Detecting Moving Objects and their Shadows: an evaluation with the PETS2002 dataset

Rita Cucchiara, Costantino Grana, Andrea Prati
Dipartimento di Ingegneria dell'Informazione
Università di Modena e Reggio Emilia
Modena, Italy, 41100

Abstract

This work presents a general-purpose method for moving visual object segmentation in videos and discusses results attained on sequences of PETS2002 datasets. The proposed approach, called Sakbot, exploits color and motion information to detect objects, shadows and ghosts, i.e. foreground objects with apparent motion. The method is based on background suppression in the color space. The main peculiarity of the approach is the exploitation of motion and shadow information to selectively update the background, improving the statistical background model with the knowledge of detected objects. The approach is able to detect Moving Visual Objects (MVOs), and stopped objects too, since the motion status is maintained at the level of tracking module. HSV color space is exploited for shadow detection in order to enhance both segmentation and background update. Time measures and precision performance analysis in tracking and counting people is provided for surveillance and monitoring purposes.

1. Introduction

A robust tracking of objects in video streams requires a 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 to different scenarios (indoor, outdoor) or to different light conditions; and efficiency, in order to have an high frame rate. In particular, 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 more rapidly pruned). In addition, if object classification is required by the application, precise detection substantially supports correct classification.

Therefore, this work addresses the problem of an accurate moving visual object detection for people tracking, dealing with some very general scenarios as:

- *Unknown objects* whose speed and trajectory is a priori unknown;
- *Unknown background* possibly changing due to two factors: a) light condition variations; b) objects that modify their status from stopped to moving or vice versa. If the background model is neither accurate nor reactive, background suppression could cause the detection of false objects, here referred to as "ghost" objects.
- *Unknown illumination* causing shadows whose direction, shape and strength are unknown.

In the absence of any a priori knowledge about target and environment, the most widely adopted approach for moving object detection is based on *background suppression* [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Approaches based on frame difference only [13, 14] should calibrate difference in dependence on object speed and are not suitable when the model of object's motion is unknown.

The object segmentation by background suppression with unknown illumination is affected by the problem of shadows [15, 16, 17] since objects and cast shadows share the most important feature (i.e. motion) and are often spatially connected. Often the points in motion of both objects and their shadows are merged together and the appearance and geometrical properties of the object are distorted. Moreover the probability of object's under-segmentation (where more objects are detected as a single one) increases due to connectivity via shadows between different objects. In [15] we proposed an approach using color appearance to detect shadows and compared it with other methods.

In this paper, we report results of PETS2002 tests using our approach called Sakbot (Statistical And Knowledge-Based Object deTector). The main feature of the approach is the integration of the knowledge of detected and classified objects, shadows and ghosts in the segmentation process, both to enhance segmentation and to improve future detection and background update. The approach we propose fully exploits both motion and color information to detect and classify foreground objects. In [18, 19] we described the approach and some performance measures.

Sakbot describes detected moving objects by means of features such as geometrical measures (area, perimeter, ...) texture (the colors and the gray-level pattern), the spatial position (centroid and extent) and the motion status (moving, still, stopped). These features could be further exploited in a more or less sophisticated tracking module. In [14] we have proposed a tracking module managing visual data at a symbolic level. It is a production rule system with forward chaining, formalizing the environment knowledge and the relationships between the extracted objects. Working at a symbolic level allows the system with flexible object tracking over a variety of applications by simply devising adequate rule sets. When the application is simpler and the environment is constrained enough, the tracking module could be substituted by a simpler process which only keeps the history of detected objects, their position and area. In this work we do not present a high-performance tracking module: therefore, although we provide complete results of tracking over the PETS2002 sequence, we aim at focusing on the accuracy of the low-level only. Main features are: i) the reactivity of background that in the sequence is useful to eliminate reflection of people on the shop window, ii) the improvements due to the use of color and shadow detection and iii) the good time performance. Since the processing time is not so high (and could also be improved by optimizing the code), there is a large space for adding a tracking system with improved performance, if it is required by the application.

