2010 IEEE/ASME International Conference on
Advanced Intelligent Mechatronics
Montréal, Canada, July 6-9, 2010

Real-time video surveillance by a hybrid static/active camera mechatronic system

Alexandros Iosifidis, Spyridon G. Mouroutsos, and Antonios Gasteratos, *Member, IEEE*

Abstract— **In this paper we present an effective real–time video surveillance system for real-life outdoor surveillance scenarios. The system is an integration of two subsystems: the static camera and moving one. The approaches employed address properly the challenges that might arise in a typical outdoor scene, such as local and global lighting changes, variations in objects' appearance and occlusions. Our aim is to detect and follow abnormal behaviors, which may occur when vehicles and pedestrians interact typical urban environment. Static camera subsystem operates multiple object tracking and classification. In case that an object of special interest is identified, the operation of the moving camera subsystem initiates, in order to track this object and record its activity.**

I. Introduction

As fixed and wireless broadband IP-networking technologies, digital video sensors and fast processors are a part of our everyday lives, automatic video surveillance is in the center of research due to its increased importance in safety and security issues [1][2]. Traditional systems need human operators to understand activities of objects of interest (humans, vehicles, etc.) and take decisions. On the other hand, automatic systems are capable of detecting objects of interest, track them from one frame to another and describe their activities with pre-learned patterns, in order to detect any typical or suspect action and take decision.

In indoor tracking scenarios, the scene is normally constrained and objects' appearance is clear due to the small camera-object distance. On the other hand, in outdoor tracking scenarios the scene is unconstrained and objects appear in small sizes. This makes the later procedure more difficult and unreliable.

Another categorization of systems is due to the capability of the camera to move. Most systems premise static cameras. On the other hand, many researchers have proposed several methods to track objects of interest with cameras capable to move, by means of servomechanisms, on two (Pan/Tilt cameras) or three (Pan/Tilt and Transfer) degrees of freedom. The integration of those two functionalities can lead to a hybrid system that is capable to detect, track and follow objects of interest in a wide area. This capability is very useful in security applications, where e.g. the entrance in a depository is the area of main interest.

Related work

Most systems described in literature utilize static cameras, which exhibit the advantage of a stationary background. An effective segmentation technique is capable to extract objects of interest. The most popular method is the one proposed in [3], where each background pixel is modeled by a mixture of Gaussians. Other techniques that for background segmentation and object tracking can be found in [4][5][6]. However, an open problem with foreground regions is that they often include shadowed pixels. Shadow elimination is an important background treatment technique [7][8]. After the extraction of foreground regions a tracking algorithm performs inter-frame correspondences in order to keep the id of every object of interest. An effective algorithm has to be able to handle changes in object's appearances, partial and/or complete occlusions, etc [9][10][11].

Systems designed for moving cameras can be categorized in Pan/Tilt camera systems and freely moving camera systems. In Pan/Tilt camera systems the concept of a relatively static background model is valid. The main approach is to shift the background information in regard with the camera rotation [12][13]. In freely moving camera systems, the most popular method models, the appearance of objects in the first frame and future appearances are obtained using Bayesian filters via sampling in the every frame [14][15].

In this paper, we present a system that integrates the functionality of a static-camera system and a moving-camera one, with applications mainly into outdoor scenarios. In the first case the background is modelled with a fast and effective technique and objects of interest are tracked using Bayesian filters. Single objects are tracked using high-order Kalman filters while directed Particle filters treat the merged objects. Objects are classified using a heuristic method. In the second case, the initialization is performed automatically with the appearance of an "object of special interest". This object is tracked and recorded and a number of instructions are produced in order to move the camera. The experimental result show that proposed system is accurate and fast enough to be utilized in real-life cases.

Manuscript received January 27, 2010.

Alexandros Iosifidis and Spyridon G. Mouroutsos are with the Electrical and Computer Engineering Department, Democritus University of Thrace, Xanthi, GR-671 00, Greece, (e-mail: sgmour@duth.gr).

Antonios Gasteratos is with the Production and Management Engineering Department, Democritus University of Thrace, Xanthi, GR-671 00 Greece.

