footprint.

In Fig. 16 the PSNR is plotted as a function of BRP for all considered error resilience improvements (i.e., concealment technique and integrity checks). The black concealment produces the worst performance; yet the two curves related to the integrity check techniques (i.e., header check and packet check) are clearly separated and highlight the impact of PLR in PSNR (having a minimum of about 20dB in case of packet check, and 25dB in the other case). From the same figure it is possible to state that the best approach in terms of reconstructed image quality is the one based on header check and copy-image concealment. However both the copy-image and copy-background concealments are very effective and produce PSNR ranging from 30dB to 35dB in both header and packet check configurations. Consequently the optimal version of the system is based on header check, because it guarantees better performance in terms of PSNR without any complexity overhead. About concealment the copy-image and the copy-background techniques are both effective: the first one is the optimal in terms of PSNR, and the second one, having a slightly lower PSNR but less computational overhead, represents the best option from cost-benefit analysis arguments.

TABLE 9. Performance results comparison between the different composition of the error resilience improvements.

	Header check					**Packet check**				
BRP	PSNR black [dB]	PSNR bgnd [dB]	PSNR copy [dB]	Data rate [Kbps]	PLR [%]	PSNR black [dB]	PSNR bgnd [dB]	PSNR copy [dB]	Data rate [Kbps]	PLR [%]
10	25.47	34.13	34.21	46.99	1.44	20.14	34.03	34.22	46.99	4.55
20	25.12	33.43	33.53	40.65	1.46	20.00	33.20	33.46	40.65	4.60
30	25.29	33.24	33.39	40.04	1.44	19.99	32.95	33.27	40.04	4.54
40	25.00	32.92	33.06	38.88	1.43	20.02	32.64	32.89	38.88	4.55
50	25.18	32.61	32.80	38.19	1.43	20.05	32.21	32.54	38.19	4.55
60	24.70	32.48	32.64	39.07	1.49	19.98	32.09	32.47	39.07	4.58
70	25.02	32.43	32.60	40.56	1.44	20.01	31.97	32.31	40.56	4.55
80	25.47	32.37	32.53	39.81	1.44	19.92	32.00	32.33	39.81	4.57
90	24.86	32.28	32.46	43.33	1.42	19.79	31.78	32.22	43.33	4.53
100	24.59	31.52	31.81	42.26	1.47	19.70	30.95	31.48	42.26	4.55

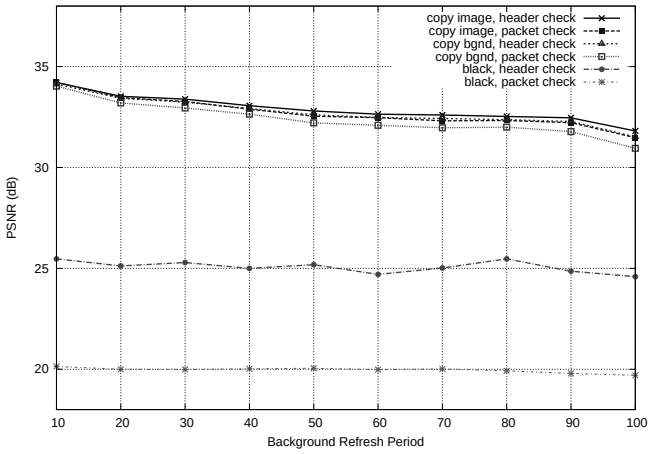


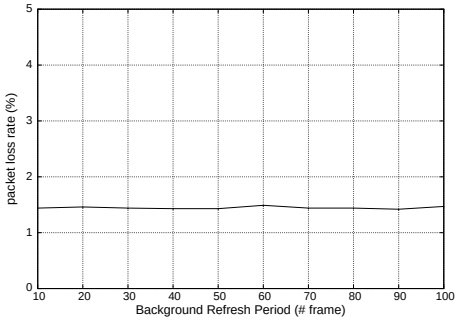
FIGURE 16. Performance comparison for the different improvements of the proposed codec.

In the following, a performance comparison between the proposed codec and the modified JPEG version described in Section 4.4.3 is presented. In Fig. 17a and 17b PLR is plotted versus BRP and JPEG Quality indicator respectively. The values are constant and rather compatible because the header-to-payload ratio are quite similar in the two cases (i.e., 0.31 for the proposed

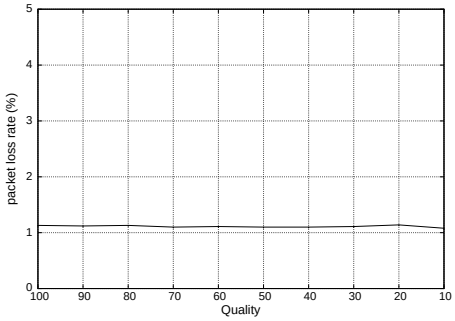
codec and 0.24 for JPEG). This comparable performance is not translated in the perceived quality of image reconstruction (see Fig. 17c and 17d) since a single bit flip either in the IPv6 or JPEG headers corrupts the full image; moreover the high correlation within a single JPEG block and among displaced blocks affects the quality of full image reconstruction also in cases where sporadic bit flips have occurred. When high quality indicators are considered, PSNR has a catastrophic drop because at high sampling frequencies all local defects are propagated to the full image. In our compressor the PSNR is systematically higher than JPEG and shows a very weak dependence on BRP. Concerning data rate, as shown in Fig. 17e and 17f, our compressor is weakly dependent on BRP; JPEG instead shows a very sensitive drop when lower quality is required. At $Q = 70$, the data rate is lower than in our case ($33kbps$ instead of $38kbps$); if this residual gain is not considered crucial, the JPEG compressed image can be protected by FEC or EC techniques at the cost of higher complexity. Nonetheless although totally protecting information the result in terms of PSNR is quite unsatisfactory if compared with ours.

TABLE 10. Performance results comparison between the proposed codec and JPEG.

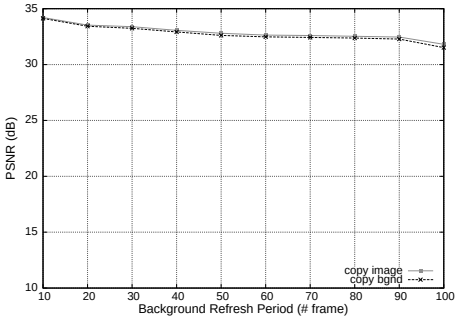
Proposed technique					**JPEG**			
BRP	PSNR copy [dB]	PSNR bgnd [Kbps]	Data rate [Kbps]	PLR [%]	Quality	PSNR copy [dB]	Data rate [Kbps]	PLR [%]
10	34.21	34.13	46.99	1.44	100	21.85	143.46	1.13
20	33.53	33.43	40.65	1.46	90	30.86	58.66	1.12
30	33.39	33.24	40.04	1.44	80	31.21	40.78	1.13
40	33.06	32.92	38.88	1.43	70	31.20	33.31	1.10
50	32.80	32.61	38.19	1.43	60	31.12	28.41	1.11
60	32.64	32.48	39.07	1.49	50	30.81	25.63	1.10
70	32.60	32.43	40.56	1.44	40	30.55	22.72	1.10
80	32.53	32.37	39.81	1.44	30	30.28	19.73	1.11
90	32.46	32.28	43.33	1.42	20	29.58	16.12	1.14
100	31.81	31.52	42.26	1.47	10	27.83	11.75	1.08



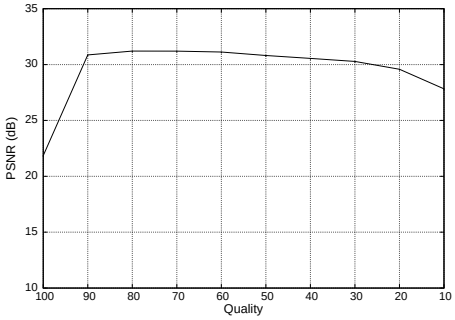
(A) Proposed codec: BRP vs. PLR



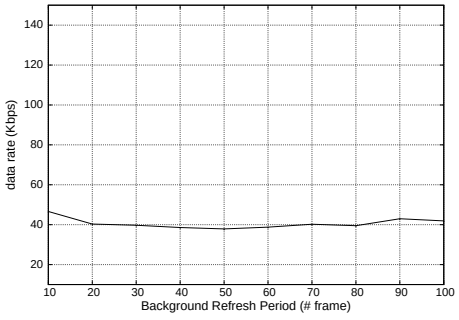
(B) JPEG: Quality vs. PLR



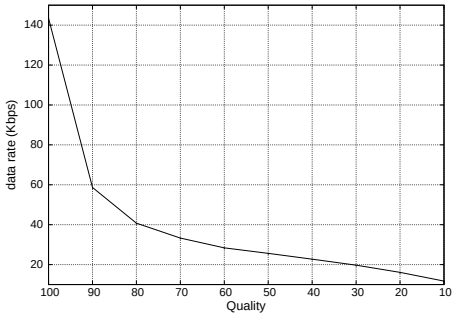
(C) Proposed codec: BRP vs. PSNR



(D) JPEG: Quality vs. PSNR



(E) Proposed codec: BRP vs. data rate



(F) JPEG: Quality vs. data rate

FIGURE 17. Comparison of PSNR, data rate and PLR between the proposed codec and JPEG.

CHAPTER 5

Embedded implementations of computer vision algorithms

IN this chapter our vision on how to deploy computer vision algorithms on SC based on low-complexity and low-memory micro-controllers with the lack of *Floating Point Unit* (*FPU*) is detailed. In the previous chapters, the classical direction on deploying state of the art computer vision algorithms on low-complexity and low-memory micro-controllers is described: to enhance the processing performance state-of-the-art algorithms process a smaller amount of data considering only part *Regions of interest* (*RoI*) of an entire image. However the above mentioned approach are suitable only for few applications that need to monitor only sub-set of pixels of an image (e.g., parking spaces, see also Chap. 3.2) or exploit certain environmental characteristics (e.g., Linesensor, see also Chap. 3.3). All the other cases (e.g., objects tracking, actions recognition, ...) require to consider the entire *Field of View* (or *FOV*) of the camera. In this situation, we propose the idea to optimise and/or approximate the state of the art computer vision algorithms in order to both reduce the memory footprint (compressing the representation of the commonly used data types) and speed up the data processing (by the use of integer representation) maintaining comparable performance.

In this chapter an integer-based *Gaussian Mixture Model* (*GMM*) [**SG99**] is described, not only to optimise computational complexity and memory footprint, but also to allow deployment on micro-controllers without an FPU (see [**SMP$^+$12**]). The methods presented in the following can be applied on any mixture of Gaussians and for any architecture. Specifically, we propose two methods to approximate the GMM: *iGMM-l* and *iGMM-s*, where the integer weight quantisation is performed by a linear or a staircase function, respectively. In what follows we will use the term iGMM-x to generally refer to both variations above.

5.1. Background

In [**SG99**] *Gaussian Mixture Method* algorithm is detailed. This technique explicitly provides rigorous statistical models for both background and foreground by representing each pixel by a set of adaptive G Gaussians. Although it allows performing solid and accurate background subtraction, this method is characterized by a high memory footprint, as each Gaussian is represented by three floating point parameters, and a high computation cost, as each parameter is updated using a Finite Impulse Response (or FIR) filter.

There have been only few reported attempts to optimize the above mentioned approach for embedded systems. An efficient implementation of the GMM algorithm is described in [**SHL$^+$12**]. The image is segmented into 8x8 blocks, and their projections (based on *compressive sensing*, proposed by [**CRT06**] and [**Don06**]) are modelled as a GMM, so as to label each one

as background or foreground. Finally, each foreground block is refined to generate a pixel-wise binary map. This approach obtains very good performance in terms of processing speed (five times faster than the original GMM) but only a modest result in terms of memory (a quarter of the original GMM).

5.2. Methodology

5.2.1. Introduction to Gaussian Mixture Model (GMM). GMM is a technique to model the static background and separate it from foreground in video sequences acquired by static cameras. The temporal variation of each pixel of the image is modeled by the weighted sum (or mixture) of G Gaussians, each one of them described by three parameters: *mean value* (μ), *variance* (σ^2) and *weight* (w). Each parameter takes values from a certain range as presented in Tab. 1. Particularly, constraining the range of variance prevents Gaussians from degenerating into a *Dirac delta* (if $\sigma^2 \to 0$) or into a *uniform distribution* (if $\sigma^2 \to \infty$).

μ	σ^2	w
[0, 255]	$[\sigma^2_{min},\sigma^2_{MAX}]$	[0, 1]

TABLE 1. Gaussian parameters ranges

The GMM adaptation procedure is based on the computation of the *difference* (see Eq. 14) of the pixel $p_i(j)$ (where i represents the pixel position and j the given frame) from the mean value of each g Gaussian, $\forall g \in [1, G]$.

$$d_{i,g}(j) = p_i(j) - \mu_{i,g}(j) \tag{14}$$

A *membership criterion* (see Eq. 15) is defined to update the Gaussians with the corresponding pixel value, given a certain value of *learning rate* α ($\alpha \in [0, 1]$) and a certain threshold T. Thus, if a pixel satisfies the membership criterion for a certain Gaussian, its mean value, variance and weight are updated using three different FIR filters described respectively by equations Eq. 16, Eq. 17 and Eq. 18. Otherwise, the weight of the considered Gaussian, is updated using Eq. 19.

$$\text{if } d^2_{i,g}(j) < T\sigma^2_{i,g}(j) \tag{15}$$

$$\mu_{i,g}(j+1) = \mu_{i,g}(j) + k_{i,g}(j)d_{i,g}(j) \tag{16}$$

$$\sigma^2_{i,g}(j+1) = \sigma^2_{i,g}(j) + k_{i,g}(j)(d^2_{i,g}(j) - \sigma^2_{i,g}(j)) \tag{17}$$

$$w_{i,g}(j+1) = (1-\alpha)\cdot w_{i,g}(j) + \alpha \tag{18}$$

$$\text{else}$$

$$w_{i,g}(j+1) = (1-\alpha)\cdot w_{i,g}(j) \tag{19}$$

where $k_{i,g}(j) = \frac{\alpha}{w_{i,g}(j)}$.

For each pixel i and each frame j, the weights of Gaussians satisfy the following equation:

$$\sum_{g=1}^{G} w_{i,g}(j) = 1 \tag{20}$$

The role of the weight is to discriminate between foreground and background Gaussians. For every pixel, the Gaussians are ordered accordingly to a *sorting rule*, based on the value of ρ ([**SG99**]) that is directly proportional to the squared value of the weight, as described in Eq. 21. If the pixel belongs to the g-th Gaussian, where g satisfies the equation $\sum_{m=1}^{g} w_m(j) \leq T_W$ (where T_W is a threshold given by the user), the pixel is labeled as background, otherwise as foreground.

$$\rho(j) = \frac{w^2(j)}{\sigma^2(j)} \tag{21}$$

5.2.2. Overview of iGMM-x. In the following sections we propose integer-GMM or iGMM, a method to port the GMM to architectures that lack a FPU, aiming to minimise both memory footprint and processing time. To this purpose, we use *sub-integer representation*, where the Gaussian parameters are represented using a number of bits less than the size of the smallest data integer type (namely *uint8_t*). Thus, the mean value and the variance updating that depend directly on the pixel values, will be handled defining a *generalised round* operation (see Sec. 5.2.3). On the other hand, weight updating, dependent only on matching a certain Gaussian, will be handled using a stochastic approach (see Sec. 5.2.5).

5.2.3. Mean value and variance updating. This section describes the proposed updating operations of the mean value and the variance of a Gaussian, represented by sub-integer variables. To permit this operation, we have an *updating methodology* based on three phases, as shown in Fig. 1: (i) the *scaling transform*, (ii) the *additive updating function*, and (iii) the *scaling inverse transform*.