2. Related work

Most of the proposals adopt background suppression and update by using *statistical* functions on a sequence of the most recent sampled frames. A very simple but very effective statistic is that proposed in [1]: as a background model, the minimum and maximum values and the median of inter-frame differences of the pixels' intensities are used. In [2], the *mode* is used, under the hypothesis that background values should be the most frequent over time. Pfinder [3] exploits the assumption that the distribution of background values is Guassian and therefore each pixel is modeled with a Guassian. Since the distribution is actually not unimodal in many real situations, models based on a mixture of Gaussians have been proposed [4, 5]. However, the use of multiple functions may affect video-rate performance, since the computational load increases with the number of combined functions. If a single function is used, instead, the background can be straightforwardly updated by linearly combining the current image and the previous background [3, 6]; it is often referred to as adaptive background update. Other proposals make use of more complex models, exploiting multi-variated Gaussians with PCA [7] or maximum likelihood estimator [8]. In our work, we use a median function in place of the mode, the Gaussian, or some higher-order statistics. Although the median filter might be less sensitive in detecting low contrast objects than other more complex statistics such those proposed in [4, 5], it is much less computationally expensive thus easing real-time execution. Moreover, with respect to other methods that use the median operator [9, 10, 11, 12], our method exploits adaptivity, since the median also considers the previous background with an adequate weight.

Since moving objects are not part of the background, their inclusion in the background update function leads to errors. Thus some methods propose a *selective* background update to exclude those pixels detected as moving points. However, the use of selectivity can carry further problems, when objects originally motionless in the background scene, start their motion. When an object starts moving, an apparent object called ghost (some times called "negative" as in [10]) appears in the position where the object was located, due to the difference between the current image and the old background value. If the ghost's area is excluded from the background update, the background will never be correctly estimated causing deadlock [4]. In Sakbot, we propose to perform this verification on the whole object containing the pixel, since the information on the whole object is supposed to be more reliable. Thus we improve the statistical background model by exploiting the knowledge of previously segmented objects.

Another major aspect of a background suppression approach is how to actually remove the background from the current image. The simplest approach is to threshold the difference between the current image and the background model with a fixed threshold. In multiple valued background approaches, the most probable background is subtracted from the image [5] and the difference is thresholded. In [4] probability is used over N Gaussian distributions. In [6], this difference is computed separately for the three color components. Similarly to [6] and [12], here the difference is computed separately for the three color components, and the maximum difference retained; however, two thresholds (low and high) are used, and thresholding is performed with hysteresis; this approach provides good detection results as will be shown in the following.

Lastly, a peculiar aspect of many proposals is the coping with the shadow problem. See [15] for a comparison of some methods. In [16], the authors propose to compute the ratio of the luminance between the current frame and the previous frame; a point is marked as shadow if the local variance (in a neighborhood of the point) of this ratio is small (this criterion is then followed by further validation). In [4], too, the ratio of the luminance of the current frame and the background model is used. An improvement of this method is proposed in [17] based on the observation that shadows are semitransparent, retaining features of the covered surface such as patterns, color, textures; therefore, the authors propose an analysis of the chromaticity in the R,G,B color space. In Sakbot shadow detection is performed on the H,S,V color space, which is more similar to human perception of colors.

3. The Sakbot approach

Sakbot works on color frames: at each time t, it uses the color frame I^t and the current background model B^t. A sequence of steps is computed to extract, classify and track the set of *known objects* KO^t associated to the frame t. The peculiar features of these steps are here outlined (for details see [18, 19]):

a) Background suppression: it is performed with a two-level threshold on the distance in the RGB color space between current frame and background; it is defined for each p image point as[1]:

$$\begin{aligned} DB^t(p) &= Distance(I^t(p), B^t(p)) = \\ &= max(|I^t(p).c - B^t(p).c|)\ , c = R, G, B \end{aligned} \quad (1)$$

