Detecting Moving Objects, Ghosts and Shadows in Video Streams

Rita Cucchiara* [1],Costantino Grana[1], Massimo Piccardi[2], Andrea Prati[1]

Abstract

Background subtraction methods are widely exploited for moving object detection in videos in many applications, such as traffic monitoring, human motion capture and video surveillance. How to correctly and efficiently model and update the background model and how to deal with shadows are two of the most distinguishing and challenging aspects of such approaches. This work proposes a general-purpose method which combines statistical assumptions with the object-level knowledge of moving objects, apparent objects (ghosts) and shadows acquired in the processing of the previous frames. Pixels belonging to moving objects, ghosts and shadows are processed differently in order to supply an object-based selective update. The proposed approach exploits color information for both background subtraction and shadow detection to improve object segmentation and background update. The approach proves fast, flexible and precise in terms of both pixel accuracy and reactivity to background changes.

Keywords

Background modeling, color segmentation, reactivity to changes, shadow detection, video surveillance, object-level knowledge

I. Introduction

DETECTION of moving objects in video streams is the first relevant step of information extraction in many computer vision applications, including video surveillance, people tracking, traffic monitoring and semantic annotation of videos. In these applications, robust tracking of objects in the scene calls for a reliable and effective moving object detection that should be characterized by some important features: high precision, with the two meanings of accuracy in shape detection and reactivity to changes in time; flexibility in different scenarios (indoor, outdoor) or different light conditions; and efficiency, in order for detection to be provided in real-time. In particular, while the fast execution and flexibility in different scenarios should

* Corresponding author.
[1]Dipartimento di Ingegneria dell'Informazione, Università di Modena e Reggio Emilia, Via Vignolese, 905 - 41100 Modena - Italy - phone: +39-059-2056136 - fax: +39-059-2056126 - e-mail:{rita.cucchiara/andrea.prati}@unimo.it, grana@dsi.unimo.it
[2]Department of Computer Systems, Faculty of IT, University of Technology, Sydney - Broadway NSW 2007 - Australia - phone: +61-2-9514-7942 - fax: +61-2-9514-1807 - e-mail: massimo@it.uts.edu.au

be considered basic requirements to be met, precision is another important goal. In fact, a precise moving object detection makes tracking more reliable (the same object can be identified more reliably from frame to frame if its shape and position are accurately detected) and faster (multiple hypotheses on the object's identity during time can be pruned more rapidly). In addition, if object classification is required by the application, precise detection substantially supports correct classification.

In this work, we assume that the models of the target objects and their motion are unknown, so as to achieve maximum application independence. In the absence of any *a priori* knowledge about target and environment, the most widely adopted approach for moving object detection with fixed camera is based on *background subtraction* [1][2][3][4][5][6][7][8][9]. An estimate of the background (often called a *background model*) is computed and evolved frame by frame: moving objects in the scene are detected by the difference between the current frame and the current background model. It is well known that background subtraction carries two problems for the precision of moving object detection. The first problem is that the model should reflect the real background as accurately as possible, to allow the system accurate shape detection of moving objects. The detection accuracy can be measured in terms of correctly and incorrectly classified pixels during normal conditions of the object's motion (i.e. the "stationary background" case). The second problem is that the background model should immediately reflect sudden scene changes such as the start or stop of objects, so as to allow detection of only the actual moving objects with high reactivity (the "transient background" case). If the background model is neither accurate nor reactive, background subtraction causes the detection of false objects, often referred to as "ghosts" [1][3]. In addition, moving object segmentation with background suppression is affected by the problem of *shadows* [4][10]. Indeed, we would like the moving object detection to not classify shadows as belonging to foreground objects, since the appearance and geometrical properties of the object can be distorted, which in turn affects many subsequent tasks such as object classification and the assessment of moving object position (normally considered to be the shape centroid). Moreover, the probability of object undersegmentation (where more than one object is detected as a single object) increases due to connectivity via shadows between different objects.

Feature	Systems
Statistics	• Minimum and maximum values [1] • Median [11][12], * • Single Gaussian [5][4][13] • Multiple Gaussians [14][10][3] • Eigenbackground approximation [15][6] • Minimization of Gaussian differences [7]
Adaptivity	[1][6][5][8][16][2], *
Selectivity	[10][2][8][1], *
Shadow	[4][10], *
Ghost	[1][3], *
High-frequency illumination changes	• Temporal filtering [14][15][6] • Size filtering *
Sudden global illumination changes	[1], *

TABLE I
COMPARED BACKGROUND SUBTRACTION APPROACHES. OUR APPROACH IS REFERRED WITH *.