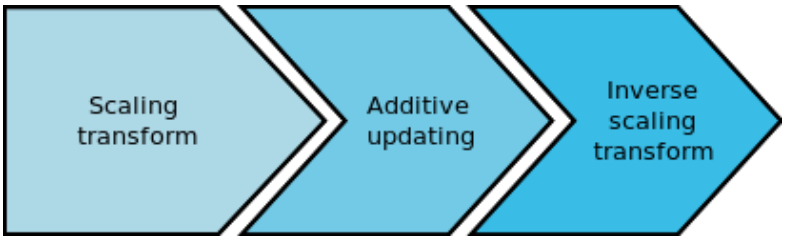


FIGURE 1. Mean value and variance updating chain.

This updating methodology can be seen as a common pipeline used in several signal processing applications: data defined in the integer space $\mathbb{V}$ (in this case the domain represented in memory) are transformed to another space $\mathbb{U}$ (by using the scaling transform) where it is simpler to process, and then converted back to $\mathbb{V}$ (with scaling inverse transform). Moreover, the processing operation, performed in the GMM case in the real domain (namely $\mathbb{R}$), is ported in the $\mathbb{U}$ domain using a method based on the additive updating function and the generalised round. All the above mentioned phases are described in the following sections.

Sub-integer operations. To achieve sub-integer accuracy in Eq. 16 and Eq. 17, all their operands are converted appropriately using a scaling transformation. Accuracy is specified by a pre-defined parameter P and the conversion operation $(\chi(\circ) : \mathbb{V} \rightarrow \mathbb{U})$ for a given operand $V \in \mathbb{V}$

is defined as:

$$\chi(V) = V \cdot P \tag{22}$$

After the updating process, the resulting parameters of applying the operation in the new space are converted back to be saved in the memory (namely from $\mathbb{U}$ to $\mathbb{V}$), using the inverse transform shown in Eq. 23 ($(\chi^{-1}(\circ) : \mathbb{U} \to \mathbb{V})$).

$$\chi^{-1}(E) = \frac{E}{P} \tag{23}$$

with $E \in \mathbb{U}$ being the transformed operand. Both transformations in Eq. 22 and 23 are linear.

According to the above considerations, a *generalised-round* operation *G_ROUND* is defined (see also [**SMP$^+$12**]) to enable a satisfactorily accurate representation of floating-point numbers in the integer domain $\mathbb{U}$, with $\mathbb{U} \subseteq \mathbb{Z}$. The above mentioned integer domain $\mathbb{U}$ can be described as the *discrete* and *ordered* set of numbers $\mathbf{B} = \{b_m \in \mathbb{U} | b_{m+1} - b_m = \gamma\}$, and it can be characterized by two parameters: the *granularity* γ, and the *updating step* ξ, defined as a function of γ:

$$\xi = \mathcal{F}(\gamma) = R\gamma \tag{24}$$

where $R \in (0, 1)$ is called *rounding parameter*. Therefore, it is possible to define the γ-floor operator $\lfloor \circ \rfloor_\gamma$ as:

$$\lfloor \delta \rfloor_\gamma = \lfloor \frac{\delta}{\gamma} \rfloor \gamma \tag{25}$$

and finally the *generalised-round* operator *G_ROUND* as:

$$G\_ROUND(\delta, \xi, \gamma) = \begin{cases} \lfloor \delta \rfloor_\gamma & \text{if } |\delta| \geq \gamma \\ \gamma & \text{if } (|\delta| < \gamma) \wedge (|\delta| \geq \xi) \\ 0 & \text{if } (|\delta| < \gamma) \wedge (|\delta| < \xi) \end{cases} \tag{26}$$

where $\delta \in \mathbb{Z}$ is the value to be rounded.

Generalised additive updating function. The second updating block in Fig. 1 corresponds to Eq. 16 and Eq. 17, which defines *additive updating operations* for both the mean value and the variance. However, since our proposed system is based on integer representation, (see Sec. 5.2.3), these formulas need to be redefined appropriately to emulate the original floating point formulation. Let $U : \mathbb{R} \to \mathbb{D}$ (in this case $\mathbb{D} \subseteq \mathbb{R}$) an *additive updating function* of the parameter a, such that $a(j+1) = U(a(j), \delta(j)) = a(j) + \delta(j)$, where $\delta(j)$ is the *updating contribution* for the parameter a. Therefore, Eq. 16 and Eq. 17 can be rewritten as:

$$\mu(j+1) = \mu(j) + \delta_\mu(d(j)) \quad \text{where } \delta_\mu(d(j)) = k(j)d(j) \tag{27}$$

$$\sigma^2(j+1) = \sigma^2(j) + \delta_{\sigma^2}(d^2(j)) \quad \text{where } \delta_{\sigma^2}(d^2(j)) = k(j)(d^2(j) - \sigma^2(j)) \tag{28}$$

According to the first block of Fig. 1, a conversion in the space $\mathbb{U}$ is needed:

$$\chi[a(j+1)] = \chi[a(j) + \delta(j)] = \chi[a(j)] + \chi[\delta(j)] \tag{29}$$

Consequently, to achieve the floating point accuracy with a sub-integer representation, the *generalised additive updating function* $\bar{U}$ is defined in the integer space $\mathbb{U}$ ($\bar{U}(\circ) : \mathbb{U} \to \mathbb{U}$), taking into account the *generalised-round* operator G_ROUND defined in Eq. 26, as follows:

$$\begin{aligned}\bar{U}(a(j), \delta(j)) &= G\_ROUND[\chi[a(j+1)]] = \\ &= \chi[a(j)] + \{G\_ROUND[\chi[\delta(j)]]\}\end{aligned} \tag{30}$$

Finally, the overall result of all the chain in Fig. 1 can be summarized by the following relation:

$$a(j+1) = a(j) + \chi^{-1}\{G\_ROUND[\chi[\delta(j)]]\} \tag{31}$$

and the parameters γ and ξ for the G_ROUND operator should have appropriate values to produce similar results to the original updating function $U(a(j), \delta(j))$.

5.2.4. Calculating the updating step ξ. As described in our previous work [**SMP$^+$12**], using a fixed value of rounding parameter R in Eq. 24 (in that case $R = 0.5$) limits the learning rate range, since updating the mean value and the variance is possible for only a sub-set of learning rate values. In this section a more general technique is suggested. It computes the value of the updating step ξ as a function of the learning rate α, to ensure appropriate updating of both parameters while maintaining the membership ranges of Gaussians similar to the ones estimated by the floating-point approach.

According to Eq. 15, 27 and 28, both the updating contributions $\delta_\mu(d_{i,j})$ and $\delta_{\sigma^2}(d^2_{i,j})$ are bounded inside *updating contribution ranges*, as defined by the following formulas:

$$\delta^{\mu}_{min}(\sigma^2) = -\sigma\sqrt{T} \leq \delta_\mu(d_{i,j}) \leq \sigma\sqrt{T} = \delta^{\mu}_{MAX}(\sigma^2) \tag{32}$$

$$\delta^{\sigma^2}_{min}(\sigma^2) = -k\sigma^2 \leq \delta_{\sigma^2}(d^2_{i,j}) \leq k\sigma^2(T-1) = \delta^{\sigma^2}_{MAX}(\sigma^2) \tag{33}$$

Thus, Eq. 32 and 33 can be seen as generic ranges bounded by a maximum (δ_{MAX}) and a minimum (δ_{min}), and they can be represented using a couple of parameters: the center C (see Eq. 34) and the radius r (see Eq. 35), as shown in Fig. 2.

$$C = \frac{\delta_{MAX} + \delta_{min}}{2} \tag{34}$$

$$r = \frac{\delta_{MAX} - \delta_{min}}{2} \tag{35}$$

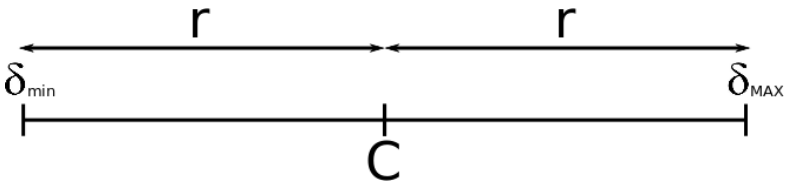


FIGURE 2. Range center (C) and radius (r).

In Tab. 2 both the intervals of Eq. 32 and 33 are represented in terms of centers and radii as a function of σ^2.

$C_\mu(\sigma^2) = 0$	$r_\mu(\sigma^2) = \alpha\sigma^2\sqrt{T}$
$C_{\sigma^2}(\sigma^2) = \dfrac{\alpha T\sigma^2 - 2\alpha\sigma^2}{2}$	$r_{\sigma^2}(\sigma^2) = \dfrac{\alpha\sigma^2 T}{2}$

TABLE 2. Centers and radii for mean value and variance updating contribution range.

Because of the bounded nature of the variance and the direct proportionality of the above mentioned parameters on its value, it is possible to show that both the centers and the radii are bounded into the following intervals:

$$C_\mu = 0 \qquad \forall\sigma^2 \tag{36}$$

$$r_\mu \in [r_\mu(\sigma^2_{min}), r_\mu(\sigma^2_{MAX})] \tag{37}$$

$$C_{\sigma^2} \in [C_{\sigma^2}(\sigma^2_{min}), C_{\sigma^2}(\sigma^2_{MAX})] \tag{38}$$

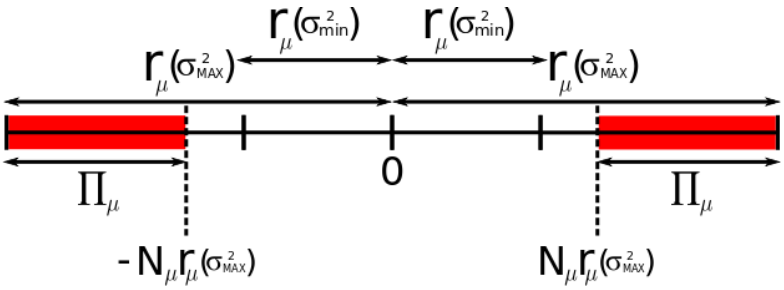
$$r_{\sigma^2} \in [r_{\sigma^2}(\sigma^2_{min}), r_{\sigma^2}(\sigma^2_{MAX})] \tag{39}$$

From the previous considerations it is possible to state that the updating contribution ranges have a variable width, dependent on the variance. Consequently, the idea is to define *critical intervals* close to the border of both the above mentioned ranges, where the updating contribution is considered sufficiently large to increase the counter. In the following the criteria to estimate these intervals are described. By comparing Eq. 16 and 17, it is noted that the former depends directly on the difference value $d_{i,g}(j)$ (where $d_{i,g}(j) \in \mathbb{Z}$) but the latter evolves with its square value. Consequently, in case of a large value of $d_{i,g}(j)$, the variance update will be significantly higher than the mean value update, and then the above mentioned critical intervals can be estimated as:

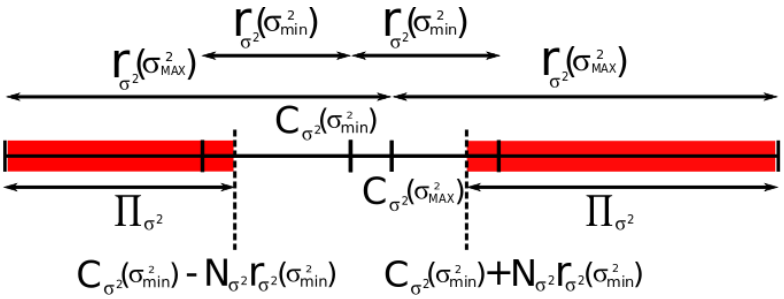
- **(a):** the critical interval Π_μ is the zone inside the maximum width membership interval (namely the interval $[-r_\mu(\sigma^2_{MAX}), r_\mu(\sigma^2_{MAX})]$) where the updating contribution δ_μ is considered sufficiently large to update the mean value of its granularity;
- **(b):** the critical interval Π_{σ^2} is the zone inside the minimum width membership interval (namely the interval $[C_{\sigma^2}(\sigma^2_{min}) - r_{\sigma^2}(\sigma^2_{min}), C_{\sigma^2}(\sigma^2_{min}) + r_{\sigma^2}(\sigma^2_{min})]$) where the updating contribution δ_{σ^2} is considered sufficiently large to update the variance of its granularity.

In Sec. 5.4.1 the impact of both of these principles will be shown and evaluated in a real test case.

Following the above considerations, two new parameters, *minimum updating numbers* $N_\mu \in \mathbb{R}$ and $N_{\sigma^2} \in \mathbb{R}$ (with $N_\mu \in (0,1)$ and $N_{\sigma^2} \in (0,1)$), can be used to code the width of the critical intervals Π_μ and Π_{σ^2} respectively (see red zones in Fig. 3a and Fig. 3b). The introduction of these two parameters allows us to define the intervals while using only integers instead of real numbers for our calculations. Thus, according to Eq. 15, 16 and 17, the following two conditions describe these critical intervals:



(A) Mean value case.



(B) Variance case.

FIGURE 3. Updating contributions ranges and *minimum updating number*.

$$\Pi_\mu = \{\delta_\mu \in \mathbb{Z} : |\delta_\mu - C_\mu| \geq \lfloor N_\mu r_\mu(\sigma^2_{MAX}) \rfloor\} \tag{40}$$

$$\Pi_{\sigma^2} = \{\delta_{\sigma^2} \in \mathbb{Z} : |\delta_{\sigma^2} - C_{\sigma^2}| \geq \lfloor N_{\sigma^2} r_{\sigma^2}(\sigma^2_{min}) \rfloor\} \tag{41}$$

Thus, it is possible to rewrite the generalised round operation (Eq. 26) as:

$$G\_ROUND(\delta, \xi, \gamma, C) = \begin{cases} \lfloor \delta \rfloor_\gamma & \text{if } |\delta| \geq \gamma \\ \gamma & \text{if } (|\delta| < \gamma) \wedge (\delta \in \Pi) \\ 0 & \text{if } (|\delta| < \gamma) \wedge (\delta \notin \Pi) \end{cases} \tag{42}$$

where ξ is the updating step, and Π is the critical interval. Thus, from Eq. 40, Eq. 41 and Eq. 42 it is possible to define the updating steps ξ as:

$$\xi_\mu = N_\mu \cdot r_\mu(\sigma^2_{MAX}) \tag{43}$$

$$\xi_{\sigma^2} = N_{\sigma^2} \cdot r_{\sigma^2}(\sigma^2_{min}) \tag{44}$$

Finally, the values of N_μ and N_{σ^2} are calibrated iteratively to minimize the difference of the iGMM-x and GMM intervals on a validation data-set, as depicted in Sec 5.4.1.

5.2.5. Weight updating. The weight updating rule, in contrast to the mean and variance updating rules, does not depend directly on the value of the pixel p_i, but only on the membership to a certain Gaussian: if the pixel belongs to a Gaussian, its weight is increased by a given value, otherwise it is decreased. In the original approach described in Sec. 5.2.1, the weights

are represented as floating points between 0 and 1 and updated using the formulas in Eq. 18 and Eq. 19 to increase and decrease the weight respectively. To avoid the use of floating point weights, Eq. 18 and Eq. 19 are solved as a function of both the learning rate α and the number of iterations s needed to reach a certain value of weight w_g starting from 0 and 1 respectively. The derived equations have an exponential form as described in Eq. 45 and 46 (see also Fig. 4a), where g is the considered Gaussian index ($g \in [1, G]$):

$$w_g(s) = 1 - (1 - \alpha)^s \tag{45}$$

$$w_g(s) = (1 - \alpha)^s. \tag{46}$$

Under this formulation, solving Eq. 45 as a function of both the learning rate α and the weight, Eq. 47 is obtained: this relation specifies how many iterations $s(w, \alpha)$ are needed to reach a certain weight value starting from 0.

$$s(w, \alpha) = \frac{\log_{10}(1 - w)}{\log_{10}(1 - \alpha)} \tag{47}$$

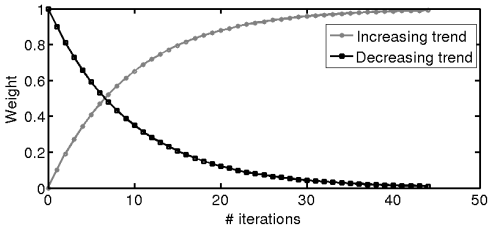
Consequently, it is possible to represent a weight as an integer counter and represent it using a fixed number of l bits, able to cover the range of values $[0, L]$ (where $L = 2^l - 1$).

