2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance

Performance Evaluation of a People Tracking System on PETS2009 Database *

D. Conte, P. Foggia, G. Percannella and M. Vento
Dipartimento di Ingegneria dell'Informazione ed Ingegneria Elettrica
Via Ponte Don Melillo, 1 - 84084 Fisciano (SA) - Italy
{dconte,pfoggia,pergen,mvento}@unisa.it

Abstract

In this paper a system for autonomous video surveillance in relatively unconstrained environments is described.

The system consists of two principal phases: object detection and object tracking. An adaptive background subtraction, together with a set of corrective algorithms, is used to cope with variable lighting, dynamic and articulate scenes, etc. The tracking algorithm is based on a matrix representation of the problem, and is used to face splitting and occlusion problems. When the tracking algorithm fails in following actual object trajectories, an appearance-based module is used to restore object identities.

An experimental evaluation, carried out on the PETS2009 dataset for tracking, shows promising results.

1. Introduction

In the last decade the great advances in electronic and telecommunication technology and the increasing amount of visual material stimulated the scientific interest in the research of automatic video interpretation. Among the most important applications of video analysis we can find: automatic interpretation of multimedia, traffic monitoring, visual surveillance. Visual surveillance is a major research area in computer vision. The recent rapid increase in the number of surveillance cameras has led to a strong demand for automatic methods of processing their outputs. The scientific challenge is to devise and implement automatic systems for obtaining detailed information about the activities and behaviors of people observed by a single camera or by a network of cameras.

As it is often the case with complex systems, the complexity of a video analysis system is usually dealt with by dividing it into loosely coupled layers, although in many proposals the boundaries between layers tend to get blurred. A coarse division that fits the great majority of the proposed methods involves three layers. Going from low level processing to high level, the three layers are:

- *object detection*, whose responsibility is to detect and to segment the moving objects (also called foreground) from the background, looking at a single frame.
- *object tracking*, that is aimed at preserving the identity of an object across a sequence of frames, following the movements and the changes in the appearance (due for example to a change of orientation or posture) of the object itself. Often this problem is faced by dividing into short-term or frame-to-frame tracking, that preserves object identity between adjacent frames, and long-term tracking, built upon the latter, that considers longer sequences to deal with total or partial occlusions or to face the re-identification problem.
- *application event detection*, that uses the results of object tracking to recognize the events that must be handled in some way by the application. This part is obviously highly dependent on the application domain, and can range from a simple processing of the tracking data (e.g. counting the number of objects) to complex classification or learning tasks.

Several methods have been proposed for the object detection layer, but none of them is up to now considered as a definitive solution. There are several criteria according to which the object detection algorithms could be classified. A possible taxonomy of the main algorithms divides them into two approaches:

- *derivative algorithms* ([1, 12, 22]), that work by comparing adjacent frames of the video, under the assumption that foreground objects correspond to rapidly changing areas, while the background is either static or slowly changing;
- *background subtraction algorithms* ([18, 19, 20, 21, 9, 10]), were the current frame of the video is compared with a background model, that is a (usually compact) representation of the set of the possible images observable when the scene does not contain foreground objects.

*This research has been partially supported by A.I.Tech s.r.l., a spin-off society of the University of Salerno.

978-0-7695-4264-5/10 $26.00 © 2010 IEEE
DOI 10.1109/AVSS.2010.87

A major challenge of the tracking problem is the occurrence of occlusions, such those caused by people interacting with each other, or static occlusions caused by background elements lying in front of foreground objects. One solution for managing occlusions utilizes multiple views to compensate for the inadequate visibility of a single view ([17]). However for some applications, multiple views are not always available. A popular solution strategy for detecting and tracking multiple people in surveillance situations is the use of probabilistic appearance models ([15, 6, 23]). These characterize the appearance of a person using probabilistic model and track the targets by localizing the maximum likelihood of the compounding models. The occlusion is solved by divide a single foreground blob in several objects according to each person's appearance model. However, the problem of occlusions is still considered open and there are several authors that propose some improvements to the appearance model strategy to cope with it.

In this paper we will focus our attention on the first two layers (object detection and object tracking), since the third one is application dependent. The remainder of the paper is organized as follows. In section 2 the overall architecture is described. Subsection 2.1 describes the object detection layer while in subsection 2.2 an object tracking layer is proposed. Section 3 presents results achieved on PETS2009 [1] dataset. Finally, in section 4 there are some concluding remarks.