Many works have been proposed in the literature as a solution to an efficient and reliable background subtraction. Table I is a classification of the most relevant papers based on the features used. Most of the approaches use a statistical combination of frames to compute the background model (see Table I). Some of these approaches propose to combine the current frame and previous models with recursive filtering (adaptivity in Table I) to update the background model. Moreover, many authors propose to use pixel selectivity by excluding from the background update process those pixels detected as in motion. Finally, problems carried by shadows have been addressed [4][10][17]. In this paper we propose a novel simple method that exploits all these features, combining them so as to efficiently provide detection of moving objects, ghosts and shadows. The main contribution of this proposal is the integration of knowledge of detected objects, shadows and ghosts in the segmentation process to enhance both object segmentation and background update. The resulting method proves to be accurate and reactive, and at the same time fast and flexible in the applications.

II. DETECTING MOVING OBJECTS, GHOSTS AND SHADOWS

The first aim of our proposal is to detect real moving objects with high accuracy, limiting false negatives (object's pixels that are not detected) as much as possible. The second aim is to extract pixels of moving

objects with the maximum responsiveness possible, avoiding detection of transient spurious objects, such as cast shadows, static objects or noise.

To accomplish these aims, we propose a taxonomy of the objects of interest in the scene, using the following definitions (see also Fig. 1):

- *Moving visual object* ($\mathbf{MVO}$): set of connected points belonging to object characterized by non-null motion.
- *Uncovered Background*: the set of visible scene points currently not in motion.
- *Background* ($\mathbf{B}$): is the computed model of the background.
- *Ghost* ($\mathbf{G}$): a set of connected points detected as in motion by means of background subtraction, but not corresponding to any real moving object.
- *Shadow*: a set of connected background points modified by a shadow cast over them by a moving object. Shadows can be further classified as *MVO shadow* ($\mathbf{MVO_{SH}}$), that is, a shadow connected with an MVO and hence sharing its motion, and *ghost shadow* ($\mathbf{G_{SH}}$), being a shadow not connected with any real MVO.

Static cast shadows are neither detected nor considered since they do not affect moving object segmentation if background subtraction is used: in fact, static shadows are included in the background model. A ghost shadow can be a shadow cast either by a ghost or an MVO: the shape and/or position of the MVO with respect to the light source can lead to the shadow not being connected to the object that generates it.

Our proposal makes use of the the explicit knowledge of all the above five categories for a precise segmentation and an effective background model update. We call our approach *Sakbot* (Statistical And Knowledge-Based ObjecT detection) since it exploits statistics and knowledge of the segmented objects to improve both background modeling and moving object detection. Sakbot is depicted in Fig. 1, reporting the aforementioned taxonomy. Sakbot's processing is the first step for different further processes, such as object classification, tracking, video annotation and so on.

Let us call *p* a point of the video frame at time t ($\mathbf{I^t}$). $\mathbf{I^t}(p)$ is the value of point *p* in the color space. Since images are acquired by standard color cameras or decompressed from videos with standard formats, the basic

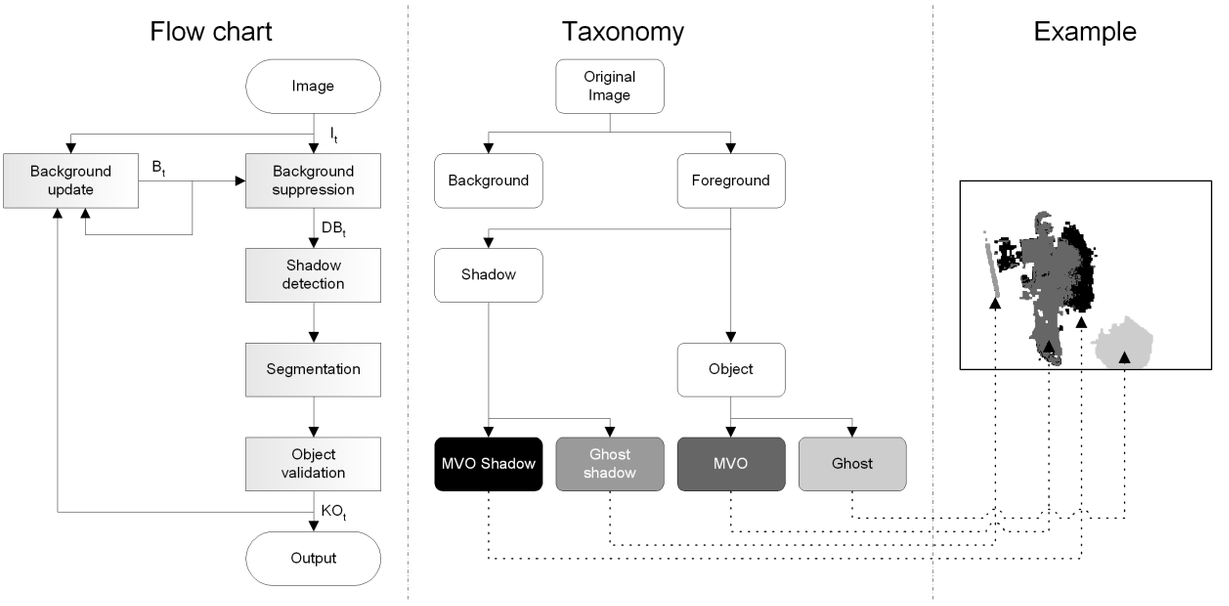


Fig. 1. Sakbot architecture

color space is RGB. Thus $\mathbf{I^t}(p)$[1] is a vector with R,G,B components. The goal is to compute, at each time t, both the set $\mathbf{KO^t}$ of known objects and the background model $\mathbf{B^t}$; in accordance with the taxonomy, $\mathbf{KO^t}$ is defined as:

$$\mathbf{KO^t} = \{\mathbf{MVO^t}\} \cup \{\mathbf{MVO^t_{SH}}\} \cup \{\mathbf{G^t}\} \cup \{\mathbf{G^t_{SH}}\} \tag{1}$$

$\mathbf{B^t}$ is the background model at time t and is defined for each point of the image. If p is a point of the uncovered background then $\mathbf{B^t}(p)$ should correspond to its value in the current frame; however, if p is a point of a known object (i.e. that has been segmented and classified), $\mathbf{B^t}(p)$ is an estimation of the value of background covered by the object itself.

If point p does not belong to any known object, the background value in p is predicted using only statistical information ($\mathbf{B_s^{t+\Delta t}}(p)$) on the following set S of elements:

$$S = \{\mathbf{I^t}(p), \mathbf{I^{t-\Delta t}}(p), ..., \mathbf{I^{t-n\Delta t}}(p)\} \cup w_b\{\mathbf{B^t}(p)\} \tag{2}$$

As it is possible to note from Eq. 2, in order to improve the stability of the model we exploited *adaptivity* too. We include an adaptive factor by combining the n sampled frame values and the background past values (with an adequate weight w_b). The n frames are sub-sampled from the original sequence at a rate of one

[1]We use a bold notation to represent the vectors (like images) and a non-bold notation to indicate the single vector element.

every Δt (typically one every ten). Then, the statistical background model is computed by using the median function (as in [11][12]) as follows:

$$\mathbf{B_s^{t+\Delta t}}(p) = \underset{i=1,\ldots,k}{arg\ min} \sum_{j=1}^{k} Distance(\mathbf{x_i}, \mathbf{x_j}) \qquad \mathbf{x_i}, \mathbf{x_j} \in S \tag{3}$$

where the distance is a *L-inf distance* in the RGB color space:

$$Distance(\mathbf{x_i}, \mathbf{x_j}) = max(|x_i.c - x_j.c|) \quad \text{with } \mathrm{c} = \mathrm{R}, \mathrm{G}, \mathrm{B}. \tag{4}$$

In our experiments, the median function has proven effective while, at the same time, of less computational cost than the Gaussian or other complex statistics.

Foreground points resulting from the background subtraction could be used for the selective background update; nevertheless, in this case, all the errors made during background subtraction will consequently affect the selective background update. A particularly critical situation occurs whenever moving objects are stopped for a long time and become part of the background. When these objects start again, a ghost is detected in the area where they were stopped. This will persist for all the following frames, preventing the area to be updated in the background image forever, causing *deadlock* [10]. Our approach substantially overcomes this problem since it performs selectivity not by reasoning on single moving points, but on detected and recognized moving objects. This object-level reasoning proved much more reliable and less sensitive to noise than point-based selectivity. Therefore, we use a knowledge-based background model defined as:

$$\mathbf{B_k^{t+\Delta t}}(\mathrm{p}) = \begin{cases} \mathbf{B^t}(\mathrm{p}) & if\ \ p \in \mathbf{O}, \mathbf{O} \text{ in } \{\mathbf{MVO^t}\} \cup \{\mathbf{MVO_{SH}^t}\} \\ \mathbf{B_s^{t+\Delta t}}(\mathrm{p}) & if\ \ p \in \mathbf{O}, \mathbf{O} \text{ in } \{\mathbf{G^t}\} \cup \{\mathbf{G_{SH}^t}\} \end{cases} \tag{5}$$

The knowledge of the scene's components in the current frame will be used to update the background model:

$$\mathbf{B^{t+\Delta t}}(p) = \begin{cases} \mathbf{B_s^{t+\Delta t}}(p) & if\ \nexists \mathbf{O} \in \mathbf{KO^t} :\ p \in \mathbf{O} \\ \mathbf{B_k^{t+\Delta t}}(p) & otherwise \end{cases} \tag{6}$$

The expression in Eq. 6 defines a selective background update, in the sense that a different background model is selected whether the point belongs to a known object or not. Differently from other proposals

([1][10][2][8]), selectivity is at *object-level* and not at pixel-level only, in order to modify the background in accordance with the knowledge of the objects detected in the scene. The advantage is that the background model is not "corrupted" by moving objects and thus it is possible to use a short Δt and a small n so as to also achieve reactivity.

