An Efficient Multiple Object Detection and Tracking Framework for Automatic Counting and Video Surveillance Applications

Carlos R. del-Blanco, Fernando Jaureguizar, and Narciso García

Abstract — *Automatic visual object counting and video surveillance have important applications for home and business environments, such as security and management of access points. However, in order to obtain a satisfactory performance these technologies need professional and expensive hardware, complex installations and setups, and the supervision of qualified workers. In this paper, an efficient visual detection and tracking framework is proposed for the tasks of object counting and surveillance, which meets the requirements of the consumer electronics: off-the-shelf equipment, easy installation and configuration, and unsupervised working conditions. This is accomplished by a novel Bayesian tracking model that can manage multimodal distributions without explicitly computing the association between tracked objects and detections. In addition, it is robust to erroneous, distorted and missing detections. The proposed algorithm is compared with a recent work, also focused on consumer electronics, proving its superior performance*[1].

Index Terms — Moving object detection, multiple object tracking, object counting, video surveillance applications, particle filtering, IP cameras, real-time applications.

I. INTRODUCTION

Automatic visual detection, tracking, and counting of a variable number of objects are crucial tasks for a wide range of home, business, and industrial applications, such as security, surveillance, management of access points, urban planning, traffic control, etc. However, these applications have not still played an important part in consumer electronics. The main reason is that they need strong requirements to achieve satisfactory working conditions: specialized and expensive hardware, complex installations and setup procedures, and supervision of qualified workers.

Some works have focused on developing automatic detection and tracking algorithms that minimizes the necessity of supervision. They typically use moving object detectors based on background subtraction techniques [1] because they usually do not need a training stage, nor complex system parameter settings. However, they have several drawbacks that complicate the tracking stage [2]: false alarms, noisy detections, missing detections, and split and merged detections. In addition, the correspondence between detections and tracked objects is unknown. To solve these problems, different data association techniques have been proposed. In [3] the set of detections is augmented with virtual detections to represent possible split and merged events. A similar strategy is followed in [4], which uses an overlapping criterion to simplify the generation of virtual detections. In [5], a probabilistic model for simulating split and merged detections is introduced, which uses a Markov Chain Monte Carlo method (MCMC) to compute association hypotheses in a batch process. A sequential approach, based also on MCMC sampling, is proposed in [6] for a fixed number of objects, and a similar strategy is described in [7] for a variable number of objects. The main problem of the previous approaches is that they have a high computational cost, and therefore require specialized and expensive hardware to work in real-time.

Other algorithms [8],[9] use prior information about the geometry of the scene, such as the floor position and the camera calibration to restrict the data association and tracking problems. However, this approach makes more complex the system installation and setting, since it is necessary to compute the camera calibration and estimate the 3D plane of the floor, which in turn depends on the camera location.

In this paper, an automatic visual object detection and tracking framework is proposed to reliably introduce video surveillance and counting-based applications in the consumer electronics environment. It is based on off-the-shelf equipment, such as IP, web cameras, and PCs, and does not need especial installation and configuration requirements. The detection stage is based on a parametric background subtraction technique that detects the moving regions in the input video flow. A postprocessing stage refines the detection by estimating and fitting a set of ellipses that represent the moving objects to the previous set of moving regions. The tracking stage uses a Bayesian model to simulate the object trajectories. For this purpose, a particle filtering technique is used to predict a set of hypotheses that represent the most probable object locations. These hypotheses are verified using a novel likelihood

[1] This work has been partially supported by the Ministerio de Ciencia e Innovación of the Spanish Government under the project TEC2010-20412 (Enhanced 3DTV).

Grupo de Tratamiento de Imágenes, ETSI Telecomunicación, Universidad Politécnica de Madrid, Madrid, 28040, Spain (emails: [C.R. del-Blanco] cda@gti.ssr.upm.es; [F. Jaureguizar] fjn@gti.ssr.upm.es; [N. García] narciso@gti.ssr.upm.es).

Contributed Paper
Manuscript received 04/20/12
Current version published 09/25/12
Electronic version published 09/25/12.

function that evaluates each hypothetical object configuration with the set of available detections without to explicit compute their data association. Thus, a considerable saving in computational cost is achieved. In addition, the likelihood function has been designed to account for noisy, false and missing detections. Fig. 1 shows a block diagram of the proposed algorithm.

