A Robust Approach to Motion Detection and Tracking in Indoor Video Surveillance

Andrea Bonarini, Matteo Matteucci, Davide Migliore, and Matteo Naccari

Artificial Intelligence and Robotics Laboratory
Politecnico di Milano, Milan, Italy

Abstract. In this paper we propose a robust approach to object detection and tracking for video surveillance. To have a strong and fast pixel classification, an innovative algorithm that uses both frame by frame difference and background subtraction is used for motion detection; tracking is then performed using a Kalman filter that exploits color "blueprint" to perform robust data association. With the proposed algorithm, a correct identification of objects is then possible without a previous background learning or explicit object model. A novel approach to frame by frame blob matching (i.e., data association), named "Relative Maximum Matching" (RMM), based on the comparison between relative maxima of objects histogram, is also presented. Experiments are given with 15 fps tracking of moving people on 640×480 unsampled color images.

1 Introduction

One of the artificial vision goals is to emulate some features of the human visual system, like the skill of recognizing object movement and tracking its behavior in a complex environment. The main issues in this task usually arise when no a-priori knowledge about the scene or about the objects to be tracked is available. The common approach, in such situations, is to gather pixels in groups named as "blobs" by distinguishing them in foreground pixels and background pixels. Doing this way, a connection between blobs and real moving objects in the environment is created and objects behavior is replaced by blobs behavior. In this process classical issues of tracking arise: *motion detection*, *tracking* and *data association* (i.e., blobs in different frames must be connected to the same moving object in order to retrace the real path of moving entities).

Our aim, in this work, is to develop a strong tracking system that does not require any major assumption about the objects to be tracked or the environment they move in. In doing this a novel approach to motion detection named *Joint Difference* is presented to obtain a robust foreground object detection. Moreover we introduce a data association algorithm that extends classical Kalman prediction with objects color "blueprints" to recover from occlusions and obtain objects re-acquiring when they move in and out of the field of view. A brief review of classical motion detection algorithm is presented Section 2. Section 3 describes the whole tracking algorithm while Section 4 presents its experimental validation. Conclusion and future works are presented in Section 5.

2 Previous Work

When dealing with visual tracking of moving objects two main issues need to be tackled: motion detection and objects following. These two activities are in fact the basic steps of a complex tracking system like the one presented in this paper. In Section 3.1 and 3.2 we propose two novel algorithms to implement respectively robust motion detection and data association. These algorithms are aimed at solving classical drawbacks of the technique presented hereafter to implement a robust tracking process.

As far as motion detection task is concerned, two main algorithm have been proposed in scientific literature: image difference and background subtraction. The former consists of a thresholded difference between frame at time t and frame at time $t-1$; this method is very performant in computational terms and grants a prompt object motion detection between two frames, but it suffers two well-known drawbacks: foreground aperture and ghosting [7]. To solve these issues, Cucchiara and Piccardi [11] propose a variation on this method: the double difference. This approach operates a thresholded difference between frames at time t and $t-1$ and between frames at time $t-1$ and $t-2$, combining them with a logical AND. However if the moving objects have not enough texture this procedure does not allow an accurate motion detection. Kanade et al. describe the VSAM projects [7], this algorithm exploits image difference between frames at time t and $t-1$ and the difference between t and $t-2$ to erase ghosting; it also keeps in memory a background model to solve the foreground aperture problem. This system is widely used in outdoor environments with a low depth of field images however suffers a few drawbacks on variable depth shots.

On th other hand, background subtraction uses a background model to be compared with the actual frame. This algorithm solves issues of image difference and gives good results in the motion detection task. However, luminance variations and objects that begin or end their motion in the scene – *waking person* – require a proper background update procedure. On this, many techniques have been developed; the most simple ones update the background by a convex composition of background pixels at time $t-1$ and image pixels at time t; update weight is eventually variable with pixel classification (light weight for background pixels and heavy weight for foreground pixels). Wren et al. describe the PFinder system [3], where there are a person and a background model previously defined; each background pixel is modeled by a Gaussian by its mean and variance. Adaptive background modeling [9] and W^4 [4] are the most important studies in the background modeling field. In the former, Stauffer and Grimson use a Gaussian mixture for each pixel: this is noise proof but also expensive in computational effort; in the latter Haritaoglu et al. use a bimodal distribution to describe background pixels: this solution can detect waking persons, but needs again to learn the background model before tracking can start.