In our approach, after background subtraction, a set of points called foreground points is detected and then merged into labeled blobs according to their connectivity. An initial camera motion compensation might have been performed previously, should the application require it (for example, to compensate small camera vibrations due to non-ideal operational conditions). This step is based on the choice of a fixed reference in the scene assumed to be never occluded at run time. In order to improve detection, background subtraction is computed by taking into account not only a point's brightness, but also its chromaticity, as in Eq. 4:

$$\mathbf{DB^t}(p) = Distance(\mathbf{I^t}(p), \mathbf{B^t}(p)) \tag{7}$$

The L-inf distance has proven effective in our experiments, while at the same time being less computationally expensive than other distances. In fact, other metrics can be used as the Euclidean distance, or the Mahalanobis distance used in [5], but this last is computationally more severe since it associates the correlation between parameters using the covariance matrix.

The selection of the initial set of foreground points is carried out by selecting the distance image $\mathbf{DB^t}$ defined in Eq. 7 with an adequately low threshold T_L. Among the selected points, some are discarded as noise, by applying morphological operators. Then, the shadow detection process is applied (as described in Section III) and the detected points are labeled as shadow points. A region-based labeling is then performed to obtain connected blobs of candidate moving objects and shadows. Eventually, blob analysis validates the blobs of candidate moving objects as either moving objects or ghosts. MVOs are validated by applying a set of rules on *area*, *saliency* and *motion* as follows:

- The MVO blob must be large enough (greater than a threshold T_A that depends on the scene and on the signal-to-noise ratio of the acquisition system); with this validation, blobs of a few pixels (due, for instance,

to high frequency background motion, like movements of tree leaves) can be removed;

- The MVO blob must be a "salient" foreground blob, as ascertained by a hysteresis thresholding. The low threshold T_L set on the difference image $\mathbf{DB^t}$ inevitably selects noise together with all the actual foreground points. A high threshold T_H selects only those points with a large difference from the background and validates the blobs which contain at least one of these points;
- The MVO blob must have non negligible motion. To measure motion, for each pixel belonging to an object we compute the spatio-temporal differential equations for optical flow approximation, in accordance with [18]. The *average optical flow* computed over all the pixels of an MVO blob is the figure we use to discriminate between MVOs and ghosts: in fact, MVOs should have significant motion, while ghosts should have a near-to-zero average optical flow since their motion is only apparent.

Optical flow computation is a highly time-consuming process; however, we compute it only when and where necessary, that is only on the blobs resulting from background subtraction (thus a small percentage of image points). The same validation process should also be carried out for shadow points, in order to select those corresponding to the set of *MVO shadows* and those belonging to *ghost shadows*. However, computing the optical flow is not reliable on uniform areas such as shadows. In fact, the spatial differences in the optical flow equation is nearly null because shadows smooth and make uniform the luminance values of the underlying background. Therefore, in order to discriminate MVO shadows from ghost shadows, we use information about connectivity between objects and shadows. Shadow blobs connected to MVOs are classified as shadows, whereas remaining ones are considered as ghost shadows. The box of Fig. 2 reports the rules adopted for classifying the objects after blob segmentation. All foreground objects not matching any of the rules in Fig. 2 are considered background and used for background update.

$< MVO >\longleftarrow$ (foreground blob) $\wedge \neg$ (shadow) $\wedge$ (large area) $\wedge$ (high saliency) $\wedge$ (high average optical flow)
$< Ghost >\longleftarrow$ (foreground blob) $\wedge \neg$ (shadow) $\wedge$ (large area) $\wedge$ (high saliency) $\wedge \neg$ (high average optical flow)
$< MVO\ shadow >\longleftarrow$ (foreground blob) $\wedge$ (shadow) $\wedge$ (connected with MVO)
$< Ghost\ shadow >\longleftarrow$ (foreground blob) $\wedge$ (shadow) $\wedge \neg$ (connected with MVO)

Fig. 2. Validation rules

In conclusion, by including Eq. 5 in Eq. 6, *the background model remains unchanged for those points that*

belong to detected MVOs or their shadow. Instead, points belonging to a ghost or ghost shadow are considered potential background points and their background model is updated by use of the statistic function.

III. SHADOW DETECTION

By *shadow detection* we mean the process of classification of foreground pixels as "shadow points" based on their appearance with respect to the reference frame, the background. The shadow detection algorithm we have defined in Sakbot aims to prevent moving cast shadows being misclassified as moving objects (or parts of them), thus improving the background update and reducing the undersegmentation problem. The major problem is how to distinguish between moving cast shadows and moving object points. In fact, points belonging to both moving objects and shadows are detected by background subtraction by means of Eq. 7. To this aim, we analyze pixels in the Hue-Saturation-Value (HSV) color space. The main reason is that the HSV color space explicitly separates chromaticity and luminosity and has proved easier than the RGB space to set a mathematical formulation for shadow detection.

For each pixel belonging to the objects resulting from the segmentation step, we check if it is a shadow according to the following considerations. First, if a shadow is cast on a background, the hue component changes, but within a certain limit. In addition, we considered also the saturation component, which was also proven experimentally to change within a certain limit. The difference in saturation must be an *absolute* difference, while the difference in hue is an *angular* difference.

