Computer Science Department of The University of Auckland
CITR at Tamaki Campus (http://www.citr.auckland.ac.nz)

CITR-TR-78 November 2000

The Background Subtraction Problem for Video Surveillance Systems

Alan McIvor[1], Qi Zang[2] and Reinhard Klette[2]

Abstract

This paper reviews papers on tracking people in a video surveillance system, and it presents a new system designed for being able to cope with shadows in a real-time application for counting people which is one of the remaining main problems in adaptive background subtraction in such video surveillance systems.

[1] Reveal Ltd., Level 1, Tudor Mall, 333 Remuera Rd, Auckland, New Zealand

[2] CITR, Department of Computer Science,Tamaki Campus, University of Auckland, Private Bag 92019, Auckland, New Zealand

The Background Subtraction Problem for Video Surveillance Systems

Alan McIvor[a], Qi Zang[b], and Reinhard Klette[b]

[a] Reveal Ltd., Level 1, Tudor Mall, 333 Remuera Rd.
[b] CITR, University of Auckland, Tamaki Campus, Building 731
Auckland, New Zealand

Abstract. This paper reviews papers on tracking people in a video surveillance system, and it presents a new system designed for being able to cope with shadows in a real-time application for counting people which is one of the remaining main problems in adaptive background subtraction in such video surveillance systems.

Keywords: video surveillance, background subtraction, counting people

1 Introduction

Video surveillance systems seek to automatically identify events of interest in a variety of situations. Example applications include intrusion detection, activity monitoring, and pedestrian counting. The capability of extracting moving objects from a video sequence is a fundamental and crucial problem of these vision systems. For systems using static cameras, background subtraction is the method typically used to segment moving regions in the image sequences, by comparing each new frame to a model of the scene background, see, e.g., [6, 13].

1.1 Classification of Existing Approaches

There are many different approaches to solving the problem of tracking people in a video surveillance system. These can be classified into three main groups: feature-based tracking, background subtraction, and optical flow techniques. Feature based tracking can be further divided depending on the scale of the features. These various approaches are discussed below. Note that this classification is based on the methods used for detecting objects of interest. The other component of a successful system is the methodology used for solving the correspondence of features between frames. In almost all approaches, this is solved using a method based on Kalman Filter tracking [1].

Large Scale Feature Based Tracking In this approach, a large scale feature of an object is identified and subsequently tracked. Typically, this is the bounding silhouette of the object and this is tracked with using a snake-type tracker, e.g., [3]. In the context of tracking people, another useful feature is to detect faces by skin-tone, etc. and track these features.

Small Scale Feature Based Tracking In this approach, features such as edge fragments or corners are identified in the images and these are independently tracked through the image sequence. Then these features have to be grouped into clusters that correspond to independent moving objects. For example, in [2], corner points are tracked in a road surveillance application and then grouped together on the basis that all corners on a vehicle will have the same motion, and that this motion will be significantly different from that of other vehicles over the scene, even when travelling in the same highway lane. Such an approach would be difficult to transfer to a pedestrian tracking application because people are highly articulated objects, and hence individual components exhibit significant relative motion. But, moreover, given the smoother exterior of people, the extracted corner points will not have stable surface generators, and they will tend to be viewpoint sensitive.

The significant problem in this approach is the clustering of the tracked primitive features into groups that correspond to objects. One possible approach is to use a perceptual grouping technique [15].

Optical Flow Based Techniques The basis of this approach is to estimate the optical flow at all points of the image plane. Significant points are then grouped based on principles of motion coherence, etc. This approach is closely related to the small scale feature based tracking approach.

Background Subtraction Background subtraction is a region-based approach where the objective is to identify parts of the image plane that are significantly different to the background, i.e., the scene as would appear if there were no foreground objects in it. The significant problem to be addressed in this approach is that of estimating the background image, especially when the illumination levels change. The method described in this paper falls into this category. Another important aspect of the background subtraction approach is the definition of what constitutes a significant difference from the background. This is a threshold selection problem [19].

