International Journal of Computer Vision 69(3), 277–293, 2006

DOI: 10.1007/s11263-006-7067-x

Robust Tracking Using Foreground-Background Texture Discrimination

HIEU T. NGUYEN*
Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, NY-12180, USA
nguyeh2@rpi.edu

ARNOLD W. M. SMEULDERS
Intelligent Sensory Information Systems, University of Amsterdam, Faculty of Science, Kruislaan 403, NL-1098 SJ, Amsterdam, The Netherlands
smeulders@science.uva.nl

Received January 11, 2005; Revised January 5, 2006; Accepted January 5, 2006

First online version published in May, 2006

Abstract. This paper conceives of tracking as the developing distinction of a foreground against the background. In this manner, fast changes in the object or background appearance can be dealt with. When modelling the target alone (and not its distinction from the background), changes of lighting or changes of viewpoint can invalidate the internal target model. As the main contribution, we propose a new model for the detection of the target using foreground/background texture discrimination. The background is represented as a set of texture patterns. During tracking, the algorithm maintains a set of discriminant functions each distinguishing one pattern in the object region from background patterns in the neighborhood of the object. The idea is to train the foreground/background discrimination dynamically, that is while the tracking develops. In our case, the discriminant functions are efficiently trained online using a differential version of Linear Discriminant Analysis (LDA). Object detection is performed by maximizing the sum of all discriminant functions. The method employs two complementary sources of information: it searches for the image region similar to the target object, and simultaneously it seeks to avoid background patterns seen before. The detection result is therefore less sensitive to sudden changes in the appearance of the object than in methods relying solely on similarity to the target. The experiments show robust performance under severe changes of viewpoint or abrupt changes of lighting.

Keywords: visual tracking, foreground/background discrimination, texture, linear discriminant analysis

1. Introduction

Tracking an object is one of the basic tasks of computer vision. Many approaches to tracking focus on following the dynamics of the target alone or on detecting outliers in a background model. For such approaches, the implicit assumption is that some properties of the foreground or of the background are constant or at least predictable. In the reality of complex scenes, however, any constancy assumption can be violated by a wide range of common circumstances. They include partial or complete occlusion, changes in illumination, and rotation of the target yielding a different facet of the object into view. In these cases, it is best to combine information of both layers and to view tracking as a foreground-background classification problem. In this

*This work was done while the first author was at the Intelligent Sensory Information Systems group, Faculty of Science, University of Amsterdam, The Netherlands.

paper, tracking is conceived as dynamic discrimination of the target against the local background learned online during the progression of the tracking.

In general, tracking is easy when the appearance of the target may be assumed constant from one frame to the other. Then, a good approach is to search for the almost identical region in the next frame. The region is then classified as target and the remaining area as background. This well-known template matching may be too static when the target undergoes variation in its appearances. One solution is to rely on invariant features like the histogram of the template Comaniciu *et al.* (2000) or color invariants Nguyen and Smeulders (2004). Another approach updates the template in answer to changes in target's appearance (Jepson *et al.*, 2001; Nguyen and Smeulders, 2004; Matthews *et al.*, 2004; Ross *et al.*, 2004). The third group of methods use a complete target model which covers the variety of variations in the appearance of the target (Black and Jepson, 1996; Cootes *et al.*, 2002; Ravela *et al.*, 1996). This approach requires the model to be learned in advance.

While template trackers focus on the object appearance, they tend to ignore background data which are equally important. Information taken from the background can serve as negative examples to be avoided by the target. The target is effectively identified as anything outside the background class. A similar behavior is observed in the human vision system where surrounding information is very important in localizing an object (Torralba and Sinha, 2001; Davon, 1977).

To date background information is employed mainly by trackers assuming a fixed camera. Some recent methods relieve the assumption and demonstrate success in using background information for a moving camera (Collins and Liu, 2003; Avidan, 2004; Tao *et al.*, 2000; Wu and Huang , 2000). The remaining limitations, however, include the poor discriminatory power of features used, the sensitivity to the Gaussian model assumption and failure in dealing with drastic changes in the appearance of the target and/or the background.

In extension of preliminary work (Nguyen and Smeulders, 2004), this paper provides a novel tracking method using foreground/background discrimination paradigm. Unlike the aforementioned methods, our method uses texture features for the discrimination and thereby gaining better discriminatory power than color. Rather than maximizing a similarity measure with the target as in common template matching, the target is distinguished from the background by maximizing a discriminant score. Hence, the method will be more robust to changes in the scene parameters. We aim at an algorithm which is robust to severe changes in viewpoint and lighting conditions. Our method does not require any prior knowledge of the target. In addition, we also make no assumptions on the background neither in homogeneity or constancy. The background is permitted to show internal variations as long as target is distinguishable.

An overview of related work is given in Section 2. Section 3 presents our discriminative approach for the target detection. The section discusses the representation of object appearance and how object matching is performed. Section 4 describes the tracking algorithm, the online training of foreground/background texture discriminant functions, and the updating of object and background texture templates. Section 5 presents the specialization of the algorithm for the case of tracking in a confined area. Section 6 shows the tracking results with a comparison with two other state-of-the-art trackers.

2. Related Work

Tracking by discriminating the foreground from the background has been proposed mostly when the camera is fixed. In this case, the background is stationary or changing slowly. Many methods also require the appearance of the target to be known.

In the common background subtraction, the foreground is segmented by thresholding the error between an estimate of the background image and the current image (Wren *et al.*, 1997). Another common background model is a mixture of Gaussians representing multiple states of background color (Friedman and Russell, 1997; Stauffer and Grimson, 1999; Hayman and Eklundh, 2003). In the method of Stauffer *et al.* in Stauffer and Grimson (1999), the background color at each pixel is modelled by a mixture of Gaussians that can change with time to accommodate changes in lighting, repetitive motions, and long term scene changes. This model is further extended in Hayman and Eklundh (2003) where the pixel likelihood is composed of Gaussians learned from the neighboring pixels. In Jojic *et al.* (2003), model the pixel intensity by a mixture of Gaussians taken from an epitome, a repository of Gaussians learned in advance. By allowing pixels at different locations to share common Gaussian components, the epitome provides a condensed representation of the image. In Rittscher *et al.* (2000), propose a rather different approach that uses the hidden Markov model to classify the pixel intensity to various states including foreground, background and shadow. The model is capable of representing the dynamic transition of a pixel between states. It may impose a temporal continuity constraint on the duration of a pixel in a specific state.

Apart from scalar features like intensity or color of individual pixels, recent methods tend to use features composed of responses to a bank of neighborhood filters. Such features are more powerful for the representation of complex image structures. In Sidenbladh and Black (2003), various types of Gaussian derivative filters are designed to extract edge, ridge and motion information. Other methods use Gaussian filters Isard and MacCormick (2001), or Laplacian of Gaussians Sullivan *et al.* (2000) to characterize image textures. The distribution of the filter responses are learned a priori for each location of the background and the foreground. Probabilistic models in use include histograms Sidenbladh and Black (2003), mixture of Gaussians Isard and MacCormick (2001), the Laplacian distribution Sullivan *et al.* (2000) and the Gibbs model Roth *et al.* (2004). The last model can handle some dependency of the filter responses. The likelihood model learned for each layer is then used to compute the image likelihood or the posterior probability of the foreground, to be maximized for tracking. Such maximization implies the maximization of the likelihood ratio between the foreground and the background at the location of the target. Note that in this approach, the appearance of the foreground is assumed to be known.