We define a shadow mask SP^t for each point p resulting from motion segmentation based on the following three conditions:

$$SP^t(p) = \begin{cases} 1 & if \quad \alpha \leq \frac{I^t(p).V}{B^t(p).V} \leq \beta \wedge \ |I^t(p).S - B^t(p).S| \leq \tau_S \wedge \ D_H \leq \tau_H \ ; \quad \alpha \in [0,1], \ \beta \in [0,1] \\ 0 & otherwise \end{cases} \tag{8}$$

where the $.H$ denotes the hue component of a vector in the HSV space and is computed as:

$$D_H^t(p) = min(\ |I^t(p).H - B^t(p).H|, \ 360 - |I^t(p).H - B^t(p).H|\ ) \tag{9}$$

The lower bound α is used to define a maximum value for the darkening effect of shadows on the background, and is approximately proportional to the light source intensity. Instead the upper bound β prevents the system from identifying as shadows those points where the background was darkened too little with respect to the expected effect of shadows. Approximated values for these parameters are also available based on empirical dependence on scene luminance parameters such as the average image luminance and gradient which can be measured directly. A preliminary sensitivity analysis for α, β, τ_H and τ_S is reported in [9]. A detailed comparison of this method with others proposed in the literature is reported in [17].

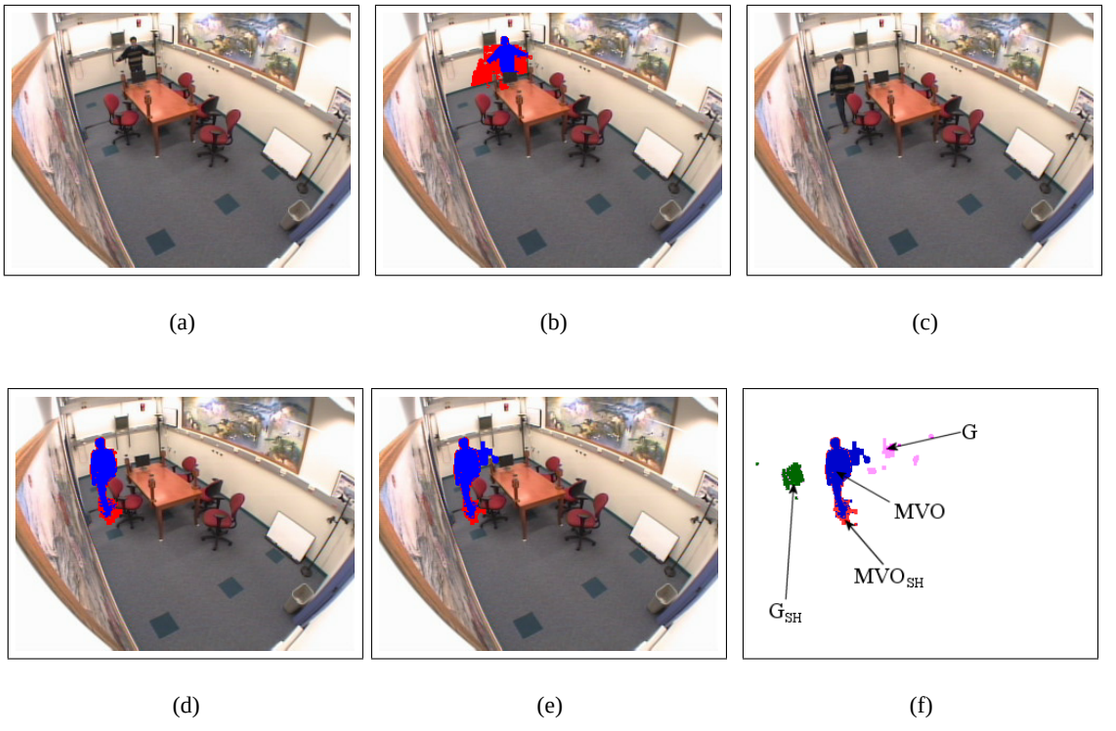


Fig. 3. The effects of shadow classification on the background modeling. Frame #180 is reported in (a) a raw image and in (b) with the detected MVOs and MVO shadows. In frame #230 (c) the detected classes are reported in figure (f). Sakbot is able to correctly segment the image (figure d), while using shadow suppression only the result is incorrect, as reported in figure e.