1.2 Evaluation Criteria

When the above methods are applied to a video surveillance problem, there are a number of key attributes and scenarios that must be handled. These are:

- The choice of models for key components.
- Initialization of the model parameters when the system is first started.
- Distinguishing objects of interest from illumination artifacts such as shadows and highlights.
- Handling uncontrollable illumination level changes, such as occur in outdoor scenes.
- Adapting to changes in the scene, such as when a new chair is introduced into a restaurant.
- Detecting when something is wrong with the processing, and re-initialization in response.

The rest of this paper will concentrate on approaches based on background subtraction. Firstly, existing methods will be reviewed with respect to the above criteria. Then a new method will be described and its performance evaluated.

2 Review of Existing Methods

In a background subtraction based approach, the field of view can be divided into three components:

background = the parts of the static scene that are still visible.
objects = things of interest to the application, e.g., pedestrians.
artifacts = image changes such as shadows and highlights.

The combination of the "objects" and the "artifacts", i.e., everything that is not "background", is often called the *foreground*. In most existing methods, only the background is explicitly modeled. The foreground is divided into object and artifact on the basis of various image properties. However, [16] takes the opposite approach of explicitly modelling the color distribution of each object using a mixture of Gaussians and uses this to detect the objects. In [18,22], both the background and the foreground are modelled as Gaussian distributions.

In this section, we discuss existing methods with respect to the choice of a background model and how it is maintained in the face of illumination changes, how the foreground is differentiated from the background, and what strategies are used to correct classification errors. Finally, the classification of foreground into objects and artifacts is discussed. The following notation will be used throughout this section:

$\mathbf{I}_t$ = the incoming image at time t.
$\mathbf{B}_t$ = the background estimate at time t.

The coordinates associated with a pixel are not shown in the equations unless necessary.

2.1 Background Models

The simplest and most common model for the background is to use a point estimate of the color at each pixel location, e.g., [12]. This point estimate is usually taken to be the mean of a Gaussian distribution. In some systems, e.g., [17], the variance of the intensity is also modelled.

To cope with variation in the illumination, the background estimate has to be continually updated. In [12], this is updated using

$$\mathbf{B}_{t+1} = \alpha \mathbf{I}_t + (1-\alpha)\mathbf{B}_t \tag{1}$$

where α is kept small to prevent artificial "tails" forming behind moving objects. In [5], the background is maintained as the temporal median of the last N frames, with typical values of N ranging from 50 to 200. The updating of the background

estimate is often restricted to pixels which have been classified as background. In [9–11], three parameters are estimated at each pixel: $\mathbf{M}$, $\mathbf{N}$, and $\mathbf{D}$, which represent the minimum, maximum, and largest interframe absolute difference observable in the background scene.

Such simple background estimates fail to cope with scenes which contain regularly changing intensities at a pixel, such as occurs with flashing lights and swaying branches. Several more complex background models have been developed to handle such scenarios. In [7, 20, 14], each pixel is separately modeled by a mixture of K Gaussians

$$P(\mathbf{I}_t) = \sum_{i=1}^{K} \omega_{i,t} \cdot \eta(\mathbf{I}_t; \mu_{i,t}, \Sigma_{i,t}) \tag{2}$$

where $K = 4$ in [14] and $K = 3 \ldots 5$ in [20]. In [7,20], it is assumed that $\Sigma_{i,t} = \sigma_{i,t}^2 \cdot \mathbf{I}$. The background is updated, before the foreground is detected, as follows:

1. If $\mathbf{I}_t$ matches component i, i.e., $\mathbf{I}_t$ is within λ standard deviations of $\mu_{i,t}$ (where λ is 2 in [7] and 2.5 in [14,20]), then the ith component is updated as follows:

$$\omega_{i,t} = \omega_{i,t-1} \tag{3}$$
$$\mu_{i,t} = (1-\rho) \cdot \mu_{i,t-1} + \rho \mathbf{I}_t \tag{4}$$
$$\sigma_{i,t}^2 = (1-\rho) \cdot \sigma_{i,t-1}^2 + \rho(\mathbf{I}_t - \mu_{i,t})^T(\mathbf{I}_t - \mu_{i,t}) \tag{5}$$

   where $\rho = \alpha \cdot \Pr(\mathbf{I}_t | \mu_{i,t-1}, \Sigma_{i,t-1})$.
2. Components which the incoming image doesn't match are updated by

$$\omega_{i,t} = (1-\alpha)\omega_{i,t-1} \tag{6}$$
$$\mu_{i,t} = \mu_{i,t-1} \tag{7}$$
$$\sigma_{i,t}^2 = \sigma_{i,t-1}^2 \tag{8}$$

3. If $\mathbf{I}_t$ does not match any component, then the least likely component (the one with smallest $\omega_{i,t}$) is replaced with a new one which has $\mu_{i,t} = \mathbf{I}_t$, $\Sigma_{i,t}$ large, and $\omega_{i,t}$ low.

The weights then have to be renormalized.

In [4], three background models are simultaneously kept, a primary, a secondary, and an old background. They are updated as follows:

1. The primary background is updated as

$$\mathbf{B}_{t+1} = \alpha \mathbf{I}_t + (1-\alpha)\mathbf{B}_t \tag{9}$$

   where the pixel is not marked as foreground, and is updated as

$$\mathbf{B}_{t+1} = \beta \mathbf{I}_t + (1-\beta)\mathbf{B}_t \tag{10}$$

   where the pixel is marked as foreground. In the above, α was selected from within the range $[0.0000610351 \ldots 0.25]$, with the default value $\alpha = 0.0078125$, and $\beta = 0.25 \cdot \alpha$.

2. The secondary background is updated as

$$\mathbf{B}_{t+1} = \alpha \mathbf{I}_t + (1-\alpha)\mathbf{B}_t \tag{11}$$

at pixels where the incoming image is not significantly different from the current value of the secondary background, where α is as for the primary background. At pixels where there is a significant difference, the secondary background is updated by

$$\mathbf{B}_{t+1} = \mathbf{I}_t \tag{12}$$

What constitutes a significant difference is not defined. It is also noted that there were problems with this background estimator.

3. The old background is a copy of the incoming image from 9000 to 18000 frames ago. It is not updated.

They claim that adding more than one secondary background actually reduces the sensitivity of the system because there is a greater range of values that a pixel can take on without being marked as foreground.

In [21], two background estimates are used which are based on a linear predictive model:

$$\mathbf{B}_t = -\sum_{k=1}^{p} a_k \mathbf{I}_{t-k} \quad \text{and} \quad \hat{\mathbf{B}}_t = -\sum_{k=1}^{p} a_k \mathbf{B}_{t-k} \tag{13}$$

where the coefficients a_k are reestimated after each frame is received so as to minimize the prediction error. The second estimator $\hat{\mathbf{B}}_t$ is introduced because the primary estimate $\mathbf{B}_t$ can become corrupted if a part of the foreground covers a pixel for a significant period of time. As well as the above, an estimate of the prediction error variance is also maintained:

$$\mathcal{E}(e_t^2) = \mathcal{E}(\mathbf{I}_t^2) + \sum_{k=1}^{p} a_k \mathcal{E}(\mathbf{I}_t \mathbf{I}_{t-k}) \ . \tag{14}$$

2.2 Foreground Detection

The method used for detecting foreground pixels is highly dependent on the background model. But, in almost all cases, pixels are classified as foreground if the observed intensity in the new image is substantially different from the background model, e.g.,

$$|\mathbf{I}_t - \mathbf{B}_t| > \tau \tag{15}$$

where τ is a "predefined" threshold, or a factor dependent on the variance estimate also maintained within the background model. In systems that maintain multiple models for the background, then a pixel must be substantially different from all background values to be classified as foreground. In [4], an adaptive thresholding with hysteresis scheme is used.