Background information has appeared useful for tracking with a moving camera. In Tao *et al.* (2000), propose an expectation-maximization (EM) algorithm to cluster image pixels into the two layers using similarity based on intensity, shape and motion. In Wu and Huang (2000), Wu and Huang employ a different EM algorithm for the foreground/background classification integrated with dynamic feature transformation. The transformation is trained by Linear Discriminant Analysis (LDA) on the currently segmented layers. It maps the original color of pixels into a space with better distinction between the foreground class and the background class. In Avidan (2004), tracking is performed by maximizing the score of a Support Vector Machines classifier trained to detect a given category of targets. A large number of training examples of potential targets are needed here to achieve a good classification, requiring offline training. In Collins and Liu (2003), Collins and Liu explicitly note the importance of the foreground/background contrast for tracking performance. They propose to switch the mean-shift tracking algorithm between various combinations of the three color channels as to select the color features that best distinguish the object histogram from the background histogram. The features are ranked by a variance test for the separability of the two histograms. The method uses background data in a local context window only. The authors report improved performance compared to the original mean-shift algorithm. A similar scheme of online feature selection is applied for particle filter tracking in Chen *et al.* (2004). Here the separability of the histograms is measured by the Kullback-Leibler distance.

In conclusion, most methods considering the maximization of the difference between the foreground and the background require a fixed camera and a priori learning of the target's appearance. Few methods can track with a moving camera, but their limitations can be noted. First, the pixel color has limited discriminatory power, while extension of methods to higher dimensional features is not obvious, nor will the use of more dimensions guarantee better results. Secondly, while many methods use EM for density estimation, it produces accurate results only when the assumption on the data distribution is valid in each layer. In particular, EM segmentation works well for statistically simple target and background composition with texture homogeneity. It often fails for cluttered scenes, where the distribution of observations is quickly changing and multimodal. In addition, all the methods appear ineffective in dealing with drastic changes in the appearance of the target and/or the background.

3. Discriminative Target Detection Using Texture Features

In the presented algorithm, the tracked object is detected by maximizing the distinction against the background in the space of texture features. Compared to intensity and color, textures have a higher discriminatory power, while preserving a good locality. The foreground/background distinction is quantified by a discriminant function that is trained online.

3.1. Object Appearance Representation

First we consider the representation of object textures. Let $I(\boldsymbol{p})$ denote the intensity function of the current frame. Assume that the target region is mapped from a reference region Ω via a coordinate transformation φ with parameters $\boldsymbol{\theta}$. Object textures are then analyzed for the transformation compensated image $I(\varphi(\boldsymbol{p};\boldsymbol{\theta}))$ using Gabor filters (Jain and Farrokhnia, 1991). These filters have been used in various applications for visual recognition (Daugman, 1993; Gong *et al.*, 1996) and tracking (Chomat and Crowley, 1999). Each pair of Gabor filters has the form:

$$\begin{aligned} G^s(\boldsymbol{p}) &= \cos\left(\frac{\boldsymbol{p}}{r}\cdot\boldsymbol{n}_\nu\right)\exp\left(-\frac{\|\boldsymbol{p}\|^2}{2\sigma^2}\right) \\ G^a(\boldsymbol{p}) &= \sin\left(\frac{\boldsymbol{p}}{r}\cdot\boldsymbol{n}_\nu\right)\exp\left(-\frac{\|\boldsymbol{p}\|^2}{2\sigma^2}\right), \end{aligned} \tag{1}$$

where σ, r and ν denote the scale, central frequency, and orientation, respectively, and $\boldsymbol{n}_\nu = \{\cos\nu, \sin\nu\}$. Setting these parameters to a range of values creates a bank of filters. Denote them $G_1, \ldots, G_K$. The object texture at pixel $\boldsymbol{p} \in \Omega$ is characterized by vector $\boldsymbol{f}(\boldsymbol{p}) \in \mathbb{R}^K$ which is composed of the response of image $I(\varphi(\boldsymbol{q};\boldsymbol{\theta}))$ to the Gabor filters:

$$[\boldsymbol{f}(\boldsymbol{p})]_k = \sum_{\boldsymbol{q}\in\mathbb{R}^2} G_k(\boldsymbol{p}-\boldsymbol{q})I(\varphi(\boldsymbol{q};\boldsymbol{\theta})), \tag{2}$$

where $[\boldsymbol{f}(\boldsymbol{p})]_k$ denotes the kth component of $\boldsymbol{f}(\boldsymbol{p})$. When necessary, we also use the notation $\boldsymbol{f}(\boldsymbol{p};\boldsymbol{\theta})$ to explicitly indicate the dependence of $\boldsymbol{f}$ on $\boldsymbol{\theta}$. The appearance of a candidate target is represented by the ordered collection of the texture vectors at n sampled pixels $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n \in \Omega$, see Fig.1:

$$\mathcal{F} = [\boldsymbol{f}(\boldsymbol{p}_1), \ldots, \boldsymbol{f}(\boldsymbol{p}_n)]. \tag{3}$$

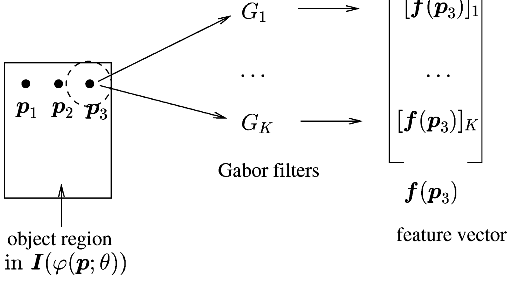

G_1
...
G_K
Gabor filters
$[\boldsymbol{f}(\boldsymbol{p}_3)]_1$
...
$[\boldsymbol{f}(\boldsymbol{p}_3)]_K$
$\boldsymbol{f}(\boldsymbol{p}_3)$
feature vector
$\boldsymbol{p}_1$ $\boldsymbol{p}_2$ $\boldsymbol{p}_3$
object region
in $\boldsymbol{I}(\varphi(\boldsymbol{p};\theta))$

Figure 1. Illustration for the representation of object appearance.

In implementation, $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$ is a down-sampled version of the object region.

3.2. Object Matching

Target detection amounts to finding the parameters $\boldsymbol{\theta}$ that give optimal $\mathcal{F}$ according to two criteria:

1. The similarity between $\mathcal{F}$ and a set of object features $\mathcal{F}_O$, computed as in Eq. (2):

$$\mathcal{F}_O = \left[\boldsymbol{f}_1^o, \ldots, \boldsymbol{f}_n^o\right]. \tag{4}$$

   The order of the vectors in $\mathcal{F}$ and $\mathcal{F}_O$ is also valuable information. That is, $\boldsymbol{f}(\boldsymbol{p}_i)$ should match specifically to $\boldsymbol{f}_i^o$, since both of them represent $\boldsymbol{p}_i$. This order information is ignored in the related approach (Collins and Liu, 2003), as it is based on histogram matching.
2. The contrast between $\mathcal{F}$ and a set of background template features:

$$\mathcal{F}_B = \left\{\boldsymbol{f}_1^b, \ldots, \boldsymbol{f}_M^b\right\}, \quad \boldsymbol{f}_j^b \in \mathbb{R}^K. \tag{5}$$

   These are the texture vectors of the background patterns in the vicinity of the current target position. The set of background patterns is obtained online by sampling in a context window surrounding the object, as the example in Fig. 2. This criterion implies that each $\boldsymbol{f}(\boldsymbol{p}_i)$ in the target window should be distinguished from all $\boldsymbol{f}_j^b$.