2. System Architecture

The proposed system is sketched out in Fig. 1. Here we briefly present the procedures present into the system; for details of each algorithm, please refer to our recent papers [5, 3, 4].

The Object Detection layer processes the input frames producing semantically separated objects, each included in the smallest enclosing rectangles. Then an object tracking procedure preserves the identity of objects across the frames by assigning them univocal IDs. In this way we obtain the trajectories of every object and after a perspective correction [2] a classification of the behaviors can be done. If some behavior is classified as an interesting event, the system reacts appropriately on the basis of the application context.

2.1. Object Detection

An adaptive background image difference based algorithm [5] has been implemented for detection moving objects. For each pixel if the difference between background and current frame pixel intensity is greater than a threshold the pixel is labeled as foreground. Obviously the threshold is dependent on the scene context, in fact: a low threshold results too noise sensitive, on the other hand a high threshold causes a loss of sensitivity. After the connected components labeling, the objects are identified calculating the smallest rectangles containing every pixels blob. On each blobs some heuristics are then applied to improve the quality of the detection.

In a realistic environment there are some classical problems that affect the performance of an object tracking system (see [20]): camouflage, foreground aperture, light of day, shadows, sleeping persons, waking persons, waving trees.

Unfortunately no one of the plain approaches results able to solve at the same time all these kinds of problems: the cause of this is, essentially, that detectors work at pixel level ignoring high level information. In order to make the system robust in realistic environments we have proposed a set of specialized procedures:

- *Adaptive Threshold*: In the standard algorithms the threshold for the pixel segmentation is statically defined depending on the scene. Evidently in outdoor locations, as the scene conditions are variable, many foreground segmentation errors pop up. Our approach, instead, uses a variable threshold dynamically adapting to the actual scene conditions. The threshold is increased or decreased on the basis of the changes of average scene illumination, measured as the frame average pixels energy.

- *Grouping*: Camouflage is an intrinsic and hardly faceable problem occurring when the pixel characteristics of a foreground object are too similar to the background to be discerned, as happens when a person is wearing clothes having similar colors to the background. The effect is that the difference of these pixels from the background model is under the threshold, and consequently incorrectly considered as background pixels. The errors, consisting in the fragmentation of the actual object in the scene, are detected and corrected by a grouping phase performed on the basis of a model of the shape to be recognized. The algorithm has been devised so as to make it possible the recursive merging of blobs, on the basis of geometrical considerations, so as to allow the possibility of recovering highly critical situations caused by camouflage, as the split of a single objects in a plurality of small parts, otherwise considered as noise (and removed by the filter described below). In Fig. 2 an example of the application of the algorithm is sketched.

- *Noise Filtering*: In the pixel analysis often some conditions cause little isolated background areas to be detected as foreground pixels because they are enough different from reference image. The causes are various

[1] http://www.cvg.rdg.ac.uk/PETS2009/a.html

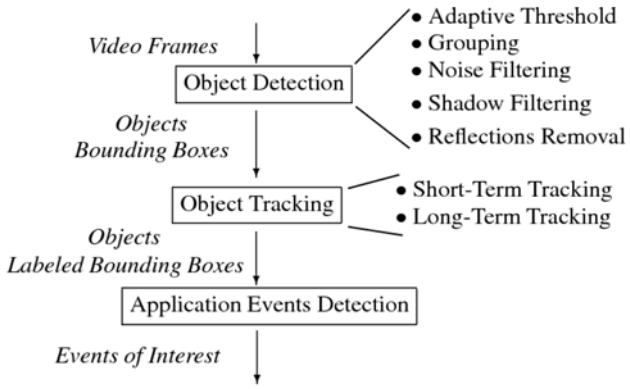

Figure 1. System architecture.

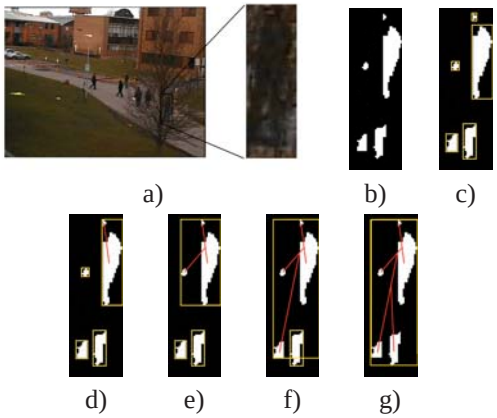

Figure 2. An example of the grouping algorithm's processing: a) original frame and the portion under analysis; b) the resulting foreground detection and the c) resulting bounding boxes; d), e), f), g) the steps of the algorithms on the considered portion of the frame.