In the mixture of Gaussians approach, the foreground is detected as follows. All components in the mixture are sorted into the order of decreasing $\omega_{i,t}/\|\Sigma_{i,t}\|$. So higher importance gets placed on components with the most evidence and lowest variance, which are assumed to be the background. Then let

$$B = \text{argmin}_b \left(\frac{\sum_{i=1}^{b} \omega_{i,t}}{\sum_{i=1}^{K} \omega_{i,t}} > T \right) \tag{16}$$

for some threshold T. Then components $1, \ldots, B$ are assumed to be background. So if $\mathbf{I}_t$ does not match one of these components, the pixel is marked as foreground.

In [21], which uses two predictions for the background value, a pixel is marked as foreground if the observed value in the new image differs from both estimates by more than a threshold $\tau = 4\sqrt{\mathcal{E}(e_t^2)}$, where $\mathcal{E}(e_t^2)$ is calculated in (14).

2.3 Error Recovery Strategies

The section above on background models describes how the various methods are designed to handle gradual changes in illumination levels. Sudden changes in illumination must be detected and the model parameters changed in response. In [21], the strategy used is to measure the proportion of the image that is classified as foreground and if this is more than 70%, then the current model parameters are abandoned and a re-initialization phase is invoked. In [12], if a pixel is marked as foreground for most of the last couple of frames (the values of 'most' and 'couple' are not given), then the background is updated as $\mathbf{B}_{t+1} = \mathbf{I}_t$. This correction is designed to compensate for sudden illumination changes and the appearance of static new objects.

Another common problem are regular changes in illumination level, such as that from moving foliage. Multiple model schemes such as the mixture of Gaussians explicitly handle such problems. However, they present a problem for single model schemes. In [12], to compensate for these problems, a pixel is masked out from inclusion in the foreground if it changes state from foreground to background frequently.

2.4 Foreground Classification

The purpose of foreground classification is to distinguish objects of interest such as pedestrians from illumination artifacts such as shadows and highlights. This is usually accomplished by evaluating the color distribution within a foreground region [13]:

- A *shadow* region has similar hue and saturation to the background but a lower intensity (see Fig. 2.4.
- A *highlight* region has similar hue and saturation to the background but a higher intensity.

and, consequently, an object region has a different hue and saturation to the background.

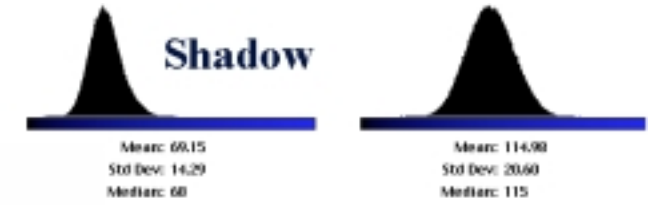

Fig. 1. Value distribution in the Blue channel of a region of the background: left - with shadow, right - without. This shows a typical situation: shadow means reduced intensity and reduced 'dynamics'.

3 An Approach Incorporating Foreground Classification

The section before has indicated that there are many background subtraction methods. Most of them are only concerned about detecting pixels where the background is no longer visible, without attempting to understand what is 'covering' such pixels. Because background subtraction is based on observed intensities, such techniques normally misclassify illumination artifacts such as shadows and highlights. If these are considered as objects of interest in a surveillance application, then this results in three types of errors:

- False alarm due to a visible shadow but not an object (that generated it).
- Shadows increase the apparent size of an object, perhaps resulting in a false labelling as multiple objects.
- Missing separation of objects because a shadow bridges the visible area between them.

In this section, a method is described which addresses these problems. In the following section its performance is evaluated.