As we consider the tracking in the condition of varying appearances of both foreground and background, the two above sets of features are dynamic quantities requiring updating over time.

The search for an image region with the features $\mathcal{F}$ satisfying the two mentioned criteria is performed by maximizing the sum of a set of discriminant functions each computed for one vector in $\mathcal{F}$:

$$\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} g_i(\boldsymbol{f}(\boldsymbol{p}_i;\boldsymbol{\theta})). \tag{6}$$

Here, $g_i(\boldsymbol{f}(\boldsymbol{p}_i;\boldsymbol{\theta}))$ is the discriminant function discriminating the object texture at pixel $\boldsymbol{p}_i$ from all

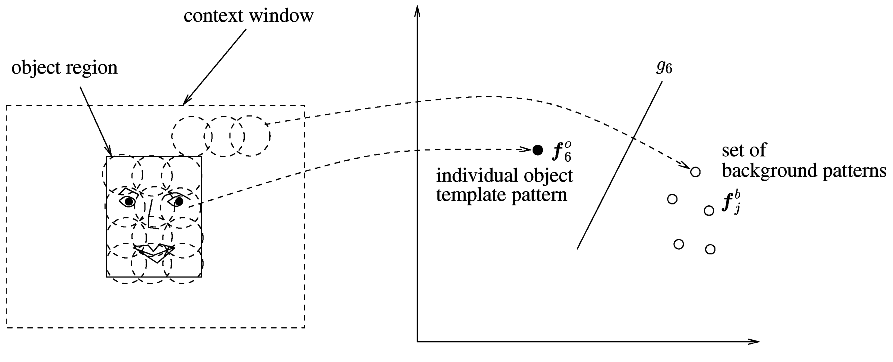

context window
object region
g_6
f_6^o
individual object template pattern
set of background patterns
f_j^b
texture space

Figure 2. Illustration for the construction of target/background texture discriminant functions. The target is then detected by maximizing the discriminant score.

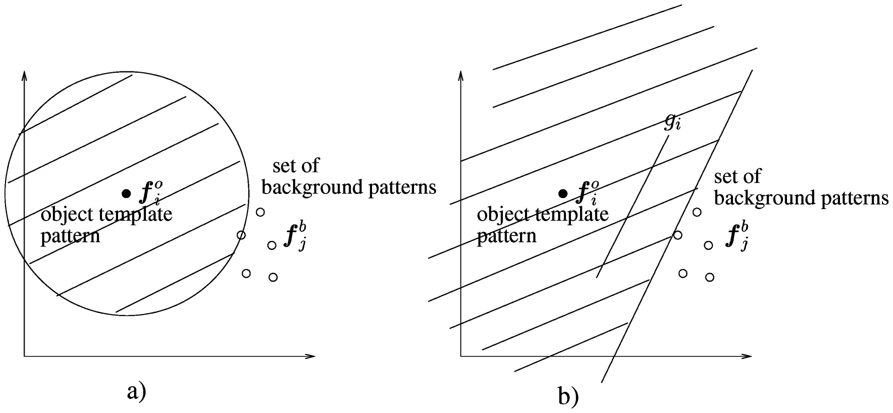


Figure 3. The allowed region for target patterns undergoing appearance changes in (a) the method minimizing the distance to the target model, and (b) the method maximizing the target/background discriminant score.

background textures. We choose g_i to be a linear function:

$$g_i(\boldsymbol{f}) = \boldsymbol{a}_i^T \boldsymbol{f} + b_i, \tag{7}$$

where $\boldsymbol{a}_i \in \mathbb{R}^K, b_i \in \mathbb{R}$ are parameters. Each g_i is trained such that:

$$g_i(\boldsymbol{f}_i^o) > 0 \quad \text{and} \quad \forall \boldsymbol{f} \in \mathcal{F}_{\mathcal{B}} : g_i(\boldsymbol{f}) < 0, \tag{8}$$

see Fig. 2.

For target detection, the method based on maximization of g_i can tolerate more variations in the target's appearance than the common approach that minimizes a distance measure to the target model $\boldsymbol{f}_i^o$. This is illustrated in Fig. 3. In case of distance minimization, the appearance of the target should not vary outside the sphere centered at $\boldsymbol{f}_i^o$ and passing through the nearest background pattern. Otherwise, the algorithm will lock on the nearest background pattern instead of the target. For the proposed discrimination-based method, the allowed region for target patterns undergoing appearance changes is much larger, being the half space resulting from the thresholding of g_i at the maximal score of g_i among the background patterns, as shown in Fig. 3(b).

Furthermore, since g_i is linear with respect to image intensities following from Eqs.(2) and (7), the solution of the maximization of g will remain the same when the intensities in the image are multiplied with any factor. This implies invariance of the tracker even to abrupt changes in global illumination.

With the definition of g_i in (7), Eq. (6) is rewritten as:

$$\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \boldsymbol{a}_i^T \boldsymbol{f}(\boldsymbol{p}_i; \boldsymbol{\theta}) + b_i. \tag{9}$$

The constant parameters b_i do not affect the maximization result, and hence are removed from Eq. (9). Plugging Eq. (2) into Eq. (9) and rearranging terms, we can rewrite the equation for target search as:

$$\max_{\boldsymbol{\theta}} \sum_{\boldsymbol{q} \in \mathbb{R}^2} I(\varphi(\boldsymbol{q}; \boldsymbol{\theta})) w(\boldsymbol{q}), \tag{10}$$

where

$$w(\boldsymbol{q}) = \sum_{i=1}^{n} \sum_{k=1}^{K} a_{ik} G_k(\boldsymbol{p}_i - \boldsymbol{q}), \tag{11}$$

and a_{ik} denotes the kth component of $\boldsymbol{a}_i$. As observed, (10) is the inner product of image $I(\varphi(\boldsymbol{q}; \boldsymbol{\theta}))$ and function w. In particular, if only translational motion is considered, $\varphi(\boldsymbol{q}; \boldsymbol{\theta}) = \boldsymbol{q} + \boldsymbol{\theta}$, and hence, object matching boils down to the maximization of the convolution of the image $I(\boldsymbol{q})$ with the kernel w over possible positions of the target.