The paper is organized as follows. In section 2, we describe the functionality of our system. Details about the procedures of every subsystem are presented and discussed. In section 3, we present some quantitative and functional results. Finally, concluding remarks are apposed in section 4.

II. System description

Our system consists of two subsystems due to the camera motion capability. Figure 1 shows the block diagram of its operation. The diagram shows the relation between those two subsystems: Static-camera subsystem uses a background subtraction technique to separate foreground regions, a procedure which tracks objects of interest, and a procedure which classifies objects of interest. Moving camera subsystem is initialized using the region of interest (ROI) that belongs to the object of special interest, tracks it and generates a number of instructions for camera's movement system.

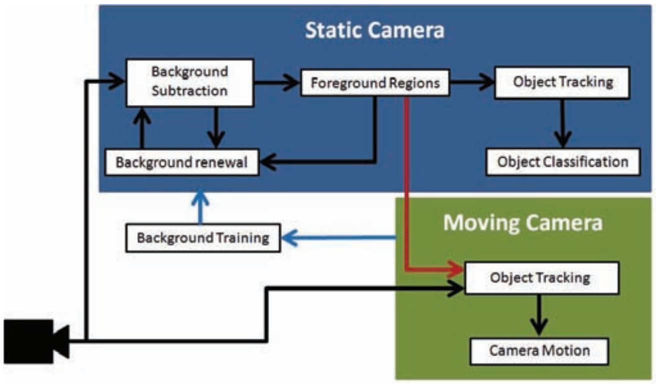

Fig. 1 System's block diagram

A. Static-Camera Subsystem

This subsystem is based on the well-know technique of background subtraction. An adaptive technique is used to create an image of the scene that does not contain any object of interest. The difficulties that arise are the treatment of shadows and illumination changes, which generates false foreground regions. Areas of interest resulted from this technique correspond to one or multiple objects. We use filters to predict and/or estimate the appearance of every object of interest in order to track every object separately.

Foreground Extraction

Foreground extraction is achieved via a pixel-based technique that integrates chromatic and texture information of the observed scene. It is a generalization of the model described in [7]. The addition of texture information is used to restrict false foreground regions created from quick illumination changes.

The initialization includes a training procedure for a number of frames, which can be described from:

$$BCK(x,y)_C^t = \alpha_C * BCK(x,y)_C^{t-1} + (1-\alpha_C) * I(x,y)^t \tag{1}$$

$$BCK(x,y)_D^t = \alpha_D * BCK(x,y)_D^{t-1} + (1-\alpha_D) * dI(x,y)^t \tag{2}$$

where $BCK(x,y)_C$ is the chromatic background image, $BCK(x,y)_D$ is the texture background image and α_C, α_D are the absorbent factors.

Because of the expectation that the majority of pixels belongs to the background, we model the absolute difference of pixel's colors (R,G,B) with three Gaussian distributions. We calculate the maximum standard deviation (S_{max}) of these distributions in a small period (10 – 15 seconds). Using this value we can decide if a pixel belongs to background, i.e. if its three values of absolute deference with the corresponding background pixel are smaller than a threshold, it is a background pixel. This threshold changes adaptively using information obtained from the scene, e.g. a threshold equal to $2*S_{max}$ indicates that 68% of pixels belong to the background.

For those pixels that do not belong to the background we calculate two factors: its chromatic and its brightness distortion, respectively:

$$\delta Br(x,y) = \|BCK(x,y)\| - \frac{I(x,y)*BCK(x,y)}{\|BCK(x,y)\|} \tag{3}$$

$$\delta Cr(x,y) = \cos^{-1}\left(\frac{I(x,y)BCK(x,y)}{\|I(x,y)\|\|BCK(x,y)\|}\right) \tag{4}$$

where $I(x,y)$ denotes the incoming frame and $BCK(x,y)$ models the background image. If the absolute values of these factors are smaller than predefined thresholds the pixel is characterized as a shadow. All the other pixels belong to foreground areas. Every background pixel is updated with a procedure similar to the training one.

The usage of this model generates a number of false foreground regions as a result of spurious background motion (e.g. waving tree leaves). Those regions can be rejected in the procedure of object tracking, which follows.