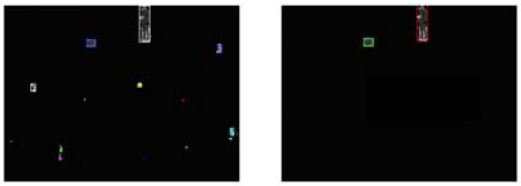

Figure 3. Effects of the noise blob filter.

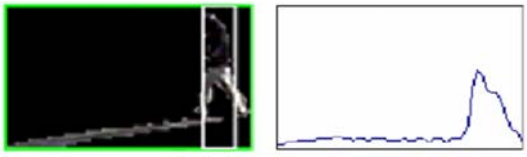

Figure 4. Shadows filtering: the illustration of the basic idea of the algorithm of the shadows filter.

and they cannot be eliminated at pixel level; the results, instead, affect the system performance because of many meaningless blobs. For this reason we have added a filter that operates at blob level to remove the spurious objects. In Fig. 3 the effect of the filter is shown.

- *Shadow Filtering*: Video surveillance systems works 24 hours per day; this means that, because of sun position, variable object shadows are projected on the ground. These shadows may alter the objects dimensions, causing errors in a possible successive object classification based on dimensions or aspect ratio features. Another problem related to shadows is that two semantically distinct objects may be detected as one blob because they results united by the shadow of one of them. We have proposed an algorithm to remove shadows that is based on the column-wise histogram of the foreground pixels (e.g. see Fig. 4).

- *Reflections Removal*: Besides shadows, an object is affected by reflection phenomena. Shadows and reflections differ under several respects; the most important differences are in position and color. The position of a shadow depends on the light sources, while reflections (assuming that the reflecting surface is a horizontal floor) are always located below the corresponding object. As regards the color, a shadow depends only on the color of the background and on the light sources; on the other hand, the color of a reflection also depends on the color of the object. As a consequence of these differences, methods for shadow removal cannot be effectively applied for removing reflections. We have proposed a method for reflection removal [4] that is based on chromatic properties of the reflections and does not require a geometric model of the objects (see Fig. 5).

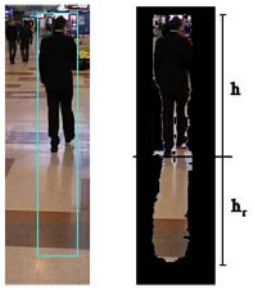

Figure 5. Detection before and after reflection removal.

2.2. Object Tracking

Tracking algorithms are aimed at reconstructing the trajectories of moving objects within a scene.

The simplest tracking approaches are based on the exploitation of spatio-temporal continuity in the motion of physical objects,and can be very effective when the objects of interest are well separated and always visible in the scene. Unfortunately, in many real-world settings the spatio-temporal continuity assumption, while generally valid for most of the time, can be often violated.

Here we present an algorithm that deals with occlusions (two objects merge to form an unique blob) and splits (one object splits in several parts).

Furthermore the algorithm is provided with an annexed module that uses a variation of the appearance model presented in [9] to cope with the failure of the short-term tracking algorithm.

Tracking objects in a video-sequence can be described as follows. If two frames are in succession, it is highly probable that the objects identified in the first frame have a correspondence with the objects identified in the second frame. The tracking is the identification of this correspondence. Corresponding objects can be more or less translated, and the amount of translation depends on both the object speed and the frame rate. Moreover, we can also have objects appearing in only one of the two frames, because they are entering or leaving the scene; furthermore we can have objects that interact with each other forming a unique object.

Formally a tracking algorithm can be described in the following way. We have a set $B^t = \{b_1^t, \cdots, b_n^t\}$ of boxes belonging to the frame t and a set $O^{t-1} = \{o_1^{t-1}, \cdots, o_m^{t-1}\}$ of labeled boxes which identities are determined at the previous frame $t-1$. The problem at hand can be represented by using a matrix M whose rows and columns are respectively used to represent the boxes of the set B^t, and the boxes of the set O^{t-1} (correspondence matrix). Each element M_{ij} of the matrix M represents a similarity measure between the box b_i^t and the labeled box o_j^{t-1}. Many methods have been proposed (e.g. [6, 8, 16, 13]) to build categories of similarity measures that are suitable for the object-tracking problem. It is possible to identify at least 3 categories of similarity measures: position, shape and visual. The position similarity measures consider the similarity between boxes according to their relative distance. Shape similarity measures consider the similarity of box shapes (e.g. dimension, aspect ratio), independently from their location in the frame. Visual similarity measures consider two boxes similar if the corresponding image regions look close from the perceptual point of view (e.g. with respect to brightness and color).

