A Middleware of Things for supporting distributed vision applications

Paolo Pagano*, Claudio Salvadori, Simone Madeo, Matteo Petracca, Stefano Bocchino, Daniele Alessandrelli, Andrea Azzarà, Marco Ghibaudi, Giovanni Pellerano, Riccardo Pelliccia
National Laboratory of Photonic Networks, CNIT, Pisa, Italy
TeCIP institute, Scuola Superiore Sant'Anna, Pisa, Italy

***Abstract*— What is a smart camera? Which capabilities has it in terms of on-board processing and networking? In this paper we discuss about design and implementation issues of Wireless Multimedia Sensor Networks exploiting the potential of autonomous low power devices in terms of flexibility and reconfigurability. Considering a large scale scatter of camera nodes connected to the Internet via an IPv6-like suite of protocols we can enable these devices with a set of computer vision primitives, composable as required by the system. In this scenario a WMSN does not represent an application-specific entity (like a system specialized on tracking), but a framework providing a set of diversified services derived by the composition of those primitives. We will discuss some results obtained implementing computer vision techniques over microcontroller-based embedded systems. We suggest a possible architecture for a Middleware of Things able to handle data, events, and code; this design will permit the deployment of flexible distributed algorithms, dependent on the environmental context including directives by operators and the occurrence of specific events.**

I. Introduction

The application domains where Wireless Sensor Networks (WSNs) are being deployed are becoming more and more complex including monitoring and actuation capabilities 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.

Wireless Multimedia Sensor Networks (WMSNs) 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.

Consider for example a monitoring service in a parking lot [1]: the WSN nodes are initialized and configured to monitor the status of parking spaces (i.e., to detect the change from free to occupied and viceversa). In this case the time of service is relatively long (in the order of minutes), and the information is generally targeted to infomobility, non-critical applications like that of refreshing the display of Variable Message Signs.

All in all, state of the art applications kept simple: (i) the on-board logic in order to match the resource constraints of popular motes platforms; (ii) the topology of the network (e.g., avoiding overlap between the "fields of view" of cameras thus avoiding any need of consensus-based reconstruction protocols); (iii) the communication pattern (for instance adopting a many-to-one data transfer). In WSN scenarios any computer vision protocol must take into account the constraints in network bandwidth (up to 250 Kbps at the physical layer as standardized in IEEE802.15.4) thus preferring to encode reports calculated on board rather than streaming flat images.

As an example we report about the IPERMOB project [2] where a WMSN is 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 [3]. Some examples are shown in Figure 1.

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

The whole camera network is managed by a custom middleware developed as ad-hoc solution, Scantraffic [4] 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 Figure 2), as well as the period of a time-driven

* Corresponding author e-mail address: paolo.pagano@cnit.it

Fig. 2. A snapshot of the remote service for definining the ROI for the WMSN devices.

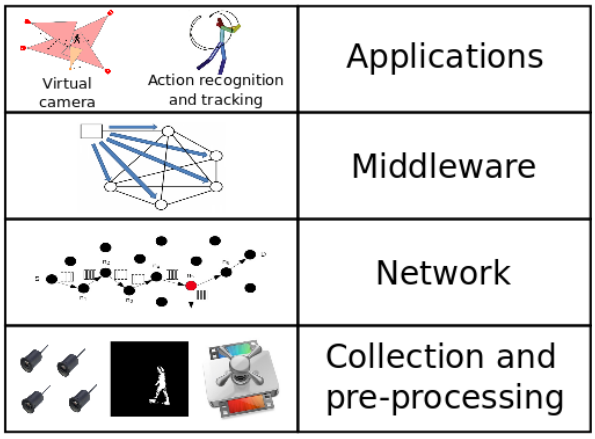

Fig. 3. A layered architecture for an information system based on WMSN.

monitoring service in which only reports about occupancy are transmitted.