Though these considerations simplify the weight updating problem, the operation to retrieve the weight value continues to be floating point based because of the exponential nature of both Eq. 45 and Eq. 46. To deal with this issue two options are considered, where the weight trend is simplified and approximated to: (i) a line (see Fig. 4b and Sec. 5.2.5) or (ii) a staircase (see Fig. 4c and Sec. 5.2.5). These two approximation methods, also called *iGMM-l* and *iGMM-s* respectively, derive directly from Eq. 47 and they allow the computation of: (i) the number of steps needed to reach a certain value of weight starting from 0 (or 1); (ii) the number of steps needed to reach a certain value of weight from another one.

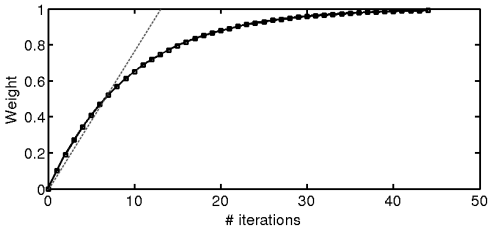
However, from Eq. 47, it is possible to show that the number of steps to retrieve the highest value of weight (namely $1 - \epsilon$ where ϵ is a very small positive number, e.g. $\epsilon = 0.01$) is inversely proportional to the learning rate, and consequently the lower the learning rate, the higher the number of bits needed to represent the weight. Because of the limited number of available bits, the deterministic calculation of both the linear and staircase trends could be efficient only for large values of α.

To overcome such restriction, we propose a stochastic method in Sec. 5.2.5 that increases the counter with a certain probability to reach the highest value of weight (namely $1 - \epsilon$) in a certain number of steps A in average using the above mentioned number of bits l, such that $A > L$ (where $L = 2^l - 1$).

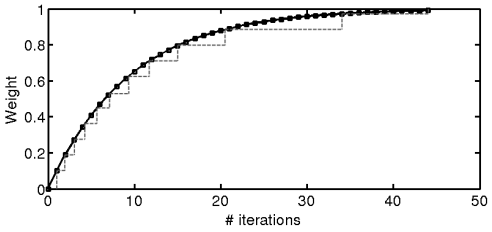
Stochastic updating. This section describes a method to generalise both the proposed weight approximations (i.e. iGMM-l and iGMM-s), to emulate floating point updating. Particularly, in the case of iGMM-l, it allows to overcome the learning rate limitation described in [**SMP$^+$12**]. In iGMM-s, stochastic updating is essential since it allows emulating a logarithmic trend using a set of weight values linearly sampled.



(A) Weight trends.



(B) iGMM-l.



(C) iGMM-s.

FIGURE 4. The weight trend and its approximations.

As stated in the previous section, the idea is to represent the weight using an integer counter and to retrieve its real value using one of the above mentioned approximations. Consequently, in an ideal scenario with no memory constraints, dealing with the weight updating means increasing the above mentioned counter by a certain value, called $STEP$. However, in case where weights are represented using l bits this approach is not feasible: reaching the highest value of weight when learning rate is relatively small may require A iterations, such that $A > L$ (where $L = 2^l - 1$). Thus a stochastic approach is proposed in order to overcome this limitation: the integer counter s is updated a $STEP$ amount with a certain probability to reach the highest value of weight in a certain A number of steps in average, as described in the following equation.

$$s_n = s_n + \mathbf{H}_n \tag{48}$$

where $\mathbf{H}_n \in \{0, STEP\}$ is a set of independent binary random numbers, such that:

$$E\left[\sum_{n=1}^{A} \mathbf{H}_n\right] = L \tag{49}$$

If the probability of the values $STEP$ and 0 are $P_{STEP}(n)$ and $P_0(n)$ respectively, from discrete random number theory and because of the linearity of the expected value $E[\circ]$, it is simple to obtain the following relation:

$$STEP \cdot \sum_{n=1}^{A} P_{STEP}(n) = L \tag{50}$$

The value of $P_{STEP}(n)$ is known and derives from the chosen weight approximation (see Sec. 5.2.5 and 5.2.5). The idea is to implement the logic derived from Eq. 48 making use of the set of the values $P_{STEP}(n)$, a uniformly distributed random number $\mathbf{X}_n$ defined in the interval

$[0, MAX]$, and a threshold T_X, such that:

$$P_{STEP}(n) = P(\mathbf{X}_n > T_X) \tag{51}$$

It is possible to demonstrate (see Appendix A) that the threshold T_X is simply specified using the following formula:

$$T_X = [1 - P_{STEP}(n)] \cdot MAX \tag{52}$$

iGMM-l: Linear approximation. In this configuration, the relation to calculate the weight value w from the number of iterations s (Eq. (45)) is approximated by a line passing through the origin and the point $P_0 = (s(w_0, \alpha), w_0)$ as shown in Fig. 4b. This can be rewritten as the equation of the line that approximates the weight trend, depicted in Eq. 53.

$$w(s) = s\frac{w_0}{s(w_0, \alpha)} \tag{53}$$

Assuming the values of P_0 are known, the weight can be represented as a counter, and it is updated adding the value $STEP$ (usually equal to $G - 1$). Consequently, the updating rules of using the linear approximation are the following: (i) the counter related with the g-th Gaussian has to increase by the updating step $STEP$; (ii) all the weights w_k (with $k \neq g$) have to decrease by the constant value $\frac{STEP}{G-1}$.

In this method it is important to estimate the optimal position of the point P_0, which can be calculated as the point where the line intersects the logarithmic trend of Eq. 45. This point represents an estimation where a weight swap may happen during the sorting operation. Because a precise estimation of the above mentioned swap point is impossible (the sorting depends on ρ, see Eq. 21), the considered value w_0 is the middle point of the weight range, namely 0.5.

As shown in [**SMP$^+$12**], because the weight counter is represented using a finite number of bits l, the method is only able to handle a subset of the learning rate, that satisfies this condition:

$$\frac{s(w_0, \alpha^w_{min})}{w_0} \leq \frac{L}{STEP} \tag{54}$$

Consequently, using Eq. 47, the retrieved interval of learning rate is:

$$\alpha \geq 1 - (1 - w_0)^{\frac{STEP}{w_0 \cdot L}} \tag{55}$$

To increase the range of this interval, the stochastic updating method described in Sec. 5.2.5 is applied. Since we are using a linear approximation, the probability $P_{STEP}(n)$ to update the weight (counter) is constant and independent to n (it depends only on the learning rate α). Consequently, it is possible to compute its value solving Eq. 50:

$$P_{STEP}(\alpha) = \frac{L}{STEP \cdot A} \tag{56}$$

Thus, in this case $P_{STEP}(n)$ represents the ratio between the maximum value $\frac{L}{STEP}$ that can be represented in l bits (namely $L = 2^l - 1$), and the number of steps A needed to reach the value $w = 1$ using Eq. 53.

iGMM-s: Staircase approximation. This technique is based on the assumption that all the possible weight values may be represented by an exponential distribution as defined by Eq. 45 or Eq. 46 that has been uniformly sampled over the iterations. For illustrative purposes, only the increasing function is considered in a first instance, being the decreasing function symmetrical as described later. Such approach emulates completely the weight trend of Eq. 45, but the above counter may require more levels than what is available in the system. Therefore, we propose an approximated solution. The increasing exponential curve in Fig. 4a can be calculated as a linear function sampled using an exponentially distributed sampling period over the iterations (see dotted line in Fig 5b).

$$\bar{\nu}(s,\alpha) = w(s) - w(s-1) = (1-\alpha)^{(s-1)}\alpha \tag{57}$$

Following this logic, the weight is a linear function that can be represented as an integer counter c .

$$w(c) = c \cdot \nu \tag{58}$$

Thus, assuming that the available number of bits to represent the weight is l, and consequently the number of available levels to represent the weight are $L = 2^l - 1$ (0 represents the value $w = 0$), the idea is to sample the interval $[0, 1]$ using at least L points. To optimise both memory footprint and processing complexity, the learning rate (namely the maximum value in Eq. 57 when $s = 1$) is chosen as sampling period ν, and consequently the counter c is updated stochastically, in order to approximate the trend of Eq. 45. Therefore, the stochastic updating method of Sec. 5.2.5 is the core mechanism used for updating the weight counter for iGMM-s: the weight is updated with a exponential distributed probability dependent on the value of the counter c ($P_{STEP}(c)$). Because the chosen sampling period is the maximum inside the set described in Eq. 57, this probability is defined as how many iterations are compressed at every counter updating of the value $STEP$:

$$P_{STEP}(c,\alpha) = \frac{STEP}{\bar{\nu}(c,\alpha)} \tag{59}$$

Thus, a *Look-Up Table* (or LUT) is instantiated inside a vector of L elements to contain the thresholds computed as a function of $P_{STEP}(c,\alpha)$ using Eq. 52: we call this data structure *threshold-LUT* or $LUT_T(\alpha)$. Therefore, from this vector indexed by a counter c (namely $LUT_T(\alpha)[c]$), it is possible to obtain the current threshold value for the stochastic updating technique (see also Sec. 5.2.5).

All these considerations only refer to the weight increasing operations and they do not take into account the weight decreasing case. As seen in Fig. 4a the increasing trend curve is symmetric to the decreasing trend curve with respect to the horizontal line through 0.5. Therefore, it is simple to conclude that the *updating step function* has the same form in both the increasing and the decreasing trend, with the difference that the first one maps the update from 0 to 1, the second one the inverse, namely from 1 to 0. It is then possible to state that the same

threshold LUT can be used for both the increasing and the decreasing trend by simply inverting the access index. Summarizing:

(1) if the pixel p_i belongs to the current Gaussian, the counter c is increased by $STEP$ if and only if the random value $\mathbf{V} > LUT_T(\alpha)[c]$;
(2) if the pixel p_i does not belong to the current Gaussian, the counter c is decreased by $STEP$ if and only if the random value $\mathbf{V} > LUT_T(\alpha)[\bar{S} - c]$, where $\bar{S}$ is the number of iterations needed to reach the weight value 0.99 from 0;
(3) otherwise no operation is performed.

5.3. Implementation

This section discusses the memory and processing requirements of our methods, deployed on the selected platforms (i.e., the SeedEye, described in Sec. 1.1.1, and the Raspberry Pi, in Sec. 1.1.2), are discussed.

5.3.1. Algorithm optimizations. Usually micro-controller/micro-processors for embedded systems are characterized by both limited memory and computational capability to reduce the size of the boards, the energy consumption and the cost. Therefore, our application code has been appropriately optimised using standard techniques (i.e., *loop-unroll* and *word-unroll*), to further reduce the computational cost. In addition, the implementation of the stochastic approach (section) is achieved by a pseudo-random number generator [**EN86**], which is used only once per frame: the same pseudo-random value is used to update all pixel weights of the same frame.

All the missing implementation details are available by request from the authors.

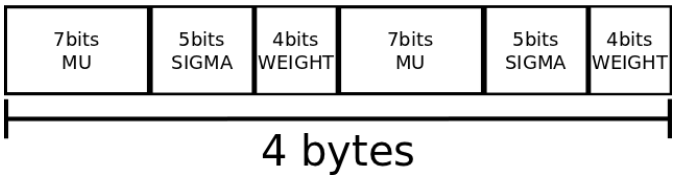
GMM representation. In this section the memory footprint of the data structure used to store the background model is discussed. In Tab. 3 the memory footprint of the GMM background model for different data types is shown. Using a standard integer type (*uint8_t*) memory usage is reduced by a factor 4 and 8, compared to floating point types such as (*float*) and (*double*) respectively. to permit the deployment of the GMM algorithms over the above mentioned memory constrained architectures, we propose to represent each Gaussian in a compressed version, namely using only 2 bytes, distributed between its three parameters.

Thus, the mean value, usually defined in the range $[0, 255]$, is represented using 7 bits and with a granularity $\gamma_\mu = 2$, such that only the even numbers are considered. Such a choice is derived from the noise introduced by the hardware components, characterized by a standard deviation of $2 - 2.5$ ([**HKK07**]): a granularity smaller than the noise is pointless.

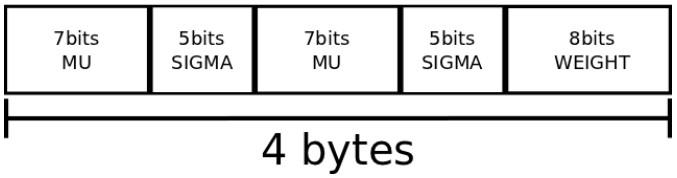
For the same reason,we propose the interval $[5, 36]$ that represents a reasonable set of variance values with respect to the background pixels distribution acquired with real cameras under no changing light conditions. Therefore, the variance is sufficiently represented using 5 bits and with a granularity of 1. Finally, the weight has two different representations depending on the chosen proposed approach. iGMM-l associates a weight (represented using 4 bits) to each Gaussian as in the original GMM. iGMM-s instantiate $G - 1$ weights represented using a minimum of 6 bits, where the last G-th weight is retrieved as a function of the previous $G - 1$, using Eq. 20. Fig. 5a and Fig. 5c shows the bit distribution for the two and three Gaussians mixture examples.

TABLE 3. Algorithm footprint (QQ-VGA images)

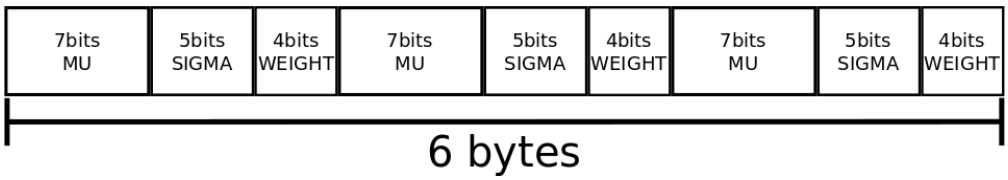
Precision	Bytes per Gaussian	Footprint 2G (Bytes)	Footprint 3G (Bytes)
Double	24	921,600	1,382,400
Float	12	460,800	691,200
uint8_t	3	115,200	172,800
iGMM-x	2	76,800	115,200



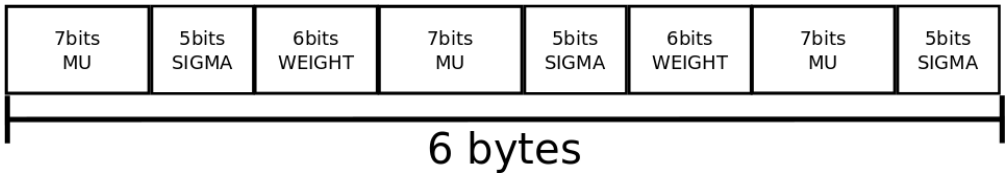
(A) iGMM-l data representation: 2 Gaussians.



(B) iGMM-s data representation: 2 Gaussians.



(C) iGMM-l data representation: 3 Gaussians.



