1

Smart Cameras for ITS in Urban Environment

Massimo Magrini[1], Davide Moroni[1], Gabriele Pieri[1] and Ovidio Salvetti[1]

[1]Signals and Images Lab, Institute of Information Science and Technologies (ISTI), National Research Council of Italy (CNR), Via Moruzzi 1, 56124, Pisa

1.1 Introduction

Fully automatic video and image analysis from traffic monitoring cameras is a fast-emerging field based on computer vision techniques with a growing impact on Intelligent Transport Systems (ITS).
Indeed the decreasing hardware cost and, therefore, the increasing deployment of cameras and embedded systems have opened a wide application field for video analytics both in urban and highway scenarios. It can be envisaged that several monitoring objectives such as congestion, traffic rule violation, and vehicle interaction can be targeted using cameras that were typically originally installed for human operators [1].
On highways, systems for the detection and classification of vehicles have successfully been using classical visual surveillance techniques such as background estimation and motion tracking for some time. Nowadays existing methodologies have good performance also in case of inclement weather and are operational 24/7. On the converse, the urban domain is less explored and more challenging with respect to traffic density, lower camera angles that lead to a high degree of occlusion and the greater variety of street users. Methods from object categorization and 3-D modelling have inspired more advanced techniques to tackle these challenges. In addition, due to scalability issues and cost-effectiveness, urban traffic monitoring cannot be constantly based on high-end acquisition and computing platforms; the emerging of embedded technologies and pervasive computing may alleviate this issue: it is indeed challenging yet definitely important to deploy pervasive and untethered technologies such as Wireless Sensor Networks (WSN) for addressing urban traffic monitoring.
On the basis of these considerations, the aim of this chapter is to introduce scalable technologies for supporting ITS-related problems in urban scenarios; in particular we survey embedded solutions for the realization of smart cameras that can be used to detect, understand and analyse traffic-related situation and events thanks to an on-board vision logics. Indeed, to suitably tackle scalability issues in the urban environment, we propose the use of a distributed, pervasive system consisting in a Smart Camera Network (SCN), a special kind of WSN in which each node is equipped with an image sensing device. For this reason, SCN are also known as Visual Sensor Networks (VSN). Clearly, gathering information from a network of scattered cameras, possibly covering a large area, is a common feature of many video surveillance and ambient intelligence systems. However, most of classical solutions are based on a centralized approach: only sensing is distributed while the actual video processing is accomplished in a single unit. In those configurations, the video streams from multiple cameras are encoded and conveyed (sometimes thanks to multiplexing technologies) to a central processing unit which decodes the streams and perform processing on each of them. With respect to those configurations, the need to introduce distributed intelligent system is motivated by several requirements, namely [2]:

- Speed: in-network distributed processing is inherently parallel; in addition, the specialization of modules permits to reduce the computational burden in the higher level of the network: in this way the role of the central server is relieved and it might be actually omitted in a fully distributed architecture.
- Bandwidth: in-node processing permits to reduce the amount of transmitted data, by transferring only information-rich parameters about the observed scene and not the redundant video data stream.
- Redundancy: a distributed system may be re-configured in case of failure of some of it components, still keeping the overall functionalities.
- Autonomy: each of the nodes may process the images asynchronously and may react autonomously to the perceived changes in the scene.

In particular, these issues suggest moving a part of intelligence towards the camera nodes. In these nodes, artificial intelligence and computer vision algorithms are able to provide autonomy and adaptation to internal conditions (e.g. hardware and software failure) as well as to external conditions (e.g. changes in weather and lighting conditions). It can be stated that in a VSN the nodes are not merely collectors of information from the sensors, but they have to blend significant and compact descriptors of the scene from the bulky raw data contained in a video stream.
This naturally requires the solution of computer vision problems such as change detection in image sequences, object detection, object recognition, tracking, and image fusion for multi-view analysis. Indeed, no understanding of a scene may be accomplished without dealing with some of the above tasks. As it is well known, for each of such problems there is an extensive corpus of already implemented methods provided by the computer vision and the video surveillance communities. However, most of the techniques currently available are not suitable to be used in VSN, due to the high computational complexity of algorithms or to excessively demanding memory requirements. Therefore, ad hoc algorithms should be designed for VSN, as we will explore in the next sections.
In this chapter, we first envisage applications of smart cameras and VSN to urban scenarios, highlighting specific challenges and peculiarities. Embedded vision nodes are introduced and a brief survey of existing hardware solutions is provided; the implementation of general computer vision algorithms on smart cameras and VSN is then addressed. We move further describing two sample ITS applications, namely analysis of traffic status and parking lot monitoring. In the first sample application, the estimation of vehicular flows on a lane is performed by using a lightweight computer vision pipeline that is somewhat dissimilar form the conventional one used on standard architecture. In the second sample application, an approach to parking lot monitoring is presented; here the vision nodes can collaborate each other for producing more accurate and robust results, e.g. by resorting to the middleware for VSN presented in Chapter ? by Petracca et al. A smart camera prototype designed with ITS application in mind is presented in Section 1.5, while in Section 1.6 we present its envisaged application scenarios and experimental results.

1.2 Applications to urban scenarios

According to [1], there has been an increased scope for the automatic analysis of urban traffic activity. This is partially due to the additional numbers of cameras and other sensors, enhanced infrastructure and consequent accessibility of data. In addition, the advances in analytical techniques for processing video streams together with increased computing power have enabled new applications in ITS. Indeed, video cameras have been deployed for a long time for traffic and other monitoring purposes, because they provide a rich information source for human understanding. Video analytics may now provide added value to cameras by automatically extracting relevant information. This way, computer vision and video analytics become increasingly important for ITS.
In highway traffic scenarios, the use of cameras is now widespread and existing commercial systems have excellent performance. Cameras are used tethered to ad hoc infrastructures, sometimes together with Variable Message Signs (VMS), RSU and other devices typical of the ITS domain. Traffic analysis is often performed remotely by using special broadband connection, encoding, multiplexing and transmission protocols to send the data to a central control room where dedicated powerful hardware technologies are used to process multiple incoming video streams [3]. The usual monitoring scenario consists in the estimation of traffic flows distinguished among lanes and vehicles typologies together with more advanced analysis such as detection of stopped vehicles, accidents and other anomalous events for safety, security and law enforcement purposes.
By converse, traffic analysis in the urban environment appears to be much more challenging than on highways. In addition in urban environments several extra monitoring objectives can be supported in principle by the application of computer vision and pattern recognition techniques, including the detection of complex traffic violations (e.g. illegal turns, one-way streets, restricted lanes) [4,5], identification of road users (e.g. vehicles, motorbikes and pedestrians) [6] and of their interactions understood as spatiotemporal relationships between people and vehicle or