The solution of a simple tracking algorithm, that does not deal with occlusions and splits, is the best matching between these set of boxes in a global sense, i.e. the matching is generated with the constraint that the sum of similarities for the matched pairs is maximized.

Here we present a solution to the tracking problem, based on the similarity matrix, that deals with occlusions and splits.

Given a threshold for the similarity measure between boxes, on the similarity matrix we can have 5 different conditions:

- there are no similarity measures over the threshold in correspondence of the i-th row of the matrix (see Fig. 6a). In this case b_i^t is a new object and takes a new unique label.
- in correspondence of the i-th row there is only one element at column j over the threshold; furthermore this element is the only one over the threshold of the j-th column of the matrix (see Fig. 6b). In this case there is a matching between box b_i^t and labeled box o_j^{t-1}; b_i^t takes the label of o_j^{t-1}.
- there is a set $I = \{i_1, \cdots, i_k\}$, with $k \geq 2$, of rows that have only one element over the threshold in the same column j; the other elements of the j-th column are under the threshold. This is the case of a split (see Fig. 6c): an object on the frame $t-1$ splits in two ore more objects on the frame t. In this case a procedure to resolve the split is applied: the boxes on the frame t are merged in a unique box and the latter takes the label of o_j^{t-1}.
- on the i-th row there are $k \geq 2$ elements over the threshold in correspondence of the set $J = \{j_1, \cdots, j_k\}$ of columns; each column of the set J has only one element in correspondence to the i-th row, over the threshold. This is the case when an occlusions occurs: two ore more objects on the frame $t-1$ merge into a one object on the frame t. In this case a procedure to resolve the occlusion is applied. The unique box on the frame t is split in the following way:

the position of the labeled boxes on the frame $t-1$ is estimated on the frame t in accordance of their motion vectors; the unique box on the frame t is split by overlapping on it the estimated position of the labeled boxes on the frame $t-1$ (see Fig. 6d).

- there is a submatrix, composed of rows $\{i_1, \cdots, i_n\}$ and columns $\{j_1, \cdots, j_k\}$, such as each row and each column contains at least two elements above the threshold within the submatrix. This happens when both split and merge occur. In this case, the procedure to solve the split is applied first, then the procedure to solve occlusions is applied on the merged box.

Obviously the above described tracking algorithm can fail when during an occlusion the objects belonging to it change their directions.

This condition can be recognized because the estimated bounding box on frame t does not correspond to an actual foreground blob (see Fig. 7).

When the algorithm recognizes the presence of a failure during an occlusion a further procedure is applied. This is based on the use of an appearance model [9] of the object and it is aimed at re-estabilishing the identity of the objects after the occlusion. The appearance model definition is extended with respect to [9], in order to have a separate visual aspect matrix and a separate frequency mask for each orientation of the object. We define it as *MultiView Appearance Model* (MVAM). It has to be noted that the orientations are quantized: in particular, we chose 8 equidistant $(45°)$ directions. The MVAM is built and updated by selecting the proper view on the basis of the motion vector of the object. Furthermore, we have added to the model a size normalization step that is able to counter the scaling effect due to perspective, and so the resulting system is better suited to work in scenes with a very deep field of view. The object resizing is done by a bilinear interpolation [14]. Finally, as in [16], we use an appearance model defined on three color channels.

The idea is to create a model for each object being tracked and use it, by comparing the model of the detected object to be identified with the models of the objects detected in previous frames, in order to detect the actual ID of the object exited from an occlusion. The model is initialized with the appearance of the object the first time it enters the scene, and it is updated by averaging it with the appearance seen in the subsequent frames.

The appearance model is described through two matrices of pixels (for each direction): the first matrix takes into account the visual aspect of the person, while the second one stores the number of times each pixel has been considered as foreground in the previous frames. More precisely, the visual aspect matrix represents the average luminosity of each pixel starting from the first appearance of the object in the scene, while the frequency mask is used to attribute a stability measure to each element of the first matrix.