(D) iGMM-s data representation: 3 Gaussians.

FIGURE 5. iGMM-x data representation.

Learning rate operating range. The usage of an integer based representation for native floating point based data introduces a limitation on the learning rate values due to Eq. 22. Therefore, our method operates properly only for a range of learning rate values, which we call *learning rate operating range*. This range is specified as the set of value of α such that:

$$\alpha \geq \frac{1}{P} \tag{60}$$

In the proposed implementation P is fixed to 10,000, and consequently the *learning rate operative range* is the interval $[0.0001, 1]$, which is sufficient to cover almost all practical cases and it is much wider than the one proposed in [**SMP$^+$12**].

5.4. Performance evaluation

In this section the performance of the proposed algorithm is shown. Particularly, using a simulation approach, the three Gaussian parameter trends are validated to satisfy the claims described in Sec. 5.2.3 and 5.2.5. Additionally, the performance of our approach is evaluated both qualitatively and quantitatively using two standard data-sets which comprise different kind of movement: the "IXMAS data-set" [**WRB06**] (slow movement) and the "Fudan Pedestrian data-set" [**BT11**] (fast movement). Finally the processing time of our implementations (i.e., iGMM-l and iGMM-s) is compared with an optimized version of the original GMM over the architectures described in Chap. 1.

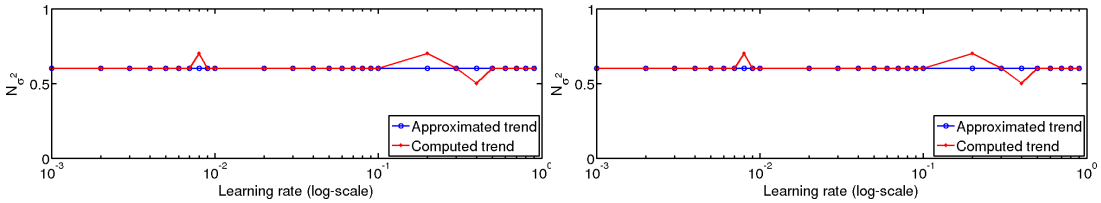
5.4.1. Validation of the parameters.

μ and σ^2 updating calibration and validation. Setting the iGMM-x technique requires only the calibration of the values N_μ and N_{σ^2} as discussed in Sec. 5.2.3. This operation maximises the overlap between the membership areas of the original GMM and the proposed iGMM-x under different illumination conditions, by varying N_μ and N_{σ^2} on their domain (namely the interval $(0, 1)$).

We simulate three different illumination conditions: (i) variable illumination (i.e. day/night transition) with a rate of 8.68 levels per minute, (ii) variable illumination (i.e. cloud/sun transition) with a rate of 4.34 levels per minute, and (iii) constant illumination. Specifically, the illumination trend is approximated as a linear transition, and the typical noise introduced by the acquisition systems is emulated using a Gaussian number generator. In this process, the iGMM-x membership area is filtered using a moving window proportional to $1/\alpha$, to reduce its fast oscillations.

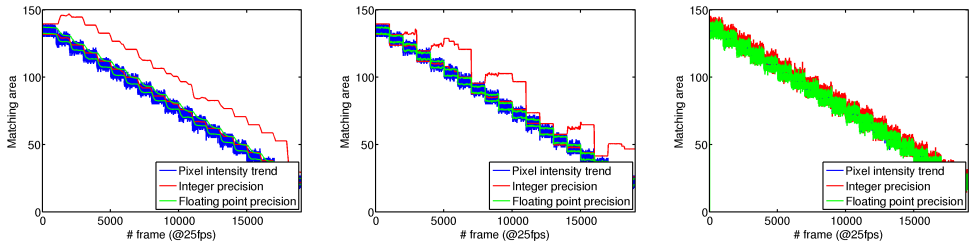
In the case of uncalibrated iGMM-x, simulation results in Fig. 7 and 8 reveal that the iGMM-x membership area (red) differs significantly from the GMM one (green), when average illumination is varying. On the other hand, when average illumination is constant, lack of calibration does not influence the membership areas (Fig. 12 and Fig. 9) as pixel matching is straightforward.

Fig. 6a and 6b depict the optimal values of N_μ and N_{σ^2} respectively that maximise the overlap between the iGMM-x membership area with the GMM one (red lines): we approximate those trends by the blue lines to correctly tune the updating step ξ (see Sec. 5.2.4).



(A) N_μ trend. (B) N_σ^2 trend.

FIGURE 6. N_μ N_σ^2 and trends.

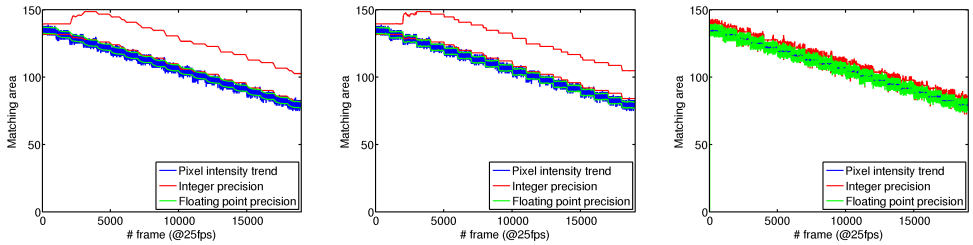


(A) $\alpha = 0.005$ (B) $\alpha = 0.05$ (C) $\alpha = 0.5$

FIGURE 7. Variable illumination @8.68 levels per minute: un-calibrated case.

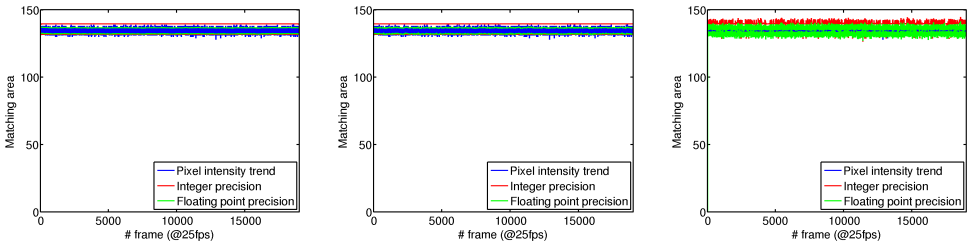
In Fig. 10 and 11 the membership area of the calibrated iGMM-x approach (red) is compared with the original GMM (green) under two different rates of illumination transitions and different values of learning rate.

Weight trend validation. In Sec. 5.2.5, two possible approximations are defined and stochastic updating techniques (see also Sec. 5.2.5) are used in both of them. To validate the proposed approximations, we propose a simulative approach of the life-time of Gaussians. As a starting condition, we consider a mixture where only the background mode MG_1 is active, and the effect of the creation and the evolution on a new Gaussian MG_2 is evaluated to understand how the MG_1 weight trend develops. Particularly, we propose to compare the evolution of the



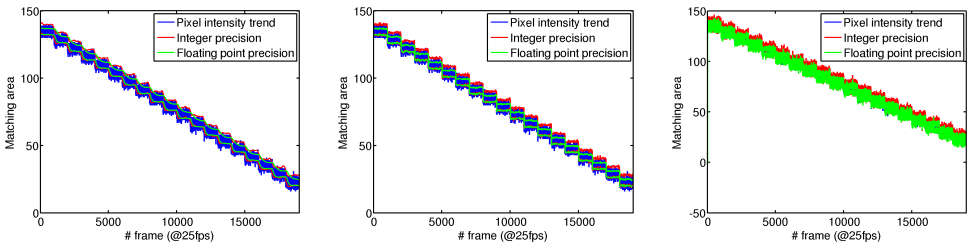
(A) $\alpha = 0.005$ (B) $\alpha = 0.05$ (C) $\alpha = 0.5$

FIGURE 8. Variable illumination @4.64 levels per minute: un-calibrated case.



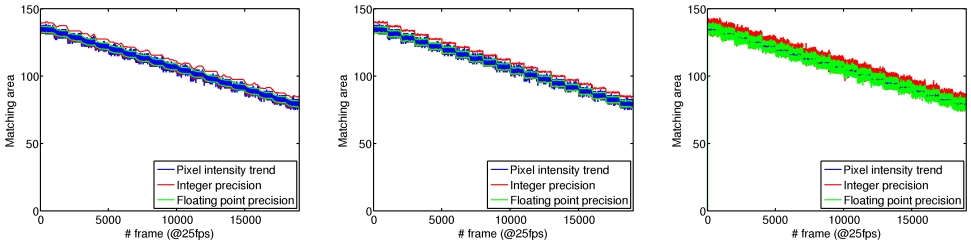
(A) $\alpha = 0.005$ (B) $\alpha = 0.05$ (C) $\alpha = 0.5$

FIGURE 9. Constant illumination: un-calibrated case.



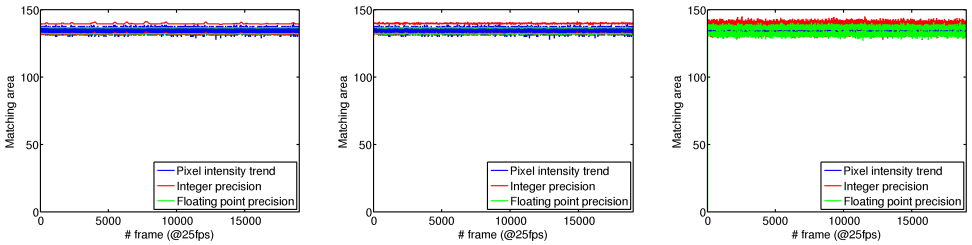
(A) $\alpha = 0.005$ (B) $\alpha = 0.05$ (C) $\alpha = 0.5$

FIGURE 10. Variable illumination @8.68 levels per minute: calibrated case.



(A) $\alpha = 0.005$ (B) $\alpha = 0.05$ (C) $\alpha = 0.5$

FIGURE 11. Variable illumination @4.34 levels per minute: calibrated case.



(A) $\alpha = 0.005$ (B) $\alpha = 0.05$ (C) $\alpha = 0.5$

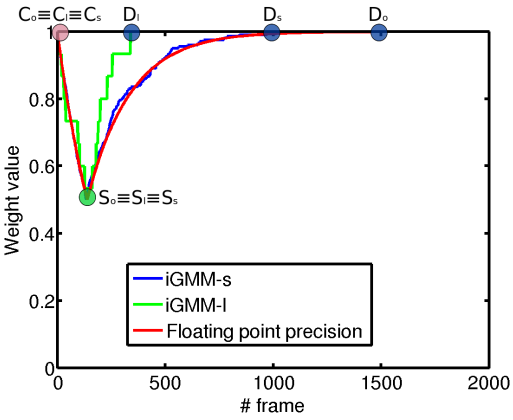
FIGURE 12. Constant illumination: calibrated case.

approximated techniques with the original GMM weight trend, especially taking into account the occurrence of the following three points: (i) the C point where the MG_2 mode is created (*creation point*), (ii) the S point, where MG_2 mode is sorted as most popular Gaussian (*swap point*), and (iii) the D point where the MG_1 is destroyed (*destruction point*). Thus, both iGMM-x techniques emulate correctly the creation point (see the C_l and C_s points in in Fig. 13). Moreover, the iGMM-l approximates linearly the weight trend (the green line in Fig. 13) and emulates correctly the Gaussian-swap point (see the point S_l), but it discards Gaussians too early (see the point D_l) in comparison with the original GMM case (the red line in Fig. 13). On the other hand, iGMM-s approximates as a staircase the weight trend (the green line in Fig. 13) and emulates correctly both the swap and the destruction point (see both the points S_s and D_s) compared with the GMM case. Consequently it is possible to state that, the iGMM-s technique approximates better the weight trend of the original approach. However, for large values of the learning rate, both cases have equivalent trends because the threshold T_X (see Sec. 5.2.5) for the staircase approximation is a very small number (see Fig 13c).

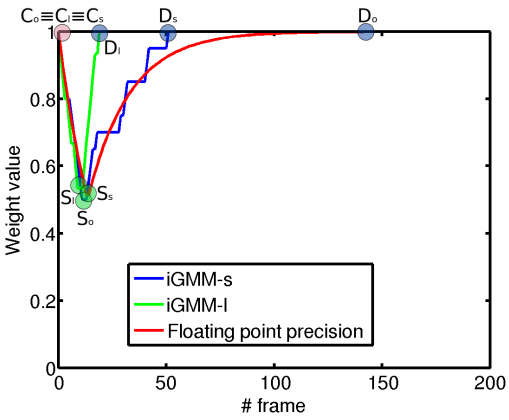
5.4.2. Comparison to the original GMM algorithm. In this section the performance for both iGMM-l and iGMM-s is compared to standard GMM by means of a qualitative analysis, i.e., a visual comparison on binarized foreground images, and quantitative analysis, i.e., an analysis on aggregated metrics. Finally in Sec. 5.4.2 the processing time of the proposed optimization (Sec. 5.2) over both the Raspberry Pi and the SeedEye boards is measured.

Qualitative and quantitative comparison. Firstly, a qualitative comparison between the binarized foreground images generated by the iGMM-x, the standard GMM and the ground-truth is shown (Tab. 4 and 5) for both data-sets: our approaches have comparable results to the original GMM method in foreground segmentation.

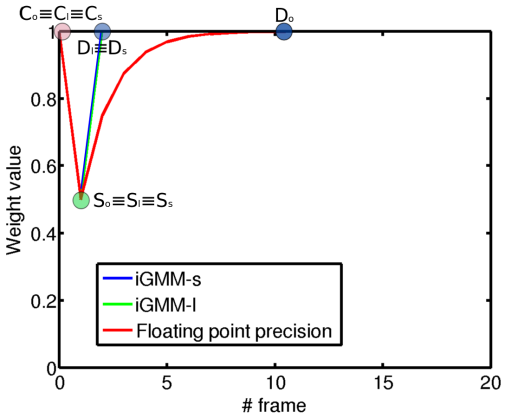
To understand the overall performance of the proposed algorithms, an aggregate analysis, based on Precision-Recall (or P-R) curves is presented. Particularly, the minimum distance of the P-R curves from the perfect classification point is computed for each value of learning rate so that the smaller the distance, the better the performance. In Fig. 14, the trends of the distance for P-R curves are shown as a function of the learning rate for both data-sets and all the cases (i.e., mixture of 2 and 3 Gaussians).



(A) $\alpha = 0.005$



(B) $\alpha = 0.05$



(C) $\alpha = 0.5$

FIGURE 13. Weight trends, where C_o, C_l and C_s are the creation points, S_o, S_l and S_s are the swap points, and D_o, D_l and D_s are the destruction points for GMM, iGMM-l and iGMM-s respectively.

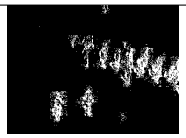
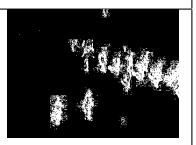
Picture	Ground-truth	GMM	iGMM-l	iGMM-s
				
				

TABLE 4. "Fudan pedestrian" data-set results using a 3 Gaussians mixture.