Object Tracking

The output of the previous procedure is a binary mask that indicates the ROIs. These regions may consist of one or more objects of interest. Our tracking system separates those two cases (single object tracking and multiple objects tracking, respectively). Every object of interest is described by its approximate size (width – height), its position (center of its bounding box) and its color (histogram). In most real scenes the appearance and the position of objects of interest alters dynamically. This generates the necessity to predict the state of every object at every time. Our system uses two high order linear Kalman filters (4th order) to predict the position and the size for every object of interest. This selection was made in order to obtain quick and accurate predictions. In every frame, predictions are made for every object of the previous frame. A flood fill algorithm extracts the ROIs from the foreground mask. The position, the size and the histogram of every ROI is compared with the corresponding values of every object. Histograms are equalized in order to correctly match objects that enter in or exit from shadowed regions.

As an object is matched with its appearance in the previous frame, its histogram and prediction filters are updated. If a region does not match with any object, a new candidate object is created to represent a new object of interest. If this region is matched correctly for a number of frames (four in our system), the object is indicated as "visible", else it is deleted. If an object does not match with any region, it is indicated as "invisible" and is updated with the estimations created by the Kalman filter. If an object is invisible for a number of frames it is omitted. Regions that include more than one object are indicated as a "group" (case of merged objects). This initializes the procedure of multiple objects tracking. To handle the case of false foreground regions generated by spurious background motion we use the "motion factor" used in [11]:

$$s_m = \frac{\left(\dfrac{\sigma_{cx}^2}{\sigma_{ux}^2 + \tau} + \dfrac{\sigma_{cy}^2}{\sigma_{uy}^2 + \tau}\right)}{2} \qquad (5)$$

where σ_{xx}^2 (σ_{cy}^2) is the x-(y) – directional variance of centroid of the ROI, σ_{ux}^2 (σ_{uy}^2) is the corresponding x-(y) – directional velocity variance, τ is a small constant to prevent an absolute standing – still object exploding s_m. If this factor is smaller than a threshold the object is rejected (spurious background motion).

If a "group" is observed then every object which belongs to this group is modelled with a directed Particle filter. We use the information of the velocity of the object in the previous frame in order to direct the filter. This gives us the advantage to use a small number of particles for every object and speed up the process. If the velocity is bigger than a value, then its new state will be in a sector which is indicated by the direction of its velocity. This value indicates small object motion. Sector's radius is equal to 1/4 of object's biggest dimension and the angle equal to π/4 (rad). Figure 2 shows the region of action for every filter.

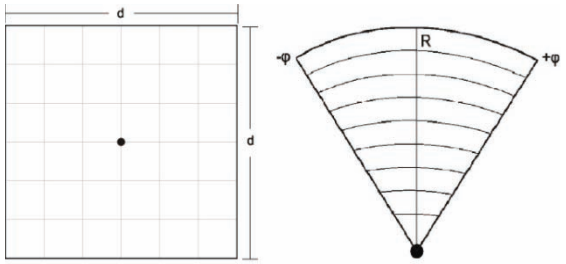

Fig. 2 Active region of directed Particle filter

The appearance of the object in the previous frame is used for the initialization. The input in every filter is the grayscale, hue and first spatial derivative histogram of the appearance of the object. This selection was made in order to keep object's color and texture information. To link backward this procedure with the procedure of Single Object Tracking, the coefficients of the Kalman filters are updated with the estimations of the Particle filters. Eventually, the group of merged objects are split and each object is observed as a single one.

Object Classification

Our classification procedure uses the velocity information of an object to classify it as a rigid or floppy one. We define the "angle of non-rigid parts" as follows:

$$\theta_\alpha = \frac{\sum_{C=1}^{n} |\phi|}{C_{o\alpha}} C_\alpha \qquad (6)$$

where θ_α is the angle of non-rigid parts of object α, φ is the angle between a corner and object's directions, C_α denotes the number of object's corners that their velocity vector creates an angle with object's velocity vector bigger than a threshold and $C_{o\alpha}$ is the total number of object's corners. The selection of the threshold is important and defines the degree of rigidness of an object which is considered as rigid or not.