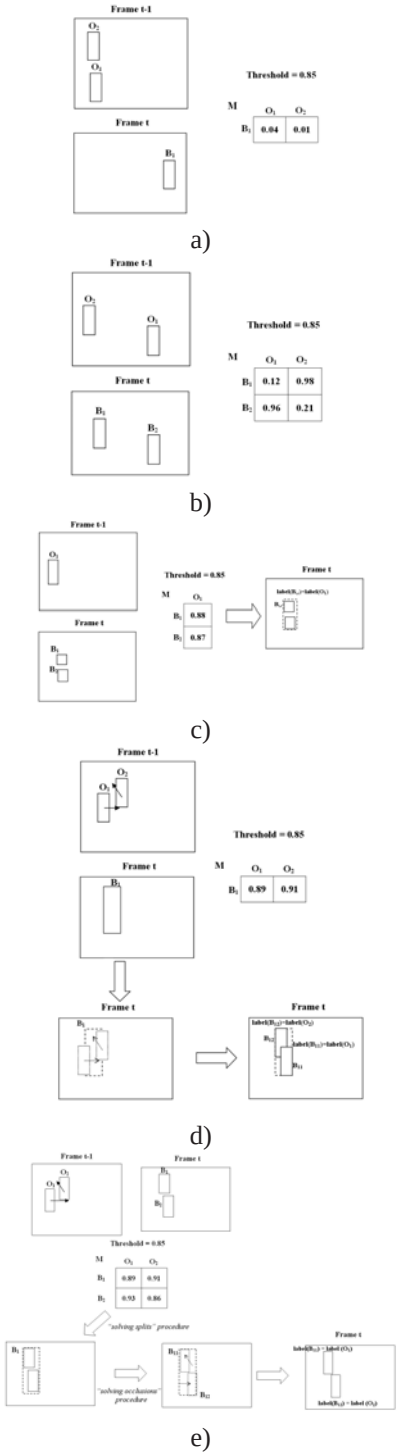

Figure 6. The tracking algorithm: a), b), c), d) and e) show the 5 possible configurations of the matrix similarity and the applied tracking procedures.

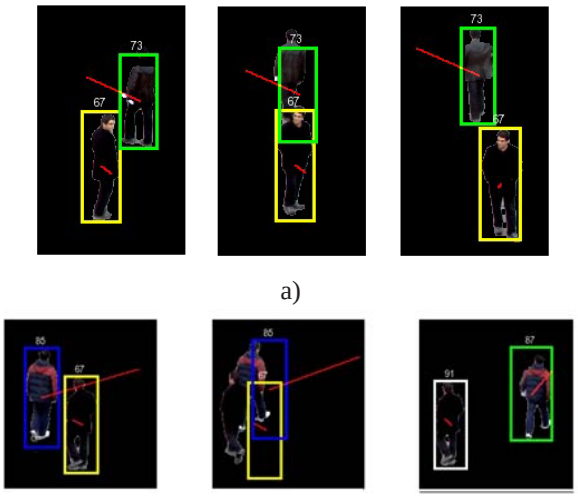

Figure 7. An example of the tracking algorithm: a) a success case and b) a failure case due to a rapid change of direction in the walking of per.

More formally, the appearance model for a direction d can be expressed in terms of a function $\psi_d^t(x,y)$ that associates a luminosity value to the blob pixel having coordinates (x,y) at time t, and a function $\omega_d^t(x,y)$ that represents the corresponding frequency mask:

$$\psi_{c,d}^t : (x,y) \in (N \times N) \rightarrow \psi^t(x,y) \in [0,255] \qquad (1)$$

$$\omega_{c,d}^t : (x,y) \in (N \times N) \rightarrow \omega^t(x,y) \in N \qquad (2)$$

with $d \in \{0°, 45°, \ldots, 315°\}$ and $c \in \{R,G,B\}$.