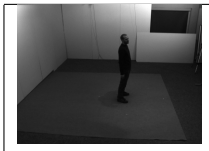
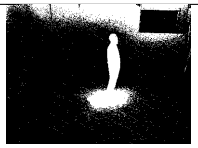
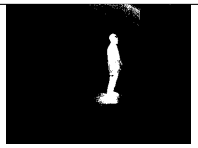
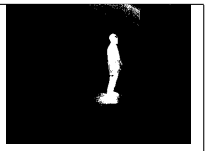
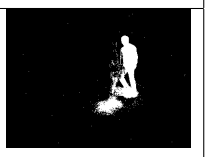
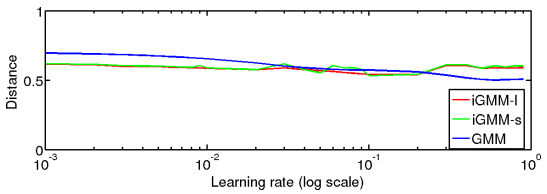
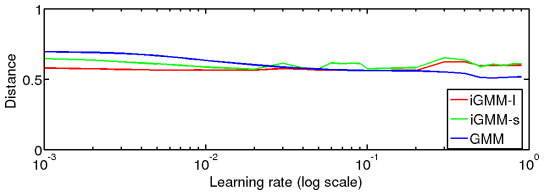
Picture	Ground-truth	GMM	iGMM-l	iGMM-s
				
				

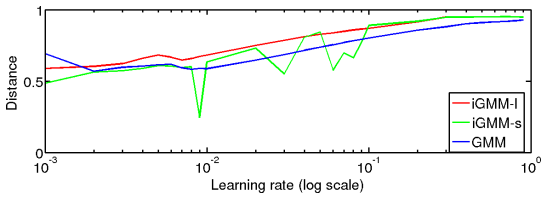
TABLE 5. "IXMAS" data-set results using a 3 Gaussians mixture.



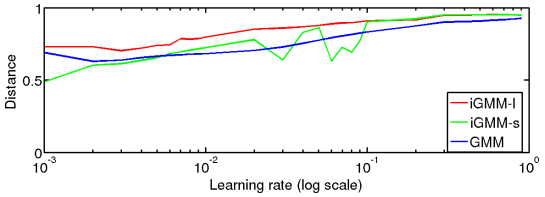
(A) Using "Fudan pedestrian" data-set (2 Gaussians).



(B) Using "Fudan pedestrian" data-set (3 Gaussians).



(C) Using "IXMAS" data-set (2 Gaussians).



(D) Using "IXMAS" data-set (3 Gaussians).

FIGURE 14. Minimum distance from the P-R curves to the perfect classification (1,1)

In Fig. 14c and 14d the evaluation on the fast movement data-set (namely Fudan data-set) are shown: in this case the considered metric has a constant trend because only few objects are absorbed by the background. On the other side, in Fig. 14a and 14b the impact of the considered techniques on the slow movement data-set (namely using IXMAS data-set) is shown. In this case the segmentation performance decrease with the increase of the learning rate, due to the foreground objects absorption inside the background. Apart the different trends, all the results shown in these figures demonstrate that the iGMM-x has comparable performance with original GMM.

Processing time comparison. In this section, the processing time of iGMM-l, iGMM-s and GMM techniques are compared to measure the real impact of implementations over the two boards described on Chap. 1. Specifically, on the Raspberry Pi board two different resolutions

TABLE 6. Processing times

	GMM [s]	iGMM-l [s]	iGMM-s [s]
Rasp. Pi 2G QQ-VGA	0.033	0.007	0.010
Rasp. Pi 3G QQ-VGA	0.036	0.013	0.025
Rasp. Pi 2G Q-VGA	0.136	0.031	0.041
Rasp. Pi 3G Q-VGA	0.148	0.057	0.104
SeedEye 2G (40x40)	0.084	0.002	0.003
SeedEye 3G (40x40)	0.115	0.003	0.006

are tested (Q-VGA and QQ-VGA) and on the SeedEye board only the 40x40 resolution is used in order to fit the scarce amount of PIC32 RAM for configurations of 2 and 3 Gaussians.

In Tab. 6, the mean values of the processing times are shown for all the different configurations in terms of used platform, resolutions and number of Gaussians. iGMM-l is $3 \div 5$ times faster than GMM and iGMM-s $1.5 \div 3$ times on the Raspberry Pi board.These ratios increase by a factor 10 (iGMM-l is around 40 times faster than GMM and iGMM-s $20 \div 28$ times) on the SeedEye board. The obtained results are consistent with the data shown in Tab. 1 in Sec. 5.3. Processing time of iGMM-s is longer than the one of iGMM-l due to the higher complexity of the staircase approximation (e.g., iGMM-s needs the use of "read" operations from the LUT_T buffer for every pixel). Moreover comparing the experiments with 2 and 3 Gaussians (in the same platform and with a constant resolution), the complexity of iGMM-s increases significantly with the number of Gaussians with respect to iGMM-l: the latter is about 1.5 times faster than the former in case of two Gaussians mixture, but about 2 times in case of three Gaussians.

CHAPTER 6

Next generation SC

Considering a SCN organized as described in Chap. 2, one of the biggest effort is the definition of a flexible and reconfigurable SCN node able to support the IoT requirements. This dynamic configuration is usually addressed by using software oriented solutions. In such a vision it is straightforward to modify both configuration parameters and applications at run-time, at the cost of avoiding possible low-level optimizations.

On the other side, using low-end devices as SC introduces a big effort in the porting (or the redefinition) of complex PC-based computer vision algorithms for embedded, as described in Chap. 5. In the above mentioned paper a GMM-based background subtraction algorithm is implemented over a low-complexity, low-memory and low-power micro-controller while analyzing both the performance gap with respect to a state-of-the-art implementation (based on a floating point arithmetic), and its capability in performing real-time image processing (namely the capability to process images at around 25 fps) in isolation.

However, the deployment of more complex pipelines (e.g., Histogram of Gradient [**DT05**]) in low-memory and low-power micro-controllers is not feasible: the memory is scarce and the time requirements cannot be satisfied. In respect of such problems, we propose a SC architecture based on (i) a micro-controller, able to handle both the high level operations and the network communications, and (ii) an FPGA, devoted at processing complex computer vision algorithms (i.e., pixel-wise operations and/or machine learning blocks). The proposed architecture can have benefit from the high degree of parallelism that an FPGA-based solution makes available, thus permitting to optimise the algorithms at very low level (i.e., hardware level). In this sense it is possible to use the programmable logic to perform elaboration directly on the data stream, as a pixel appears at input port, and without buffering frames. This behaviour might be defined as a *streaming* method because the data is treated as a continuous flow of information.

The only drawback of using FPGAs is that usually are exploited in the realisation of dedicated solutions to optimise the low level operations, neglecting both the reconfigurability and flexibility issues (derived from the IoT paradigm). These facts are demonstrated by the available scientific production that proposes only non-reconfigurable custom hardware based solutions targeted to specific applications: in the works of [**NDSO11**, **CDG12**, **MTT$^+$12**, **MTCM11**], the hardware is programmed from scratch in order to find the most optimised solution for a specific problem. More in detail, [**NDSO11**, **CDG12**, **MTT$^+$12**] present an implementation of the Histogram of Oriented Gradients (HOG) on streaming video flow, while [**MTCM11**] extracts aggregated features into a covariance matrix.

To fit the IoT paradigm requirements described in Chap. 2, we propose an innovative and reconfigurable FPGA internal architecture based on System on a Programmable Chip (SoPC) design technique able to overcome these limitations abstracting each functional block as a network resource [**MSP$^+$13**]. In this scenario, after the instantiation of a fixed set of hardware blocks (namely a set of computer vision algorithms able to define a certain group of consistent pipelines) our architecture permits at run-time to (i) change one block in the pipeline, (ii) instantiate a new application in parallel with another one already running, (iii) remove unused pipelines, and finally (iv) configure certain blocks with new parameters. All the above features give the possibility of creating a flexible and configurable solution without changing the whole FPGA bitstream. The paradigm that permits to implement all these feature is called "hardware software" co-design [**CGJ$^+$94**], and represents the state-of-the-art of the FPGA programming because merges the flexibility of software programming with the parallel computation allowed by hardware modules.

6.1. Architecture description

Every complex computer vision application can be seen as a pipeline of functional blocks. Following this view, and by using a set of hardware modules (i.e., hardware computer vision library) which implement some elaboration functions, we provide a reconfigurable solution able to combine pipeline items to implement different computer vision applications (e.g., tracking, recognition tasks, etc.): in this section, firstly we describe the architecture of the new generation SC (see Sec. 6.1.1), and then the FPGA internal architecture in order to fit the IoT paradigm requirements.

6.1.1. Architecture of a SC. The proposed SCN node is mainly based on a micro-controller and an FPGA. The former is in charge of all network communications (i.e., data transmission and feature sharing among nodes), moreover it plays the role of high level controller able to manage remote configuration requests. The latter implements the reconfigurable hardware pipeline. This hybrid structure allows to speed-up the elaboration performance with respect to pure software oriented solutions, while keeping the possibility of a remote configuration.

The proposed SCN node architecture is depicted in Figure 1, where it has been reported the high level architectural view of the node, as well as a more detailed representation in which the FPGA communicates with software components run by the micro-controller. The dual layer structure realises a separation between the computer vision heavy processing operations, performed by the FPGA, and the remote node interface, represented by the software abstraction into the micro-controller. In the FPGA, reconfigurable hardware modules that perform optimised pixel elaborations are implemented, as well as a SoftCore in charge of controlling modules parameters and the whole computer vision pipeline. The micro-controller, instead, gives a high level abstraction of the whole system by providing network based interfaces to control configuration parameters of elementary blocks, pipeline composition, and data transmission. Where IoT compliant SCN nodes are necessary, the micro-controller provides an additional abstraction

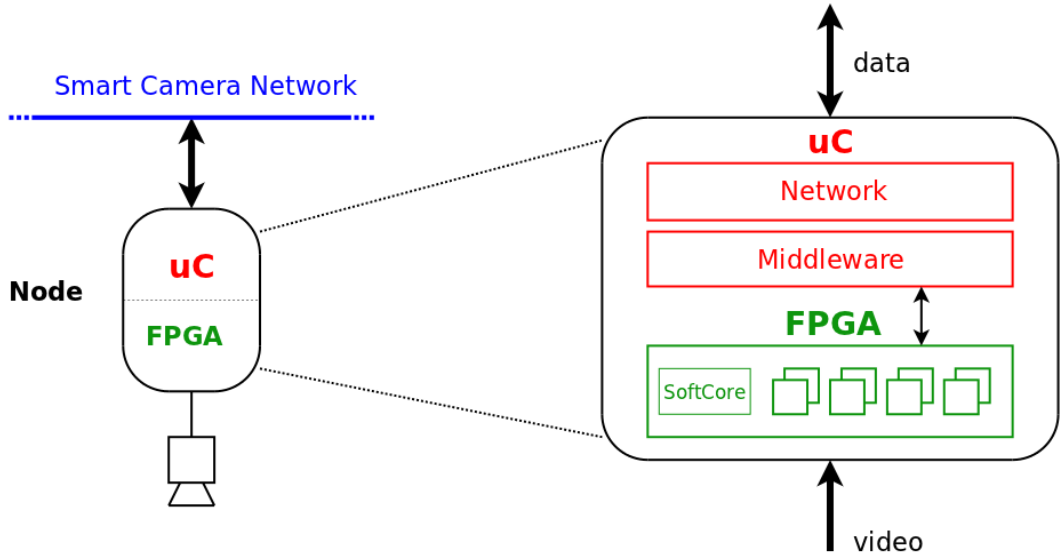


FIGURE 1. Proposed smart camera network node.

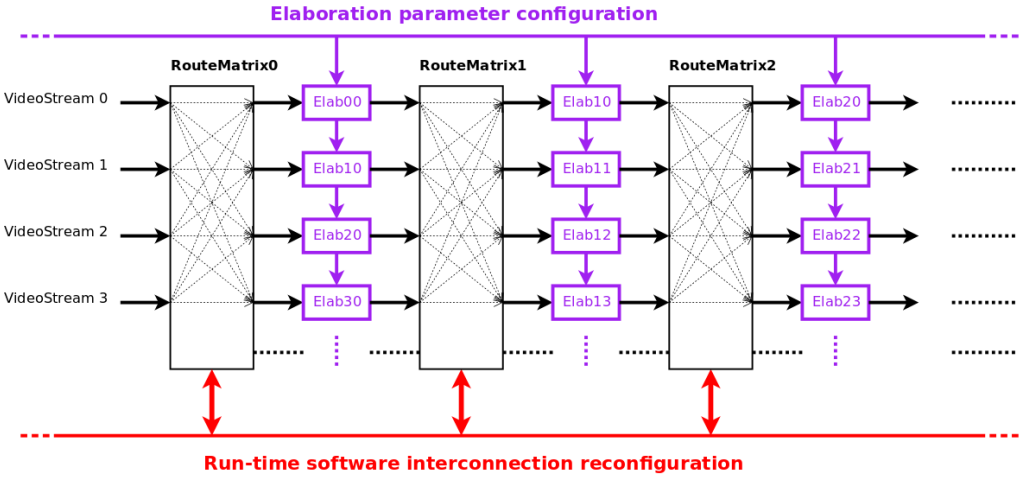


FIGURE 2. Camera_OneFrame architecture.

of the system through a middleware solution, thus permitting to show all the internal FPGA modules as resources for other network devices.

6.1.2. Internal FPGA architecture. The proposed FPGA architecture called *Camera_OneFrame*,is detailed in Figure 2. It permits to create a full reconfigurable pipeline in a SCN node. The whole architecture is designed focusing on interconnection modules, called *RouteMatrix*, and functional blocks, called *Elab*. The former realises the connections between the functional blocks, while the latter implement basic computer vision algorithms that can be part of a specific elaboration pipeline tunable at run-time. In the left side of the figure four video input ports, labeled as

VideoStream are shown. This architecture allows to have a multi-camera video inputs that can be parallel processed as a continuous pixel flow. The data flow, composed by one or more video streams, is captured and then processed by successive steps, represented by Elab blocks. In this respect, the captured flow has not an associated semantic, so that an user interacting with a configuration manager can select and compose the appropriate modules suited to the desired application. For instance, through the IoT middleware the desired pipeline can be remotely selected. Every parameter of each Elab block in the architecture as well as the connection among the blocks are managed by the SoftCore created into the FPGA.

The described internal FPGA structure can be expanded as a function of a set of possible applications. Indeed, the number of input and output ports, even the number of elaboration steps can be abstracted as parameters and then configured during the hardware compilation. This allows to modify the number of the parallel data flow, the amount of pipeline steps, or the elaboration blocks, without modifying the HDL instances, but easily inserting them as a new Elab instance using for example a graphic interface tool.

An image processing application has to be divided into functional blocks, algorithm steps, to unveil the internal data-path requested. Each functional block performs an associated algorithm, implemented by a hardware module available into a Hardware Library tool. This library contains a certain amount of functional blocks, defined using HDL, and then collected into a package made available as high level resource. After the instantiation phase, the system is ready to be compiled and programmed in the FPGA as a bit-stream. Afterward, though the FPGA bit-stream is statically programmed, the system still keeps the flexibility through the software configuration of blocks and connections. This leads to a dynamic and adaptive system, but optimised at the same time, because of the software re-configuration.

The proposed internal FPGA architecture introduces two degrees of freedom: (i) it is possible to grab the requested functional operation from a library, without any knowledge of HDL languages as it was in a model-based unit, and (ii) it is possible to configure the system at run-time to perform new pipelines with the already installed hardware modules configured and connected through the SoftCore.

6.1.3. RouteMatrix module. RouteMatrix is the core of the internal FPGA architecture: it is the connection point between hardware configuration and software programming. The logic behind our approach can be seen as a 3D-multiplexer, having a $N \in \mathbb{N}$ inputs and $M \in \mathbb{N}$ outputs (Figure 3), and configured by a $N \times M$ matrix through a dedicated bus (the red lines in Figure 3). In this way, the RouteMatrix internal logic guarantees that each output is connected to a selected $m - th$ input vector (with $m \in [1, M]$), to avoid data collision. On the other side every input vector can be connected to several outputs, in order to generate two or more twin sub-pipelines from the same source.
More in details, the RouteMatrix module permits to:

- define the routing path of a data stream (Figure 4a);
- handle the execution of parallel pipelines with different data sources (Figure 4b);
- split a data stream in two or more pipelines (Figure 4c).