A coarse grain foreground point selection is made by thresholding the DB^t image with a low threshold T_L. Among the selected points, some are discarded as noise applying some morphological opening operators.

b) Shadow detection: we do not provide shadow suppression but shadow detection, since shadows are not simply discarded but are labeled and then used in background update. Shadow detection uses the HSV color space [15]. A point is classified as shadow point by the following mask:

$$SP^t(p) = \begin{cases} 1 & if \quad \alpha \leq \frac{I^t(p).V}{B^t(p).V} \leq \beta \\ & \wedge |I^t(p).S - B^t(p).S| \leq \tau_S \\ & \wedge Diff_H^t \leq \tau_H \\ 0 & otherwise \end{cases} \quad (2)$$

where:

$$\begin{aligned} Diff_H^t(p) = min(\ &|I^t(p).H - B^t(p).H|, \\ &360 - |I^t(p).H - B^t(p).H|\ ) \end{aligned} \quad (3)$$

being H an angular value.

The rationale is that the ratio between the pixel's Value component in the current frame and in the background model must be less than one. In fact, a cast shadow darkens the background point, while an object point might darken it or not, depending on the object's color texture; the lower the ratio, the larger is the darkening effect. We approximate the luminance with the value of the V component in HSV space of point p in frame t. Then, if a shadow is cast on a background, the H (hue) component changes within a certain limit, as assumed by the threshold on the expression of Eq. 2. In addition, we introduce the use of the saturation component, which also was proven experimentally to change within a certain limit, as is indicated in Eq. 2. In Fig. 1 (bottom-right window) shadows are shown in white. The dark gray parts (e.g., the face of the man identified by the id #37) are foreground object that only the color analysis prevents from being misclassified as shadow.

c) Foreground blob computation: A region-based labeling computes connected blobs of candidate moving points (by means of the 8-connectivity) and shadows (with a distinguishing shadow label). To each blobs, some image analysis operators associate selected visual features (such as area, perimeters, texture, colors, ...). Moreover also an intrinsic object speed is computed by means of the average optical flow, aOF. This feature is not evaluated for shadow's blobs, where optical flow cannot be correctly computed due to the low contrast between shadows and background.

d) Object validation and classification: Using the blob's features we can validate a blob as an actual Moving Visual Object (MVO), distinguishing it from other classes. The taxonomy divides blobs into subset of Known Objects at instant time t (KO^t) as follows:

$$\begin{aligned} \mathbf{KO^t} = &\{MVO^t\} \cup \{MVO\ shadow^t\} \\ &\cup \{ghost^t\} \cup \{ghost\ shadow^t\} \end{aligned} \quad (4)$$

A candidate MVO must have an area large enough (w.r.t. a threshold) and be salient enough (with at least one point with a high $Distance$ of Eq. 1). Moreover is validated by a sufficiently high average optical flow that asserts its motion (see rules in Fig. 2). The bottom-left window in Fig. 1 shows MVOs only.

If the object has not a significant aOF, it could be due either to an error in the background model, i.e. a ghost, or to a stopped object. The *Match()* function verifies whether its shape matches a MVO detected at the previous frame or not. This is the only rule that has been included for tracking since needs a history list of detected objects. The man identified with id #9 in Fig. 3(a) is currently still and is not classified as MVO but as StoppedMVO (note the blue box around it in the top-left image and that it is not reported in the bottom-left). Actually the *Match()* function takes into account both MVOs and StoppedMVOs of previous frames, with a *Timeout* flag; thus if an object is detected as stopped for a time greater than a timeout, it is immediately included in the background.

Finally, shadows are classified as belonging to an MVO if they are connected to a MVO or a Stopped MVO, whereas are considered ghost shadows if they can not be associated to any real MVO.

[1]Working in a vector space (either RGB or HSV), the notation $X.y$ means the y component of the X vector.

