Computer Vision on Embedded Sensors for Traffic Flow Monitoring

Massimo Magrini, Davide Moroni, Giovanni Palazzese, Gabriele Pieri, Giuseppe Riccardo Leone, and Ovidio Salvetti
Institute of Information Science and Technologies "A. Faedo" – ISTI
National Research Council of Italy, CNR
Pisa, Italy
{massimo.magrini, davide.moroni, giovanni.palazzese, gabriele.pieri, g.leone, ovidio.salvetti}@isti.cnr.it

Abstract—**Capillary monitoring of traffic in urban environment is key to a more sustainable mobility in smart cities. In this context, the use of low cost technologies is mandatory to avoid scalability issues that would prevent the adoption of monitoring solutions at the full city scale. In this paper, we introduce a low power and low cost sensor equipped with embedded vision logics that can be used for building Smart Camera Networks (SCN) for applications in Intelligent Transportation System (ITS); in particular, we describe an ad hoc computer vision algorithm for estimation of traffic flow and discuss the findings obtained through an actual field test.**

Keywords—Real-Time Imaging, Embedded Systems, Intelligent Transport Systems (ITS).

I. INTRODUCTION

Thanks to computer vision techniques, fully automatic video and image analysis from traffic monitoring cameras is a fast-emerging field based 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 scenario for video analytics in both 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 [2].

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.

Based on these considerations, the aim of this paper is to introduce a scalable technology for supporting ITS applications in urban scenarios; in particular, we propose an embedded solution for the realization of a smart camera that can be used to detect, understand and analyze traffic-related situation and events making use of 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. 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, which poses several scalability concerns. As argued in [8], the SCN approach offers several advantages besides scalability, such as less stringent bandwidth requirements and the possibility to add autonomy features to the nodes. Indeed, on-board 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). It can be stated that in a SCN the nodes are not merely collectors of information from the sensors, but they have to distill brief, high-significant descriptors of the scene from the bulky 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 SCN, due to the high computational complexity of algorithms or to excessively demanding memory requirements. Therefore, ad hoc algorithms should be designed for SCN, as we will explore in the next sections. In particular, after describing the possible role of SCN in urban scenarios, we present in Section III a sample application, namely the estimation of vehicular flows on a road, proposing a lightweight method suitable for embedded systems. Then, we introduce the sensor prototype we designed and developed in Section IV. In Section V, we report the experimental results gathered during a test field and we finally conclude the paper in Section VI.

II. SCN IN URBAN SCENARIOS

According to [2], 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. However, the push towards this direction is motivated by the increasing interest in providing more sustainable and efficient models to cities. In this context, 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 [6]. 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, further monitoring objectives can be supported (at least in principle) by the application of computer vision and pattern recognition techniques. For example these include the detection of complex traffic violations (e.g. illegal turns, one-way streets, restricted lanes) ([4], [11]), identification of road users (e.g. vehicles, motorbikes and pedestrians) [1] and of their interactions understood as spatiotemporal relationships between people and vehicle or vehicle-to-vehicle [3]. 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 is possible to target using video-surveillance based systems, namely: i) safety and security, ii) law enforcement, iii) billing, and iv) traffic monitoring and management. In this paper, we just consider the last item, which that does not require safety critical features and, thus, can be addressed using non-certified sensors. Indeed, traffic monitoring and management is related to extraction from urban scenes of informative content 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) or 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 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 a growing interest in video-based counting system also based on embedded devices. Some solutions, such as [10], 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.

Finally, we might argue that pervasive technologies based on vision turn out to be of interest when i) there is some complex 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 SCN 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 simply by 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.

III. TRAFFIC IMAGE PROCESSING

In this Section, a sample ITS applications based on computer vision over SCN is reported. It regards the estimation of vehicular flows and is based on a lightweight computer vision pipeline that differs from the conventional one used on standard architectures.

More precisely, 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 transited 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 SCN, instead, it is convenient to adopt a lightweight approach; in particular, only data in predefined

Region of Interests (RoIs) are processed, to detect the presence of a vehicle. Based on these detections, then, flow information is derived without making explicit use of classical tracking algorithms.

A. Background Subtraction

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 only 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.

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 before 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 [9]. Indeed, in those cases pixels clearly exhibit a multimodal distribution. However, GMM are computationally more intensive than a single Gaussian. Codebooks [5] are another adaptive background modelling techniques presenting computational advantages for real-time background modelling with respect to 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.

B. Transit Detection

The transit detection 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 a vehicle occupies the RoI R_k or not. The decision is based on the ratio of pixels changed with respect to the total number of pixels in R_k, i.e.:

$$a_k(t)=\#(\text{changed pixels in } R_k)/\ \#(\text{pixels in } R_k) \qquad (1)$$

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_0 and R_1, lying on the same lane but translated by a distance Δ, to estimate the vehicle speed (Fig. 1). If there is a delay of δ frames, the vehicle speed can be estimated as $v=(\Delta*\nu)/\delta$, where ν is the frame rate. The vehicle length can in turn be estimated as $l=v*k/\nu$.