6.1.4. Elab module. In this situation, we define a hardware block as the implementation of a certain computer vision algorithm in any HDL language (e.g., Verilog, VHDL, CAPH [**SBA11**]). Every block requires at least one input and one output with the related data-valid signals, that notify the validity of the data to the following blocks. The data-valid signals are necessary for enabling the streaming paradigm: in this direction the proposed architecture does not require to specify any data latency and processing time. Finally, the proposed architecture provides a dedicated bus for managing the FPGA region mapped as internal memory of the system. Such a memory can be directly addressed by the SoftCore for elaboration blocks configuration purposes.

6.2. Implementation

As described in the previous sections, the SCN architecture is based on a micro-controller and a FPGA: the former is in charge of managing the network communication and acts as high level resource controller, while the latter performs heavy processing operations by providing a reconfigurable internal architecture. To really evaluate the benefits of such architecture a real implementation has been performed by using two commercial off the shelf boards: the SeedEye (see Sec. 1.1.1) and the Terasic DE0-nano (see Sec. 1.1.3) boards.

Into the Altera Cyclone IV FPGA we have included an Altera NIOS-II SoftCore as data flow controller embedded into the elaboration logic: connecting such soft-core to the Elab and RouteMatrix blocks by using the Avalon-MM bus, the architecture permits to have a greater flexibility with respect to a pure hardware based solution. In fact, every elaboration block can be mapped on the Avalon-MM bus to be addressed from the NIOS-II as a standard memory location.

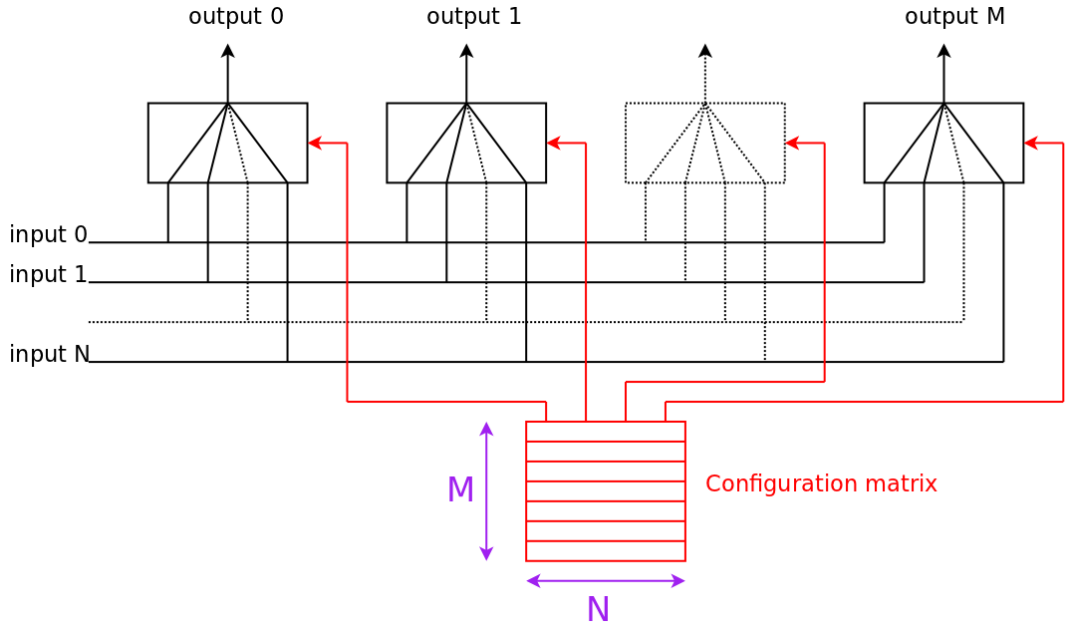
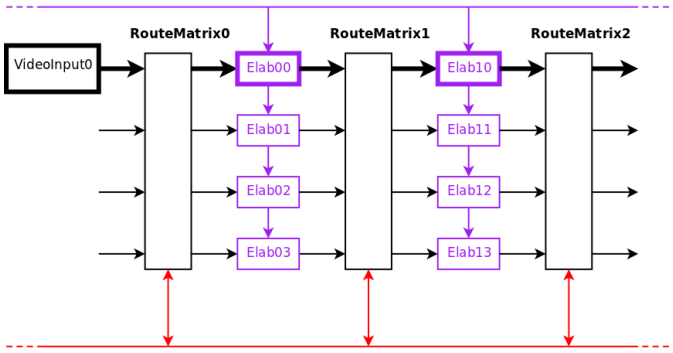
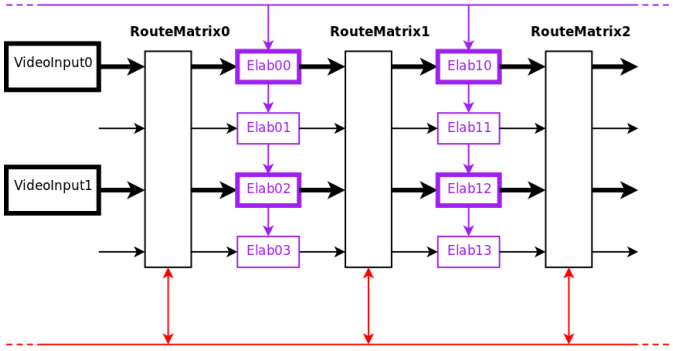


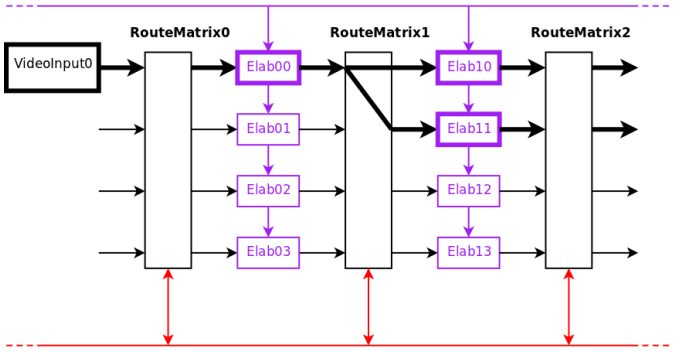
FIGURE 3. RouteMatrix internal architecture.



(A) Single pipeline.



(B) Parallel pipelines.



(C) Pipeline splitting.

FIGURE 4. RouteMatrix functionalities.

This main feature allow us to: (i) set up the block interconnection at run-time, using some dedicated registers in the RouteMatrix instances, and (ii) control the elaboration parameters for every block inserted into the elaboration pipeline.

6.2.1. Hardware library. The *Camera_OneFrame* architecture allows a user to instantiate hardware elaboration blocks without modifying their internal HDL behaviour. This result is realized according to the model-based design view as a consequence of the hardware abstraction. To the end of validating this claim a minimal HDL library has been developed. The library implements the following functions: (i) the *VideoSampler*, which realises the CMOS camera interface; (ii) *RemoteImg*, which performs a serial acquisition of an image from an UART link; (iii) *GradientHW*, that implements a spatial gradient extraction, and (iv) *HistogramHW*, which extracts the histogram of an image divided in cells of a configurable size (as used in the HOG algorithm [**DT05**]). Each block optimises a specific function by exploiting the advantages given by the hardware parallelism and the internal DSP modules embedded into the FPGA. Moreover, a complete run-time reconfiguration of all block parameters is allowed through a connection with the Avalon-MM bus. In Tab. 1 the FPGA occupancy of every block is described in terms of LE, on-chip RAM, and DSP elements by considering the EP4CE22 FPGA embedded on the Terasic DE0-nano board.

All the developed hardware library functions are compliant with the SoPC Builder [**Alt10**] tool provided by Altera as a part of the Quartus II software edition [**Alt12**]. SoPC Builder abstracts every HDL module as a functional entity into a graphical interface, thus helping its instantiation in the proposed architecture. In Figure 5 the design flow for a generic computer vision application is shown using the SoPC Builder tool: (i) the application is chosen (ii) and divided into functional elements; then (iii) the specific HDL blocks are instantiated using the above mentioned tool, and finally (iv) the code is compiled into a bit-stream.

6.3. Evaluation and Results

In this section the performance of the proposed architecture is evaluated in terms of data output latency. Particularly, we propose a suite of test cases aimed at validating all the claims

Table 1. Hardware library occupancy.

	Logic elements	RAM footprint (Bytes)	DSP (9x9 bit)
VideoSampler	200 (0,9%)	512	0
RemoteImage	200 (0,9%)	1	0
GradientHW	1200 (5,4%)	640	32
HistogramHW	850 (4,0%)	16384	0
RouteMatrix (x1)	400 (1,8%)	40	0

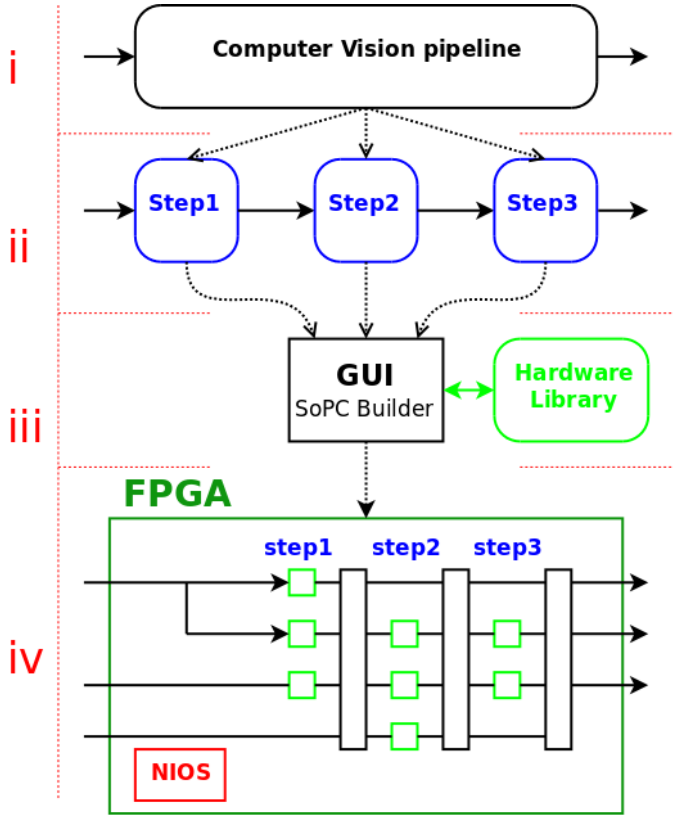


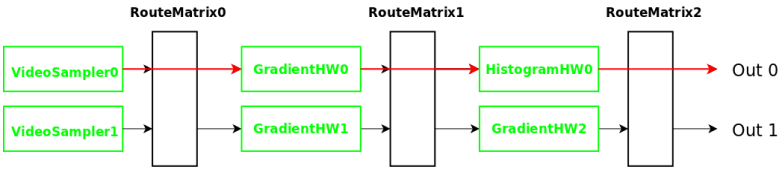
FIGURE 5. Example of the architecture design flow.

described in the previous sections by using the modules presented in Section 6.2. More in detail, in the test cases (shown in Figure 6) the following modules have been instantiated: two VideoSampler, three GradientHW, a HistogramHW, three RouteMatrix instances and finally a NIOSII SoftCore. In this scenario the proposed architecture permits to generate several combinations of the considered blocks, thus performing a set of applications fitting inside a rather small FPGA size. In fact, considering the occupancy data reported in Tab. 1, and considering the additional overhead due to the NIOSII SoftCore and the Avalon-MM bus, the whole bit-stream occupies only 31% of the LE, the 38% of on-chip memory and the 51% of the 9x9 DSP modules.

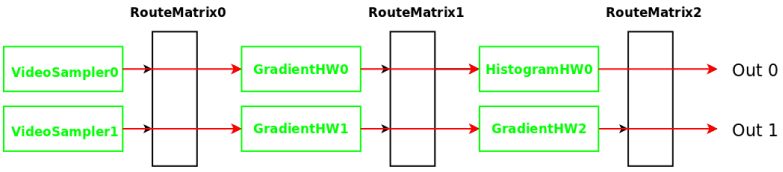
The all possible cases of the test suite are reported in Figure 6, where it is shown that the proposed architecture can: (i) handle a single pipeline (Figure 6a), (ii) handle two parallel pipelines using two cameras (Figure 6b) and (iii) split a data stream to follow two pipelines (Fig. 6c).

In order to evaluate the output latency of each pipeline, the time latency of each architecture block is shown in Tab. 2 in terms of FPGA clock cycles. These values measure the amount of clock cycles needed by data to flow inside each block, considering that they are implemented according to the streaming paradigm, without buffering a whole image.

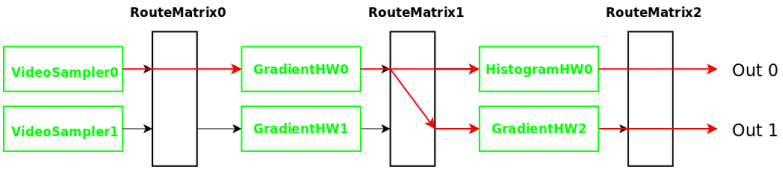
As reported in the table above, every module has a constant data delay, as a consequence of the hardware realisation based on the HDL description. More in detail, the VideoSampler



(A) Test 1: handling a single pipeline.



(B) Test 2: handling two pipelines.



(C) Test 3: split a data stream to follow two pipelines.

FIGURE 6. Test cases.

TABLE 2. Hardware module processing latency.

	Latency (clock cycles)
VideoSampler	0
RemoteImage	0
GradientHW	2
HistogramHW	2560
RouteMatrix	1

and RemoteImage modules do not introduce any latency time, because they do not perform any elaboration, but only manage clock speed conversions. The GradientHW, instead, introduces a latency time 2 clock cycles. The largest latency values are related to the HistogramHW module, that introduces a latency dependent on image size and cell size: in the contingent case, using 8x8 pixels cells and Q-VGA images, it is of 2560 clock cycles. The last implemented block, RouteMatrix, only introduces a latency of 1 clock cycle.

The overall system performance in time delay can be evaluated as sum of the latencies of the modules inserted into a specific pipeline plus the one introduced by the architecture structure elements (namely the RouteMatrix), as shown in the following equation:

TABLE 3. Tests case results.

	Latency (clock cycles)		Time (μs)	
	Out_0	Out_1	Out_0	Out_1
Test 1	2565	-	102,600	-
Test 2	2565	9	102,600	0,360
Test 3	2565	9	102,600	0,360

$$L_{total} = \sum_{i=0}^{N-1} L_i^B + M \cdot L^{RM} \tag{61}$$

where L_i^B is the latency of the i-th elaboration block, L^{RM} the latency of RouteMatrix, M the number of RouteMatrix iterations, and N the depth of the considered pipeline.

The latency time value in number of clock cycles can be easily converted in time delay by defining the frequency of the FPGA master clock. In our design, all the system runs at a frequency of 50MHz, thus the values of Tab. 2 can be expressed as delay time by multiplying them for the clock period (in this case $40ns$). Tab. 3 shows the delays for all the test cases depicted in Figure 6. For all the performed experiments the latency in terms of clock cycles and delay time is evaluated following the rule described in Equation 61.

CHAPTER 7

Conclusions

THIS thesis deals with a notable set of issues related to Smart Camera Networks; more specifically we introduced and discussed in one side the architectural organisation of these systems, and in the other the embedded implementation of the computer vision algorithms.

We generally followed a two-fold schema, looking at theoretical and simulation studies on one hand, and looking at real-world implementation on the other hand. This approach has provided concreteness to this research and contributed to the early exploitation of our experimental results.