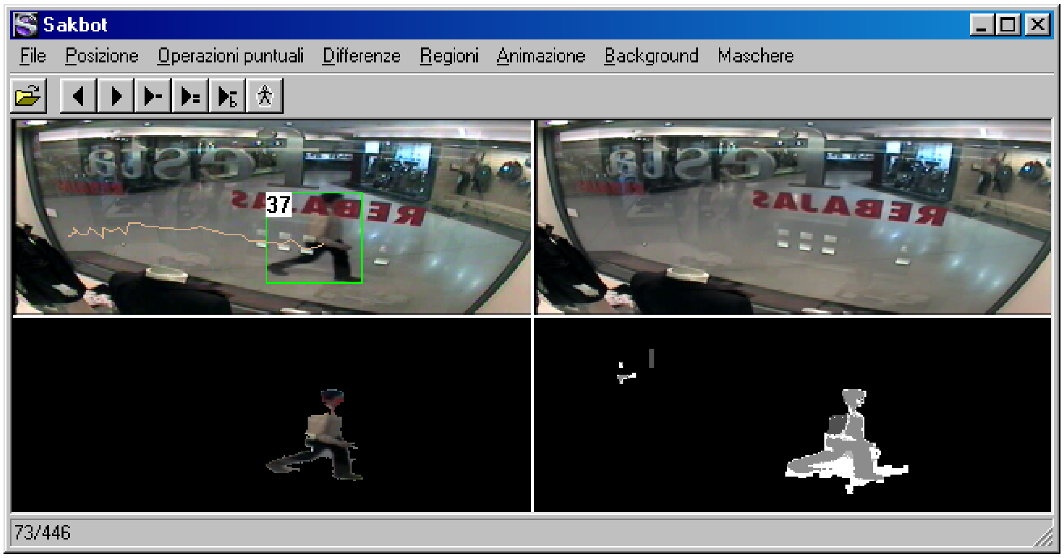


Figure 1: Interface of the Sakbot system

$<candidateVO^t>$	$\leftarrow$	$(ForegroundBlob^t) \wedge \neg(Shadow^t) \wedge (LargeArea^t) \wedge (HighSaliency^t)$
$<MVO^t>$	$\leftarrow$	$<candidateVO^t> \wedge(HighAverageOpticalFlow^t)$
$<StoppedMVO^t>$	$\leftarrow$	$<candidateVO^t> \wedge\neg(HighAverageOpticalFlow^t) \wedge Match(<MVO^{t-1}>)$
$<Ghost^t>$	$\leftarrow$	$<candidateVO^t> \wedge\neg(HighAverageOpticalFlow^t) \wedge \neg Match(<MVO^{t-1}>)$
$<MVOShadow^t>$	$\leftarrow$	$(ForegroundBlob^t) \wedge (Shadow^t) \wedge ConnectedWith(<MVO^t>, <StoppedMVO^t>)$
$<GhostShadow^t>$	$\leftarrow$	$(ForegroundBlob^t) \wedge (Shadow^t) \wedge \neg ConnectedWith(<MVO^t>, <StoppedMVO^t>)$

Figure 2: Classifying rules

e) *MVO Tracking*: we added a simple tracking module that keeps track of the detected MVOs and StoppedMVOs, correlates their extent, centroid position and expected position in order to follow their motion during the frames. If no error occurs, the MVO maintains the same identifier (the process accepts to lose them for a limited number of frames) and the centroid position can be tracked, as in the top-left window of Fig. 1. In Fig. 3 the green and blue extents indicate moving objects and stopped objects, respectively.

f) *Background update*: this part is the key strength of Sakbot and has been defined in order to cope with background changing in a very reactive way. This feature is not exploited in substantially static scenes as the ones of PETS2002 datasets, but it has been used in outdoor surveillance when background changes often. Our approach combines three issues:

- the *statistical* combination of a number of sampled frames, which implies the exploitation of the information on the history of a pixel in a finite time window;

- the *adaptability* of the model to slow changes in the scene, keeping the knowledge of previous backgrounds;

- the *knowledge-based selectivity* to improve the accuracy and to relax the constraint on the minimum window width.