Although any system making use of computer vision techniques in WSN scenarios is challenging in itself, the previous set-up poses several experimental problems severely limiting the industrial impact of the results obtained so far.

The sensor planning together with the configuration of ROIs is done at configuration time and is expected to change very seldom along the lifetime of the set-up; the total number of nodes is minimized for design requirements though system is weak against local points of failure; the system does not exploit the WSN potential of hosting collaborative applications and permitting in-network processing of partial records.

A. *Contribution of this paper*

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 Internet of Things (IoT) and Service Oriented Architectures (SOA); the camera nodes (together with gateways and high-end PCs) will be part of a Middleware of Things, set in between of the collection layer and the final users (or third party systems).

In Figure 3 the system is represented in a layered structure consisting of "Collection and pre-processing", "Network", "Middleware" and "Applications" functional blocks.

In this architecture light functions (e.g. data collection and on-board processing) will be implemented on low-end devices whereas computational and data intensive tasks (e.g. image composition and rendering) will be run on PCs; pervasive network topologies and middleware services will ease the sharing of the data and multi-node collaboration patterns.

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 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.

B. *Related work*

All the works proposed in WMSN literature describe system able to solve specific monitoring problems (i.e., tracking, video-streaming, etc.) without considering the benefit of a middleware capable to instantiate and compose a configurable and generic visual application in order to exploit a multi-node system.

The definition, development and eventual porting of computer vision algorithms over embedded systems is a challenging research topic.

From the hardware side, the market of consumer electronics components lets available low cost and low power micro-controllers suited to perform on-board several preprocessing operations (i.e., image compression, segmentation, etc.); from the algorithm perspective, research studies start proposing solutions for this type of platforms like for instance, Gaussian Mixture Modeling of background [5] and simple compression techniques [6].

Furthermore the integration of the above mentioned microprocessors over low-cost board equipped with wireless transceivers and CMOS cameras stimulates interest on distributed computer vision algorithms like recognition and tracking. In [7] a distributed and hierarchical action recognition algorithm is described for smart-home applications to localize a person at home and to recognize both coarse-level and fine-level activities using different machine learning and data fusion methods. In [8] a distributed tracker based on particle filter is discussed. More in detail the above mentioned tracking algorithm is tested over a realistic networking scenario using a network simulator engine, called "WiSE-MNet" [9], based on Castalia [10] and Omnet++ [11]. Finally, in [12] a noteworthy video coding paradigm is discussed; this work exploits the theory of source coding with side information (see also [13]) in order to deploy a distributed video compressor able to address: (i) error resiliency, (ii) compression efficency, and (iii) low complexity encoding.

In the remainder of the paper we will first discuss the ingredients to deploy a large scale WMSN: in Section II we will introduce and comment upon hardware platforms, firmware and software solutions for embedded devices, in Section III we will discuss networking and SOA-based transaction protocols. In Section IV we will propose a middleware design (based on [14]) and depict a sample application from the Smart Cities target domain. Finally in Section V we will drop our conclusions.

II. Camera nodes

A major issue in deploying Wireless Multimedia Sensor Network 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.

A. 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 [15] manufactured a new camera chipset (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[16]), tested on the field for the early detection of fire in forests.

Another approach, resulting from the IPERMOB project, is that of decoupling [18] 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. 4). A reduced version of the board is also available with no CPLD, suited for more simple applications and reducing a lot energy consumption.

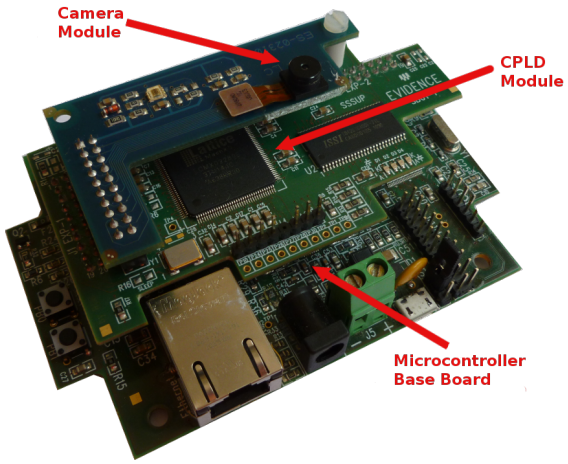

Fig. 4. The SEED-EYE board (with CPLD), distributed by Evidence [17]

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.