vehicle to vehicle [7]. For these reasons, it is worthwhile to apply the wireless sensor network approach to the urban scenario.
Generally, we may identify four different scopes that can be targeted thanks to video-surveillance based systems, namely i) safety and security, ii) law enforcement, iii) billing and iv) traffic monitoring and management. Although in this chapter we focus mostly on the latter, we give a brief overview of each of them.
Safety and security relate to the prevention and prompt notification both of proper traffic events and of roadside events typical of urban environment. From one side, detection of events like car accidents, stopped vehicles, general obstacles, tunnel accidents, floods and landslides is of fundamental importance: real time detections allows for immediate response that might be life-saving. In most cases, the information obtainable thanks to visual nodes most be usefully complemented with other detectors. For example, smoke detectors play a more crucial role than video sensors for dangerous tunnel accidents involving fire. In general, visual information turns out to be essential when complex scenes with non-trivial semantics should be understood. For instance, in case of landslides and obstacle detection, technologies based on radar might provide extended reliability and be fully operational also in case of adverse meteorological conditions (e.g. rain and snow) and low visibility situation (e.g. foggy weather). However, also in this case, integration of video information might be useful in reducing false positives by using object recognition methods thus improving the overall performance. Safety in urban environment regards also the detection of roadside events like crimes and vandalisms. For instance the commercial available solution [8] includes methods for detecting car park surfing, that is the act of a pedestrian getting out as passenger of one car and moving to another. This is indeed the usual hunting behaviour of car thieves.
Law enforcement is based on the detection of unlawful acts and to their documentation for allowing the emission of a fine. Besides well-known and established technologies e.g. for streetlight violations, vision based systems might allow for identification of more complex behaviour e.g. illegal turns or trespassing on a High Occupancy Vehicle (HOV) lane. For instance, Xerox has recently produced a vehicle passenger detection system that uses geometric algorithms to detect whether a seat is vacant or occupied without using facial recognition [9]. Documentation of unlawful acts is usually performed by acquiring a number of images sufficient for representing the violation, combined with automatic number plate recognition (ANPR) for identifying the offender vehicle.
ANPR is also a common component of video-based billing and tolling. Also in this case there are a number of established technologies provided as commercial solutions by many vendors [10]. A peculiarity of urban billing systems with respect to highways is the non-intrusiveness requirement: it is not possible to alter the normal vehicular flow but a free-flow tolling must be implemented. Technologies satisfying this requirement are already available and used in cities such as London, Stockholm and Singapore but their actual cost prevents their massive deployment in medium-size or low-resource cities. Nevertheless, the availability of such billing technologies at a lower cost may pave the way to the collection of fine-grained data analytics of vehicular flows, road usage and congestions, allowing for the implementation of adaptive Travel Demand Management (TDM) policies aimed at a more sustainable, effective and socially acceptable mobility applied to urban and metropolitan contexts. It is likely that other technologies not based on video but for instance on NFC might become widespread in the near future to fill this gap.
Finally, traffic monitoring and management is related to extraction information from urban observed scenes that might be beneficial in several contexts. For instance, real-time vehicle counting might be used to assess level of service on a road and detecting possible congestions. Such real-time information might then be used for traffic routing; either by providing directly suggestion to user (e.g. by VMS) of by letting a trip planner deploys these data to search for an optimal path. Finally, statistics on vehicular flows may be used to understand mobility patterns and help stakeholders to improve urban mobility. Usually, vehicle count is performed by inductive loops which provide precise measurements and some vehicle classification. The major drawback of inductive loops is that they are very intrusive in the road surface and therefore require a rather long and expensive installation procedure. Furthermore, maintenance also requires intervention on the road pavement and therefore is not sustainable in most urban scenarios. Radar-based sensing systems are also used for vehicle counting and simple analytics but in cases of congestions they generally exhibit deteriorated performance. In the last years there has been interest in video-based counting system based on imaging devices, also embedded. Some solutions, such as Traficam [11], are commercially available and provide vehicle count in several lanes at an intersection. A version of Traficam working in the infrared spectrum is also available. Besides vehicle counting, traffic management can include the extraction of other flow parameters, e.g. discriminating the components of flow generated by different vehicle classes (car, track, buses, bike and motorbikes) and assessing the transit speed of each detected vehicle.
Another interesting topic is the monitoring of parking slots. Indeed, although there are several commercial parking slot monitoring solutions, most of them are only suitable for structured and closed parking lots, often requiring great installation costs to be adapted to already existing parking facilities. Visual nodes, instead, are flexible for

application to several scenarios, including roadside parking spaces. The visual nodes can provide then information pertaining the availability or not of a single parking space. This might be useful for example in the monitoring of special spaces, such as disabled space or spaces featuring electric vehicle charging station.
From this brief survey of urban scenario applications, we might argue that pervasive technologies based on vision turn out to be of interest when i) there is some semantics to be understood that cannot be acquired solely on the basis of scalar sensors, ii) there is no possibility or no sufficient revenue in actuating installation of tethered technologies, such as intrusive sensor or high-end devices and iii) there is the need of a scalable architecture, capable of covering a metropolitan area. Since computer vision is not application specific, an additional feature of a VSN is represented by the fact that it can be re-adapted to the changing urban environment and reconfigured even for supporting new scene understanding tasks by just updating the vision logics hosted in each sensor. On the converse, scalar sensors (like inductive loops) and specific sensors like radar have no flexibility in providing information different form the one they were built for.
In summary, with respect to more conventional ITS, that are often are limited to close and rich systems, pervasive technologies based on VSN can thus provide a cost-effective collaborative sensing infrastructure which has intrinsic scalability features (since the architecture is made out of logical islands corresponding to VSN segments), can be adapted to several -even unstructured- scenarios and employs advanced yet low cost technologies. Thus VSN may be exploited at several levels, impacting on transportation systems to be set up in small, mid-size and big cities as well as in unstructured road networks.

1.3 Embedded vision nodes

Following the trends in low-power processing, wireless networking and distributed sensing, VSN are experiencing a period of great interest, as shown by the recent scientific production [12]. A VSN consists of tiny visual sensor nodes called camera nodes, which integrate the image sensor, the embedded processor and a wireless RF transceiver. The large number of camera nodes forms a distributed system where the camera nodes are able to process image data locally (in-node processing) and to extract relevant information, to collaborate with other-cameras – even autonomously – on the application specific task, and to provide the system user with information-rich descriptions of the captured scene.