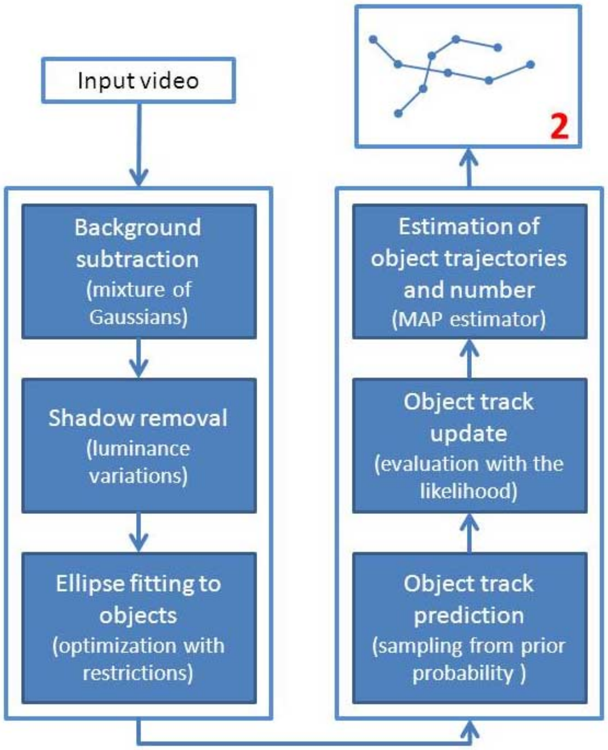
Input video
Background subtraction (mixture of Gaussians)
Shadow removal (luminance variations)
Ellipse fitting to objects (optimization with restrictions)
Object track prediction (sampling from prior probability)
Object track update (evaluation with the likelihood)
Estimation of object trajectories and number (MAP estimator)
2

Fig. 1 System block diagram.

The proposed detection and tracking algorithm has been compared with a recent approach that is also oriented to the consumer electronics.

II. Moving Object Detection

The detection of moving objects is based on an off-the-shelf background subtraction technique [10] that performs an online learning of the background of a scene. The learnt background is used to estimate the foreground, i.e. the moving objects, by detecting those image regions that are not compatible with the background model. The background is modeled pixel by pixel using a mixture of Gaussians that can even represent non static background scenes (for example moving tree leaves).

This moving object detector has also the ability to detect and remove shadows, a great source of false alarms in this kind of visual detectors.

The output of the detector is a set of independent image regions that ideally have a one-to-one correspondence with a moving object. However, likewise any other background subtraction technique, the resulting moving object detection can contain two or more regions that in fact correspond to only one object, which is called split detections. Also, it can contain regions that correspond to two or more objects, which is called merged detections. These both events, split and merged detections, make the trajectory estimation and object counting a challenging task. Fig. 2 shows the result of applying the background subtraction technique to the above image. It can be observed that there is no one-to-one correspondence between the detected moving regions and the objects presented in the image.

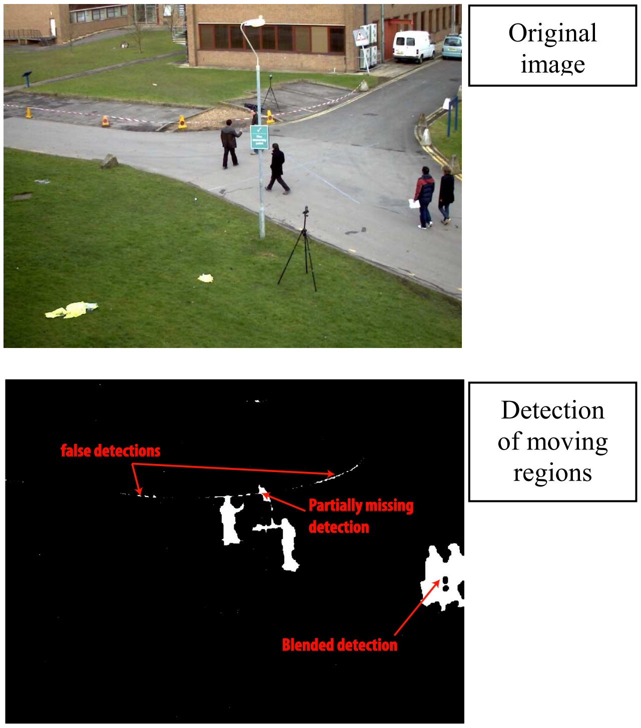

Fig. 2 Detection of moving regions. There is no one-to-one correspondence with the real objects: noisy, false, missing, split, and merged detections.