3.3. *Construction of the Discriminant Functions*

In principle, any linear classifier can be used for training g_i. However, in view of the dynamic characteristic of the training set, the selected classifier should allow for training in incremental mode. Also it should be computationally tractable in real-time tracking. To this end, we adopt the LDA (Linear Discriminant Analysis) Duda *et al.* (2001). Function g_i minimizes the cost function:

$$\min_{\boldsymbol{a}_i, b_i} \left(\boldsymbol{a}_i^T \boldsymbol{f}_i^o + b_i - 1\right)^2 + \sum_{j=1}^{M} \alpha_j \left(\boldsymbol{a}_i^T \boldsymbol{f}_j^b + b_i + 1\right)^2 + \frac{\lambda}{2} \|\boldsymbol{a}_i\|^2, \tag{12}$$

over $\boldsymbol{a}_i$ and b_i. Here, α_j are the weighting coefficients of the background patterns normalized so that $\sum_{j=1}^{M} \alpha_j = 1$. They are introduced in the general mode of tracking in a non-confined area, where the background set is constantly expanded, to put emphasis on recently observed patterns over old patterns. The regularization term $\frac{\lambda}{2} \|\boldsymbol{a}_i\|^2$ is added in order to overcome the numerical instability due to high-dimensionality of texture features.

The solution of Eq. (12) is obtained in closed form:

$$\boldsymbol{a}_i = \kappa_i [\lambda \mathbf{I} + \mathbf{B}]^{-1} \left[\boldsymbol{f}_i^o - \bar{\boldsymbol{f}}^b\right] \tag{13}$$

where

$$\bar{\boldsymbol{f}}^b = \sum_{j=1}^{M} \alpha_j \boldsymbol{f}_j^b, \tag{14}$$

$$\mathbf{B} = \sum_{j=1}^{M} \alpha_j \left[\boldsymbol{f}_j^b - \bar{\boldsymbol{f}}^b\right]\left[\boldsymbol{f}_j^b - \bar{\boldsymbol{f}}^b\right]^T, \tag{15}$$

$$\kappa_i = \frac{1}{1 + \frac{1}{2}\left[\boldsymbol{f}_i^o - \bar{\boldsymbol{f}}^b\right]^T [\lambda \mathbf{I} + \mathbf{B}]^{-1} \left[\boldsymbol{f}_i^o - \bar{\boldsymbol{f}}^b\right]}. \tag{16}$$

The discriminant functions depend only on the object features $\boldsymbol{f}_i^o$, the mean vector $\bar{\boldsymbol{f}}^b$ and the covariance matrix $\mathbf{B}$ of all texture patterns in the background. These quantities can efficiently be updated during tracking.

Note that the background is usually non-uniform. Therefore one mean pattern $\bar{\boldsymbol{f}}^b$ is unlikely to be sufficient for an accurate representation of background textures. The diversity of background patterns is encoded in the covariance matrix $\mathbf{B}$ instead. This representation model would be a crude approximation for a large set of background patterns. However, it can represent provides the small set of patterns in the context window with reasonable accuracy.

4. Tracking in a Non-Confined Area

This section describes the algorithm for the general case without any restriction on the area where the tracking takes place. On the other hand, Section 5 will give a version of the algorithm for the tracking in a confined area and in the condition where a priori learning of a background model is possible.

The data flow of the algorithm is given in Fig. 4.

4.1. *Updating of the Foreground Model*

The feature vectors $\boldsymbol{f}_i^o$ need to be updated constantly to follow up the varying appearance of the foreground. On the other hand, a hasty updating is

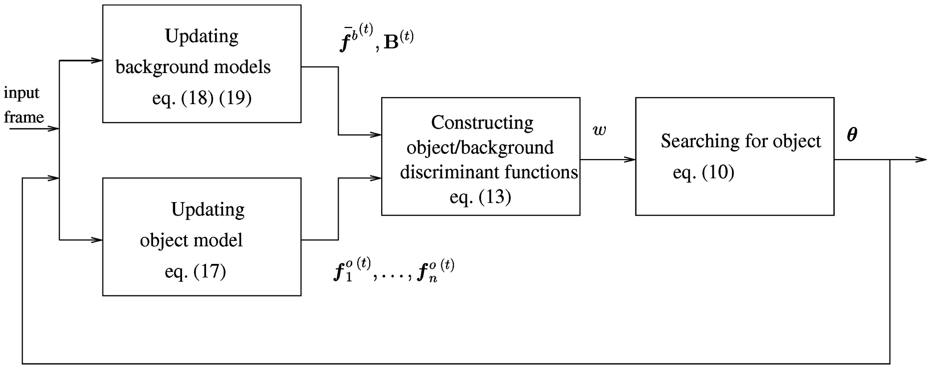


Figure 4. The flow diagram of the tracking algorithm.

sensitive to sudden tracking failure and stimulates drift of the target window. A slower updating can be obtained by a compromise between the latest model and the new data. This is done by the simple decay:

$$\boldsymbol{f}_i^{o\,(t)} = (1-\gamma)\boldsymbol{f}_i^{o\,(t-1)} + \gamma \boldsymbol{f}(\boldsymbol{p}_i;\boldsymbol{\theta}) \tag{17}$$

where the superscript (t) denotes time, and $0 < \gamma < 1$ is the predefined decay coefficient.

4.2. Updating of the Background Model

The set of background patterns $\mathcal{F}_{\mathcal{B}}$ is sampled online in a context window surrounding the target window, see Fig. 2.

As the background moves, constantly new patterns enter the context window and some other patterns leave the window. $\mathcal{F}_{\mathcal{B}}$ is expanded to include newly appearing patterns. Each new pattern $\boldsymbol{f}_j^b$ is given a time-decaying weighting coefficient α_j which controls the influence of the patterns in Eq. (12). In other words, we keep all observed patterns in $\mathcal{F}_{\mathcal{B}}$ but gradually decrease their weights α_j with time to enable the tracker to forget obsolete patterns that have left the context window.

At every tracking step, the Gabor filters are applied for image $I(\boldsymbol{p})$ at m fixed locations in the context window, yielding m new background texture vectors denoted by $\boldsymbol{f}_{M+1}^b, \ldots, \boldsymbol{f}_{M+m}^b$, where M is the current number of background patterns. The weighting coefficients are then distributed over the new and the old elements in $\mathcal{F}_{\mathcal{B}}$ so that the total weight of the new patterns amounts to γ while that of the old patterns is $1-\gamma$. Therefore, each new pattern is assigned an equal weighting coefficient $\alpha_j = \gamma/m$. Meantime, the coefficient of every existing pattern in $\mathcal{F}_{\mathcal{B}}$ is re-scaled with the factor $1-\gamma$. Let $\bar{\boldsymbol{f}}^b{}_{\text{new}} = \frac{1}{m}\sum_{j=M+1}^{M+m} \boldsymbol{f}_j^b$. The update equations for $\bar{\boldsymbol{f}}^b$ and $\boldsymbol{B}$ are:

$$\bar{\boldsymbol{f}}^{b^{(t)}} = (1-\gamma)\bar{\boldsymbol{f}}^{b^{(t-1)}} + \gamma \bar{\boldsymbol{f}}^b_{\text{new}}, \tag{18}$$

$$\begin{aligned} \boldsymbol{B}^{(t)} &= (1-\gamma)\boldsymbol{B}^{(t-1)} + (1-\gamma)\bar{\boldsymbol{f}}^{b^{(t-1)}}\bar{\boldsymbol{f}}^{b^{(t-1)}\,T} \\ &\quad - \bar{\boldsymbol{f}}^{b^{(t)}}\bar{\boldsymbol{f}}^{b^{(t)}\,T} + \frac{\gamma}{m}\sum_{j=M+1}^{M+m} \boldsymbol{f}_j^b \boldsymbol{f}_j^{b\,T}. \end{aligned} \tag{19}$$

The equations allow for efficient updating of the background model in the incremental mode.