3.1 The Background Model

The background observed at each pixel is modelled by a multidimensional Gaussian distribution in RGB space. This requires estimates of the mean and standard deviation. The estimates of these parameters are updated with each new frame, but only at pixels that have been classified as background. The updating is done with the following linear filter:

$$\mu = (1-\alpha)\mu + \alpha Ii$$
$$\sigma^2 = (1-\alpha)\sigma^2 + \alpha(Ii-\mu)^2$$

Here, α is the predefined learning rate.

Initially the distributions of the background are not known. We initialize all unknown variables in the following way:

Mean value = the pixel's RGB value of the first frame.

Standard deviation = a high value (the actual value will be obtained during updating).
Brightness distortion = a high value (the actual value will be obtained during updating).

3.2 Pixel Classification

Each pixel in a new image is classified as one of background, object, shadow, or highlight. Pixels are classified as background if the observed color value is within the ellipsoid of probability concentration (the region of minimum volume, centered around the background mean, that contains 99% of the probability mass for the Gaussian distribution).

The distinction between objects, shadows, and highlights among the pixels not classified as background is made using the brightness distortion [13]. This detects shifts in the RGB space between a current pixel and the corresponding background pixel. Let E_i represent the expected background pixel's RGB value, I_i represents the current pixel's RGB value in the current frame, σ_i represents its standard deviation, with

$$E_i = [E_r(i), E_g(i), E_b(i)],\ I_i = [I_r(i), I_g(i), I_b(i)],\ \text{and}\ \sigma_i = [\sigma_r(i), \sigma_g(i), \sigma_b(i)]\ .$$

The brightness distortion is a scalar value that brings the observed color close to the expected chromaticity line (line from origin to E_i). It is obtained by minimizing

$$\phi(\alpha_i) = (I_i - \alpha_i E_i)^2\ ,$$

where α_i represents the pixel's strength of brightness with respect to the expected value: α_i is 1 if the brightness of the given pixel in the current image is the same as in the reference image, and α_i is less than 1 if it is darker, and greater than 1 if it becomes brighter than the expected brightness. If the current pixel's intensity is greater than the background intensity and the current pixel's brightness distortion value is less than the corresponding background pixel's brightness distortion value, this pixel will be marked as highlight. If the current pixel's intensity is less than the background intensity and the current pixel's brightness distortion value is less than the corresponding background pixel's brightness distortion value, this pixel will be marked as shadow. Otherwise the pixel is marked as an object of interest. In the figures below the produced output is marked as follows: moving objects (people) in light green, and shadow or highlight in dark red.

4 Our Results

The algorithm as discussed above has been tested for indoor lab scenes. Figure 2 illustrates the effects of different lighting on a static scene (i.e. no moving people). It shows how different lighting affects our test area. Figure 3 shows that the algorithm works accurately in a uniformly lighted scene: The moving

Fig. 2. Different background lighting: left - uniform lighting with all curtains closed , right - lighting coming just from window.

Fig. 3. Backgrond subtraction for uniform lighting: left - the original frame, right - the result after background subtraction.

person is marked as light green, the shadow is marked as dark red successfully. Figure 4 illustrates that the algorithm still works fine after adding one extra light. Figure 5 shows how the algorithm is able to cope with global illumination changes after adding two extra lights. Figure 6 shows a different lighting situa-

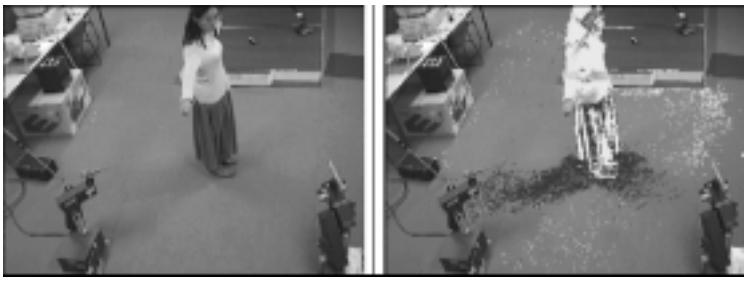

Fig. 4. Backgrond subtraction for uniform lighting plus one extra light: left - the original frame, right - the result after background subtraction.

tion. The algorithm can not only detect moving people and shadows accuratly,

Fig. 5. Backgrond subtraction for uniform lighting plus two extra lights: left - the original frame, right - the result after background subtraction.