For each new frame the appearance model of the objects already present in the scene is updated according to the following procedure:

- a view, between the defined views of the object's appearance model, is chosen according to its motion vector;
- a registration is performed between the blob at current frame and the current visual aspect of the considered appearance model; i.e. the blob at current frame is translated so that it have the best correspondence with the visual aspect of the appearance model;
- the visual aspect is updated according to the following formula:

$$\psi_{c,d}^t(x,y) = \frac{I_{c,d}^t(x,y) + \omega_d^{t-1}(x,y) \times \psi_{c,d}^{t-1}(x,y)}{\omega_d^{t-1}(x,y) + 1} \quad \text{with } c \in \{R,G,B\} \text{ and } d \in \{0°, 45°, \ldots, 315°\} \qquad (3)$$

$$D_c(p,m_d) = \alpha \cdot \sum_{(x,y)\in\psi_{c,d}^t \cap B_p^t} \Delta C(x,y) + \beta \cdot \frac{\sum_{(x,y)\in\psi_{c,d}^t - B_p^t} \omega_d^t(x,y)}{\sum_{(x,y)\in\psi_{c,d}^t} \omega_d^t(x,y)} + \gamma \cdot \frac{\sum_{(x,y)\in B_p^t - \psi_{c,d}^t} 1}{area(B_p^t)} \qquad (4)$$

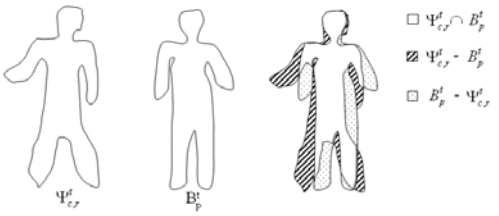

Figure 8. The parts of the blob and of the model that are considered in the calculation of their distance.

where $I_{c,d}^t(x,y)$ is the luminosity of the pixel (x,y) in the current frame (at time t) for each color and for the chosen direction d. It is worth noting that the coordinate spaces for the functions $\psi_{c,d}^t$ and $I_{c,d}^t$ are the same due to the translation of the current blob done at the previous step;

- the frequency mask is updated on the basis of the foreground mask of the blob in the current frame.

In order to determine the identity of a detected object it is necessary to give the definition of the distance between an object and an appearance model. Here we present a new distance measure with respect to that defined in [9]. If we denote with m_d a view in the direction d of the visual part of the appearance model and with p the detected blob, the distance $D(p,m_d)$ is calculated as the average of the distances $D_c(p,m_d)$ calculated on the three color channels. $D_c(p,m_d)$ is obtained as a weighted sum of three contributions: the first contribution takes into account the color difference between p and m_d calculated on $p \cap m_r$, the second considers the area of the model in $m_d - p$ weighted by the frequency mask and the last accounts for the area of the detected blob in $p - m_d$. The formula is shown in Eq. 4 where B_p^t is the set of pixels of the blob p at time t and $\alpha + \beta + \gamma = 1$ (see Fig. 8).

3. Experimental Results

We evaluate our system on the PETS2009 dataset S2.L1, using the sequences from view 001, view 003 and view 005 (see Fig. 9). In order to give a preliminary evaluation and

a) View_001

b) View_003

c) View_005

Figure 9. An example of a frame from the three considered views of the PETS2009 dataset.

validation of the system, the sequences have been manually ground-truthed.

The evaluation was based on the framework by Kasturi et al. [11], which is a well established protocol for performance evaluation of object detection and tracking in video sequences.

In Table 1 and Table 2 we report the results obtained using the tool USF-DATE (the USF Detection and Tracking Evaluation) provided by Kasturi et al. Table 1 presents the results of the detection phase according to the indexes: Sequence Frame Detection Accuracy (SFDA), Multiple Object Detection Accuracy (MODA) and Multiple Object Detection Precision (MODP). Table 2 presents the results of the tracking phase according to the indexes: Average Tracking Accuracy (ATA), Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP).

Compared with the results presented in the previous

Sequence	SDFA	MODA	MODP
View001	0.594	0.833	0.645
View003	0.400	0.630	0.502
View005	0.521	0.663	0.616

Table 1. Performance Evaluation of the Detection Layer on PETS2009 S2.L1 video sequences.

Sequence	ATA	MOTA	MOTP
View001	0.092	0.830	0.638
View003	0.026	0.625	0.506
View005	0.113	0.648	0.607

Table 2. Performance Evaluation of the Tracking Layer on PETS2009 S2.L1 video sequences.

Measure	ATA	MOTA	MOTP
Average	0.077	0.701	0.584

Table 3. Average of each metric measurement for the Tracking Layer on View1, 3 and 5 of PETS2009 S2.L1 video sequences.