1.3.1 Features of available vision nodes

In the last years, several research projects produced prototypes of embedded vision platforms which may be deployed to build a VSN. Among the first experiences, Panoptes project [13] 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 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, and 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 [14] provided another representative smart camera for sensor networks. The camera nodes is equipped with a low-performance ATmega128 8-bit RISC microcontroller. 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 colour, and 16-bit YCbCr colour at CIF resolution (352x288). In the Cyclops board, the camera module contains a complete image processing pipeline for performing demosaicing, image size scaling, colour correction, tone correction and colour space conversion. In the MeshEye project [15] an energy-efficient smart camera mote 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 colour 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 colour 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-cost embedded vision system is represented by the CMUcam series [16], developed at the Carnegie Mellon University. More precisely the third generation of the CMUcam series 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-cost embedded system with simple vision capabilities, so as to obtain smart sensors. The hardware platform is constituted by a CMOS camera, an ARM7 processor and a slot for MMC cards. Standard RF transceiver (e.g. TELOS mote) can be easily integrated. CMUcam4 is now on the market, featuring a Parallax P8X32A and an Arduino compatible shield. More recently, the CITRIC platform [17] integrates in one device a camera sensor, a CPU (with frequency scalable up to 624MHz), 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 VSN. The design of the CITRIC system allows performing 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 background subtraction and iii) self-localization of the camera nodes in the network. The aforementioned electronics projects are examples of existing devices that can be turned into sensor nodes of a visual wireless sensor network. In Section 1.5 we will present an alternative smart camera prototype.

1.3.2 Computer vision on embedded nodes

Embedded nodes equipped with an image sensor need special computer vision algorithms to be turned into actual smart cameras. The versatility of computer vision offers the possibility to tackle a great range of problems, by drawing on the extensive literature on the subject. Indeed for most of computer vision tasks such as change detection, object detection, object recognition, tracking, and image fusion for multi-view analysis there exists an arsenal of already implemented methods (see [18] for a survey of change detection algorithms); however most of the techniques currently available are not suitable to be used in VSN. Indeed, as shown by the examples reported in previous sections, embedded nodes have usually very constrained memory and computational power. Sometimes microcontroller architectures are also used, where floating point operation are not natively supported. In addition, power consumption is often limited in self-powered or battery-powered sensors: intensive operations might reduce autonomy below acceptable levels. For these reasons, conventional computer vision pipeline used in standard centralized infrastructure cannot be used on VSN, but a redesign of the employed algorithms is necessary. The redesign may range from an optimization for the embedded architecture (use of lookup tables, approximation in computations and introduction of heuristics) to more drastic changes in the pipeline in order to implement a more lightweight approach. Some attempts to employ non-trivial image analysis methods over VSN have been done. For example [19] presents a VSN 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 [20]. SIFT descriptors are indeed 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 particular, using SIFT descriptors allows retrieving the object of interest from the scene, no matter at which scale it is imaged. Interesting computer algorithms are also provided on the CMUcam3 vision system. Besides basic image processing filters (such as convolutions), methods for real-time tracking of blobs on the base either of colour 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 [21], 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. Machine learning classifiers, such as the Viola-Jones detector, are very useful for deployment on embedded nodes, in order to provide a semantic interpretation of the scene. Indeed, the automatic detection of semantic concepts in videos and images represents an attempt to overcome the semantic distance between machines and humans, a distance that can be defined as the lack of coincidence between the information that one can extract from the sensor data and the interpretation that this same data can give a user in a given situation [22]. Bridging this gap with vision logics that can automatically recognize certain semantic concepts (such as *car*, *person*, or *obstacle*) is the real strength that makes VSN unparalleled with respect scalar sensors.

The detection of basic concepts can be performed using supervised learning methods, where a sufficient set of labelled data (annotated so that they contain, or do not contain, the concept to be detected) is used in a training phase to learn a model of the concept. The system learned in a supervised manner (e.g. a Support Vector Machine, SVM [23]) extracts some features of the images and their labels as input and learns a relationship model between these visual features and the concept. We can then classify new images not used in the training process using the learned model. The low-level visual features typically used are the colour histograms depicting the image or parts of the image, histograms of gradients, points of interest [20], edges, motion and depth to cite a few.

In more complex cases, it is required to detect events instead of simple objects; events can be formally represented and recognized as a set of objects (including people) interacting in time and space, e.g. a group of pedestrian crossing the street, a load loss, a car accident and car park surfing for instance. The regions in images of a video sequence are labelled by objects, and the spatial relationships between objects changes between images as a result of their interactions. Machine learning algorithms require a preliminary (and generally computational intensive) learning phase to produce a trained classifiers. Clearly, when the methods are to be deployed on a VSN, the preliminary learning phase may be accomplished off-site, while only the already trained detectors need to be ported to the visual 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 – if available –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). For example, since license plates have standard sizes, if we know roughly the scale of the image, we could expect to observe a plate only if the size in pixel of the area is compatible with the actual physical size. For what regards the binary classifiers itself, among various possibilities, the Viola-Jones method is particularly appealing for use on VSN. Indeed, such classifier is based on the use of the so-called a rejection cascade. A 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. The cascade 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 in 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 maybe interrupted at an earlier stage in the cascade, thus producing a quick even though less reliable output, which may be anyhow sufficient for solving the current decision making problem. In the same way, by controlling the threshold in the last stage of the cascade, the VSN may dynamically select the optimal trade-off between false alarm rate and detection rate needed in a particular context.

One advantage of VSN with respect monocular systems is the fact that they can inherently exploit multi-view information. 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 over a VSN.

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. Such ideas are used for tackling the parking lot monitoring problem by using multiple cameras and a special middleware layer for complex event composition (Section 1.4).

1.4 Implementation of computer vision logics on embedded systems for ITS

In this Section, two sample ITS applications based on computer vision over VSN are reported. The first one regards the estimation of vehicular flows and is based on a lightweight computer vision pipeline that is dissimilar form the conventional one used on standard architectures. In the second sample application, an approach to parking lot monitoring is presented; here the vision nodes can cooperate among each other for producing more accurate and robust results, performing an inter-node decision regarding parking slot occupancy status. The inter-node decision logics can be implemented in an Internet of Things (IoT) framework, e.g. by using the middleware for VSN presented in Chapter ? by Petracca et al. The presented applications are based and extend previous work [25, 26].

1.4.1 Traffic status and level of service

The analysis of traffic status and the estimation of level of service are usually obtained by extracting information on the vehicular flows in terms of passed vehicles, their speed and typology. Conventional pipelines start with i) background subtraction and move forward to ii) vehicle detection, iii) vehicle classification, iv) vehicle tracking and v) final data extraction. On VSN, instead, it is convenient to adopt a lightweight approach; in particular data only in Region of Interest (RoI) is processed, where the presence of a vehicle is detected. On the basis of these detections, then, flow information is derived without making explicit use of classical tracking algorithms.

More in detail, background subtraction is performed only on small quadrangular RoIs. Such shape is sufficient for modelling physical rectangles under perspective skew. In this way, when low vision angles are available (as common in urban scenarios), it is possible to deal with a skewed scene even without performing direct image rectification, which can be computationally intensive on an embedded sensor. The quadrangular RoI can be used to model lines on the image (i.e. a 1 pixel thick line) either.