5. Tracking in a Confined Area

This section presents the specialization of the algorithm for tracking in a confined area. In many surveillance applications, the background is limited to a confined area like a room, a corridor or a courtyard. In this case, the set of background patterns is fixed. This allows for the construction of a stable and complete representation model for all background samples. Such a background model needs no updating, making the tracking more robust to drifts compared to the short-term model used in the previous section.

The algorithm in this case differs from the general version in Section 4 by the construction of the background model only, see Fig. 5.

We learn a complete background model from a few training images which are taken a priori and cover all main views of the area. Note that recording background images in advance is often more feasible than the common approach that records views of the target (Black and Jepson, 1996). One simple global representation

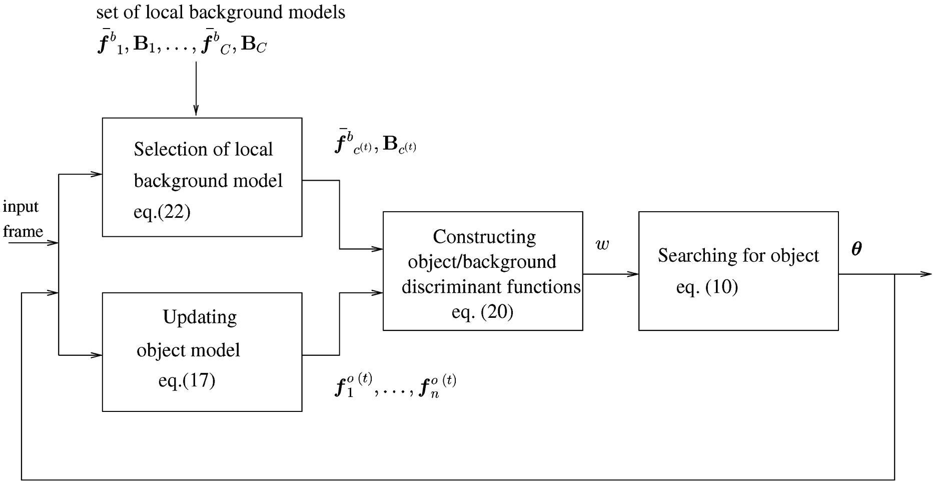

set of local background models
$\bar{\boldsymbol{f}}^b{}_1, \mathbf{B}_1, \dots, \bar{\boldsymbol{f}}^b{}_C, \mathbf{B}_C$
Selection of local background model eq.(22)
$\bar{\boldsymbol{f}}^b{}_{c^{(t)}}, \mathbf{B}_{c^{(t)}}$
input frame
Constructing object/background discriminant functions eq. (20)
w
Searching for object eq. (10)
$\boldsymbol{\theta}$
Updating object model eq.(17)
$\boldsymbol{f}_1^{o\,(t)}, \dots, \boldsymbol{f}_n^{o\,(t)}$

Figure 5. The flow diagram of tracking in a confined area.

model will not suffice for the usually large number of all possible background patterns. Therefore, we represent the patterns by a set of local models. From the training images, a repository of overlapping contexts are extracted, each a rectangular region in a background training image. We call them reference contexts. For each reference context, a mean vector $\bar{f}^b_c$ and a covariance matrix $\mathbf{B}_c$ of the Gabor feature vectors are calculated as in Eqs. (14) and (15), respectively, with equal weights for all patterns in the context. Let C be the number of the contexts. We then have $c = 1, \dots, C$ background models: $\bar{f}^b_1, \mathbf{B}_1, \dots, \bar{f}^b_C, \mathbf{B}_C$. In each tracking step, the algorithm selects just one local model with index $c^{(t)}$ to construct the vectors $\boldsymbol{a}_i$:

$$\boldsymbol{a}_i^{(t)} = \kappa_i^{(t)}\left[\lambda^{(t)}\mathbf{I} + \mathbf{B}_{c^{(t)}}\right]^{-1}\left[f_i^{o\,(t)} - \bar{f}^b_{c^{(t)}}\right]. \qquad (20)$$

The context indicated by this index is considered most similar to the image data in the surrounding of the target window. The use of the reference contexts reduces the number of background patterns at t while allowing for a flexible representation of the background under all camera viewpoints.

To find the optimal $c^{(t)}$, each background pattern in the current context window is matched to every reference context in the repository by the Mahalanobis distance. The relevance of the current context to a reference context is measured by the sum of Mahalanobis distances of all background patterns in the current context to the reference context:

$$\mathcal{M}_c = \sum_{f^b \,\in\, \text{current context}} \left(f^b - \bar{f}^b_c\right)^T \mathbf{B}_c^{-1}\left(f^b - \bar{f}^b_c\right). \qquad (21)$$

The reference context with the minimal $\mathcal{M}_c$ is selected:

$$c^{(t)} = \arg\min_{1 \le c \le C} \mathcal{M}_c. \qquad (22)$$

The mean vector and the covariance matrix of this index are then used for the computation of the foreground/backround discriminant function as in Eq. (20).

6. Results

This section shows tracking results for real video sequences, demonstrating the capabilities of the proposed algorithm. We also provide insight in the limitations by actively seeking for conditions of failure.

In the current implementation, translational motion is assumed. For the extraction of texture features, the algorithm uses a set of twelve Gabor filters created for scale $\sigma = 4$ pixels, and $r = \frac{1}{2}\sigma$ and six directions of ν equally spaced by 30°. The increase of the number of directions does not change the tracking performance. The selection of the scale is based on a trade-off between the discrimination power and the representation accuracy. Large size filters have more power in discriminating foreground textures from background textures but also decrease the representation accuracy due to the overlap with the background. The target region is set to a rectangle although it is not essential. Object pixels $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$ are sampled with a spacing by σ. The same spacing is applied for the background pixels in the context window. For the updating of the object and background texture templates, we set the decay coefficient $\gamma = 0.05$ in accordance with the appearance change. The regularization coefficient λ is set to a fraction of the trace of the covariance matrix: $\lambda = \lambda' \mathrm{tr}(\mathbf{B})$, making the resulting discriminant functions invariant to illumination changes. Experiments show that satisfactory regularization is achieved with the values of λ' in the range 0.001–0.1. We have used $\lambda' = 0.004$.

6.1. *Tracking under Severe Variations of Target Appearance*

This subsection shows the tracking performance of the algorithm under severe changes of lighting and viewpoint. We also compare the proposed algorithm with three other state-of-the-art trackers, namely:

1. an intensity sum-of-squared-differences (SSD) tracker (Lucas and Kanade , 1981) modified to achieve better performance,
2. the tracker of Collins and Liu (2003),
3. the WSL tracker by Jepson *et al.* (2001).

Note that the SSD tracker uses a fixed template, while the proposed algorithm updates the foreground model during tracking. So, to remove any advantage of the proposed algorithm that might come from the model updating, we also update the template of the SSD tracker in the same way. In every frame the template is recalculated as a weighted average between the latest template and new intensity data, where the weight of the new data is $\gamma = 0.05$. The averaging operation results in a smoothed template which is also resilient to viewpoint changes in some degree. Unlike the proposed approach, the SSD algorithm does not use background information.