The proposed approach addresses the previous problems by fitting a set of ellipses to the output of the background subtraction algorithm, in such a way that each ellipse represents an independent object. An ellipse can contain one or several independent moving regions of the foreground detections, accounting for the split detections. On the other hand, a moving region of the foreground can be associated to two or more ellipses, which accounts for merged detections and also objects occlusions.

The process of fitting ellipses to the foreground detection is carried out by an algorithm similar to [11], which combines the Expectation-Maximization algorithm and the Levenberg-Marquardt one to estimate the number of ellipses and their parameters. However, the algorithm has been modified to achieve two goals. The first one is to limit the number of ellipses per object to one, since the original algorithm uses a hierarchical set of connected ellipses to represent object silhouettes. The second one is to assist in the estimation of the ellipse parameters restricting their values according to a predetermined range of possible object sizes and orientations. In addition, the two previous adaptations reduce the complexity of the fitting process, making the algorithm more suitable for fulfill real-time restrictions. Fig. 3 illustrates the process of ellipse fitting over the previous moving region detection.

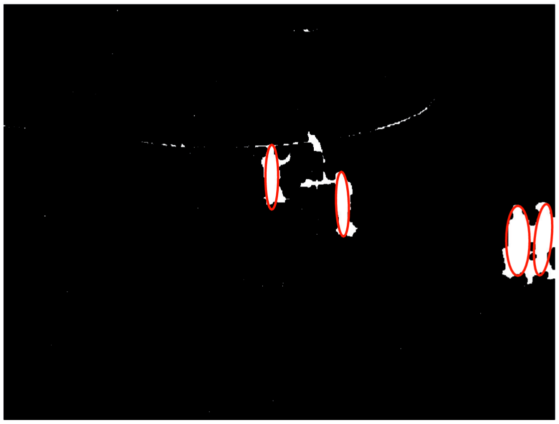

Fig. 3 Ellipse fitting to the moving regions resulting from the background subtraction.

III. Bayesian tracking model

Tracking information about the moving objects is represented using a vector state notation by

$$x_t = \left[x_{t,n} \mid n = 1, \ldots, N_o \right], \quad (1)$$

where N_o is the number of moving objects at time step t. The n^{th} component

$$x_{t,n} = \left[r, v, e \right]_{t,n} \quad (2)$$

contains the object position, the velocity, and the elliptical bounding of an object, respectively. Object position and velocity are expressed relatively to the image plane. The image region taken up by the object is approximated by an ellipse, whose geometric center is $r_{t,n}$, and the rest of parameters (major and minor axes, and orientation) is stored in $e_{t,n}$.

The vector state x_t is estimated using a Bayesian approach, which computes the posterior probability density function (pdf) of the vector state, $p\left(x_t \mid z_{1:t}\right)$, using the sequence of available measurements until the current time step

$$z_{1:t} = \left\{ z_1, \ldots, z_t \right\}. \quad (3)$$

In each time step, z_t represents the set of object measurements that have been obtained by the detection stage of moving objects.

The posterior pdf can be recursively expressed using the Bayes' theorem as [12]

$$p\left(x_t \mid z_{1:t}\right) = \frac{p\left(z_t \mid x_t\right) p\left(x_t \mid z_{1:t-1}\right)}{p\left(z_t \mid z_{1:t-1}\right)}, \quad (4)$$

where $p\left(x_t \mid z_{1:t-1}\right)$ is the prior probability, $p\left(z_t \mid x_t\right)$ is the likelihood, and $p\left(z_t \mid z_{1:t-1}\right)$ is just a normalization factor.