Research has flourished also on the data association problem and in particular regarding blob matching. On this topic, Teknomo et al. [6], with an aerial point of view hypothesis, suggest the use of some statistical indexes combined in a unique index and an algorithm for the management of objects apparitions and disappearances. Often blob matching phase is improved by a Kalman filter prediction; Masoud et al. [5] use this prediction to associate an object between a frame and the next one: besides, they state some rules for objects split and merge across the entire video sequence.

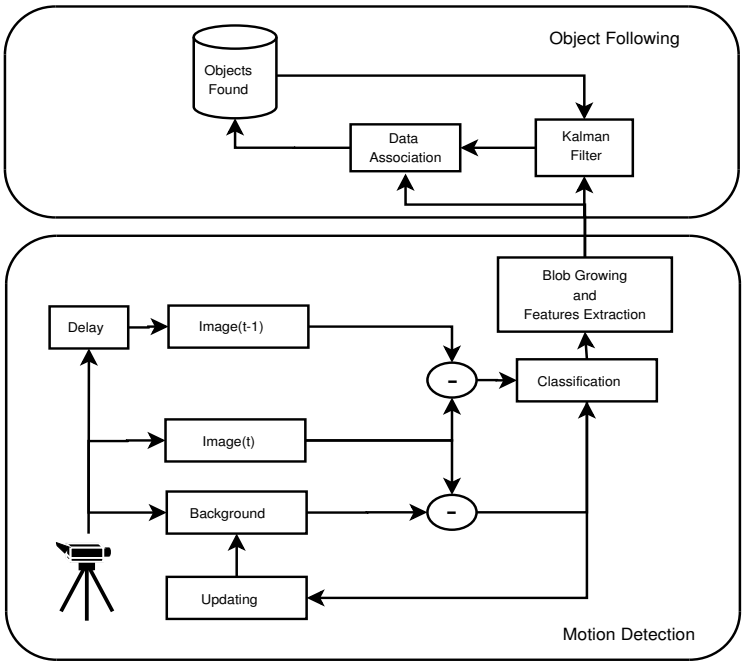


Fig. 1. The overall schema of the tracking system

3 The Tracking System

To perform a robust tracking of moving object we would like to have an algorithm requiring only weak assumption about the working environment. The only hypothesis we assume in the following is the absence of unexpected sudden illumination changes (i.e., no one is going to switch off the light) and fixed camera[1]. Objects are allowed to start their motion from the very first frame, no background needs to be learnt in advance and no object model is given.

To solve these issues we implemented a novel motion detection algorithm named *Joint Difference* (see Fig. 1) that uses both frame difference and background subtraction to perform a first pixel classification in: foreground (FG), background (BG), ghosting (GH), foreground aperture (FA), and waking person (WP). Pixel are then combined in connected regions, shadows removal is performed and features for data association are extracted (Section 3.1). Data association is performed by extending the information provided by the Kalman filter prediction with another algorithm named *Relative Maxima Matching* and the information gathered by this association is used in a classical Kalman Filter to perform object following as described in Section 3.2.

[1] The fixed camera hypothesis is a temporary hypothesis since we are currently working on including camera motion estimation in the system.

3.1 Motion Detection by the Joint Difference

To classify pixels in a robust fashion, at time t we combine the difference between the actual frame F_t and previous frame F_{t-1} with the difference between actual frame and the actual model of background B_t. To bootstrap this process, the first image of the sequence is kept as initial background model and it is dynamically updated with the new frame according to the following classification of its pixels.

As already introduced, five classes can be distinguished in this process: *foreground* (FG), *background* (BG), *ghosting* (GH), *foreground aperture* (FA), and *waking person* (WP). A foreground pixel belongs effectively to the moving object. On the contrary a background pixel belongs to the background of the scene. The other three classes are more interesting. To better describe the ghosting problem let be p a moving pixel at location (x, y) in F_{t-1}; in F_t it will be in a different location, so the image difference in (x, y) will present an object pixel instead of a background one. In Fig. 2 we can notice that the ball is moving from right to left and in the difference image ghosting is present on the right. On the other hand, foreground aperture is due to pixels with similar texture. In Fig. 2 the white zone between the two black circles shows this problem.

The *waking person* is a different problem due to background subtraction; an object that belongs to the background model suddenly starts to move at time t. From this frame, motion detection identifies a second object in the position where previously was the moved object. This is due to the difference between the old background and the actual frame where the waking person is moving. This phenomenon is shown in Fig. 3.