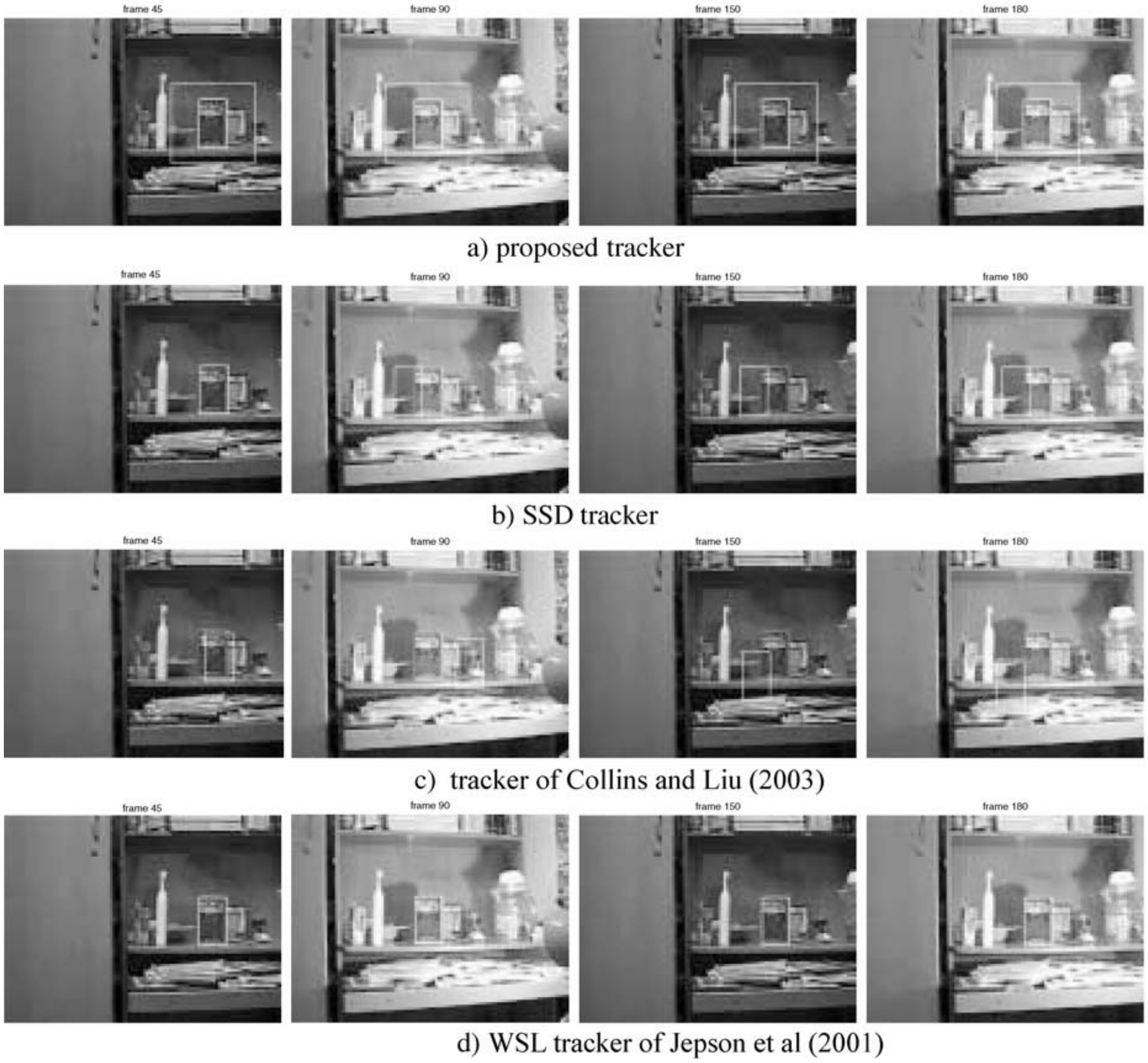
The method of Collins and Liu (2003) involves a set of mean-shift trackers (Comaniciu *et al.*, 2000). All the mean-shift trackers use the same scale parameter that is set equal to the height of target window. As suggested in Collins and Liu (2003), in each frame we update the reference histogram as a weighted average of the histogram of the current target region and the original histogram of the target in the first frame. In the experiment, we used equal weights for both histograms.

The results of the WSL algorithm were provided by the authors of Jepson *et al.* (2001) themselves.

The above algorithms have been applied for several test image sequences, and the results are shown in Figs. 6, 7, 8, and 9. In this figures, sub-figures a, b, c, d show the results of the proposed algorithm, the SSD tracker, the method of Collins and Liu (2003), and the WSL tracker in Jepson *et al.* (2001) respectively.

6.1.1. Severe Changes of Lighting. In Fig. 6, the target is a book placed upright on a shelf. The camera pans back and forth. The table is lit with two light sources: a main lamp of the room and a smaller table lamp placed nearby. In the beginning, the table lamp is off. When the table lamp turns on, the target becomes brighter. As a consequence, the SSD tracker locks on the shadow of the book on the wall, because this region is more similar to the current template, see frame 90 Fig. 6(b). The tracker of Collins and Liu (2003) also lost the track. Furthermore, it developed a drift even before the illumination change, possibly due to the low discriminatory power of the color histogram. The change of illumination did not affect the proposed tracker together with the WSL tracker as shown in Fig. 6(a) and (d). Despite many more sudden changes which were created by switching one of the lamps on and off, the results of both trackers remain accurate. The success of the WSL tracker is due to the illumination independence of the appearance model proposed by this method, as well as the adaptiveness to new appearances of the target.

6.1.2. Severe Changes of Viewpoint. Figure 7 shows an example of head tracking. Initially the head is at the frontal view pose. The background is non-uniform, and the camera pans back and forth, keeping the head in the center while showing completely different views of the head. The proposed tracker could capture even the back view of the head previously unseen. As shown in Fig. 7(b), the SSD tracker also exhibits a robust performance under slight pose changes of the head, but it gives wrong results when the head pose changes severely as in frames 26 and 52. Nevertheless, the SSD tracker did not lose track and well recovered from the drift when the head returned back to the frontal view. This success can be explained by the uniqueness of the black hair in the scene. Similar results are observed in



a) proposed tracker

b) SSD tracker

c) tracker of Collins and Liu (2003)