B. Moving-Camera Subsystem

The second subsystem initiates as soon as an "object of special interest" emerges. The indication of this object may be done automatically by a decision extraction system; however, in our current implementation this is indicated by the user during the static camera object tracking operation. The goal of this function is to keep the object in the camera's field of view as long as possible and record its activity during its appearance in the scene. As in literature this task is mainly approached by the usage of Bayesian filters, we use a Particle filter to model object's appearance and track it until it exits the scene or until the user chooses to terminate the procedure.

Object Tracking

The initialization of this function is made in the selection frame. Using object's foreground region created by the static camera object tracking procedure, we use object's position and size, and calculate histograms of objects intensity, hue and first spatial derivative images, respectively, to describe its appearance. By retaining these three images the object's dominant color, a linear combination of object's colors and object's texture information are preserved. These are the inputs of a Particle filter which uses second-order autoregressive dynamics for its propagation:

$$x^t = A_1\left(x^{t-1} - x^0\right) + A_2\left(x^{t-2} - x^0\right) + B_0 R_x + x^0 \quad (7)$$

$$y^t = A_1\left(y^{t-1} - y^0\right) + A_2\left(t^{t-2} - y^0\right) + B_0 R_y + y^0 \quad (8)$$

$$s^t = A_1\left(s^{t-1} - 1\right) + A_2\left(s^{t-2} - 1\right) + B_0 R_s + 1 \quad (9)$$

where (x^t, y^t) and s^t denote the object's position and scale in the current frame; (x_0,y_0) is the object's position in initialization frame; A_i,B_i are the models coefficients and R_i denotes random numbers (Gaussian distribution).

Camera motion

In every frame object's position is compared with frame's center and a number of instructions are generated in order to keep the object in camera's field of view. In other words, the vector denoting the object's translation on the 2D image plane is transformed into motors commands into the actuators' space. While in our implementation we employed a Pan/Tilt mechanism, a similar procedure as the one used by our moving-camera subsystem can be utilized by any visual-servoing mechatronic mechanism, e.g. a camera rotation-translation one. When the function of this procedure ends, the system returns in the static camera procedure. This enables the training of the background model at the new scene, which resulted by the movement of the camera.

III. EXPERIMENTAL RESULTS

Our software system operates in video frame size of 320X240 on a 2.5GHz PC without any optimization. Single object tracking runs at about 31 frames per second, whilst, merged objects tracking runs at about 20 frames per second. The process of classification slows down the process at 6 frames per second and is optional.

Our system has been tested on four well-known video sequences of real-world surveillance scenarios [16][17][18]. Instances of those videos are shown in Figure 3. In Figures 4 and 5, a number of example results are given, showing the system handling successfully typical problems encountered in the outdoor urban surveillance scenario of AVSS 2007. In Figure 4 green rectangles show regions of interest detected from background model and red rectangles tracked objects of interest.

We also present some quantitative analysis of a few test results. We count the number of objects tracked correctly during their appearance in the scene in the case of low occlusion (lower than 25%). In this process we do not consider border areas of each frame and very small objects (those less than 50 pixels).

Table 1 lists the analysis results of tested videos. Briefly, PETS'2000 scene shows a small portion of a crossroad, PETS'2001 scene contains a number of small items (humans and vehicles) in a very noisy environment, Visor's scene shows a parking lot and AVSS'2007 scene shows an urban road. These results are very encouraging since Visor's and AVSS'2007 scenes represent real common urban surveillance scenes.

TABLE I
EVALUATION RESULTS

Video	Frames	Objects	Success (%)
PETS 2000	1451	6	100
PETS 2001	1245	11	72.7
Visor	1500	10	90
AVSS 2007 (Easy)	3000	55	92.7
AVSS 2007 (Medium)	3000	35	92.1

We also have compared our system with those found in [11] and [19], as exhibited in Table 2. These results show that our system is comparative in performance. Furthermore, it performs in real time, as the system found in [11] does, while the frame rate of the system in [19] is low.