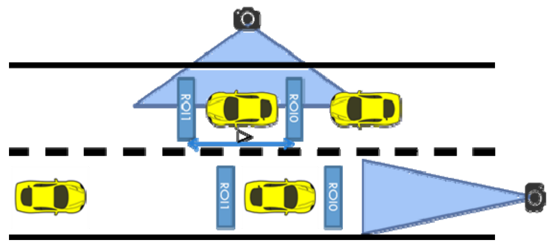

Fig. 1. RoI configuration for traffic flow analysis.

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 analyzing the level of service of the road, early identifying possible congestions.

IV. Sensor Prototype

In this section the design and development of a sensor node prototype based on SCN concepts is presented. This prototype is particularly suited for urban application scenarios [7]. 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. For the design of the prototype, an important issue to follow has been the use of low cost technologies. 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. The single sensor node has a main board that manages both the vision tasks and the networking tasks thanks to an integrated wireless communication module (RF Transceiver). 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 (Fig. 2).

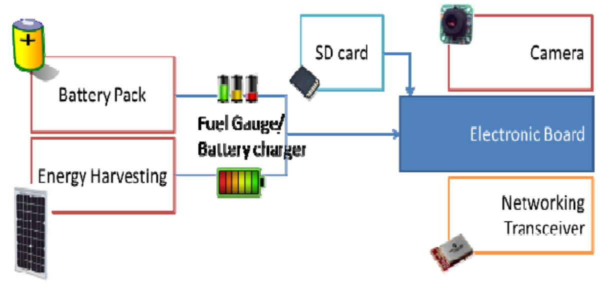

Fig. 2. Architecture of the sensor node.

A. The Main 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 has been evaluated with respect to computing power, flexibility/expandability, price/performance ratio and support. They were all found to have as common disadvantages high power consumption and the presence of electronic parts that are not useful for the tasks of a smart camera node.

It has been therefore decided to design and realize a custom vision component by designing, printing and producing a new Printed Circuit Board (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. In addition, the CPU package of type TQFP128 helped us 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 board has several communication interfaces including RS232 serial port for communication with the networking board, SPI, I2C and USB. For radio communication, a transceiver compliant with IEEE 802.15.4 has been integrated in line with modern approaches to IoT. A suitable glue has been used to integrate the transceiver with the IPv6 stack, also containing the 6LoWPAN header compression and adaptation layer for IEEE 802.15.4 links. Therefore, the operating system is well capable of supporting ETSI M2M communications over the SCN.

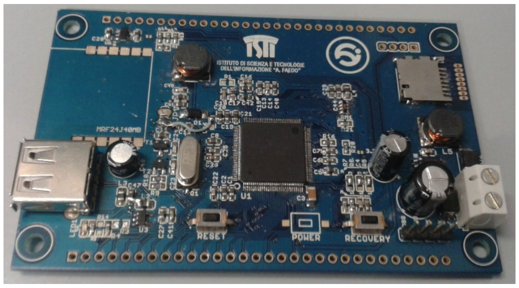

Fig. 3. The PCB showing its design and main features

B. Sensor, Energy Harvesting and Housing

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.

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 Fig. 4, the general setup of a single node for urban monitoring with the electric connections for the involved components is shown.

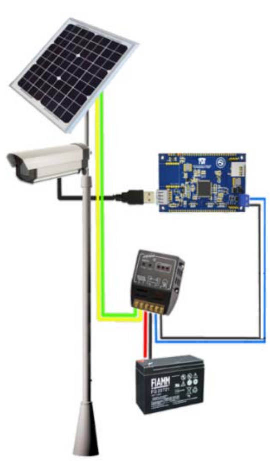

Fig. 4. General setup of a single node.

V. Experimental Results

For the traffic flow, the set-up consists in a small set of SCN 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.

Two versions of the algorithm were implemented. In the first, the solutions uses frame differencing as a background subtraction method, obtaining a binary representation of the moving objects in the RoI frame. In the second, an adaptive background modelling based on Gaussian distribution has been employed using a weighted mixture of previous backgrounds. This means that previous backgrounds are used with a heavier weight in case of no-event occurring (i.e. no transit of car), while they are used with light or no-weight in case there is an event of transit occurring.

Test sequences have been acquired under real traffic conditions and then used for testing both algorithms. The test sequences have been acquired mounting the sensor nodes on poles along urban traffic roads. Sequences were acquired in from October to December 2014 and then processed offline on the sensors. The ground-truth total for these sequences was the following:

- 124 vehicles transited,

having the following length estimation subdivision:

- 11 with length between 0 and 2 meters
- 98 (between 2 and 5 meters)
- 15 (5 and more meters)

Moreover, the algorithms yield a speed class estimate, but for this type of data there is no ground truth available. The total classification results are shown in the following Table I:

TABLE I. Resume of Classification Results

	Ground truth	Algorithm 1 Frame diff.	Algorithm 2 Adaptive
Total transited vehicles	124	140	121
Correctly identified vehicles		124 **(100%)**	118 **(95.2%)**
False positives		16 **(12.9%)**	3 **(2.4%)**