To solve these problems we have exploited a joint background subtraction and frame by frame difference to properly classify pixels. A pixel belongs to foreground when frame by frame difference is above a given threshold τ_I and difference between background and actual frame is above threshold τ_B. Instead, a pixel belongs to background if frame by frame difference is below τ_I and difference between background and actual frame is below τ_B. Moreover we can classify ghosting by exploiting the fact that frame by frame difference is above τ_I and difference between background and actual frame is below τ_B. However, with this simple reasoning we can not distinguish between waking person and foreground aperture; we just know that these two phenomenon appear when frame by frame difference is below than τ_I and background difference is above τ_B. We can thus summarize this simple classification procedure as:

$$\Delta_I > \tau_I \wedge \Delta_B > \tau_B \Rightarrow FG \tag{1}$$

$$\Delta_I < \tau_I \wedge \Delta_B > \tau_B \Rightarrow FA \vee WP \tag{2}$$

$$\Delta_I > \tau_I \wedge \Delta_B < \tau_B \Rightarrow GH \tag{3}$$

$$\Delta_I < \tau_I \wedge \Delta_B < \tau_B \Rightarrow BG \tag{4}$$

being Δ_F and Δ_B defined as $\Delta_F = |F_t(x, y) - F_{t-1}(x, y)|$ and $\Delta_B = |F_t(x, y) - B_{t-1}(x, y)|$.

At this point, to solve the ambiguity between FA and WP, pixels are collected into blobs without doing any distinction among foreground, foreground aperture and waking person. An immediate distinction between foreground aperture and waking person is impossible; once pixels are collected in blobs, we can exploit the fact that a waking person blob will contain only pixels classified as $FA \vee WP$ while a foreground blob

(a) Previous frame (b) Actual frame (c) Frame difference

Fig. 2. An example of frame difference with foreground aperture.

(a) Previous frame (b) Background model (c) Background difference

Fig. 3. An example of background subtraction with waking person.

can contain mostly foreground pixels. This can be easily obtained by comparing the ratio of foreground pixels in the blob with the total number of pixels:

$$N_{FG} \geq \gamma N_{Tot} \Rightarrow FG \tag{5}$$

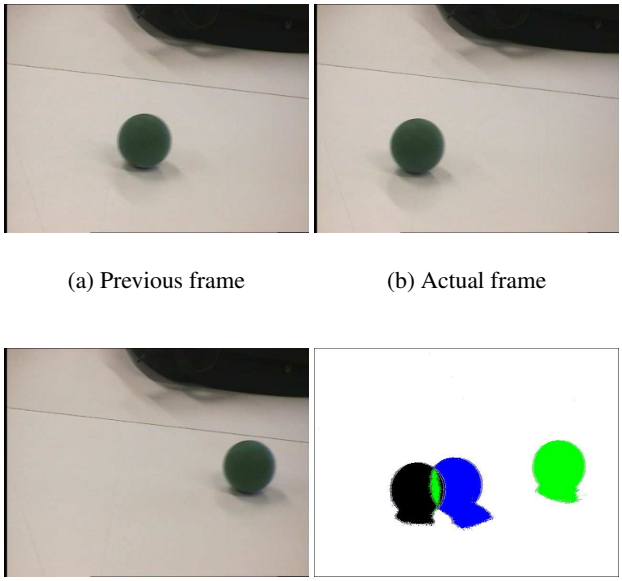
$$N_{FG} < \gamma N_{Tot} \Rightarrow WP. \tag{6}$$

To get a stronger detection of waking person a further constraint can added for those objects that started their motion into the scene; if a pixel has been classified as $FA \vee WP$ for a given number of frames it can be safely classified as a waking person since (by definition) a foreground aperture is something that moves. Fig. 4 shows the result at the end of this classification procedure in all the situations mentioned above.

This classification process is based on simple algorithms that make it very fast in performance and can be exploited both to get a more precise blob detection and an adaptive background update as described further on.