B. 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 [5] 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 cost sensor nodes for data preprocessing; (ii) more complex concentrator nodes for the computational intensive operations; (iii) low bandwidth

and noisy networks. In [19] 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 tecniques 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 wirelesss communication channel, we developed a specific compressor [6] able to extract and trasmit 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.

III. Networks

Considering Wireless Multimedia Sensor Network architectures proposed in recent works, nobody has attempted any integration with the Internet of Things. In [20] an interesting reference architecture for WMSNs is presented while mapping usual WSN node functionality into multimedia capabilities: low-end sensor nodes, tagged as *camera 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 WMSNs 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) [21], [22] represents a solution to be considered jointly with the Constrained Application Protocol (CoAP) [23], an HTTP-like protocol especially designed for constrained devices, presently in draft status along its standardization process at IETF.

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 [24].

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 WMSN, 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 [20] and taking into account the usual topology of 6LoWPAN, it is possible to define a new reference architecture (see Fig. 5) 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 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

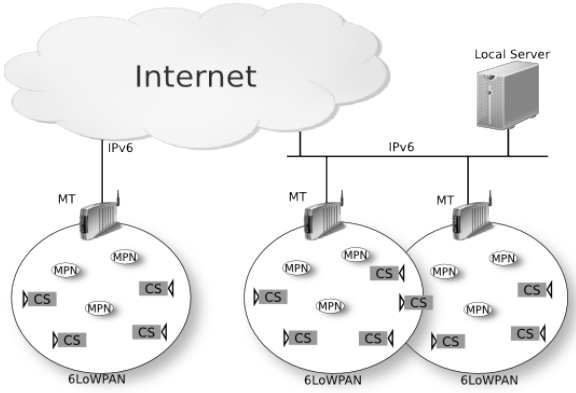

Fig. 5. WMSN architecture based on 6LoWPAN standards.

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.

IV. 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 famililies 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 bitrate, 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 Figure 6). The latter can also rely on third party systems like "clouds" which can offer storage and computing resources if required by the running application.

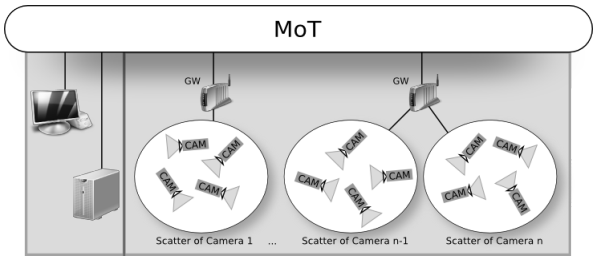

Fig. 6. WMSN architecture including a Middleware of Things.

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: For instance:
  1) switching from streaming to tracking;
  2) adding tracking as another service while keeping the streaming on;
  3) etc.
- 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 [14]. 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.

A. 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 Figure 7). 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.

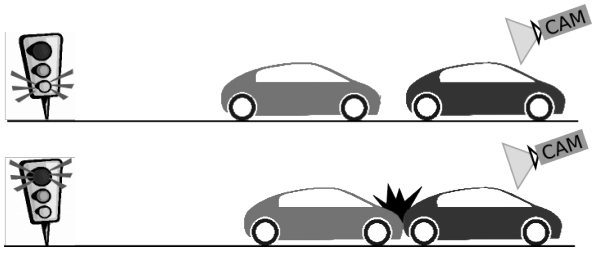

Fig. 7. A case study for Intelligent Transport System.