Figg. 3(a) and 3(c) are two frames (#180 and #230) of a video from an indoor scene with distributed light sources creating many shadows. In the scene, a person keeps on moving in the same zone for a while, and the points of his connected shadows occupy always a same area. Fig. 3(b) shows the detected MVO and its connected shadows at frame #180; shadow suppression is evidently needed for achieving precise segmentation of the MVO. Moreover, the use of shadowed areas is essential also for obtaining an accurate

and reactive background modeling. To demonstrate this, Fig. 3(c) shows a later frame of the same sequence (frame #230), where the person moves from the area. Fig. 3(d) shows the correct segmentation achieved with Sakbot, which correctly updates the background (see Eq. 5). Figg. 3(e) and 3(f) show the results achieved without exploiting the shadow classification in the background update process. The MVO's shape is evidently affected by errors, arising as follows: let us suppose that in frame #180 the background is updated using all the shadows; in frame #230, an area previously occupied by shadows is now uncovered, thus creating apparent foreground points; some of them are grouped into isolated blobs, which can be easily classified as ghosts, their average optical flow being null; however, other apparent foreground points connected with the real MVO points are instead included in the MVO segmentation, thus substantially affecting the object's shape. Fig. 3(f) shows the differently classified points. Although difficult to be quantified, the corrupting effects of including shadows in the background modeling update are relevant in real cases.

IV. Results evaluations

In the following, we describe some relevant cases. The first example measures reactivity in a limit condition, when the background reflects changes from a car that starts its motion after having previously been part of the background (a reverse out of a parking lot).

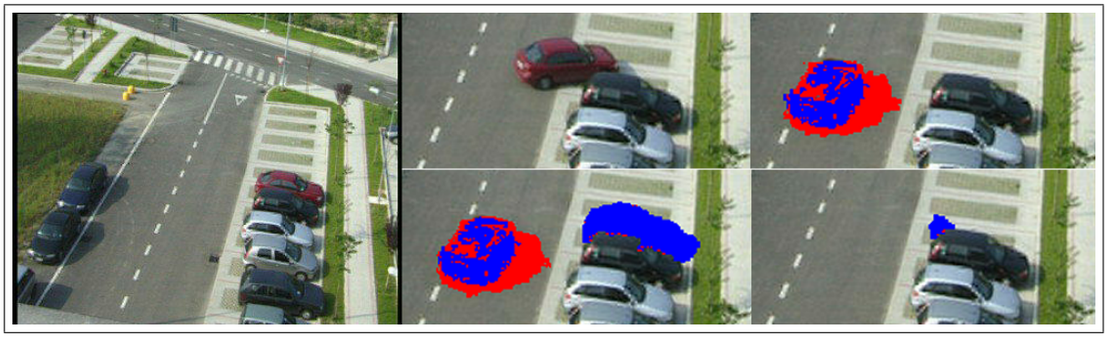


Fig. 4. The reactivity of the background model. First column contains the background model at frame #65; second column contains a zoomed detail of the frame # 100 (upper image) and of detected MVOs (in black) by using pixel selectivity only (lower image): false positives are due to the ghost; third column reports the detected MVO by using Sakbot (upper) and by using statistic background update only at frame #134 (lower)

While the car is parked, it is included in the background image. At frame #65 (Fig.4, first column), it starts reversing. Until about Frame #100 (Fig.4, second column, upper image) the moving object still substantially

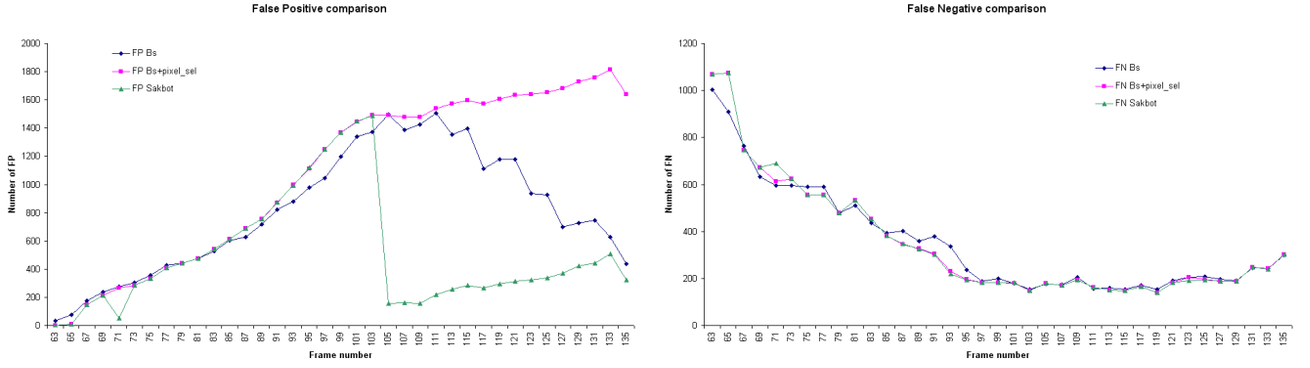
covers the area where it was stopped, preventing separation from its forming ghost. However, after a few frames, the correct background update and the correct segmentation can be achieved with Sakbot (Fig.4, second column, upper image).