On such RoI, lightweight detection methods are used to classify a pixel as *changed* (in which case it is assigned to the foreground) or *unchanged* (in which case it is deemed to belong to the background). Such decision is obtained by modelling the background. Several approaches are feasible. The simplest one is represented by straightforward *frame differencing*. In this approach, the frame previous to the one that is being processed is taken as background. A pixel is considered changed if the frame difference value is bigger than a threshold. Frame differencing is one of the fastest methods but has some cons in ITS applications; for instance a pixel is considered changed two times: first when a vehicle enters and, second, when it exits from the pixel area. In addition, if a vehicle is homogeneous and it is imaged in more than one frame, it might be not detected in the frames after the first. Another approach is given by *static background*. In this approach, the background is taken as a fixed image without vehicles, possibly normalized to factor illumination changes. Due to weather, shadow, and light changes the background should be updated to yield meaningful results in outdoor environments. However strategies for background update might be complex; indeed it should be guaranteed that the scene is without vehicles when updating. To overcome these issues, algorithms featuring *adaptive background* are used. Indeed this class of algorithms is the most robust for use in uncontrolled outdoor scenes. The background is constantly updated fusing the old background model and the new observed image. There are several ways of obtaining adaptation, with different levels of computational complexity. The simplest is to use an *average image*. In this method, the background is modelled as the average of the frames in a time window. Online computation of the average is performed. Then a pixel is considered changed if it is different more than a threshold from the corresponding pixel in the average image. The threshold is uniform on all the pixels. Instead of modelling just the average, it is possible to include the standard deviation of pixel intensities, thus using a statistic model of the background as a single *Gaussian distribution*. In this case, both the average and standard deviation images are computed by an online method on the basis of the frames already observed. In this way, instead of using a uniform threshold on the difference image, a constant threshold is used on the probability that the observed pixel is a sample drawn from the background distribution which is modelled pixel by pixel as a Gaussian. *Gaussian Mixture Models* (GMM) are a generalization of the previous method. Instead of modelling each pixel in the background image as a Gaussian, a mixture of Gaussians is used. The number k of Gaussians in the mixture is a fixed parameter of the algorithm. When one of the Gaussian has a marginal contribution to the overall probability density function, it is disregarded and a new Gaussian is instantiated. GMM are known to be able to model changing background even in cases where there are phenomena such as trembling shadows and tree foliage [27]. Indeed in those cases pixels clearly exhibit a multimodal distribution. However GMM are computationally more intensive than a single Gaussian. *Codebooks* [28] are another adaptive background modelling techniques presenting computational advantages for real-time background modelling with respect GMM. In this method, sample background values at each pixel are quantized into codebooks which represent a compressed form of background model for a long image sequence. That allows to capture even complex structural background variation (e.g. due to shadows and trembling foliage) over a long period of time under limited memory.

Several ad hoc procedures can be envisaged starting with the methods just described. In particular, one important issue concerns the policy by which the background is updated or not. In particular, if a pixel is labelled as foreground in some frame, we might want that this pixel does not contribute in updating the background or that it contributes to a lesser extent. Similarly, if we are dealing with a RoI, we might want to fully update the background only if no change has been detected in the RoI; if a change has been detected instead, we may decide not to update any pixel in the background.

The data extraction procedure starts taking in input one or more RoIs for each lane suitably segmented in foreground/background by the aforementioned methods. When processing the frame acquired at time t, the algorithm decides if the RoI R_k is occupied by a vehicle or not. The decision is based on the ratio of pixels changed with respect the total number of pixels in R_k, i.e. $a_k(t)$=#(changed pixels in R_k)/ #(pixels in R_k). Then $a_k(t)$ is compared to a threshold τ in order to evaluate if a vehicle was effectively passing on R_k. If $a_k(t) > \tau$ and at time t-1 no vehicle was detected, then a new transit event is generated. If a vehicle was already detected instead at time t-1, no new event is generated but the time length of the last created event is incremented by one frame. When finally at a time $t+k$ no vehicle is detected (i.e. $a_k(t) < \tau$) , the transit event is declared as accomplished and no further updated. Assuming that the vehicle speed is uniform during the detection time, the number of frames k in which the vehicle has been observed is proportional to the vehicle length and inversely proportional to its speed. In the same way, it is possible to use two RoIs R_1 and R_2, lying on the same lane but translated by a distance Δ, to estimate the vehicle speed. See Figure 1.1. Indeed, if there is a delay of δ frames, the vehicle speed can be estimated as $v=\Delta/(\delta*\nu)$ where ν is the frame rate. The vehicle length can in turn be estimated as $l=k/v$. Clearly the quality of these estimates varies greatly with respect to several factors, and is in particular due to a) frame rate and b) finite length of RoIS. Indeed, the frame rate generates a quantization error which leads to the estimation of the speed range; therefore the approach cannot be used to compute the instantaneous speed. For what regards b), an ideal detection area is represented by a detection line, having length equal to zero. Otherwise, a localization error affects any detection, i.e. it is not know exactly where the vehicle is inside the RoI at detection time. The use of a 1-pixel thick RoIs alleviates the problem but it results in less robust detections. This problem introduces some issues both in vehicle length and speed computations, because in both formulas we use the nominal distance Δ and not the precise (and unknown) distance between the detections. This is the drawback in not using a proper tracking algorithm in the pipeline, which would require however computational resources not usually available on embedded devices. Nevertheless, it is possible to provide a speed and size class for each vehicle. For each speed and vehicle class a counter is used to accumulate the number of detections. Temporal analysis on the counter is sufficient for estimating traffic typologies, average speed and analysing the level of service of the road identifying possible congestions.

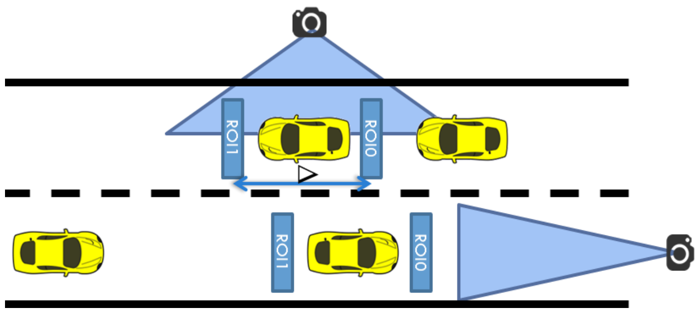

Figure 1.1 RoI configuration for traffic flow analysis.

1.4.2 Parking monitoring

As a second sample application, an algorithm suitable for deployment on VSN has been studied and designed for the analysis of parking lot occupancy status. The approach followed is based on frame differencing, in order to highlight the changes in the RoIs, with respect to a background reference image. In the following details on the change detection algorithm will be given, and then specifications regarding the modality of detecting the occupancy rate of a single parking slot will be introduced. Afterwards, the cooperation of nodes within the VSN in order to improve the detection performances is described.

In order to improve the computational efficiency, the frame differencing for detecting changes is performed only on predetermined RoI in the acquired frame. Each of the RoI corresponds to a specific parking slot, and for each of the regions the absolute accumulated pixel-wise differences is reported; such a difference is dynamically weighted in order to correct and improve the robustness of the algorithm with respect to environmental light changes.