The prior probability expresses the spatio-temporal evolution of the moving objects according to a dynamic model. It can be considered as a prediction of the current state of the moving objects. Its expression is given by

$$p\left(x_t \mid z_{1:t-1}\right) = \int p\left(x_t \mid x_{t-1}\right) p\left(x_{t-1} \mid z_{1:t-1}\right) dx_{t-1}, \quad (5)$$

where $p\left(x_{t-1} \mid z_{1:t-1}\right)$ is the posterior probability at the previous time step (the recursive step) and $p\left(x_t \mid x_{t-1}\right)$ is the transition probability that simulates the object dynamics as a constant velocity model with Gaussian uncertainty. Its mathematical expression is

$$p\left(x_t \mid x_{t-1}\right) = N\left(x_t; Ax_{t-1}, \Sigma_x\right), \quad (6)$$

where $N\left(x; \mu, \sigma\right)$ is a multivariate Gaussian function, A is matrix that encodes the constant velocity model, and Σ_x is the covariance matrix that expressed the uncertainty of the object dynamics.

The likelihood term uses the measurements at the current time step z_t to estimate the most probable object configuration. It is like an update step that refines the prediction carried out by the prior probability. The computation of the likelihood requires the estimation of the association between objects and measurements to properly update the tracking information of each object. To efficiently accomplish this task, a novel data association is proposed based on computing the best area overlapping between objects and measurements. The main idea is that the greater the overlapping between the set of ellipses corresponding to the objects and the set of ellipses corresponding to the measurements is, the more probable underlying state vector is. The area overlapping is expressed by the intersection of the union of the areas of all the ellipses of measurements and detections

$$O_v = \bigcup_{m=1}^{N_d} I_m \cap \bigcup_{n=1}^{N_o} I_n \quad (7)$$

where N_d is the number of measurements, I_m is a binary formed by setting to 1 those pixels inside the ellipse associated to the m^{th} measurement, and I_n is a binary formed by setting to 1 those pixels inside the ellipse associated to the n^{th} object.

Using this overlapping measure, the likelihood is expressed by a Gaussian function that compares the area between the overlapping and the total area of all the measurements

$$p\left(z_t \mid x_t\right) = N\left(C_d\left(O_v\right); C_d\left(\bigcup_{m=1}^{N_d} I_m \right), \Sigma_z \right), \quad (8)$$

where C_d is the cardinal function used to count the number of pixels that compose an image area, and Σ_z is a covariance matrix that expresses the uncertainty related to the area overlapping process in the likelihood..

Once the expression of the posterior pdf has been derived, an optimal estimation of the objects paths can be obtained, since this probability contains all the required information. For this purpose, the Maximum A Posteriori (MAP) estimator has been used, which obtains the state vector with maximum probability. Unlike other estimators as the Minimum Mean Squared Error (MMSE), the MAP estimator avoids erroneous estimations resulting from multimodal distributions, as this is the case.

However, the posterior pdf $p\left(x_t \mid z_{1:t}\right)$ cannot be analytically solved due to the non-linearities of the overlapping based observation model used to define the likelihood $p\left(z_t \mid x_t\right)$ [13]. To overcome this problem an approximated solution is computed by means of a particle filtering strategy, which is describe in the next section.

IV. Particle Filtering Approximation

The posterior pdf is approximated using a particle filtering [13] strategy by means of a set of weighted samples, also called particles,

$$p\left(x_t \mid z_{1:t}\right) \approx \sum_{i=1}^{N_s} w_t^i \delta\left(x_t - x_t^i\right), \tag{9}$$

where $\delta\left(x\right)$ is the Dirac delta function, N_s is the number of samples, $\left\{x_t^i \mid i=1,\ldots,N_s\right\}$ is the set of samples, and $\left\{w_t^i \mid i=1,\ldots,N\_\mathrm{S}\right\}$ their relative weights.