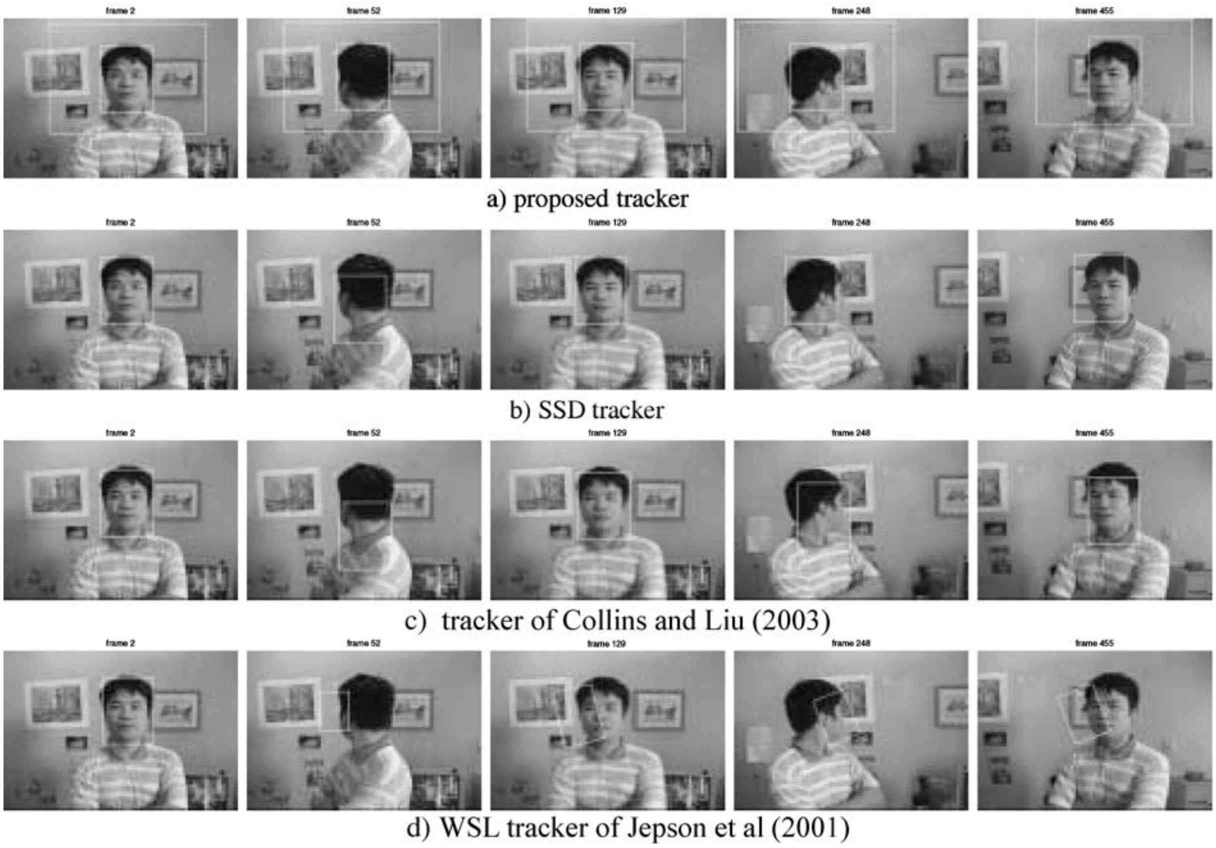
d) WSL tracker of Jepson et al (2001)

Figure 6. Tracking results of different trackers under severe change in illumination. The outer rectangle in (a) indicates the context window.

Fig. 7(c). Where in the previous sequence, the WSL-tracker followed all changes in illumination intensity, the results are not very good here since the WSL-method was not designed to handle severe changes in viewpoint.

A clear example where the proposed algorithm outperforms the other trackers is shown in Fig. 8. The figure shows the tracking result for a sequence where a mousepad is flipped around its vertical axis, switching between the light front side and the completely black back side. The proposed algorithm recovered perfectly when the unseen dark side comes into view. It could also successfully lock back on the front side as in frame 108. The results indicate that the proposed tracker prefers an unseen object region over a background region. As shown in Fig. 8(b), (c d), all the three other trackers drifted off at the first flip at frame 30, because these trackers still look for the front side of the mousepad which has a similar color as the wall.

6.1.3. Occlusions. All the four algorithms were successful in many sequences with partial and short-time occlusions. In the example shown in Fig. 9, the WSL tracker gives the best result as it tracks the rotational motion accurately. The algorithm (Collins



a) proposed tracker

b) SSD tracker

c) tracker of Collins and Liu (2003)

d) WSL tracker of Jepson et al (2001)

Figure 7. Tracking results under rotation of object.

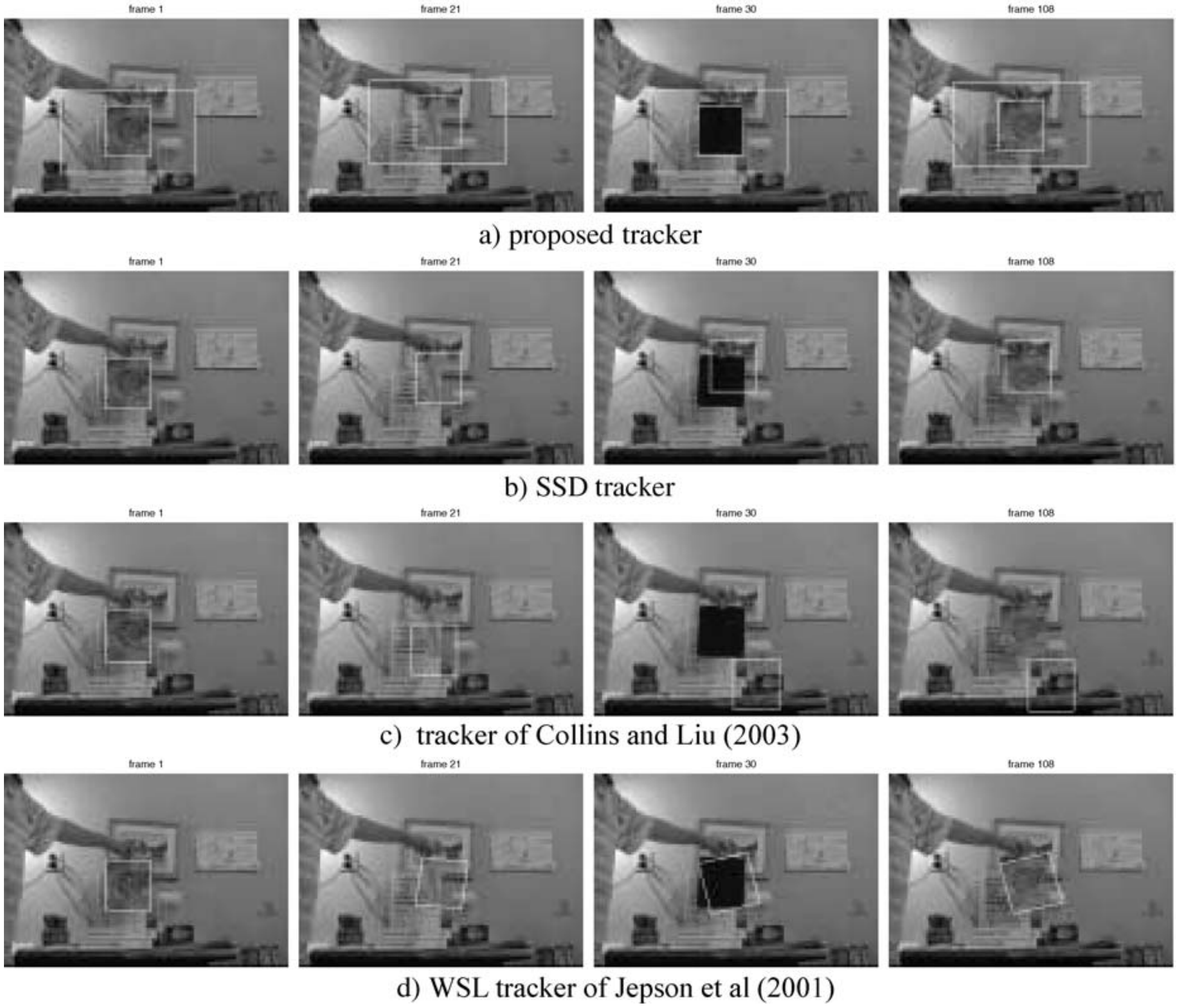
and Liu, 2003) develops minor drifts sometime. The robustness of the proposed algorithm to occlusions is explained by the temporal filter in Section 4.1 which slows the template updating and keeps the templates valid even when the target is occluded. However, slow updating appears not effective in dealing with severe occlusions. For a more sophisticated technique of occlusion handling, see also Nguyen and Smeulders (2004).