In order to perform this improvement, normalized versions of the images are computed and used, with respect to global illumination parameters (average and variance of both the current and reference image). The sum of the differences is scaled with respect to the size of the RoI, and finally it is stored in a buffer. At this point verification occurs in order to detect the eventual change. In particular, a comparison of the stored actual value with the historical values allows filtering out possible spurious values (i.e. exceeding the threshold) due to, e.g. the presence of shadows. In the same way the stored value is compared with another threshold in order to detect possible changes with respect to the reference image. At this point the algorithm yields a first outcome which is a value regarding the occupancy status of the specific parking slot.

Once the algorithm has computed values regarding the occupancy for each parking slot (corresponding to the RoIs), an intra-node occupancy detection process occurs. In order to avoid transitory events (e.g. such as walking people and shadows casted by external objects), the occupancy status becomes effective and is transmitted to the VSN, only after being observed consecutively for a specific number of acquired frames.

For each parking slot, the algorithm yields a confidence value in the range [0..255]. Meaning that values next to 0 represent an almost no-change detected with respect to the reference value, and thus the slot is likely to be free; higher values, on the other hand, indicate that main changes have occurred in the observed scene, and thus the slot is likely to be occupied. Figure 1.2 shows the flow chart of the algorithm.

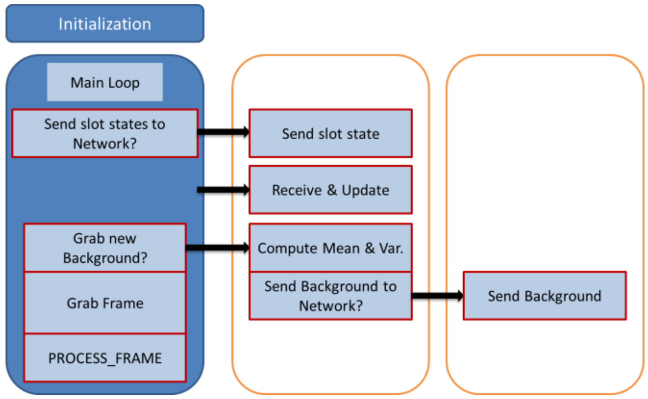

Figure 1.2 Flow chart representation of the parking lot occupancy algorithm.

At a higher level of the VSN, the confidence values produced by the single nodes as a 256 levels number should be transformed to binary values corresponding either to free or busy parking slots, thus taking a final decision regarding the parking availability.
To this end, local confidence values will be propagated through the VSN thanks to a middleware layer. In particular when a parking space is monitored by more than one sensor node, the final decision regarding its occupancy is obtained at an inter-node level.
More in detail, this final decision is obtained aggregating all the confidence values produced by the different nodes (which are statically dislocated and have static tables of the monitored parking slots). If a slot *k* is monitored by *n=n(k)* sensor nodes, and being $v_1^k(t), \cdots, v_n^k(t)$ the confidence values measurements from each single sensor node at time *t*, then the aggregated measure is computed as:

$$v^k(t) = \sum_{i=1}^{n} \omega_{i,k} v_i^k(t)$$

Where the $\omega_{i,k}$ are the non-negative weights and:

$$\sum_{i=1}^{n} \omega_{i,k} = 1$$

Thus the final decision $st^k(t)$regarding the slot *k* is obtained performing a comparison with a threshold ε:

$$st^k(t) = \begin{cases} 1 & if\ v^k(t) > \varepsilon \\ 0 & if\ v^k(t) \le \varepsilon \end{cases}$$

In order to implement a more robust algorithm, avoiding meaningless oscillations, the above decision is further improved using two levels of thresholds$\varepsilon_1 < \varepsilon_2$, and considering the status of the slot at the previous measure obtained at time *t-1*:

$$st^k(t) = \begin{cases} 1 & \text{if } v^k(t) > \varepsilon_2 \text{ or } (v^k(t) > \varepsilon_1 \text{ and } st^k(t-1) = 1) \\ 0 & \text{if } v^k(t) < \varepsilon_1 \text{ or } (v^k(t) < \varepsilon_2 \text{ and } st^k(t-1) = 0) \end{cases}$$

Weights $\omega_{i,j}$ are determined heuristically for each physical configuration of the VSN, while the thresholds $\varepsilon_1, \varepsilon_2$ are set to a common value for all the nodes, the sensors and the parking slots.

1.5 Sensor node prototype

In this section the design and development of a sensor node prototype based on VSN concepts is presented. This prototype is particularly suited for urban application scenarios. In particular, the prototype is a sensor node having enough computational power to accomplish the computer vision task envisaged for urban scenarios as described in the previous section. Along with such computational power, the prototype is completed with a networking board in order to make it included within the sensor network, and for dispatching and receiving data, through a deployed event-based middleware. Finally an energy harvesting module, implemented to keep the node autonomous, is included and described.
In the following an overview of the prototype implementation is given; starting with the description of the architecture of the implemented prototype, followed by the features of the single hardware components, namely: the

vision board, the networking board, characteristics of the acquisition sensor and the energy harvesting module. Thereafter the layout of designed and implemented board will be presented.
For the design of the prototype an important issue to follow has been the use of *low cost* technologies. In particular, the node is using sensors and electronic components at low cost, so that once engineered, the device can be manufactured at low cost in large quantities. In the design and planning of the architectural side, an important issue is represented by the ease of installation of the device, thus the protective shield that has been considered for the sensor node is compact but able to accommodate all components of the device.
Going into detail, the single sensor node can be divided into two main parts: the *vision board* equipped with the camera sensor and the logics for image analysis and the *networking board* connected to the wireless communication module (RF Transceiver).
They have respectively the following tasks: i) acquire and process images and ii) control the device to coordinate the processes of transmission of all the information extracted about the scene.
Other components of the sensor node are given by the power supply system that controls charging and permits to choose optimal energy savings policies. The power supply system includes the battery pack and a module for harvesting energy, e.g. through photovoltaic panel. Figure 1.3 shows the last design of the sensor node architecture.

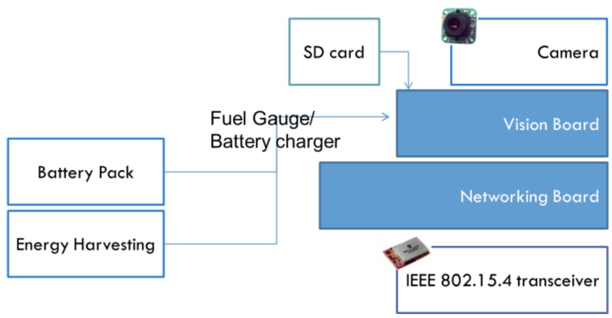

Figure 1.3 Architecture of the sensor node.