In our system, the background model is initialized with the first image of the video sequence, and then updated, as in [3], using:

$$B_t = (1 - \alpha)B_{t-1} + \alpha F_t. \tag{7}$$



(a) Previous frame (b) Actual frame

(c) Background model (d) Classification result

Fig. 4. Classification results: ghosting (blue), foreground aperture and waking person (green), background (white), and foreground(black)

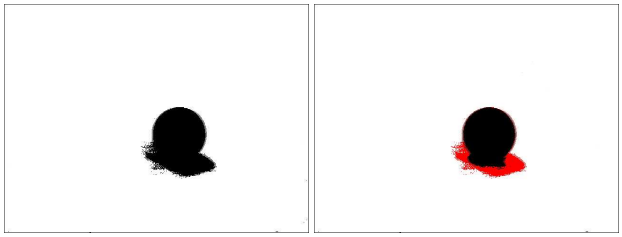
Pixel classification is used to tune background adaptation. α is an adaptive weight that has a value set according to pixel classification; if the pixel is foreground then $\alpha = 0$ to avoid background corruption, if it is background then α is set to a low level not to corrupt the model with shadows or reflections, finally, if a pixel is a waking person α is set to the highest level, so the background is quickly restored. Image difference is computed through a pixel by pixel Manhattan distance in the RGB color space [2]; after this, the image is thresholded to become binary.

Before binarization it is useful to introduce a shadows filter to get a refined blob growing. The work by Horprasert et al. [10] shows that, in the HSV space, the difference between a shaded zone and the same zone enlightened regards the luminance component, leaving the components of hue and saturation nearly unchanged. A simple filter is used in the blobs obtained during classification; we define the x and y distance between two points with constant S in the HSV cylinder as:

$$HSx_i = S_i \cos(H_i) \tag{8}$$

$$HSy_i = S_i \sin(H_i) \tag{9}$$

[2] We tested many color spaces as HSV, HSI, YUV, HLS, Ohta and Opponent, but RGB gave us good results in pixels classification with a low computational cost.



(a) Without shadow detection (b) With shadow detection

Fig. 5. An example of shadow detection.

then a shadow is identified if the three following inequalities are verified:

$$|HSx_B - HSx_F| < \tau_x \tag{10}$$

$$|HSy_B - HSy_F| < \tau_y \tag{11}$$

$$|V_B - V_F| < \tau_v \tag{12}$$

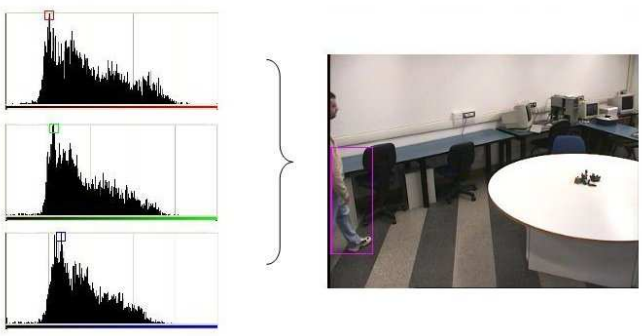
The result of this shadows filter is reported in Fig. 5, red pixels are those detected by the previous conditions.

3.2 Data Association in Tracking

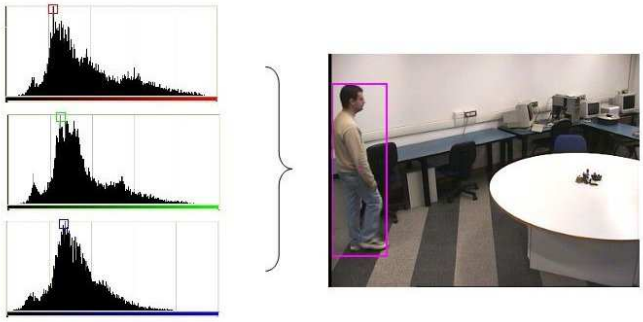
Object following is obtained by using a classical Kalman filter on the image plane to track moving objects behavior [12]. Each blob state is composed of its x and y coordinates on the image plane and its v_x and v_y velocities. This state is estimated using a Kalman filter and, to allow a good state initialization, it is activated only when the blob appears at least twice consecutively in the scene; it is deactivated and reset when the object goes out of the field of view. The novelty we introduced is the algorithm used for the data association phase; to this purpose, for each blob, some useful information are collected such as: pixels centroid, bounding box parameters, and color blueprints.

A distance comparison is made between the blob centroid and the Kalman filter prediction, when active, or with a constant threshold to select possible matching. In order to disambiguate between possible matching and be robust to Kalman filter errors, distance comparison is only one of the criteria used to perform a data association between a blob and the (possible) corresponding object. Based on the hypothesis that two identical objects have a similar RGB values we implemented also a comparison between RGB channels average and we use a threshold to check if this property holds between blobs and object. Similarly, the ratio between the variance of RGB channels and the total pixels is also used to distinguish homogeneous blobs with a compact distribution from those with a scattered distribution.