Fig. 3 Instances of evaluation videos

TABLE III
COMPARISON RESULTS

Video	Our System	System [2]	System [19]
PETS'01(DS1-C1)	95	93.4	-
AVSS'07 Easy	92.7	-	98.3
AVSS'07 Medium	92.1	-	95.2

V. Conclusions

In this paper we presented a real-time integrated video surveillance solution, which can respond in a robust way in low occlusion urban surveillance scenes. The proposed system aims at the detection of suspicious actions in camera's visual field and keep the object of "special interest" inside cameras view as long as possible. This is achieved by comprising two independent subsystems into a single surveillance system. The static camera subsystem is based on a fast and adaptive background subtraction technique with shadow detection, which uses scene's chromatic and texture information. The extracted foreground mask is utilized by a blob-based tracking algorithm, which distinguishes the procedure of object tracking into single and multiple ones. In the first case high-order Kalman filters are used to predict object's appearance (position and size) and color information is used to match objects in the consecutive frames. In the second case object's appearance is modeled with directed Particle filters in order to speed up the procedure. Classification is performed using object's motion information. The moving camera subsystem is based on Particle filters. Chromatic and texture information is used to describe object's appearance and filter's propagation process is based on second-order autoagressive dynamics. Object's position is used to produce a number of instructions into the actuators space, which direct a camera-movement system. Several scenarios of abnormal behaviors have to be considered to fully automate the transition from static-camera to moving-camera subsystem. Experimental results show that our system is fast and reliable enough to be applied into typical urban surveillance scenes.

Some useful observations are as follows: (1) Although the system responds well in small camera shake, there is a need to stabilize the videos captured prior to object detection to face severe camera shake. (2) It is necessary to extend our system to include successful tracking in the case of higher occlusion level. (3) It is necessary to introduce a better criterion to exclude false foreground regions created from moving background objects.

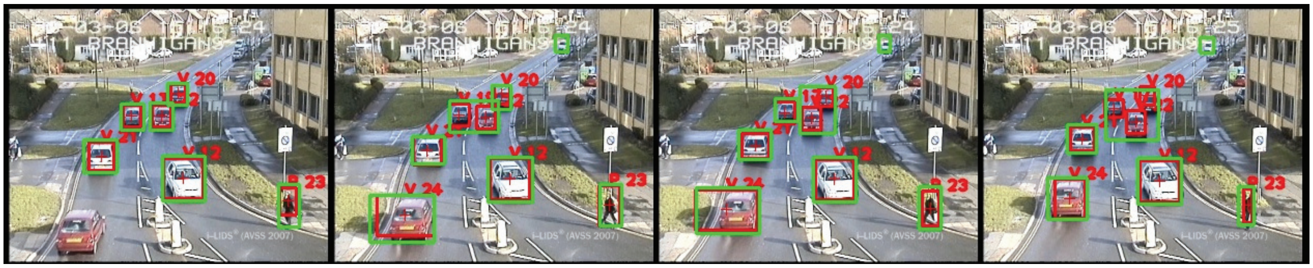

Fig. 4 Output of static camera subsystem in AVSS' scene. A typical merging – splitting sequence of objects of interest

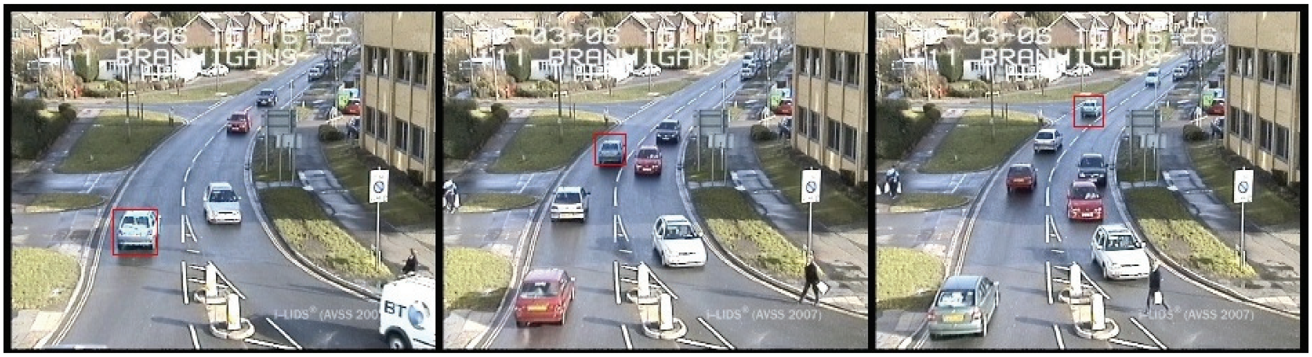

Fig. 5 Output of Moving camera subsystem in AVSS' scene. An object enters and exits in a shadowed region

VI. References

[1] W. Hu, T. Tan, L. Wang and S. Maybank "A Survey on Visual Surveillance of Object Motion and Behaviors,". IEEE Transactions On Systems, Man, and Cybernetics, 2004.