The samples are drawn from a proposal distribution $q\left(x_t \mid z_{1:t}\right)$ that ideally should be proportional to the real posterior. Since this not possible (it has not analytical expression), the prior probability is used as proposal distribution (in general is a reasonable approximation)

$$\begin{aligned} & q\left(x_t \mid z_{1:t}\right) = p\left(x_t \mid z_{1:t}\right) = \\ & = \int p\left(x_t \mid x_{t-1}\right) p\left(x_{t-1} \mid z_{1:t-1}\right) dx_{t-1} \approx \\ & \approx \int p\left(x_t \mid x_{t-1}\right) \sum_{i=1}^{N_s} \delta\left(x_{t-1} \mid x_{t-1}^i\right) dx_{t-1} = , \\ & = \sum_{i=1}^{N_s} p\left(x_t \mid x_{t-1}^i\right) \end{aligned} \tag{10}$$

where the particle filtering approximation of the posterior pdf at the previous time step has been used. Thus, the recursive structure of the Bayesian model is hold. Substituting the transition probability by its expression given in (ref), the following expression for the proposal distribution is obtained

$$q\left(x_t \mid z_{1:t}\right) = \sum_{i=1}^{N_s} N\left(x_t; A x_{t-1}^i, \Sigma_x\right) \tag{11}$$

The weights are then computed to rectify the approximation made in the previous sampling generation stage. The mathematical expression of the non-normalized weights is given by

$$\begin{aligned} \tilde{w}_t^i & \propto \frac{p\left(x_t \mid z_{1:t}\right)}{q\left(x_t \mid z_{1:t}\right)} \propto \frac{p\left(z_t \mid x_t\right) p\left(x_t \mid z_{1:t-1}\right)}{q\left(x_t \mid z_{1:t}\right)} , \\ & = p\left(z_t \mid x_t\right) \end{aligned} \tag{12}$$

resulting in that the likelihood is used to weigh the samples. The substitution of the likelihood by its expression (ref equ) gives

$$\tilde{w}_t^i \propto N\left(C_d\left(O_v\right); C_d\left(\bigcup_{m=1}^{N_d} I_m\right), \Sigma_z\right). \tag{13}$$

The computed weights are then normalized

$$w_t^i = \frac{\tilde{w}_t^i}{\sum_{i=1}^{N_s} \tilde{w}_t^i}. \tag{14}$$

Finally, to avoid the degeneracy problem, consisting in that after a few iterations all the samples except one have a negligible value, a resampling stage is performed by means of the Sampling Importance Resampling (SIR) algorithm.

V. Inputs and outputs of objects

The size of the state vector x_t can change along the time due to the input and output of objects in the scene. The entrance of new objects is associated to measurements that do not overlap with any existing tracked objects, although these measurements can be false alarms as well. This ambiguity is modeled by a Binomial distribution that estimates the probability that N_{in} new objects have entered in the scene given the set of measurements that do not overlap with any object. The parameter of the Binomial distribution is set according to the expected number of false alarms that is usually a system parameter depending on the scene characteristics and the object detector.

The estimation of objects that leave the scene is related to the number of objects that do not overlap with any existing measurements. However, the object detector could simply have failed in detecting those objects. A Gamma distribution is used to model this ambiguity. It simulates how many time steps are required to determine that one object has left the scene, considering that on average this quantity is N_t time steps. The value N_t is a system parameter that depends on the scene configuration and performance of the object detector.

VI. Results

The proposed visual detection and tracking framework for moving objects has been tested using sequences of the datasets PETS2006 and PETS2010. The sequences of both datasets have been acquired by several cameras with different points of view, and show situations with a varying number of people. The sequences belonging to PETS2006 correspond with indoor situations, whereas those belonging to PETS2010 with outdoor situations.

The first stage of experiments has evaluated the performance of the presented framework to count people. And, a second stage has carried out a comparison of the tracking results obtained by the presented framework and the approach presented in [14]. This work is also focused on automatic detection and tracking of moving objects, but with substantial differences. The most relevant ones are: the moving object detector does not include a stage for shadow removal, nor a postprocessing stage to fit ellipses (representing real objects) to the obtained moving regions; the tracking is based on a deterministic approach that reaches the closer mode, not being capable to manage different hypotheses corresponding to different modes; and there is no explicit mechanism to deal with false and missing detections.

Fig. 4 shows how the counting errors are distributed along the time for the first 1000 frames of the sequence "S1-T1-C", camera number 3, belonging to the PETS2006 dataset. The number of errors remains quite low along the sequence. In addition, an important percentage of them are only temporal, corresponding to the entrance of new objects or the exit of existing ones. The reason is that the tracking filter needs some time to converge for these transient situations. Anyway, for the purpose of counting the total number of objects in a sequence, these errors do not affect.