The novelty in this multi criteria approach is introduced to match blobs with a part of an object. In fact, a blob can sometimes identify a part of an object (e.g., an arm or a foot



(a) Previous frame



(b) Actual frame

Fig. 6. Relative Maxima Analysis: in (a) we can see the histograms of the leg and in (b) local maxima of the leg are mapped on the entire body histograms

of a person entering in the scene) and a more flexible index is needed to match the whole target with these parts. To efficiently solve this issue we propose to compare relative maxima of color histograms to check a *Relative Maxima Match* (RMM) criterion. In fact, as shown in Fig. 6, parts of an object can to be identified with a good approximation using this fast histograms analysis. The RMM criterion can be easily summarized as follows: an object and a blob are selected, one by the previous frame (or history) and the other by the actual frame and the smaller is chosen to start the matching. From each channel histogram the first maximum is taken, then it is compared to the maxima of the other blob corresponding histogram. When the difference between them is less than a (small) threshold, a matching for the channel is found; otherwise the next significative maximum of the smaller blob is chosen until all the maxima are checked or a successful matching is done. If there is a successful matching for all three channel histograms then

the blob is associated to the object. To speed-up computation for this criterion relative maxima are checked only if they are significant, i.e., if the ratio between their lever and the maximum of the histogram is high enough.

The use of many matching criteria and especially RMM grants strength and stability in the data association even with significative variations in blobs dimension and provide a solid base for the re-acquiring of object disappeared from the scene. In particular, the use of a color blueprint for the objects allows to recognize a match between a blob and an object that temporarily disappeared from the field of view. On the other hand, having many matching criteria it can happen that a blob can match with more than one object. To obtain a single data association each of the previous criterion has associated a weight and for each object O_i the highest weight of its matches is chosen as final matching degree for the couple blob-object (b_j, O_i). Blob matching is thus divided in two phases:

1. Match weight calculation for each couple $w(b_j, O_i)$ using the maximum weight of matching criteria
2. Single match assignment of blob b_i to the object O_i having the maximum $w(b_j, O_i)$

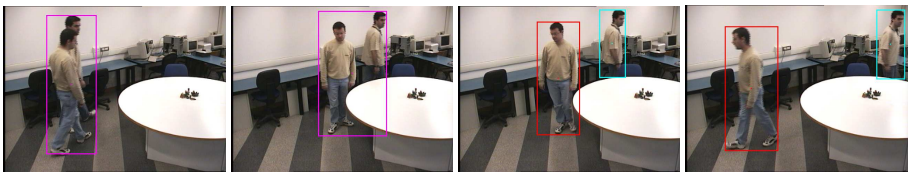
4 Experimental Validation

Our system has been tested using 640x480 color images grabbed by a Sony EVI D31 video camera at 15 fps. In this section we present five benchmark designed to test the effectiveness of the algorithm proposed[3]. We are interested in validating the robustness of the approach so we focus on difficult situations that can be found in indoor video surveillance:

- Objects splitting: two objects enter the scene together and then leave it taking two different directions
- Objects merging: two objects enter the scene from opposite directions, then they meet and finally go out of the field of view together
- Object merge and split: it is similar to Intersection but this time the two persons stay together for a long time so that the merging of the two blobs is actuated. Finally the two persons leave and go outside of the scene
- Objects crossing: it is similar to objects merging, but the persons go outside taking different trajectories
- Object re-acquiring: a few objects move through the scene out of the field of view; after a while one of these enters again the scene and has to be recognized

The first experiment is aimed at validating the objects detection capability of the overall system. As it is shown in the sequence of Fig. 7 the system is capable to detect a moving object entering the scene. Actually this object is not a single entity since it is composed by two person walking together; when these two persons split to different directions the system is not able to track them any more and a split happens. Notice that in this case two new object are created from the old one and the past trajectory is inherited by both the new objects.

[3] Video sequences are available on request.



(a) (b) (c) (d)

Fig. 7. An example of objects splitting: two persons enter together, they change their trajectories and two objects are created.

(a) (b) (c) (d)

Fig. 8. An example of objects merging: two persons enter alone, they meet and the merge is done

The second experiment (Fig. 8) verifies the opposite situation, in this case the two persons are merged together, no new object is created. This experiment has to be compare with the following one reported in Fig. 9. In this second case the merge is done, but it is followed by a split of the two object. In this case we do non want two new object to be created since they were already seen in previous frames. As it can be noticed from the images this is properly obtained.