This result could not be achieved by using statistics only. As an example, the use of a *statistic background*, using only $\mathbf{B_S}$ in equation 6 (Fig. 4, second column, lower image), almost correctly updates the new background only after about forty frames, even with still considerable errors (the black area). Moreover, results comparable with those of Sakbot cannot be achieved by only adopting selectivity at pixel-level, being the usual approach that excludes from the background update pixels detected as in motion[1][8][2]. In fact, if the value of the detected foreground points is never used to update the background, the background will never be modified; consequently, the ghost will be detected forever (Fig. 4, first column, lower image). If, instead, the value of the detected foreground points is used in the statistic update, but with a limited weight as in [8], the udpate will still be very slow.

This different reactivity is compared in the graphs in Fig. 5, where false negatives (FN) and false positives (FP) are compared against the ground-truth. The three lines show FN and FP results with the same statistical function (Eq.s 2 and 3 with n=7), comparing the statistic background (Bs curve), selectivity at pixel-level (Bs+pix_sel curve) and with the knowledge-based selectivity of Sakbot (Sakbot curve); all the approaches include shadow suppression and classification. The FN curves are similar for all three approaches, since FN accounts for false negatives that are due to incorrect segmentation (mostly some parts of the car window, erroneously classified as shadows). Instead, the FP curves account for false positives, differing much depending on the different background reactivity. The Sakbot curve proves that immediately after an object has moved away from its initial position (Frame #103 in the graph), nearly no ghost points are segmented as MVO points. Starting from frame #105 the FP increase slowly: this is due to the fact that the moving car is turning parallel to the road, increasing its size and that of its shadow; the FP increase proportionally due to the unavoidable imperfections of the shadow detection algorithm. The ghost remains forever instead in the case of pixel selectivity (Bs+pix_sel curve), while it decreases slowly in the case of pure statistics (Bs curve).

In frame #103, the Bs curve still shows a non negligible value of FP; this is due to the still partially erroneous background shown in Fig. 4, second column, lower image.

In Sakbot, high responsiveness to background changes is given by the concurrence of two features, namely the limited number of samples in the statistics and the knowledge-based selectivity at object level. Differently from purely statistical methods, the knowledge-based selectivity allows the system a limited observation window without an erroneous background update. Differently from pixel-level selective methods, the classification of ghosts is more robust and deadlock is avoided.

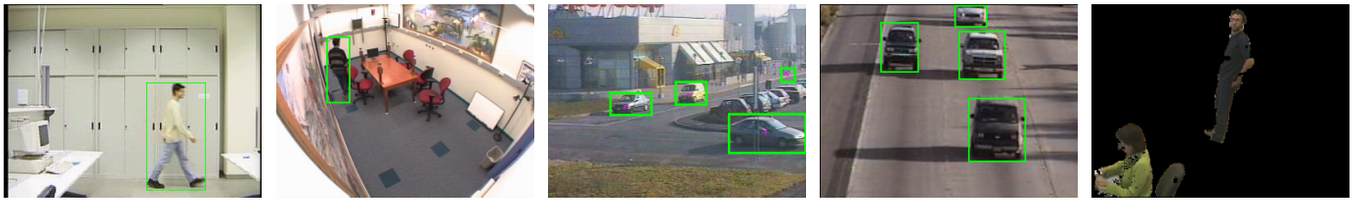


(a) False Positives

(b) False Negatives

Fig. 5. Reactivity comparison between a simply statistical background, a pixel-level selective background and the statistical and knowledge-based background models. Segmentation is provided via background subtraction including shadow detection.

V. CONCLUSIONS



(a) Domotics application

(b) US Intelligent Room

(c) Outdoor surveillance

(d) Traffic Control in US Highways

(e) Video segmentation for transcoding

Fig. 6. Examples of different applications of Sakbot.

This paper has presented Sakbot, a system for moving object detection in image sequences. This system has the unique characteristic of explicitly addressing various troublesome situations such as cast shadows and

ghosts. Cast shadows are detected and removed from the background update function, thus preventing undesired corruption of the background model. Ghosts are also explicitly modeled and detected so as to avoid a further cause of undesired background modification. Actually, in scenes where objects are in constant motion (i.e., no ghosts are present), any common background suppression algorithm already performs effectively. However, if the dynamics of the scene is more complex, with objects stopping and starting their motion, standard techniques would suffer from significant errors, due to the absence of the object-level knowledge which is instead accounted for in our proposal. With Sakbot, when an object starts to move, only initially it will be connected to its ghost. This connected object-ghost blob can be either globally accepted as a true object or rejected based on its AOF. In either case, this will cause inevitable transient errors in the background model. However, as soon as the ghost separates from the actual object, it will be quickly classified as ghost object and, unlike any other common approaches, the background will recover immediately. This will significantly reduce the impact of ghost errors in highly dynamic scenes such as dense urban traffic scenes with mixed vehicles and people. The approach proved fast, flexible and precise in terms of both shape accuracy and reactivity to background changes. These results are mainly due to the integration of some form of object-level knowledge into a statistical background model.