Fig. 6. Backgrond subtraction for lighting coming from window plus one extra light: left - the original frame, right - the result after background subtraction.

it also can cope with highlights. Figure 7 illustrates the highlight abilities after adding an extra light. It still works fairly well and robust for moving people sutraction and shadow detection. Figure 8 shows a result after initialization by using static frames without moving people. We obtain good foreground subraction and shadow detection results in real-time. Figure 9 shows that after static initialization, the background model can be updated adaptively. The algorithm updates a new background model while frames with moving people are captured. Figure 10 is to illustrate the robustness of background updating. Even when we capture a very different frame with moving people, this algorithm still can refine the background model and we obtain good subtraction results.

Fig. 7. Backgrond subtraction for lighting coming from window plus two extra lights: left - the original frame, right - the result after background subtraction.

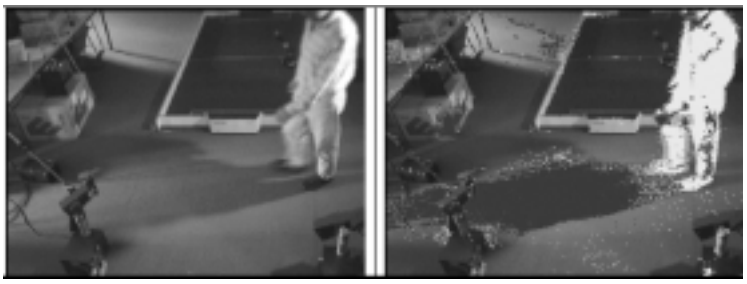

Fig. 8. Here, all frames before the current frame have been static (i.e. without moving people): left - the current frame, right - the result after background subtraction.

Fig. 9. Here, starting with static frames (i.e. without moving people) we continue with 5 frames showing moving people until the current frame: left - the current frame, right - the result after background subtraction.

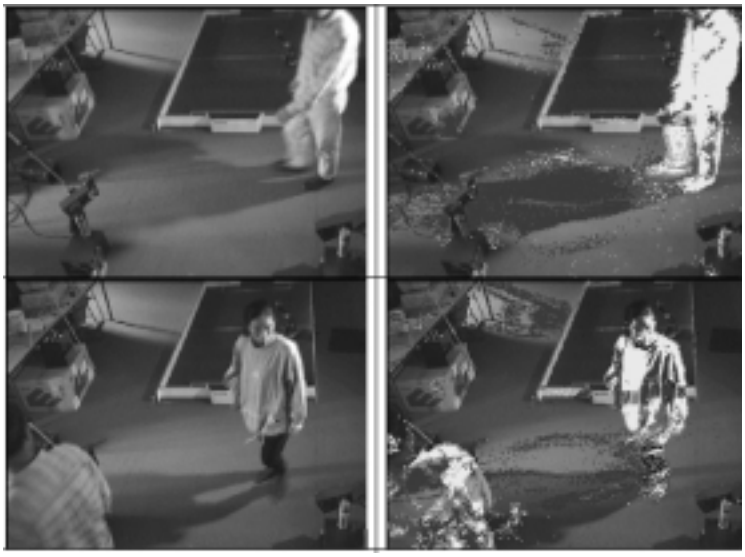

Fig. 10. Same situation as in figure before, but different people are walking in the scene before the current frame is taken: left - the current frame, right - the result after background subtraction.

5 Conclusion and Further work

This paper presents and discusses a background subtraction approach which can detect moving people on a background while also allowing for shadows and highlights. The background is adaptively updated. The approach has been tested in RGB space. The method is efficient with respect to computation time and storage. It only requires to store a background model and brightness distortion values. This allows real-time video processing.