In Chap.2, we contributed on the definition of the architecture of both Smart Camera Network and Smart Camera. More in detail, we proposed the necessity of using a Middleware of Things based on The Internet of Things paradigm, capable to solve the problems related to data sharing and aggregation together with other features related to software customization (via code update and parameters configuration).

Regarding the Smart Camera Networks applications, we looked at the problem of local on-board processing proposing several enabling application related to both Intelligent Transportation Systems and video-surveillance.

In Chap. 3 a full-edged simulation of the basic functionality of large-scale ITS, designed as a grid of services with a pervasive collection layer was described. Moreover three examples of ITS Smart Camera sensors was taken into account. Firstly, a multi-node processing method was introduced to decompose a complex computer vision task into a hierarchy of computationally simpler problems to be solved over the nodes of the network. In the same direction, we proposed a low-complexity and low-memory Smart Camera able to understand the parking spaces occupancy state, avoiding the duplication of the detection task. Both the presented techniques had good performance and was presented as an effective solution for vehicle parking lot monitoring applications in the domain of ITS. As last point of the chapter, we proposed a computer vision application enabled to count the vehicles passing in front a camera sensor, and able to be deployed on low-complexity and low-memory embedded devices. This application was based on the Linesensor theory that permitted to detect objects processing only 1D images (i.e., a line), and by exploiting the temporal redundancy of the movement. In this environment, we defined a low-complexity background model called SMM robust to illumination changes and capable to absorb standing objects. This technique was compared with a state-of-the-art algorithm (i.e., GMM) obtaining comparable results in terms of background/foreground segmentation but with a processing time reduced in average of the 93%.

Moreover in Chap. 4, we contributed on studying the problem of multimedia streaming taking into account the impact of the network noise on raw images and defining a low-complexity video

codec oriented to low-end devices. Thus, using real video and exploiting experimental loss traces gathered in a complex real-world scenario situated in the Pisa International Airport landside, the impact of IEEE802.15.4 packet losses on the received video quality was evaluated. To overcome data losses a forward error correction strategy based on BCH codes and fully compliant with the above mentioned standard was proposed, and its performance evaluated by means of a simulative study. The proposed data error recovery strategy was developed by defining a new MAC data frame which makes use of the bits kept reserved by the standard for future purposes. Performance results as a function of the error correction capacity of the code and its code rate showed how the proposed technique guaranteed to improve the final PSNR. Such a method permitted to instantiate only a video stream at 1 fps. To overcome such a limitation, we proposed a low-complexity change detection based algorithm aiming at reducing the required transmission bandwidth of video streaming applications based on raw images. In this situation, we was able to instantiate 4 video streams (@1fps) maintaining a good video quality level, in terms of PSNR.

In Chap. 5 we proposed our point of view in porting computer vision algorithms on embedded micro-controllers with the lack of FPU. Defining two new computer arithmetics to map floating point numbers into a sub-integer representation, we presented two realisations of mixture of Gaussian modelling in order to fit the strict constraints of the above mentioned embedded micro-controllers. To this purpose, we redefined the updating rules of each Gaussian parameter. Specifically, the mean value and the variance of each Gaussian was updated by a redefined and generalised "round" operation that emulates the original updating rules for a large set of learning rates. On the other hand, weights are represented by counters that was updated following stochastic rules to allow a wider range of learning rates and the weight trend is approximated by a line (iGMM-l) or a staircase (iGMM-s). Experimental results showed that both integer realisations had comparable accuracy on background/foreground segmentation compared to the original floating point precision, but significantly smaller memory footprint and lower computational cost over both the considered hardware platforms. Specifically, in our implementations, memory requirements was 6 and 12 times lower than *float* and *double* precision versions respectively. Processing time was reduced by 30%-79% on the Raspberry Pi board and by 95%-98% on the SEEDEYE board. Such a difference was justified by the usage of an optimised floating point library for the implementation of the original GMM on the Raspberry Pi board. The two proposed versions, iGMM-l and iGMM-s had similar performance and memory footprint. However, iGMM-l was faster by 50%-100% than iGMM-s, and the higher the number of Gaussians the higher the difference, due to the complexity of handling the staircase weight-trend.

Finally, in order to permit the deployment of more complex computer vision algorithm inside the camera node, we presented an innovative architecture concept for a Smart Camera. By leveraging the hardware software co-design concept we proposed a hybrid SCN node composed by a microcontroller and a reconfigurable FPGA architecture controlled by an internal SoftCore. Concerning the FPGA architecture, it represented a flexible and reconfigurable solution into a static FPGA bit-stream. The flexibility and the reconfigurability was enabled by providing the possibility of composing computer vision pipelines as well as configuring the elaboration block parameters, making this solution compatible with the above described IoT paradigm.

APPENDIX A

Proof from Sec. 5.2.5

A U niformly distributed random number $\mathbf{X} \in [0, MAX]$ is characterised by the cumulative distribution function in Eq. 62 and the probability density function in Eq. 63.

$$F_X(x) = P(\mathbf{X} \leq x) = \frac{x}{MAX} \tag{62}$$

$$f_X(x) = \frac{\partial}{\partial x} F_X(x) = 1 \tag{63}$$

On the other hand, if we consider the value $P(\mathbf{X}_n > T_X)$ and we use the probability properties, the following relation can be obtained:

$$P(\mathbf{X} > T_X) = 1 - F_X(x) = 1 - \frac{T_X}{MAX} \tag{64}$$

From the assumption in Eq. 51, Eq. 52 is true.

■

Bibliography

[A. 12] A. Azzarà, D. Alessandrelli, S. Bocchino, M. Petracca, and P. Pagano. Architecture, functional requirements, and early implementation of an instrumentation grid for the IoT. In *Proceedings of the 14th IEEE International Conference on High Performance Computing and Communications (HPCC)*, 2012. (to appear).

[AAP+12] A. Alessandrelli, A. Azzarà, M. Petracca, C. Nastasi, and P. Pagano. ScanTraffic: Smart Camera Network for Traffic Information Collection. In *Proceedings of European Conference on Wireless Sensor Networks*, pages 196–211, Trento, Italy, February 2012.

[ACm05] Hamideh Afsarmanesh and Luis M. Camarinha-matos. A framework for management of virtual organization breeding environments. In *In Proceedings of IMP group Conference*, pages 35–48. Springer, 2005.

[Agy] Kwabena W Agyeman. Low-power motion detection and building control using computer vision.

[Alt10] Altera. SoPC Builder User Guide. http://www.altera.com/literature/ug/ug_sopc_builder.pdf, 2010.

[Alt12] Altera. Quartus II Software. http://www.altera.com/products/ software/quartus-ii/about/qts-performance-productivity.html, 2012.

[AVF+09] Senyo Apewokin, Brian Valentine, Dana Forsthoefel, Linda Wills, Scott Wills, and Antonio Gentile. Embedded real-time surveillance using multimodal mean background modeling. In Branislav Kisaanin, Shuvra S. Bhattacharyya, and Sek Chai, editors, *Embedded Computer Vision*, Advances in Pattern Recognition, pages 163–175. Springer London, 2009.

[BCD+11] L. Berruti, C. Caligaris, L. Denegri, M. Perrando, and S. Zappatore. A grid approach for calibrating and comparing microscopic road traffic models. In F. Davoli, N. Meyer, R. Pugliese, and S. Zappatore, editors, *Remote Instrumentation Services on the e-Infrastructure*, pages 271–281. Springer US, 2011.

[BDVZ10] L. Berruti, F. Davoli, S. Vignola, and S. Zappatore. Performance analysis of a grid-based instrumentation device farm. In F. Davoli, N. Meyer, R. Pugliese, and S. Zappatore, editors, *Remote Instrumentation and Virtual Laboratories*, pages 23–32. Springer US, 2010.

[BT11] Liang Wang Ben Tan, Junping Zhang. Semi-supervised elastic net for pedestrian counting. *Pattern Recognition*, 2011.

[BYJK07] G. Balakrishnan, Mei Yang, Yingtao Jiang, and Yoohwan Kim. Performance analysis of error control codes for wireless sensor networks. In *Information Technology, 2007. ITNG '07. Fourth International Conference on*, pages 876–879, 2007.

[CAB+08] Phoebus Chen, Parvez Ahammad, Colby Boyer, Shih i Huang, Leon Lin, Edgar Lobaton, Marci Meingast, Songhwai Oh, Simon Wang, Posu Yan, Allen Y. Yang, Chuohao

Yeo, Lung chung Chang, J. D. Tygar, and S. Shankar Sastry. Citric: A low-bandwidth wireless camera network platform. In *In Proceedings of the International Conference on Distributed Smart Cameras*, 2008.

[CDG12] TP. Cao, D.Elton, and G.Deng. Fast buffering for fpga implementation of vision-based object recognition systems. *Journal of Real-Time Image Processing*, 7(3):173–183, 2012.

[CGJ+94] M. Chiodo, P. Giusto, A. Jurecska, H.C. Hsieh, A. Sangiovanni-Vincentelli, and L. Lavagno. Hardware-software codesign of embedded systems. *IEEE Micro*, 14(4):26–36, 1994.

[CGPP03] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1337 – 1342, oct. 2003.

[CJAS07] E. Culurciello, J.H.Park, A., and Savvides. Address-event video streaming over wireless sensor networks. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 849–852, 2007.

[CLZ+09] Mangesh Chitnis, Yao Liang, Jiang Yu Zheng, Paolo Pagano, and Giuseppe Lipari. Wireless line sensor network for distributed visual surveillance. In *PE-WASUN*, pages 71–78, 2009.

[CM02] C.F. Chiasserini and E. Magli. Energy Consumption and Image Quality in Wireless Video-Surveillance Networks. In *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 2357–2361, Lisboa, Portugal, September 2002.

[cor] Constrained RESTful Environments (CORE) working group. `http://www.datatracker.ietf.org/wg/core/charter/`.

[CRT06] E.J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489 – 509, feb. 2006.

[CSP+10] Mangesh Chitnis, Claudio Salvadori, Matteo Petracca, Giuseppe Lipari, and Paolo Pagano. Traffic related observations by line sensing techniques. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 373–374, New York, NY, USA, 2010. ACM.

[CSP+12] Mangesh Chitnis, Claudio Salvadori, Matteo Petracca, Paolo Pagano, Giuseppe Lipari, and Luca Santinelli. Distributed visual surveillance with resource constrained embedded systems. In Li-Minn Ang and Kah Phooi Seng, editors, *Visual Information Processing in Wireless Sensor Networks: Technology, Trends and Applications*. IGI Global Press, 2012.

[DBSM07] F. Dias, F. Berry, J. Serot, and F. Marmoiton. Hardware, design and implementation issues on a fpga-based smart camera. In *Distributed Smart Cameras, 2007. ICDSC '07. First ACM/IEEE International Conference on*, pages 20–26, 2007.

[Deb12] Debian project. Raspbian. http://www.raspbian.org/, 2012.

[DFL08] C. Duran-Faundez and V. Lecuire. Error resilient image communication with chaotic pixel interleaving for wireless camera sensors. In *Proceedings of the workshop on Real-world wireless sensor networks*, pages 21–25, April 2008.

[DFLL11] C. Duran-Faundez, V. Lecuire, and F. Lepage. Tiny block-size coding for energy-efficient image compression and communication in wireless camera sensor networks. *Signal Processing: Image Communication*, 26(8 - 9):466 – 481, 2011.

[Don06] D.L. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289 –1306, april 2006.

[DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 886–893, June 2005.

[EN86] Michele Elia and Fabio Neri. Generation of pseudorandom independent sequences. In *IASTED International Symposium MIC '86*, Feb. 1986.

[ETS06] ETSI. Electromagnetic compatibility and radio spectrum matters (erm); wideband transmission systems; data transmission equipment operating in the 2,4 ghz ism band and using wide band modulation techniques; harmonized en covering essential requirements under article 3.2 of the r&tte directive. *ETSI EN 300 328 V1.7.1*, 2006.

[ETS10] ETSI. Intelligent transport systems (its). communications architecture. *ETSI EN 302 665 V1.1.1*, 2010.

[FBCGCG11] J. Fernandez-Berni, R. Carmona-Galan, and L. Carranza-Gonzlez. FLIP-Q: A QCIF Resolution Focal-Plane Array for Low-Power Image Processing. *Solid-State Circuits, IEEE Journal of*, 46(3):669 –680, march 2011.

[FBCGLC+11] J. Fernandez-Berni, R. Carmona-Galan, G. Linan-Cembrano, A. Zarandy, and A. Rodriguez-Vazquez. Wi-FLIP: A wireless smart camera based on a focal-plane low-power image processor. In *Distributed Smart Cameras (ICDSC), 2011 Fifth ACM/IEEE International Conference on*, pages 1 –6, aug. 2011.

[Fen11] Dieter Fensel, editor. *Foundations for the Web of Information and Services - A Review of 20 Years of Semantic Web Research*. Springer, 2011.

[FKFB05] Wu-Chi Feng, Ed Kaiser, Wu Chang Feng, and Mikael Le Baillif. Panoptes: scalable low-power video sensor networking technologies. *ACM Trans. Multimedia Comput. Commun. Appl.*, 1(2):151–167, May 2005.

[GMBA+11] E.D. Gutierrez Mlot, S. Bocchino, A. Azzara, M. Petracca, and P. Pagano. Web services transactions in 6lowpan networks. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, pages 1–2, 2011.

[HBY08] Z. Hang, Y. Bo, and X. Yijun. Surveillance image compression based on roi and image restoration. In *Proceedings of International Conference on Embedded Software and Systems Symposia*, pages 485–489, July 2008.

[HKK07] Youngbae Hwang, Jun-Sik Kim, and In-So Kweon. Sensor noise modeling using the skellam distribution: Application to the color edge detection. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[HMM13] J. M. Hernández-Muñoz and L. Muñoz. The smartsantander project. In *Future Internet Assembly*, pages 361–362, 2013.

[HPFA07] Stephan Hengstler, Daniel Prashanth, Sufen Fong, and Hamid Aghajan. 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*, IPSN '07, pages 360–369, New York, NY, USA, 2007. ACM.

[HPH10] Mao-Hsiung Hung, Jeng-Shyang Pan, and Chaur-Heh Hsieh. Speed up temporal median filter for background subtraction. In *Proceedings of the 2010 First International Conference on Pervasive Computing, Signal Processing and Applications*, PCSPA '10, pages 297–300, Washington, DC, USA, 2010. IEEE Computer Society.

[IJG] The Independent JPEG Group's JPEG Software. http://www.ijg.org.

[ILRLB10] G. Iannizzotto, F. La Rosa, and L. Lo Bello. A wireless sensor network for distributed autonomous traffc monitoring. In *Human System Interactions (HSI), 2010 3rd Conference on*, pages 612 –619, may 2010.

[ipe] The WSN segment of the IPERMOB project. `http://ipermob.org/files/documents/003/003-3-5v1.0.pdf`.

[ipe09] The IPERMOB project: A Pervasive and Heterogeneous Infrastructure to control Urban Mobility in Real-Time. `http://www.ipermob.org`, July 2009.

[IPE11] IPERDS. http://rtn.sssup.it/index.php/projects/prjipermob/ipermobdataset, September 2011.

[ISO92] ISO/IEC. Digital compression and coding of continuous-tone still images.requirements and guidelines. September 1992. ISO/IEC 10918-1:1993.

[ISO96] ISO/IEC. MPEG-2 generic coding of moving pictures and associated audio information. *ISO/IEC 13818*, July 1996.

[ISO04] ISO/IEC. Jpeg 2000 image coding system. part 1: Core coding system. *ISO/IEC 15444-1:2000*, September 2004.

[ISO10] ISO. Communications access for land mobiles (calm) - architecture. *ISO IS 21217*, 2010.