Background is updated as follows:

$$B^{t+1}(p) = \begin{cases} Bk^t(p) & if\ \ p \in O, O\ in\ \{KO^t\} \\ Bs^{t+1}(p) & otherwise \end{cases} \tag{5}$$

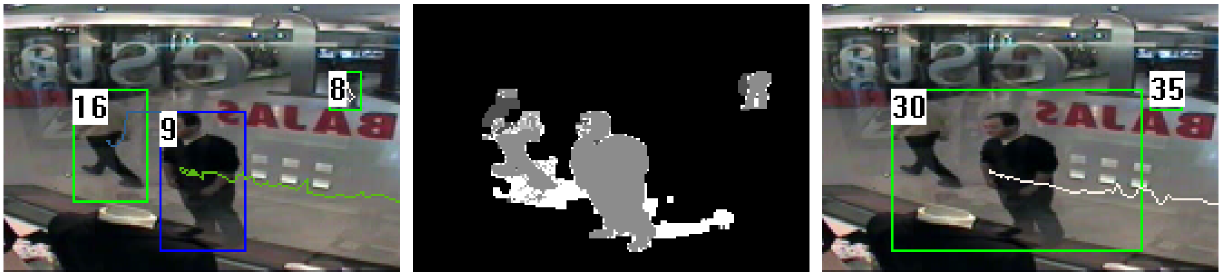
If a point p does not belong to any detected object, background in p is computed statistically: Bs is calculated as a *median* over an history set of previous frames, with a factor of adaptativity that takes the current background into account with a certain weight w_b (i.e. repeated w_b times). Given the history set $H^t = \{I^{t-\Delta t}, ..., I^{t-n\Delta t}\}$ of n samples over an observation window of $W = n\Delta t$, at the update time, the history set and the background become

$$\begin{aligned} H^{t+1} &= H^t \cup \{I^t\} - \{I^{t-n\Delta t}\} \\ Bs^{t+1} &= Median(H^{t+1} \cup w_b[B^t]) \end{aligned} \tag{6}$$

The median is computed in the RGB color space with the approximated distance of Eq. 1. The $Median(x_1, ..., x_k)$ returns x_i so that:

$$x_i = arg\ min \sum_{j=1}^{k} Distance(x_i, x_j) \tag{7}$$

In the experiments we use n=7, w_b=2 and Δt ranging between 5 and 30.



(a) (b) (c)

Figure 3: MVOs detected with the complete approach (a) (with the corresponding shadow detection results reported in (b)) and without shadow detection and the selectivity (c).

Conversely, when we know something about the object in the scene we update the background selectively. Only for those points belonging to an MVO (or a StoppedMVO) or its shadow, we adopt another background model, Bk, defined as:

$$Bk^{t+1}(p) = \begin{cases} B^t(p) & if \;\; p \in O, O \; in \; \{MVO^t\} \cup \\ & \{MVOshadow^t\} \vee \\ & O \; in \; \{StoppedMVO^t\} \wedge \\ & \neg(Timeout(O)) \\ I^{t+1}(p) & if \;\; p \in O, \\ & O \; in \; \{StoppedMVO^t\} \wedge \\ & \neg(Timeout(O)) \\ Bs^t(p) & if \;\; p \in O, O \; in \; \{Ghost^t\} \cup \\ & \{GhostShadow^t\} \end{cases} \tag{8}$$

If a point belongs to an object classified as a MVO or a StoppedMVO, the estimate of background in p does not change (B^t is not updated); instead if the StoppedMVO is in the still status for a time greater than a *Timeout* its color value is inserted in the background. If the point belongs to a ghost or a ghost shadow the statistical function is used.

Note that we use in different way the detected shadows: if a shadow is associated with a moving object, its pixel's value does not update the background. This model allows a very reactive and precise background, and, as a consequence, a very precise object segmentation at pixel level. For performance evaluation with pixel level ground-truths see [18].