B. Remarks

In the real world (often far from research projects) smart cities will be deployed as a puzzle of services, manufactured by different vendors, often set-up, operated, and maintained by others.

The effort of developing common middleware services to support present and future applications deployed by different actors permits to promote these solutions to the real world without incurring the heavy task of redesign. The more vertical implementations are avoided, the more computer vision applications, although complex, can be felt as promising by industrial partners needing to quote the market share of the proposed solutions.

Communities interested in distributed computer vision can therefore profit of complementary know-how (in software engineering and networking) for deploying new innovative systems of great industrial impact.

V. Conclusions

In this paper we presented a new approach for supporting distributed vision applications in WMSN. We discussed issues related to hardware and software options in embedded systems together with the benefits of relying on IoT-oriented solutions for the network design. We discussed the benefit of a middleware capable of solving problems related to data sharing and aggregation together with other features related to software customization (via code update and parameters configuration).

Our target is that of designing a generic WMSN, capable of being reconfigured after deployment, capable of hosting fully customized applications depending on the events characterizing the specific WSN context (for instance that of an accident in Intelligent Transport Systems).

We argue that leveraging academic research results from different domains of expertize, it is possible to prototype such a network and deploy complex computer vision applications (for instance tracking or "Virtual Cameras").

The authors have recently submitted a research proposal for setting up a Living Lab for WMSN prototyping which will be opened to the world through an IPv6 network. In that context, these ideas will be tested against real-world case studies like those related to the vehicle movements in urban areas.

References

[1] 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.

[2] "The IPERMOB project," http://www.ipermob.org.

[3] 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*, vol. 21, pp. 20–29, 2011.

[4] 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*, Trento, Italy, February 2012, pp. 196–211.

[5] C. Salvadori, D. Makris, M. Petracca, J. del Rincón, and S. Velastin, "Gaussian mixture background modelling optimisation for microcontrollers," in *Proceedings of Advances in Visual Computing*, Crete, Greece, July 2012, pp. 241–251.

[6] 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*, Vilnius, Lithuania, April 2012, pp. 8–13.

[7] C. Wu, A. Khalili, and H. Aghajan, "Multiview activity recognition in smart homes with spatio-temporal features," in *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras*, Atlanta, USA, September 2010, pp. 142–149.

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

[9] C. Nastasi and A. Cavallaro, "WiSE-MNet: an experimental environment for wireless multimedia sensor networks," in *Proceedings of Sensor Signal Processing for Defence*, London, UK, September 2011.

[10] "The Castalia simulation model," http://castalia.npc.nicta.com.au/.

[11] "OMNeT++ Project," http://www.omnetpp.org/, 2010.

[12] A. Majumdar and K. Ramchandran, "Prism: an error-resilient video coding paradigm for wireless networks," in *Proceedings of International Conference on Broadband Networks*, San Jose, USA, October 2004, pp. 478 – 485.

[13] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *Information Theory, IEEE Transactions on*, vol. 22, no. 1, pp. 1 – 10, jan 1976.

[14] 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).

[15] 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*, vol. 46, no. 3, pp. 669 –680, march 2011.

[16] 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*, aug. 2011, pp. 1 –6.

[17] "Evidence s.r.l." http://www.evidence.eu.com.

[18] "The WSN segment of the IPERMOB project," http://ipermob.org/files/documents/OO3/OO3-3-5v1.0.pdf.

[19] 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*, Corfu, Greece, July 2011, pp. 43–48.

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

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

[22] 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, Internet Engineering Task Force, May 2012.

[23] Z. Shelby, "Embedded web services," *Wireless Communications, IEEE*, vol. 17, no. 6, pp. 52 – 57, December 2010.

[24] D. Fensel, Ed., *Foundations for the Web of Information and Services - A Review of 20 Years of Semantic Web Research*. Springer, 2011.