[ISO12] ISO. Communications access for land mobiles (calm) - ipv6 networking. *ISO 21210:2012*, 2012.

[IT03] ITU-T. Advanced video coding for generic audiovisual services. *ITU-T Rec. H.264 & ISO\IEC 14496-10 AVC*, May 2003.

[ITU04] ITU. General principles and general reference model for next generation networks. *ITU-T Y.2011*, 2004.

[ITU06] ITU. Functional requirements and architecture of the ngn. *ITU-T Y.2012*, 2006.

[ITU10] ITU. Requirements for support of ubiquitous sensor network (usn) applications and services in the ngn environment. *ITU-T Y.2221*, 2010.

[KJC07] Ankur Kamthe, Lun Jiang, and Alberto Cerpa. Scopes: Smart cameras object position estimation system. Technical report, Computer Science and Engineering, School of Engineering, University of California, Merced; Merced, CA 95344, 2007.

[KKGB12] E. Kim, D. Kaspar, C. Gomez, and C. Bormann. Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing. RFC 6606, May 2012.

[KMS07] N. Kushalnagar, G. Montenegro, and C. Schumacher. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. RFC 4919, August 2007.

[LDMM+06] Benoî t Latré, Pieter De Mil, Ingrid Moerman, N VAN DIERDONCK, Bart Dhoedt, and Piet Demeester. Throughput and delay analysis of unslotted ieee 802.15.4. *Journal of Networks*, 1(1):20–28, 2006.

[LGA+05] Thomas Luckenbach, Peter Gober, Stefan Arbanowski, Andreas Kotsopoulos, and Kyle Kim. Tinyrest - a protocol for integrating sensor networks into the internet. In *in Proc. of REALWSN*, 2005.

[LL81] C. Leung and A. Lam. Forward error correction for an arq scheme. *IEEE Transactions on Communications*, 29(10):1514–1519, October 1981.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[LV01] B.P.L. Lo and S.A. Velastin. Automatic congestion detection system for underground platforms. In *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, pages 158 –161, 2001.

[MA11] T. Melodia and I. F. Akyildiz. *Research Challenges for Wireless Multimedia Sensor Networks*, pages 233–246. Springer London, July 2011.

[MATB11] M. Mowafi, F. Awad, E. Taqieddin, and O. Banimelhem. Experimental evaluation of image compression and transmission for tinyos-based imote2 platform. In *Innovations in Information Technology (IIT), 2011 International Conference on*, pages 173–178, 2011.

[MHPS12] T. Ma, H. Hempel, D. Peng, and H. Sharif. A survey of energy-efficient compression and communication techniques for multimedia in resource constrained systems. *IEEE Communications Surveys & Tutorials*, (99):1–10, 2012.

[Mic08] Microchip. PIC32MX3XX/4XX Family Data Sheet. ww1.microchip.com/downloads/en/DeviceDoc/61143E.pdf, 2008.

[MMM03] E. Magli, M. Mancin, and L. Merello. Low-complexity video compression for wireless sensor networks. In *Proceedings of International Conference on Multimedia and Expo*, pages 585–588, Baltimore, USA, July 2003.

[MMN+11] M. Magrini, D. Moroni, C. Nastasi, P. Pagano, M. Petracca, G. Pieri, C. Salvadori, and O. Salvetti. Visual sensor networks for infomobility. *Pattern Recognition and Image Analysis*, 21:20–29, 2011.

[MMP+10] Massimo Magrini, Davide Moroni, Paolo Pagano, Matteo Petracca, Gabriele Pieri, Claudio Salvadori, and Ovidio Salvetti. Image mining for informability. In *3rd Int. Workshop on Image Mining Theory and Applications (INSTICC Press, Angers, 2010)*, pages 35–44, Angers, France, 2010. INSTICC Press.

[mon] The Pentaho MONDRIAN project. `http://www.mondrian.pentaho.com`.

[Moo05] T. K. Moon. *Error Correction Coding: Mathematical Methods and Algorithms*. Wiley-Interscience, 2005.

[MRG08] S. Misra, M. Reisslein, and X. Guoliang. A survey of multimedia streaming in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 10(4):18–39, Fourth quarter 2008.

[MS95] N. J. B. McFarlane and C. P. Schofield. Segmentation and tracking of piglets in images. *Machine Vision and Applications*, 8(3):187–193, May 1995.

[MS04] E. Martinian and C.-E.W. Sundberg. Burst erasure correction codes with low decoding delay. *IEEE Transactions on Information Theory*, 50(10):2494–2502, October 2004.

[MSP+13] Luca Maggiani, Claudio Salvadori, Matteo Petracca, Paolo Pagano, and Roberto Saletti. Reconfigurable fpga architecture for computer vision applications in smart camera networks. In *Distributed Smart Cameras (ICDSC), 2013 Seventh International Conference on*, 2013.

[MTCM11] S. Martelli, D. Tosato, M. Cristani, and V. Murino. Fpga-based pedestrian detection using array of covariance features. In *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–6, August 2011.

[MTT+12] K. Mizuno, Y. Terachi, K. Takagi, S. Izumi, H. Kawaguchi, and M. Yoshimoto. Architectural study of hog feature extraction processor for real-time object detection. In *Proceedings of IEEE Workshop on Signal Processing Systems*, pages 197–202, October 2012.

[NC11] C. Nastasi and A. Cavallaro. Distributed target tracking under realistic network conditions. In *Proceedings of Sensor Signal Processing for Defence*, London, UK, September 2011.

[NDSO11] K. Negi, K. Dohi, Y. Shibata, and K. Oguri. Deep pipelined one-chip fpga implementation of a real-time image-based human detection algorithm. In *Proceedings of International Conference on Field-Programmable Technology*, pages 1–8, December 2011.

[NJS+08] L. Nachman, J.Huang, J. Shahabdeen, R. Adler, and R. Kling. IMOTE2: Serious Computation at the Edge. In *Proceedings of International Wireless Communications and Mobile Computing Conference*, pages 1118–1123, Crete Island, Greece, August 2008.

[OACM10] A. Luos Osorio, Hamideh Afsarmanesh, and Luis M. Camarinha-Matos. Towards a reference architecture for a collaborative intelligent transport system infrastructure. In Luis M. Camarinha-Matos, Xavier Boucher, and Hamideh Afsarmanesh, editors, *PRO-VE*, volume 336 of *IFIP Advances in Information and Communication Technology*, pages 469–477. Springer, 2010.

[OEL+06] Ake Ostmark, Jens Eliasson, Per Lindgren, Aart Halteren, and Lianne Meppelink. An infrastructure for service oriented sensor networks. *Journal of Computers*, 1(5), 2006.

[Par10] European Parliament. Framework for the deployment of intelligent transport system in the field of road transport and for interfaces with other modes of transport. Directive 2010/40/eu, 2010.

[PBR+11] S. Paniga, L. Borsani, A. Redondi, M. Tagliasacchi, and M. Cesana. Experimental evaluation of a video streaming system for wireless multimedia sensor networks. In *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, pages 165–170, 2011.

[PGS+11] M. Petracca, M. Ghibaudi, C. Salvadori, P. Pagano, and D. Munaretto. Performance evaluation of FEC techniques based on BCH codes in video streaming over wireless sensor networks. In *Proceedings of IEEE Symposium on Computers and Communications*, pages 43–48, Corfu, Greece, July 2011.

[PKC+05] T.R. Park, T.H. Kim, J.Y. Choi, S. Choi, and W.H. Kwon. Throughput and energy consumption analysis of ieee 802.15.4 slotted csma/ca. *Electronics Letters*, 41:1017–1019(2), September 2005.

[PKGZ08] Nissanka B. Priyantha, Aman Kansal, Michel Goraczko, and Feng Zhao. Tiny web services: design and implementation of interoperable and evolvable sensor networks. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, pages 253–266, New York, NY, USA, 2008. ACM.

[PPAS13] P. Pagano, M. Petracca, D. Alessandrelli, and C. Salvadori. Is ict mature for an eu-wide intelligent transport system? *Intelligent Transport Systems, IET*, 7(1):151–159, 2013.

[PPP+12] Matteo Petracca, Paolo Pagano, Riccardo Pelliccia, Marco Ghibaudi, Claudio Salvadori, and Christian Nastasi. On board unit hardware and software design for vehicular ad-hoc networks. In R. Daher and A. Vinel, editors, *Roadside Networks for Vehicular Communications: Architectures, Applications and Test Fields*. IGI Global Press, 2012.

[PSM+12] P. Pagano, C. Salvadori, S. Madeo, M. Petracca, S. Bocchino, D. Alessandrelli, A. Azzar, M. Ghibaudi, G. Pellerano, and R. Pelliccia. A middleware of things for supporting distributed vision applications. In *Proceedings of Workshop on Smart Cameras for Robotic Applications*, June 2012.

[PSOB05] G. Pekhteryev, Z. Sahinoglu, P. Orlik, and G. Bhatti. Image transmission over IEEE 802.15.4 and zigbee networks. In *Proceedings of IEEE International Symposium on Circuits and Systems*, pages 23–26, Kobe, Japan, May 2005.

[RAAKR05] Richard J. Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing*, 14:294–307, 2005.

[Ras11] Raspberry Pi Foundation. Raspberry Pi BOARD. http://www.raspberrypi.org/, 2011.

[RBI+05] Mohammad Rahimi, Rick Baer, Obimdinachi I. Iroezi, Juan C. Garcia, Jay Warrior, Deborah Estrin, and Mani Srivastava. Cyclops: In situ image sensing and interpretation in wireless sensor networks. In *In SenSys*, pages 192–204. ACM Press, 2005.

[RGGN07] Anthony Rowe, Adam G Goode, Dhiraj Goel, and Illah Nourbakhsh. Cmucam3: An open programmable embedded vision sensor. Technical Report CMU-RI-TR-07-13, Robotics Institute, Pittsburgh, PA, May 2007.

[RW08] B. Rinner and W. Wolf. An introduction to distributed smart cameras. *Proceedings of the IEEE*, 96(10):1565–1575, 2008.

[SAB+07] K. Shuaib, M. Alnuaimi, M. Boulmalf, I. Jawhar, F. Sallabi, and A. Lakas. Performance evaluation of ieee 802.15.4: Experimental and simulation results. *Journal of Communications*, 2(4), 2007.

[Sal06] D. Salomon. *Data Compression: The Complete Reference*. Springer-Verlag New York, Inc., Secaucus, USA, 2006.

[SBA11] Jocelyn Serot, Francois Berry, and Sameer Ahmed. Implementing stream-processing applications on fpgas: A dsl-based approach. In *Proceedings of the 2011 21st International Conference on Field Programmable Logic and Applications*, FPL '11, pages 130–137, Washington, DC, USA, 2011. IEEE Computer Society.

[Scu11] Scuola Superiore Sant'Anna and Evidence s.r.l. SeedEye board. http://www.evidence.eu.com/products/seed-eye.html, 2011.

[SG99] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 637 – 663, 1999.

[SG07] E. Setton and B. Girod. *Peer-to-Peer Video Streaming*. Spinger, 2007.

[SHBB13] Z. Shelby, K. Hartke, C. Bormann, and B.Frank. Constrained Application Protocol (CoAP). IETF Draft Version 18, Jun. 2013.

[She10] Z. Shelby. Embedded web services. *Wireless Communications, IEEE*, 17(6):52 – 57, December 2010.

[SHL+12] Y. Shen, W. Hu, J. Liu, M. Yang, B. Wei, and C.T. Chou. Efficient background subtraction for real-time tracking in embedded camera networks. In *Proceedings of the 10th ACM conference on Embedded Network Sensor Systems*, 2012.

[SMP+12] C. Salvadori, D. Makris, M. Petracca, J. Martínez del Rincón, and S. A. Velastin. Gaussian mixture background modelling optimisation for micro-controllers. In *Advances in Visual Computing*, volume 7431 of *Lecture Notes in Computer Science*, pages 241–251. Springer Berlin Heidelberg, 2012.

[SN09] Claudio Salvadori and Christian Nastasi. Cmos camera application note. `http://erika.tuxfamily.org/applicationnotes/34-flex-and-dspic/73-hv7131gp.html`, 2009.

[SPGP12] C. Salvadori, M. Petracca, M. Ghibaudi, and P. Pagano. *On-board Image Processing in Wireless Multimedia Sensor Networks: a Parking Space Monitoring Solution for Intelligent Transportation Systems*. CRC Press, December 2012.

[SPM+13] Claudio Salvadori, Matteo Petracca, Simone Madeo, Stefano Bocchino, and Paolo Pagano. Video streaming applications in wireless camera networks: A change detection based approach targeted to 6lowpan. *Journal of Systems Architecture*, (0):–, 2013.

[SPP+12] C. Salvadori, M. Petracca, R. Pelliccia, M. Ghibaudi, and P. Pagano. Video streaming in wireless sensor networks with low-complexity change detection enforcement. In *Proceedings of Baltic Congress on Future Internet Communications*, pages 8–13, Vilnius, Lithuania, April 2012.

[st103] Ieee standard for information technology – telecommunication and information exchange between systems – local and metropolitan area networks – specific requirements. part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans). IEEE Computer Society, October 2003.

[sum] The SUMO Urban Mobility simulation platform. `http://www.sumo.sourceforge.net`.

[Ter12] Terasic Technologies Inc. DE0-nano. www.terasic.com, 2012.

[TkTB05] Chen-Khong Tham, , Chen khong Tham, and Rajkumar Buyya. Sensorgrid: Integrating sensor networks and grid computing. *CSI Commun.*, 29:24–29, 2005.

[VJ04] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, May 2004.

[WC08] Z. Wu and H. Chen. *Semantic grid applications in intelligent transportation systems*, pages 210–226. Springer (USA), 2008.

[WGK+03] Von Welch, Jarek Gawor, Carl Kesselman, Sam Meder, and Laura Pearlman. Security for grid services. In *In Twelfth International Symposium on High Performance Distributed Computing (HPDC-12*, pages 48–57. IEEE Press, 2003.

[WHP+10] W. Wang, M. Hempel, Dongming Peng, Honggang Wang, H. Sharif, and Hsiao-Hwa Chen. On energy efficient encryption for video streaming in wireless sensor networks. *Multimedia, IEEE Transactions on*, 12(5):417–426, 2010.

[WPW+08] W. Wang, D. Peng, H. Wang, H.Sharif, and H.H. Chen. Energy-constrained distortion reduction optimization for wavelet-based coded image transmission in wireless sensor networks. *IEEE Transactions on Multimedia*, 10(6):1169–1180, October 2008.

[WRB06] Daniel Weinland, Remi Ronfard, and Edmond Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2-3):249–257, 2006.

[YD09] Dogan Yazar and Adam Dunkels. Efficient application integration in ip-based sensor networks. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '09, pages 43–48, New York, NY, USA, 2009. ACM.

[YGM08] Tingxin Yan, Deepak Ganesan, and R. Manmatha. Distributed image search in camera sensor networks. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, pages 155–168, New York, NY, USA, 2008. ACM.

[YYY10] Yang Yang, Chen Y, and W. Yi. Cross-layer forward error control for reliable transfer in wireless multimedia sensor networks. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1–2, 2010.

[ZEP+04] F. Zhai, Y. Eisenberg, T.N. Pappas, R. Berry, and A.K. Katsaggelos. An integrated joint source-channel coding framework for video transmission over packet lossy networks. In *International Conference on Image Processing*, pages 2531 – 2534, October 2004.

[ZLY+10] R. Zhou, L. Liu, S. Yin, A. Luo, X. Chen, and S. Wei. A VLSI design of sensor node for wireless image sensor network . In *Proceedings of IEEE International Symposium on Circuits and Systems*, pages 149–152, Paris, France, June 2010.