The Sakbot system has been tested in a wide range of different environments and applications. Fig. 6 shows some of these applications: domotics, intelligent rooms, outdoor surveillance, traffic control on US highways, and object segmentation in video for semantic transcoding [19]. Sakbot resulted to be a general-purpose approach that can be easily tuned for various contexts. This approach was intentionally designed to be completely independent of the tracking step in order to retain maximum flexibility. Since tracking is often application-dependent, this might jeopardize the generality of this moving object detection approach. Actually, if feedback from the tracking level to the object detection level could be exploited, it is likely that the object classification could be improved by verification of temporal consistency.

Finally, the method is highly computationally cost-effective since it is not severe in computational time (excluding the computation of approximate optical flow equation, which is however limited to the pixels of

foreground blobs only). Unlike other background subtraction methods which compute multiple and more complex background statistics at a time, the very simple median operator requires very limited computation. This approach consequently allows fast detection of moving objects, which for many applications is performed in real time even on common PCs; this in turn allows successive higher-level tasks such as tracking and classification to be easily performed in real time. For example, we analyzed the time performance with a Pentium 4 1.5 GHz for a video with 320 x 240 frames (24 bits per pixel) in which the Sakbot system is able to obtain an average frame rate of 9.82 fps with Δt=10, and reaches 10.98 fps (performance obtained on the video of Fig. 6(a)).

ACKNOWLEDGEMENTS

This work is partially supported by the Italian M.I.U.R., Project “High Performance Web Server”.

REFERENCES

[1] I. Haritaoglu, D. Harwood, and L.S. Davis, “W4: real-time surveillance of people and their activities,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[2] N. Amamoto and A. Fujii, “Detecting obstructions and tracking moving objects by image processing technique,” *Electronics and Communications in Japan, Part 3*, vol. 82, no. 11, pp. 28–37, 1999.

[3] C. Stauffer and W.E.L. Grimson, “Learning patterns of activity using real-time tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[4] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, “Tracking groups of people,” *Computer Vision and Image Understanding*, vol. 80, no. 1, pp. 42–56, Oct. 2000.

[5] C. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, “Pfinder: real-time tracking of the human body,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, July 1997.

[6] M. Seki, H. Fujiwara, and K. Sumi, “A robust background subtraction method for changing background,” in *Proceedings of IEEE Workshop on Applications of Computer Vision*, 2000, pp. 207–213.

[7] N. Ohta, “A statistical approach to background suppression for surveillance systems,” in *Proceedings of IEEE Int’l Conference on Computer Vision*, 2001, pp. 481–486.

[8] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russel, “Towards Robust Automatic Traffic Scene Analysis in Real-Time,” in *Proceedings of Int’l Conference on Pattern Recognition*, 1994, pp. 126–131.

[9] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, “Improving shadow suppression in moving object detection with HSV color information,” in *Proceedings of IEEE Int’l Conference on Intelligent Transportation Systems*, Aug. 2001, pp. 334–339.

[10] A. Elgammal, D. Harwood, and L.S. Davis, “Non-parametric Model for Background Subtraction,” in *Proceedings of IEEE ICCV’99 FRAME-RATE Workshop*, 1999.

[11] B.P.L. Lo and S.A. Velastin, “Automatic congestion detection system for underground platforms,” in *Proceedings of the Int’l Symposium on Intelligent Multimedia, Video and Speech Processing*, 2000, pp. 158–161.

[12] B. Gloyer, H.K. Aghajan, K.Y. Siu, and T. Kailath, “Video-based freeway monitoring system using recursive vehicle tracking,” in *Proceedings of SPIE Symposium on Electronic Imaging: Image and Video Processing*, 1995.

[13] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld, “Detection and location of people in video images using adaptive fusion of color and edge information,” in *Proceedings of Int’l Conference on Pattern Recognition*, 2000, pp. 627–630.

[14] C. Stauffer and W.E.L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proceedings of IEEE Int’l Conference on Computer Vision and Pattern Recognition*, 1999, pp. 246–252.

[15] N.M. Oliver, B. Rosario, and A.P. Pentland, “A bayesian computer vision system for modeling human interactions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831–843, Aug. 2000.

[16] N. Rota and M. Thonnat, “Video sequence interpretation for visual surveillance,” in *Proceedings of IEEE Workshop on Visual Surveillance (VS ’00)*, 2000, pp. 325–332.

[17] A. Prati, R. Cucchiara, I. Mikic, and M.M. Trivedi, “Analysis and Detection of Shadows in Video Streams: A Comparative Evaluation,” in *Proceedings of IEEE Int’l Conference on Computer Vision and Pattern Recognition*, 2001.

[18] A. Bainbridge-Smith and R.G. Lane, “Determining optical flow using a differential method,” *Image and Vision Computing*, vol. 15, pp. 11–22, 1997.

[19] R. Cucchiara, C. Grana, and A. Prati, “Semantic Transcoding for Live Video Server,” in *Proceedings of ACM Multimedia 2002 Conference*, Dec. 2002, pp. 223–226.