Measure	SDFA	MODA	MODP
Average	0.505	0.709	0.588

Table 4. Average of each metric measurement for the Detection Layer on View1, 3 and 5 of PETS2009 S2.L1 video sequences.

PETS Conference [7], our method, on the View001, outperforms the other methods whose performances were reported. The Average on Views 1, 3 and 5 (see Table 3 and Table 4) highlight that our method is slightly better than other approaches present in the literature. The very low value of ATA on View003, common to all the methods, is due to the very high fragmentation of the objects because of the tree in front of the camera.

4. Conclusions

In this paper we have discussed a video surveillance system. We have described the algorithms used in the object detection phase and in the object tracking phase. We have shown that a basic background update algorithm, together with a set of properly heuristics, can be successfully used in real environments. Furthermore, we have discussed an object tracking method based on the assignment problem framework with some modification to deal with object splits and merges. Moreover we have added a further module, based on the appearance model strategy, that re-establish objects IDs when the tracking algorithms fails in following actual objects trajectories. The experimental phase, conducted on the PETS2009 dataset using a well established protocol for performance evaluation of this kind of systems, shown very promising results.

References

[1] F. Archetti, C. Manfredotti, V. Messina, and D. Sorrenti. Foreground-to-ghost discrimination in single-difference pre-processing. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, 2006. 1

[2] M. Bertozzi and A. Broggi. Gold: a parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Transactions on Image Processing*, 7–1:62–81, 1998. 2

[3] D. Conte, P. Foggia, G. Percannella, F. Tufano, and M. Vento. An algorithm for recovering camouflage errors on moving people. In *13th International Workshop on Structural and Syntactic Pattern Recognition. To appear*, 2010. 2

[4] D. Conte, P. Foggia, G. Percannella, F. Tufano, and M. Vento. Reflection removal in colour videos. In *Twentieth International Conference on Pattern Recognition. To appear*, 2010. 2, 3

[5] D. Conte, P. Foggia, M. Petretta, F. Tufano, and M. Vento. Meeting the application requirements of intelligent video surveillance systems in moving object detection. In *Proceedings of the 3rd International Conference on Advances in Pattern Recognition and Image Analysis*, 2005. 2

[6] A. Elgammal and L. Davis. Probabilistic framework for segmenting people under occlusion. In *IEEE International Conference on Computer Vision*, 2001. 2, 4

[7] A. Ellis, A. Shahrokni, and J. Ferryman. Overall evaluation of the pets2009 results. In *Proceedings 11th IEEE International Workshop on PETS*, 2009. 7

[8] L. M. Fuentes and S. A. Velastin. People tracking in indoor surveillance applications. In *Proc. of the 2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, 2001. 4

[9] I. Haritaoglu, D. Harwood, and L. Davis. W^4: Real-time surveillance of people and their activies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22–8:809–830, 2000. 1, 4, 5, 6

[10] P. Kaewtrakulpong and R. Bowden. An improved adaptive background mixture model for realtime tracking with shadow detection. In *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS01, VIDEO BASED SURVEILLANCE SYSTEMS: Computer Vision and Distributed Processing*, 2001. 1

[11] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31–2:319–336, 2009. 7

[12] L. Li and M. Leung. Integrating intensity and texture differences for robust change detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 11–2:105–112, 2002. 1

[13] N. Peterfreund. Robust tracking of position and velocity with kalman snakes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21–6:564–569, 1999. 4

[14] W. Pratt. *Digital Image Processing*. Wiley, New York, 1978. 5

[15] A. Senior, A. Hampapur, Y. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusions handling. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2001. 2

[16] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusion handling. *Image and Vision Computing*, 24–11:1233–1243, 2006. 4, 5

[17] M. Shah and S. Khan. Tracking multiple occluding people by localizing on multiple scene planes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31–3:505–519, 2009. 2

[18] X. Song, J. Cui, H. Zha, and H. Zhao. Vision-based multiple interacting targets tracking via on-line supervised learning. In *10th European Conference on Computer Vision*, 2008. 1

[19] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22-8:747–757, 2000. 1

[20] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Seventh IEEE International Conference on Computer Vision*, 1999. 1, 2

[21] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19-7:780–785, 1997. 1

[22] J. Xia, J. Wu, H. Zhai, and Z. Cui. Moving vehicle tracking based on double difference and camshift. In *Proceedings of the International Symposium on Information Processing*, 2009. 1

[23] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26–9:1208–1221, 2004. 2