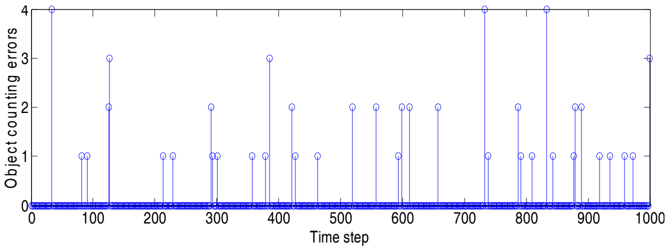

Fig. 4 Number of counting errors per time step along the first 1000 frames of the sequence S1-T1-C, camera 3, PETS 2006.

Global object counting results are provided in TABLE I using several sequences of the dataset PETS2006. The results are expressed as the percentage of counting errors in the whole sequence. As it can be observed, the presented algorithm has a great accuracy, despite the relatively simplicity of the used object detector in comparison with others available in the literature.

TABLE I
Object counting performance

Dataset	Sequence name	Camera	Sequence length	% of counting errors
PETS 2006	S1-T1-C	3	3021	5.2
	S1-T1-C	4	3021	5.7
	S1-T3-C	3	2551	5.3
	S1-T3-C	4	2551	6.5
	S1-T5-A	3	3051	5.9
	S1-T5-A	4	3051	7.3

A comparison of the tracking performance between the presented framework and the one described in [14] is provided in TABLE II. The tracking performance has been measured by means of the total number of tracking errors per video sequence. The obtained results make the superior performance of the presented detection and tracking framework manifest. This can be attributed, on the one hand, to an improved detection stage, and on the other hand, to the ability of the tracking algorithm to handle false and missing detections.

TABLE II
Object tracking performance and comparison with another approach

Dataset	Sequence name	Camera	Nº tracking errors (Proposed algorithm)	Nº tracking errors (algorithm in [14])
PETS 2010	S2-L1	1	47	237
	S2-L1	3	72	358
	S2-L1	4	56	287
	S2-L1	5	44	243
	S2-L1	6	63	289
	S2-L1	7	23	221
	S2-L1	8	68	304

Regarding the computational burden, the average execution time per frame has been measured for each sequence for both algorithms, as is showed in Table III. The results show that the execution time of the proposed algorithm is slightly greater, since its complexity is significantly higher. Nonetheless, the obtained time results allow the processing of the input video in real or quasi real time. Both implementations have been developed using a combination of a scripting language and C++, and executed in a consumer PC with a quad-core based CPU at 2.53 GHz and a system bus at 1.3 GHz.

TABLE III
Comparison of execution time

Dataset	Sequence name	Camera	Execution time (ms) (Proposed algorithm)	Execution time (ms) (algorithm in [14])
PETS 2010	S2-L1	1	55	46
	S2-L1	3	53	44
	S2-L1	4	54	45
	S2-L1	5	56	48
	S2-L1	6	52	42
	S2-L1	7	53	45
	S2-L1	8	54	44

VII. CONCLUSIONS

A visual detection and tracking framework has been proposed for surveillance and counting applications. In addition, it has been especially designed to enter in the consumer electronics market, meeting the following requirements: off-the-shelf equipment, easy installation and configuration, and unsupervised working conditions. This is achieved by the combination of a moving detection algorithm that can handle split and merged detections, and the use of a novel Bayesian tracking model that can handle multimodal distributions, false detections, and missing detections. The proposed algorithm has been compared with another approach, also oriented to consumer electronics, proving its superior performance.

REFERENCES

[1] M. Piccardi, “Background subtraction techniques: a review”, *IEEE Proc. of International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3099-3104, Oct. 2004.

[2] C. R. del Blanco, F. Jaureguizar, and N. García, “Visual tracking of multiple interacting objects through raoblackwellized data association particle filtering,” *IEEE Int. Conf. Image Processing*, pp. 821–824, 2010.