The fourth experiment is a classic benchmark to test the effectiveness of Kalman prediction in data association. As can be seen from sequence in Fig. 10 this is properly managed by our system. From this sequence it is possible to notice how occlusions can generate new objects in the tracking: the foot of the person walking from right to left is tracked up to the last frame.

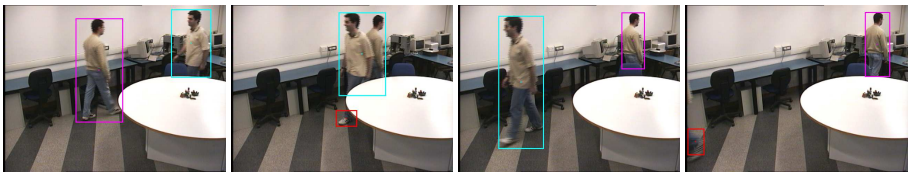
A more interesting experiment is the one described in Fig. 11; three persons walk through the room and go out, then one of them jumps back into the scene. In this case data association is performed thanks to the Relative Maxima Matching criterion and the person is properly re-acquired.

5 Conclusions and Future Works

In this paper we have presented a novel approach to object detection and tracking in indoor environments for video surveillance. To have a strong and fast pixel classification, an innovative algorithm for motion detection is used that uses both frame difference and

(a) (b) (c) (d)

Fig. 9. An example of objects merge and split: two persons enter the scene, meet for a while, and then go out taking different directions.



(a) (b) (c) (d)

Fig. 10. An example of object crossing: two persons enter the scene, they walk in the same direction and cross each other.

background subtraction. Tracking is then performed on detected blobs using a Kalman filter that exploits blob features like color "blueprint" to perform robust data association. A correct identification of objects is then possible without a previous background learning or explicit object model. Frame by frame data association different criteria and one of this (i.e., Relative Maximum Matching) has been exploited to re-acquire objects that move in and out of the field of view.

It still exist the case in which data association fail due to the possibility that two objects may have a similar color blueprint and similar size/position; in this cases we would like to exploit a multiple hypothesis approach to overcome the unimodality of Kalman filter. This system could be also improved with an adaptive threshold selection that would increase the precision of motion detection task. The integration of background alignment to fit camera motion and the integration of perspective information in the Kalman filter are work in progress.

References

1. S. Coradeschi, A. Saffiotti "An Introduction to the anchoring problem" *Robotics and Autonomous Systems*, Vol 43, No. 2-3, pp.85-96, 2003.
2. M. Rautianinen, T. Ojala, and H. Kauniskangas, "Detecting Perceptual Color Changes from Sequential Images for Scene Surveillance," *IEICE*, Vol. E84-D, December 2001.

(a) (b) (c) (d)

Fig. 11. An example of object re-acquiring: three persons enter a scene and go out then one of them enters again and he is recognized.

3. C. R. Wren, A. Azarbayejani, T. Darrel and A. P. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE*, Vol. 19, No. 7, July 1997.
4. I. Haritaoglu, D. Harwood and L. S. Davis, "W4: Real-Time surveillance of People and Their Activities," *IEEE*, Vol. 22, No. 8, July 2000.
5. H. Veeraraghavan, O. Masoud and N. P. Papanikolopoulos, "Computer Vision Algorithm for Intersection Monitoring," *IEEE*, Vol. 4, No. 2, July 2003.
6. K. Teknomo, Y. Takeyama and H. Inamura, "Frame-Based Tracing of Multiple Objects," *IEEE*, 2001.
7. R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa. P. Burt and L. Wixson, "A System for Video Surveillance and Monitoring," *Robotics Institute, Carnegie Mellon University, Technical Report*, CMU-RI-TR-00-12, 2000.
8. P. L. Rosin and T. Ellis, "Image difference threshold strategies and shadow detection," *Proceedings of the 6th British Machine Vision Conference, Birmingham, UK, September*, pp. 247-356, 1995.
9. C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE*, 1999.
10. T. Horprasert, D. Harwood and L. S. Davis, "A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection," *IEEE*, 1999.
11. R. Cucchiara, M. Piccardi: "Vehicle Detection Under Day and Night Illumination" *Proceeding of the 3rd International ICSC Symposia on Intelligent Industrial Automation and SOft computing* June 1-4, 1999, Genova, Italy, ISBN 3-906554-17-7
12. M. Kohler "Using the Kalman filter to track human interactive motion, modelling and initialization of the Kalman filter for translational motion" *Universitat Dortmund, Technical Report*, 1997.