[2] T.B. Moeslund,, A. Hilton, and V. Kruger, "A Survey of Advances in Vision-Based Human Motion Capture and Analysis," Journal of Computer Vision and Image Understanding, 104(2-3), 2006.

[3] C. Stauffer, W.E.L. Grimson, "Adaptive background mixture models for real-time tracking", Proc. of CVPR 1999, pp. 246-252.

[4] L.L.W. Huang, I.Y.H. Gu and Q. Tuan, "Foreground Object Detection from Videos Containing Complex Background", in Proc. ACM Multimedia conf., ACM, Berkeley, CA, USA, November 2-8 2003.

[5] V. Mahadevan, N. Vasconelos, "Background Subtraction in Highly Dynamic Scenes," in *Conf. on Computer Vision and Pattern Recognition, Anchorage 2008*.

[6] R.M. Luque, E. Dominguez, E.J. Palomo, J. Munoz, "A Neural Network Approach for Video Object Segmentation in Traffic Surveillance"

[7] M. Dahmane and J. Meunier "Real-Time Video Surveillance with Self-Organizing Maps"Proc. Of the Second Canadian Conference on Computer and Rovot Vision ,2005.

[8] A. Leone, C. Distante, F. Buccolieri, "A Shadow Elimination Approach in Video-Surveillance Context," Pattern Recognition Letters 27, pp 345-355, 2006 .

[9] Li, L., Huang, W., Gu, I.Y.-H., Luo, R., Tian, Q. "An efficient sequential approach to tracking multiple objects through crowds for real-time intelligent cctv systems. " IEEE Trans. on Systems, Man, and Cybernetics 38(5), 1254–1269 (2008).

[10] A.R.J. Francois, "Real-Time Multi-Resolution Blob Tracking," IRIS Techical Report, IRIS-04-422, University of Southern California, Los Angeles, USA, 2004.

[11] B. Lei and L.-Q. Xu, "Real-time outdoor video surveillance with robust foreground extraction and object tracking via multi-state transition management," Pattern Recognition. Letters., vol. 27, no. 15, pp. 1816–1825, 2006.

[12] C. Micheloni, G.L. Foresti, "Real Time Image Processing for Active Monitoring of Wide Areas," Jurnal of Visual Communication and Image Representation Special issue on Real-Imaging, No. 3, pp. 589-604, 2006.

[13] A. Janelle, "A System to Automatically Track Humans and Vehicles with a PTZ Camera," in Visual Information Processing XVI, Proc. Of the SPIE, 2007.

[14] J. Xue, N. Zheng, J. Geng, X. Zhong, "Tracking Multiple Visual Targets via Particle-Based Belief Propagation," IEEE Trans. Syst., Man, Cybern, vol. 38, no. 1, pp. 196-209, 2008.

[15] E. Maggio, F. Smeraldi, A.Cavallaro, "Adaptive Multi-Feature Tracking in a Particle Filtering Framework," IEEE Trans. Circuits and Systems for Video Technology 10, pp. 1348-1359, 2007.

[16] Video Surveillance Online Repository (VISOR), http://imagelab.ing.unimore.it/visor/video_videosInCategory.asp?idcategory=12.

[17] Performance Evaluation of Tracking and Surveillance (PETS), http://www.cvg.rdg.ac.uk/slides/pets.html.

[18] Advanced Video and Signal based Surveillance (AVSS), http://www.elec.qmul.ac.uk/staffinfo/adrea/avss2007_d.html

[19] J.T. Lee, M.S. Ryoo, M.Riley, J.K.Aggarwal, "Real-time Detection of Illegally Parked Vehicles using 1-D Transformation," Proc. of the 2007 IEEE Conf. on Advanced Video and Signal Based Surveillance, pp. 254-259, 2007