1.5.1 The vision board

For the realization of the vision board, an embedded Linux architecture has been selected in the design stage for providing enough computational power and ease of programming. A selection of ready-made Linux based prototyping boards had been evaluated with respect to computing power, flexibility / expandability, price/performance ratio and support. For example, the following candidates were considered Raspberry Pi Model B (ARM11, 700 MHz) [29], Phidget SBC (ARM9, 400 MHz) [30] and BeagleBone – TI Sitara AM3359 (Cortex A8, 720 MHz) [31].
All these candidates have as common disadvantages high power consumption and the presence of electronic parts which are not useful for the tasks foreseen here.
It has been therefore decided to design and realize a custom vision component by designing, printing and producing a new PCB. The new PCB was designed in order to have the maximum flexibility of use while maximizing the performance / consumption ratio. A good compromise has been achieved by using a *Freescale* CPU based on the ARM architecture, with support for MMU -like operating systems GNU / Linux.
This architecture has the advantage of integrating a Power Management Unit (PMU), in addition to numerous peripherals interface, thus minimizing the complexity of the board. Also the CPU package of type TQFP128 helped us to minimize the layout complexity, since it was not necessary to use multilayer PCB technologies for routing. Thus, the board can be printed also in a small number of instances. The choice has contributed to the further benefit of reducing development costs, in fact, the CPU only needs an external SDRAM, a 24MHz quartz oscillator and an inductance for the PMU.
It has an average consumption, measured at the highest speed (454MHz), of less than 500mW.
The system includes an on-board step-down voltage regulator type LM2576 featuring high efficiency to ensure a range of voltages between 6 and 25V, making it ideal for battery-powered systems, in particular for power supply by lithium batteries (7.2 V packs) and lead acid batteries (6V, 12V, 24V packs).
The board has several communication interfaces including RS232 serial port for communication with the networking board, SPI, I2C and USB.

Thanks to the GNU / Linux operating system, software development is partially relieved, relying on libraries already available for the interface to devices connected to the board. For example, it is not necessary to know the characteristics of a particular HW camera, but it is enough that it is compatible with the standard USB Video Class (UVC); through the UVC API, it is then possible to configure all the parameters available.

1.5.2 The networking board

For the realization of the networking board, it has been decided to use a microcontroller-based device with a 32bit architecture. For radio communication, a transceiver compliant with IEEE 802.15.4 has been required in line with modern approaches to IoT. For what regards the software, it has been decided to adopt Contiki [32] as operating system. Contiki provides the uIPv6 stack, which deals with IPv6 networking. The IPv6 stack also contains the 6LoWPAN header compression and adaptation layer for IEEE 802.15.4 links. Therefore the operating system is well capable of supporting an event based middleware for VSN. An analysis of the boards available on the market has shown that there exist devices satisfying all the above requirements. In particular, the Evidence SEED-EYE board [33] has been selected, which is particularly suited for implementing low-cost multimedia WSN.

1.5.3 The sensor

For the integration of a camera sensor on the vision board, some specific requirements were defined in the design stage for providing easiness of connection and to the board itself and management through it, and capability to have at least a minimal performance in difficult visibility condition, i.e. night vision. Thus the minimal constraints were to be compliant with USB Video Class device (UVC) and the possibility to remove IR filter or capability of Near-IR acquisition. Moreover, the selection of a low cost device was an implicit requirement considered for the whole sensor node prototype. Among a very large list of UVC compliant devices [34], an easy-to-buy and cheap camera was selected (TRUST SpotLight Webcam [35]). Moreover the camera is equipped with an IR filter, designed to reduce the noise from IR light sources, which is easily removable for our purposes of acquiring images even in low light conditions.

1.5.4 Energy harvesting and housing

The previously described boards and camera are housed into an IP66 shield. Another important component of the node is the power supply and energy harvesting system that controls charging and permits to choose optimal energy savings policies. The power supply system includes the lead (Pb) acid battery pack and a module for harvesting energy through photovoltaic panel.
In Figure 1.4, the general setup of a single node with the electric connections for the involved components is shown. Notice that, in order to implement energy savings policies, the vision board has also been used to measure the charging status of the batteries. To this end, an ADC Conditioning module has been used to adapt the voltage level of the power supply system to the voltage range of the vision board ADC input.

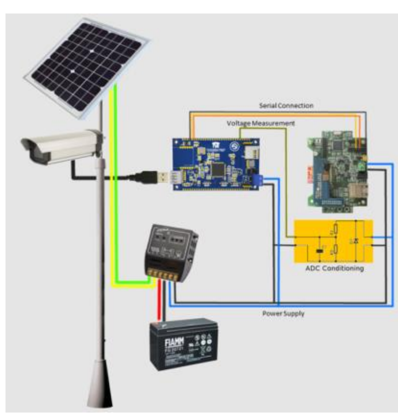

Figure 1.4 General setup of the VSN sensor node prototype with energy harvesting system.

1.5.5 The board layout

After having introduced the main features of the selected hardware, this section presents the layout of the vision board. Indeed the creation of a vision board having the basic features as previously described has required the design of a schematic in which to allocate and organize all modules and components required for its operation. As a result, in Figure 1.5 the layout of the vision board is shown, respectively through a 3D rendering of the board and a printed sample of the board.

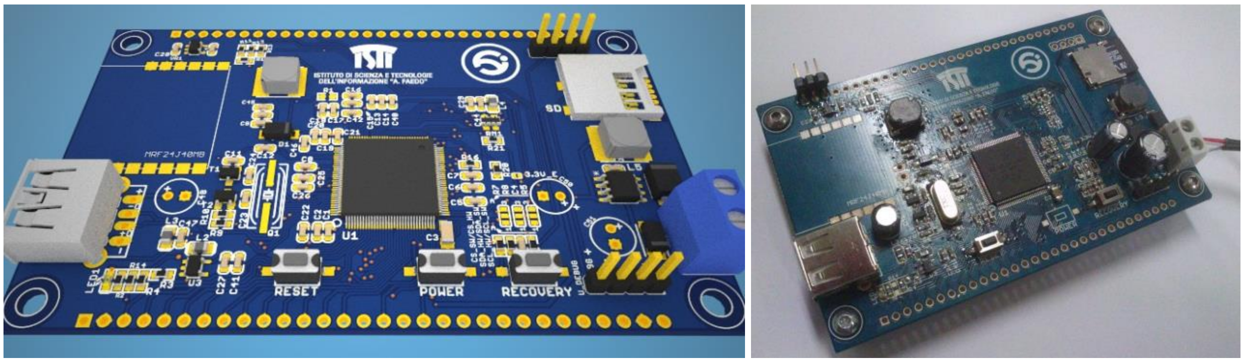

Figure 1.5 The 3D rendering of the vision board and a printed sample of the board.