4. Tests on PETS 2002

PETS2002 tests have been specifically defined with difficulties and artifacts that make tracking difficult. Sakbot is not able to overcome most of these obstacles since a sophisticated tracking module has not been added. We are unable, for instance, to divide groups of people partially overlapped as other proposals do (see e.g. Hydra [20]). These typical errors of under-segmentation are shown in Fig. 4 and Fig. 5. Nevertheless we want reports results of Sakbot since we believe that they could be interesting to suggest a low-level precise object detection and classification method that is an unavoidable and critical first step of tracking and gesture recognition.

In the Dataset1, when few groups of people move together, the system is precise at tracking level too. In the graph of Fig. 4 we indicate the number of moving people detected frame by frame in front of the window. Within the $\{MVO^t\}$ set we selected only those objects whose centroid is in a image zone defined as "in front of the window", while other little objects that Sakbot is able to detect in the higher part of the image (e.g. that identified with id #8 in Fig. 3(a)) are discarded. Sakbot detects for one frame an object that does not exist (is part of a man exiting from the scene), while has a number of under-segmentation errors, due, as above mentioned, to two people partially overlapped. Although this error should be avoided by means of a specific tracking module, the presence of a shadow detection method increases performance considerably in term of precision at pixel level: in Fig. 3(a) three objects are detected: two (labeled #8 and #16 in green) are classified as MVOs, one (#9) as a StoppedMVO. Without shadow suppression the two objects #9 and #16 are merged in a single one as in Fig. 3(c) (label #30). In Fig. 3(c) knowledge-based selectivity is not used and therefore also the error due to a reflex is considered a set of moving points and remains forever as a detected object.

Fig. 5 shows the number of people considered as StoppedMVOs in front of the window. The errors in some frames are due to the inclusion of the stopped person in the area of the moving one. Thus, in Dataset1 the errors are due to group of people only, while no errors due to reflections have been made.

Datasets 2 and 3 are more complicated, since a lot of people walk near each other, therefore our system fails (as it is obvious) lacking a specific module oriented to this problem. In Table 1 the tracking results are reported. In Dataset

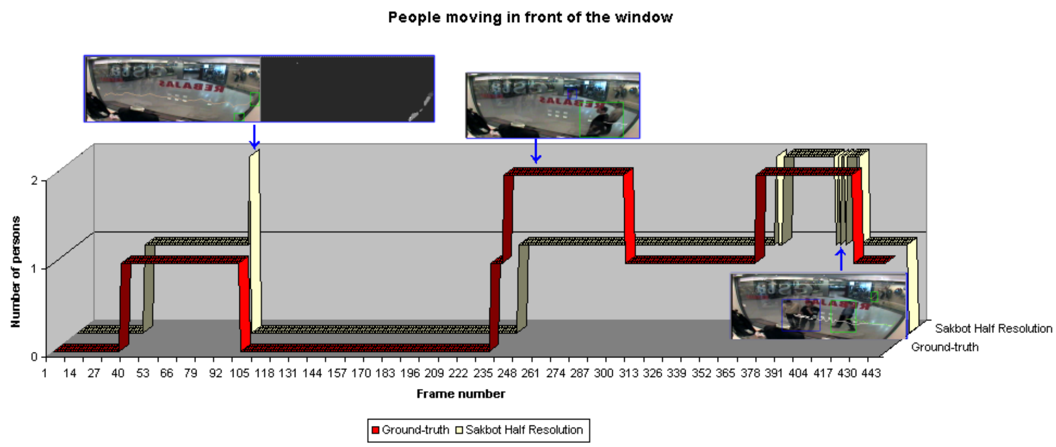


Figure 4: Moving object (MVOs) in Dataset1

Figure 5: Stopped object in Dataset1

3 the higher number of MVOs detected are due to the fact that some objects are over-segmented but, since the system is not able to divide groups some objects change their ids after some frame.