[3] A. Genovesio and J.C. Olivo-Marin, “Split and merge data association filter for dense multi-target tracking,” *IEEE Proc. of International Conference on Pattern Recognition*, vol. 4, pp. 677–680, 2004.

[4] Y. Ma, Q. Yu, and I. Cohen, “Target tracking with incomplete detection,” *Comp. Vision and Image Understanding*, vol. 113, no. 4, pp. 580–587, 2009.

[5] Q. Yu and G. Medioni, “Multiple-target tracking by spatiotemporal monte carlo markov chain data association,” *IEEE Trans. on Patern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2196–2210, 2009.

[6] Z. Khan, T. Balch, and F. Dellaert, “Multitarget tracking with split and merged measurements,” *IEEE Proc. of Computer Vision and Pattern Recognition*, vol. 1, pp. 605–610, 2005.

[7] C.R. del Blanco, F. Jaureguizar, and N. García, “Bayesian Visual Surveillance: a Model for Detecting and Tracking a variable number of moving objects”, *IEEE Proc. of International Conference on Image Processing*, pp. 1437-1440, Sep. 2011.

[8] K. Yam, W. Siu, N. Law, and C. Chan, “Effective bidirectional people flow counting for real time surveillance system,” *IEEE Proc. International Conference on Consumer Electronics*, pp. 863-864, 2011.

[9] O. Tuzel, F. Porikli, and P. Meer, “Learning on lie groups for invariant detection and tracking,” *IEEE Proc. of International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[10] Z. Zivkovic and F. van der Heijden, “Efficient adaptive density estimation per image pixel for the task of background subtraction,” *Pattern Recognition Letters*, vol. 27, pp. 773–780, 2006.

[11] R, Y. Da Xu and M. Kemp, "Fitting multiple connected ellipses to an image silhouette hierarchically". *IEEE Trans. on Image Processing.* vol. 19, no. 7, 1673-1682, Jul 2010.

[12] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174-188, 2002.

[13] A. Doucet, S. Godsill, and C. Andrieu, “On sequential monte carlo sampling methods for bayesian filtering,” *Statistics and Computing*, vol. 10, no. 3, pp. 197-208, 2000.

[14] J. S. Kim, D. H. Yeom, Y. H. Joo, "Fast and robust algorithm of tracking multiple moving objects for intelligent video surveillance systems", *IEEE Trans. on Consumer Electronics*, vol. 57, no. 3, pp. 1165-1170, Aug. 2011.

BIOGRAPHIES

Carlos R. del Blanco received the Ph.D. degree with honors in Universidad Politécnica de Madrid (Spain) in 2011,and the “Ingeniero de Telecomunicación” degree with honors from Universidad Politécnica de Madrid (Spain) in 2005. He is currently a member of the Grupo de Tratamiento de Imágenes at the Department of Signals, Systems, and Radio-communications in the Universidad Politécnica de Madrid. His research interests include signal and image processing, computer vision, pattern recognition, machine learning, and stochastic dynamic models.

Fernando Jaureguizar received the Telecommunication Engineering degree and the Ph.D. degree in Telecommunication, both from the Universidad Politécnica de Madrid (UPM), in 1987 and 1994, respectively. Since 1987 he is a member of the Image Processing Group of the UPM. In addition, since 1991 he is a member of the faculty of the E.T.S. Ingenieros de Telecomunicación at UPM, and since 1995 he is an Associate Professor of Signal Theory and Communications at the Department of Signals, Systems, and Communications. His professional interests include digital image processing, video coding, 3DTV, computer vision, and design and development of multimedia communications systems. He has been actively involved in European projects (Eureka, ACTS and IST) and national projects in Spain.

Narciso García received Telecommunication Engineering degree and the Ph.D. degree in Telecommunication, both from the Universidad Politécnica de Madrid (UPM), in 1976 (Spanish National Graduation Award) and 1983 (Doctoral Graduation Award), respectively.

Since 1977 he is a member of the faculty of the UPM, where is currently Professor of Signal Theory and Communications. He leads the Image Processing Group of the UPM. He was Coordinator of the Spanish Evaluation Agency from 1990 to 1992 and evaluator, reviewer, and auditor of European programs since 1990. His professional and research interests are in the areas of digital image and video compression and of computer vision.