1.6 Application scenarios and experimental results

Nowadays most of the available sensors for traffic monitoring are usually focused on structured environments and are based on tethered sensors. Moreover, their cost usually prevents their massive use for covering large areas, since the ratio cost/benefit is not favourable. The proposed and developed embedded system and low-cost camera sensor, make possible to conceive sensor-based pervasive intelligent systems centred on image data. Following this concept the application scenarios identified for an urban area case study are capable of covering several different situations that typically occur in this kind of area: restricted traffic area, ecological area, intermodal node services, intermodal parking, car sharing services, electric vehicle charging stations. In particular, different scenarios for the evaluation of both parking lots and traffic flow have been set-up. For the parking lot scenario the set-up consists in a set of VSN nodes equipped with cameras having partially overlapping field of views. The goal was to observe and estimate the availability and location of parking spaces. A basic assumption was made on the geometry of the parking: each camera knows the positions of the parking slots under its monitoring. In addition, we assume that a coordinator node knows the full geometry of the parking lot as well as the calibration parameters of the involved cameras, in order to properly aggregate their outcomes.

For the traffic flow, the set-up consists in a smaller set of VSN nodes, which are in charge of observing and estimating dynamic real-time traffic related information, in particular regarding traffic flow and the number and direction of the vehicles, as well as giving a rough estimate about the average speed of the cars in the traffic flow.

Regarding the experimentation results, first data from the parking lot are presented and analysed, in Figure 1.6 an example of parking lot image is reported and then in Figure 1.7 the differences between the sensor values and the ground-truth recorded for a sample parking slot acquisition is presented, showing the good separation obtained between different events.

Regarding the traffic flow monitoring, two versions of the algorithm were implemented on the VSN. In the first, the solutions used three different frames (using frame differencing) to obtain a binary representation of the moving objects in the reference frame. Analysing the connected components, blobs are detected, and then it is verified whether these can be referred to objects moving through (i.e. traffic flow) a predefined RoI [25]. The final was designed to eliminate the analysis of connected components, using the algorithm presented in Section 1.4.

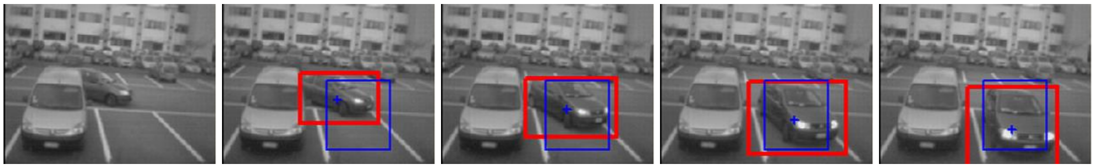

Figure 1.6 Car detection and analysis of parking lot occupancy status.

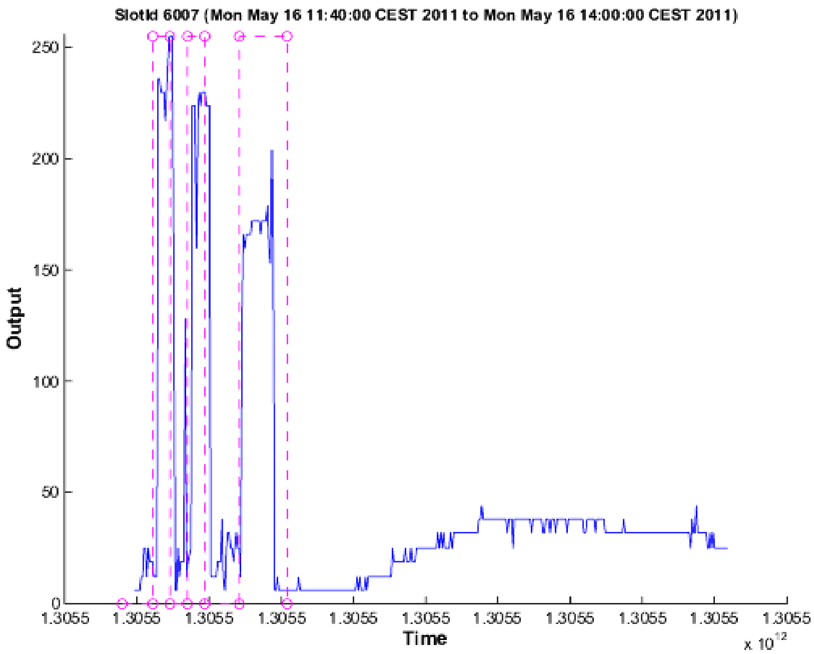

Figure 1.7 Data collected for a parking slot on the case study site. Sensor confidence values are shown in blue, while the ground-truth recorded is shown in red with circles representing change-event.

In the following Figure 1.8 a sample of the acquired and processed images for traffic flow analysis is reported and then, in Table 1.1 the results of the traffic flow case study are reported, showing the improvement in performance between the preliminary version and the final implemented solution.

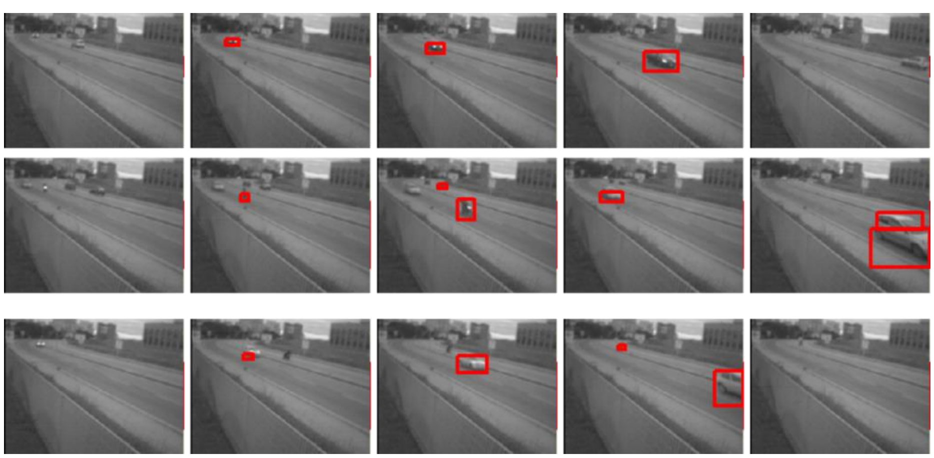

Figure 1.8 Detection of vehicles on an urban road for traffic flow analysis.

Table 1.1 Traffic flow performance comparison between preliminary (V1) and final (V2) versions.