Tests	**People moving (on the window)**	**People stopped (on the window)**	**People moving (total)**
Dataset 1	6 (4)	1 (2)	10 (6)
Dataset 2	10 (7)	3 (3)	17 (12)
Dataset 3	35 (12)	4 (6)	49 (15)

Table 1: Tracking results (in brackets the ground-truth)

In conclusion, we want to summary some key points resulting from these tests:

- Sakbot can detected and classify at each frame MVOs, stopped MVOs, ghost and shadows;

- since Sakbot gives a tuple of visual feature associated with each KO, and assuming that only persons are moving in the videos, we are able to deduce how many people pass near the windows (evaluating the centroid position), and how many stop to watch into the window (frame by frame);

- Sakbot currently does not cope with under-segmentation and over-segmentation problems, that could be solved with many methods that have been proposed in the literature. Then, people are sometimes divided in many parts or more peoples are grouped together;

- the single MVO detection is enhanced by the use of color and is strongly improved by two factors, i.e. the shadow detection and the knowledge-based background update. We provided comparison between the method with and without these improvements;

- lastly, the tuning of a correct *Timeout* for stopping MVO prevents to include them in the background model even if the background is updated frequently. Moreover, exploiting a small ΔT sub-sample rate for background update makes background model more reactive to noise and luminance changes (for instance the reflections due to peoples inside the shops are removed).

5. Execution time

The tests run on a standard Pentium III 800Mhz PC with Window2000 O.S. Programs written in ANSI C++ with Microsoft libraries are compiled without any code optimization. Performances in terms of execution time are underestimated since the measured execution time comprehends the times spent for user interface, data and partial result visualization (see Fig. 1) and the write process of a large amount of logging files. Nevertheless the system achieve good performances, since many frames are evaluated in a second.

Full resolution (640x240)	**Dataset 1**		
Average execution time	S&KB	S&KB w/o shadow	Only statistical w/o shadow
	348.32	340.60	350.50

Table 2: Execution times at full resolution

Half resolution (320x120)	**Dataset 1**	**Dataset 2**	**Dataset 3**
Average exec. time (fps)	70,82 (14,12)	80,32 (12,45)	94,20 (10,62)
Ave. exec. time w/o bkg update	58,88	68,74	83,06
Ave. exec. time only for bkg update	179,68	185,43	195,25

Table 3: Execution times

Sakbot works on full color images. Time measure have been provided working on full size frames (640x240) and on frames with half dimensions (320x120).

The average execution time with full resolution and full color on Dataset1 is reported in Tables 2 and 3 (in msec). We provide three measure: the first refers to the complete Sakbot approach, the second without shadow suppression and the third without shadow and knowledge-based background update.

Shadow suppression is a very important task, unavoidable to have a precise moving object shape but is not highly time consuming. Instead the adoption of knowledge based selectivity is more efficient, as well as more precise.

The execution time is high due to the very costly optical flow computation. If necessary it should be substituted with other more time-consuming motion verification process (e.g. block matching) in order to validate that foreground objects are actual moving objects.

In Table 3 we report also results using half resolution. In these applications of foreground people tracking (people watching the window) the full resolution is useless. Processing full and half resolution bring to the same results in terms of tracking precision.

The time performance are slightly data dependent and in the three datasets from 10 to 14 frame per second are processed. Sakbot has an internal parameter (ΔT) associated to frequency of the background update: in the experiments it is fixed to 10 so that the background is updated each 10 frames. When background is updated the time spent is obviously higher: in Table 3 you can see that an average of 59 ms is used to detect, classify and track moving objects, while 180 ms is spent if background updated is added. This high update frequency is not necessary in this experiment where the background is substantially fixed. We tested that similar results in efficacy are achieved with a $\Delta T = 100$. When ΔT is higher the average execution time (in our case 70 ms) slows down and is closer to the lower bound (58.9 ms).

These are preliminary results that are achieved without any optimization and without any specific tailoring to the dataset.