The first algorithm based on frame differencing has a significant number of false positives but it reaches a 100% identification rate, while the second adaptive algorithm has an acceptable rate of identification with a very low false positive rate. As a further step, in the following Tables II and III the classification estimates for the speeds and lengths classes for each of the implemented algorithms are shown. For a correct evaluation of these tables, it has to be taken into account the fact that length estimates were made roughly by an observer by sight, while there is no estimate at all on the ground truth regarding the speeds. Furthermore, for the first algorithm all the false positive were detected in the class having length in meters more than *5* with fastest speed, and have been identified as bugs related to the camera and its automatic setting of balance and contrast. All these issues and deeper analysis are under studying and will provide more detailed results.

Besides the offline processing of pre-acquired sequences, the sensor nodes have been tested on the field as it will be described elsewhere.

TABLE II. ALGORITHM 1 RESULTS

Algorithm 1 Frame Diff.	***Speed <20 Km/h***	***Speed between 20÷35 Km/h***	***Speed >35 Km/h***	***TOT***
L. 0÷2 m	10	8	2	**20**
L. 2÷5 m	29	27	8	**64**
L. 5+ m	0	10	46	**56**
TOT	**39**	**45**	**56**	**140**

TABLE III. ALGORITHM 2 RESULTS

Algorithm 2 Adaptive	***Speed <20 Km/h***	***Speed between 20÷35 Km/h***	***Speed >35 Km/h***	***TOT***
L. 0÷2 m	25	1	1	**27**
L. 2÷5 m	27	35	3	**65**
L. 5+ m	8	15	6	**29**
TOT	**60**	**51**	**10**	**121**

VI. CONCLUSIONS

In this paper, we have presented technologies based on computer vision for supporting urban mobility, envisaging a number of applications of interest. Then, as a sample, we introduced a specially-designed lightweight pipeline for traffic flow analysis that is suitable for embedded system with constrained memory and computational power. Such method has been tested on a prototype sensor (designed and developed at the Signals and Images Lab, ISTI, CNR) whose main features are reported in this paper. The sensor, being low cost and equipped with a wireless transceiver, is a very good candidate for becoming the key ingredient of a scalable and pervasive smart camera network for the urban environment. Its good functionalities are proved by the set of experimental results that were collected on the field in realistic conditions. In the future, besides refining the procedure for vehicle characterization in term of speed and size, a more in-depth analysis of computational constraint will be performed, comparing the proposed pipeline with other more advanced methods. In addition, we plan to extend the class of vision logics to address further applications to mobility.

ACKNOWLEDGMENT

This work has been partially supported by EU FP7 “ICSI” – Intelligent Cooperative Sensing for Improved traffic efficiency and EU CIP “MobiWallet” – Mobility and Transport Digital Wallet.

REFERENCES

[1] N. Buch, J. Orwell, and S. A. Velastin, “Urban road user detection and classification using 3D wire frame models,” IET Computer Vision, vol. 4 (2), pp. 105-116, 2010.

[2] N. Buch, S. A. Velastin, and J. Orwell, “A review of computer vision techniques for the analysis of urban traffic,” IEEE Trans. ITS, vol. 12 (3), pp. 920-939, 2011.

[3] J. Candamo, M. Shreve, D. B. Goldgof, D. B. Sapper, and R. Kasturi, “Understanding Transit Scenes: A survey on human behavior-recognition algorithms,” IEEE Trans. ITS, vol. 11 (1), pp. 206 -224, 2010.

[4] H Guo, Z Wang, B Yu, H Zhao, and X Yuan, “TripVista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection,” PacificVis, IEEE Conference, pp. 163-170, 2011.

[5] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, “Background modeling and subtraction by codebook construction,” IEEE ICIP2004, pp. 2-5, 2004.

[6] J. Lopes, J. Bento, E. Huang, C. Antoniou, and M. Ben-Akiv, “Traffic and mobility data collection for real-time applications,” IEEE Conf. ITSC, pp. 216-223, 2010.

[7] M. Magrini, D. Moroni, G. Pieri, and O. Salvetti, “Real time image analysis for infomobility,” Lecture Notes in Computer Science, vol. 7252, pp. 207 – 218, 2012.

[8] P. Remagnino, A. I. Shihab, and G. A. Jones, “Distributed intelligence for multi-camera visual surveillance,” Pattern Recognition, vol. 37 (4), pp. 675–689, 2004.

[9] C. Stauffer, and W. E. Grimson, “Adaptive background mixture models for real-time tracking,” Proc. CVPR, Fort Collins, CO, USA, vol. 2, pp. 246-252, 1999.

[10] Traficam, 2014. Available at: http://www.traficam.com/ (Last retrieved October 28, 2014).

[11] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. Van De Wetering, “Visual Traffic Jam Analysis Based on Trajectory Data,” IEEE Transactions on Visualization and Computer Graphics, Vol.: 19, Issue: 12, pp. 2159-2168, 2013.