The method has a number of limitations that will be addressed in future work. A static background without moving people is needed during the initialization phase, otherwise it takes a substantial amount of time to reliably classify pixels as foreground. Very dark parts of people are still classified as either background, or shadows. Another limitation is that shadows on very dark backgrounds or several shadows added together will not be detected effectively. Bright highlights are also a problem because of saturation in the camera sensor.

References

1. Y. Bar-Shalom, Th. E. Fortmann: *Tracking and Data Association.* Academic Press, New York (1988).
2. D. Beymer, Ph. McLauchlan, B. Coifman, J. Malik: A real-time computer vision system for measuring traffic parameters. *CVPR'97* (1997) 495–501.

3. A. Blake, M. Isard: *Active Contours.* Springer, Berlin (1998).
4. T. E. Boult, R. Micheals, X. Gao, P. Lewis, C. Power, W. Yin, A. Erkan: Frame-rate omnidirectional surveillance and tracking of camouflaged and occluded targets. *Second IEEE Workshop on Visual Surveillance.* (1999) 48–55.
5. R. Cutler, L. Davis: View-based detection and analysis of periodic motion. *Internat. Conf. Pattern Recognition* (1998) 495–500.
6. A. Elgammal, D. Harwood, L. A. Davis: Non-parametric model for background subtraction. *ICCV'99* (1999) ??–??.
7. W. E. L. Grimson, C. Stauffer, R. Romano, L. Lee: Using adaptive tracking to classify and monitor activities in a site. *Computer Vision and Pattern Recognition* (1998) 1–8.
8. W. E. L. Grimson, C. Stauffer: Adaptive background mixture models for real-time tracking. *CVPR'99* (1999) ??–??.
9. I. Haritaoglu, D. Harwood, L. S. Davis: W^4: Who? When? Where? What? A real time system for detecting and tracking people. *3rd Face and Gesture Recognition Conf.* (1998) 222–227.
10. I. Haritaoglu, R. Cutler, D. Harwood, L. S. Davis: Backpack: Detection of people carrying objects using silhouettes. *Internat. Conf. Computer Vision* (1999) 102–107.
11. I. Haritaoglu, D. Harwood, L. S. Davis: Hydra: Multiple people detection and tracking using silhouettes. *2nd IEEE Workshop on Visual Surveillance* (1999) 6–13.
12. J. Heikkila, O. Silven: A real-time system for monitoring of cyclists and pedestrians. *2nd IEEE Workshop on Visual Surveillance* (1999) 74–81.
13. T. Horprasert, D. Harwood, L. A. Davis: A statistical approach for real-time robust background subtraction and shadow detection. *ICCV'99 Frame Rate Workshop* (1999) 1–19.
14. T. Y. Ivanov, C. Stauffer, A. Bobick, W. E. L. Grimson Video Surveillance of Interactions. *2nd IEEE Workshop on Visual Surveillance* (1999) 82–90.
15. M.-S. Lee: Detecting people in cluttered indoor scenes. *CVPR'00* (2000) ??-??.
16. S. J. McKenna, Y. Raja, S. Gong: Tracking color objects using adaptive mixture models. *Image and Vision Computing* (1999) 780–785.
17. J. Orwell, P. Remagnino, G. A. Jones: Multi-camera color tracking. *2nd IEEE Workshop on Visual Surveillance* (1999) 14–24.
18. R. Rosales, S. Sclaroff: 3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions. *Computer Vision and Pattern Recognition* (1999) 117–123.
19. P. L. Rosin: Thresholding for change detection. *ICCV'98* (1998) 274–279.
20. C. Stauffer, W. E. L. Grimson, Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition* (1999) 246–252.
21. K. Toyama, J. Krumm, B. Brumitt, B. Meyers: Wallflower: Principles and practice of background maintenance. *Internat. Conf. Computer Vision* (1999) 255–261.
22. C. Wren, A. Azabayejani, T. Darrell, A. Pentland: Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Intelligence* (1997) 780–785.