6. Conclusion and future work

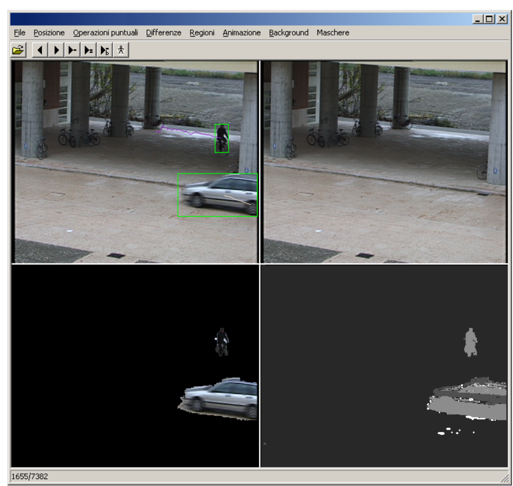


Figure 6: An example of application

Sakbot has proved to be robust in many different environment, such as outdoor traffic scenes in highways or indoor surveillance of our University Campus (Fig. 6). It can be used as a general-purpose approach for object detection in conjunction with many different higher level tracking system that can be tuned to the application.

Future works include its extension to videos taken from moving cameras. It will be used with some suitable modification to surveillance application based on moving PTZ cameras with a defined path. Surveillance via Web is now available in an experimental setup at our Lab . See `http:\\guilderstern.ing.unimo.it`.

Eventually, we intend to use Sakbot also for a general purpose operator for a semantic transcoding of videos both from live cameras or video-on-demand applications, in order to give the user only the information required, tailored with the user' needs or bandwidth requirements.

Acknowledgments

We would like to thanks Prof. Trivedi (UCSD-USA) and Prof. Piccardi (Sidney University of Technology, Australia) for their valuable help in many phases of this work.

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] A. Shio and J. Sklansky, "Segmentation of people in motion," in *Proceedings of IEEE Workshop on Visual Motion*, 1991, pp. 325–332.

[3] 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.

[4] A. Elgammal, D. Harwood, and L.S. Davis, "Non-parametric model for background subtraction," in *Proceedings of IEEE ICCV'99 FRAME-RATE Workshop*, 1999.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] N.T. Siebel and S.J. Maybank, "Real-time tracking of pedestrians and vehicles," in *Proceedings of IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, 2001.

[11] 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.

[12] Q. Zhou and J.K. Aggarwal, "Tracking and classifying moving objects from videos," in *Proceedings of IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, 2001.

[13] Y. Kameda and M. Minoh, "A human motion estimation method using 3-successive video frames," in *Proceedings of International Conference on Virtual Systems and Multimedia*, 1996, pp. 135–140.

[14] R. Cucchiara, M. Piccardi, and P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 119–130, June 2000.

[15] 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, vol. 2, pp. 571–576.

[16] J. Stauder, R. Mech, and J. Ostermann, "Detection of moving cast shadows for object segmentation," *IEEE Transactions on Multimedia*, vol. 1, no. 1, pp. 65–76, Mar. 1999.

[17] T. Horprasert, D. Harwood, and L.S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *Proceedings of IEEE ICCV'99 FRAME-RATE Workshop*, 1999.

[18] R. Cucchiara, C. Grana, G. Neri, M. Piccardi, and A. Prati, *The Sakbot system for moving object detection and tracking*, chapter 12, pp. 145–158, in 'Video-based Surveillance Systems: Computer Vision and Distributed Processing'. Kluwer Academic Publishers, Boston, Massachusetts, USA, Nov. 2001.

[19] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting objects, sahdows and ghosts in video streams by exploiting color and motion information," in *Proceedings of IEEE Int'l Conference on Image Analysis and Processing*, 2001, pp. 360–365.

[20] I. Haritaoglu, D. Harwood, and L.S. Davis, "Hydra: multiple people detection and tracking using silhouettes," in *Proceedings of IEEE Int'l Conference on Image Analysis and Processing*, 1999, pp. 280–285.