	Sequence	Hit	Miss	False Positive	Total real events	Sensitivity rate	False positive rate
V1	S1	204	24	9	228	89.0%	4.0%
	S2	234	2	10	236	99.2%	4.2%
	TOTAL	**438**	**26**	**19**	**464**	**94.3%**	**4.1%**
V2	S1	226	2	3	228	99.1%	1.3%
	S2	234	2	2	236	99.2%	0.8%
	TOTAL	**460**	**4**	**5**	**464**	**99.1%**	**1.1%**

1.7 Conclusions

In this chapter a scalable technological solution has been introduced for supporting ITS-related problems in an urban scenario. The survey mainly addressed embedded solutions for the realization of smart cameras that can be used to detect, understand and analyse traffic-related situation and events thanks to the integration of on-board vision logics. Such embedded solutions can be elements of a broader VSN with each node equipped with an image sensing device. Several issues suggest moving from a centralized solution to a distributed intelligence one, from processing speed to available bandwidth for transmission, and from capabilities of redundancy in a distributed solution to the autonomy granted to each distributed node. In such nodes, artificial intelligence and computer vision algorithms are able to provide autonomy and adaptation both to internal as well as to external conditions.
The smarter part is that these nodes are not just gatherers of information from the cameras, but they can extract significant and compact descriptors of the scene from the data acquired in the video stream. We then show that currently available techniques are not suitable to be used in those networks of vision due to the complexity of the algorithms and to the excessive hardware requirements; thus ad hoc algorithms needed to be designed and implemented.
After introducing the developed embedded vision node, the implementation of computer vision algorithms for smart cameras (in a VSN) has been addressed. In particular, the two sample ITS applications, for the analysis of traffic flow status and parking lot monitoring, were described. Lately the designed and realized smart camera prototype was presented and the envisaged application scenarios described with the promising results obtained.

References

1. N. Buch, S.A. Velastin and J. Orwell: A review of computer vision techniques for the analysis of urban traffic. IEEE Transactions on Intelligent Transportation system, Vol. 12, N°3, pp. 920-939, 2011.
2. P. Remagnino, A. I. Shihab, and G. A. Jones: Distributed intelligence for multi-camera visual surveillance. Pattern Recognition, 37(4):675–689, 2004
3. J. Lopes, J. Bento, E. Huang, C. Antoniou and M. Ben-Akiv: Traffic and mobility data collection for real-time applications, Proc. IEEE Conf. ITSC, pp.216 -223 2010
4. H. Guo, Z. Wang, B. Yu, H. Zhao, X. Yuan: TripVista: Triple Perspective Visual Trajectory Analytics and Its Application on Microscopic Traffic Data at a Road Intersection. Pacific Visualization Symposium (PacificVis), 2011 IEEE: 163-170
5. Z. Wang; Min Lu; X. Yuan; J. Zhang; Van De Wetering, H.: Visual Traffic Jam Analysis Based on Trajectory Data, Visualization and Computer Graphics, IEEE Transactions on, On page(s): 2159 - 2168 Volume: 19, Issue: 12, Dec. 2013
6. N. Buch; J. Orwel ; S.A. Velastin: Urban road user detection and classification using 3D wire frame models IET Computer Vision, Volume 4, Issue 2, June 2010, p. 105 – 116
7. J. Candamo , M. Shreve , D. B. Goldgof , D. B. Sapper and R. Kastur: Understanding transit scenes: A survey on human behavior-recognition algorithms, IEEE Trans. ITS, vol. 11, no. 1, pp.206 -224 2010
8. Ipsotek. Available at: http://www.ipsotek.com/ (Last retrieved April 28, 2014)

9. Xerox Technology Showcase at 21st ITS World Congress, Detroit, 2014 http://itsworldcongress.org/events/xerox-vehicle-passenger-detection-system/
10. Digital Recognition. Available at: http://www.digital-recognition.com/ (Last retrieved April 28, 2014)
11. Traficam. Available at: http://www.traficam.com/ (Last retrieved April 28, 2014)
12. Deepa Kundur, Ching-Yung Lin, and Chun-Shien Lu: Visual sensor networks. EURASIP Journal on Advances in Signal Processing Signal Processing, 2007:Article ID 21515, 3 pages, 2007.
13. W. Feng, B. Code, E. C. Kaiser, M., W.C. Feng, and L. Bavoil: Panoptes: scalable low-power video sensor networking technologies. In ACM Multimedia, pages 562–571, 2003
14. M. H. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcıa, J. Warrior, D. Estrin, and M. B. Srivastava: Cyclops: in situ image sensing and interpretation in wireless sensor networks. In SenSys, pages 192–204, 2005.
15. S. Hengstler et al.: Mesheye: a hybrid-resolution smart camera mote. In Proc. 6th Int. Conf. Inf. proc. in sensor networks, pages 360–369. ACM, 2007.
16. CMUcam. Available at: http://www.cmucam.org/ (Last retrieved April 28, 2014)
17. P. Chen, P et al.: Citric: A low-bandwidth wireless camera network platform. In Distributed Smart Cameras, pages 1–10, September 2008.
18. R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam: Image change detection algorithms: a systematic survey. IEEE Transactions on Image Processing, 14(3):294–307, 2005.
19. T. Yan, D. Ganesan, and R. Manmatha: Distributed image search in camera sensor networks. SenSys, pages 155–168. ACM, 2008.
20. Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoints, IJCV, 60, 2, pp. 91-110, 2004
21. P. A. Viola and M. J. Jones: Robust real-time face detection. IJCV, 57(2):137–154, 2004.
22. A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain: Content-based image retrieval, the end of the early years, IEEE Trans on PAMI 2000.
23. C. Cortes and V. Vapnik: Support-Vector Networks, Machine Learning, 20, 1995.
24. D. Ayers, R. Chellappa: Scenario recognition from video using a hierarchy of dynamic belief networks, In Proc. Of 15th Int. Conf. on Pattern Recognition, 835-838, 2000.
25. Magrini M., Moroni D., Pieri G., Salvetti O: Real time image analysis for infomobility. Lecture Notes in Computer Science, vol. 7252, 207 – 218, 2012.
26. Magrini M., Moroni D., Nastasi C., Pagano P., Petracca M., Pieri G., Salvadori C., Salvetti O.: Visual sensor networks for infomobility. In: Pattern Recognition and Image Analysis, vol 21. Springer, 2011.
27. C. Stauffer and W.E. Grimson: Adaptive background mixture models for real-time tracking, Proceedings of the Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, USA: 1999, pp. 2: 246-252
28. K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis: Background modeling and subtraction by codebook construction, IEEE International Conference on Image Processing (ICIP), 2004, pp. 2-5
29. Raspberry Pi. Available at: http://www.raspberrypi.org/ (Last retrieved April 28, 2014)
30. Phidgets board. Available at: http://www.phidgets.com/ (Last retrieved April 28, 2014)
31. Beagleboard. Available at: http://beagleboard.org/ (Last retrieved April 28, 2014)
32. Contiki OS. Available at: http://www.contiki-os.org/ (Last retrieved April 28, 2014)
33. SEED-EYE Board. Available at: http://www.evidence.eu.com/it/products/seed-eye.html (Last retrieved April 28, 2014)
34. UVC devices. Available at: http://linux-uvc.berlios.de/#devices (Last retrieved April 28, 2014)
35. Trust Spotlight. Available at :http://www.trust.com/it-it/all-products/16429-spotlight-webcam (Last retrieved April 28, 2014)