To conclude this subsection, Fig. 10 demonstrates the tracking result of the proposed algorithm for the Dudek face sequence used in Jepson *et al.* (2001). In this sequence, tracking the face of the man is difficult due to many challenging conditions including changes in illumination, head pose, facial expression, and background settings, as well as partial occlusions and zooms. However, the proposed algorithm produces good results. It does not calculate the scaling, but follows the face accurately, and for most of the time, keeps the face in the center of the target window.

6.2. *Failure Cases*

Apart from the strengths of the proposed algorithm, we deliberately searched for cases of failure:

6.2.1. Background Regions of Similar Appearance as the Target. As the algorithm is based on foreground/background discrimination, its performance can degrade in case of poor separability between two layers. As an example,Fig. 11 shows a sequence which is somewhat similar to the sequence of Fig. 8. The main difference, however, is a dark region under the table. Although the algorithm successfully detects the back side of the mousepad after the first aspect change in Fig. 11b, it did not survive the second change and locked on the dark region of the background. The explanation for the failure is the similarity in appearance of the dark region and the target before the aspect transition. It is important to note that the algorithm does not avoid this background region despite its presence in the context



a) proposed tracker

b) SSD tracker

c) tracker of Collins and Liu (2003)

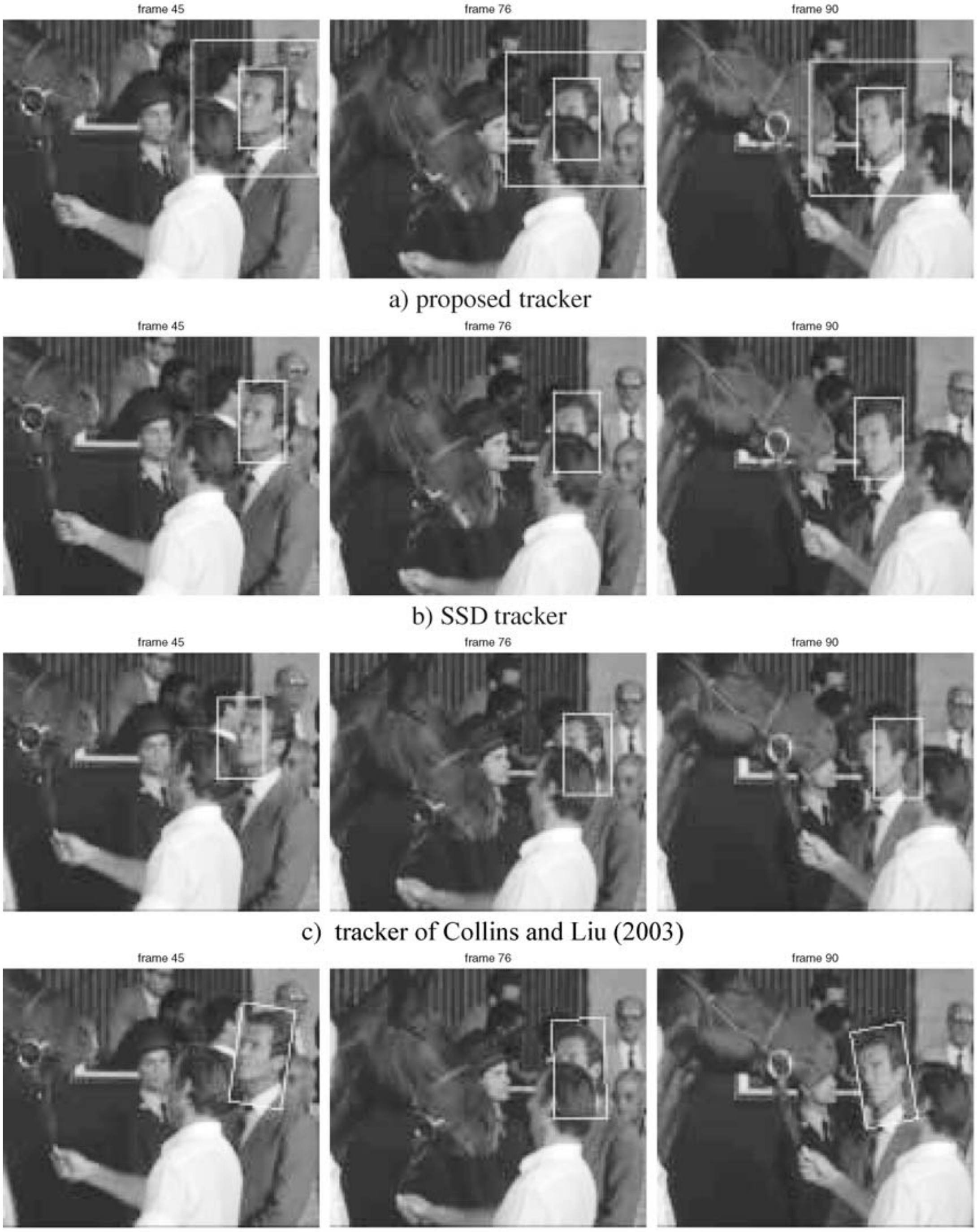
d) WSL tracker of Jepson et al (2001)

Figure 8. Tracking results under severe change of view point.

window for a long period of time. This indicates a shortcoming of LDA which allows the discriminant function to have positive score even for some negative training examples.

6.2.2. Highly Textured Object Under Viewpoint Changes. Figure 12 shows another failure case. In this sequence, the camera keeps tracking a cube-shaped child's toy while slowly moving around it. The background is highly cluttered with strong edges, making it difficult to achieve an accurate representation of the background patterns. In addition, the target is highly textured, causing drift when the camera moves to another aspect of the target as shown in Fig. 12(b). Here, the ability of the algorithm to take the spatial relationship between object patterns into account becomes a drawback as the tracker "remembers" a high energy pattern of the object too long. It sticks to the pattern during aspect transition. Once the target window has drifted, it cannot recover since the tracker now wrongly assigns the object to be a part of the background.

6.2.3. Gradual Drift. Adaptiveness can also become a problem when models are wrongly updated. While the slow updating scheme of the templates actually helps to overcome sudden appearance changes in both the layers, it appears ineffective in dealing with gradual drifts. This also causes tracking failure in other adaptive tracking algorithms. The problem can also be observed



a) proposed tracker

b) SSD tracker

c) tracker of Collins and Liu (2003)

d) WSL tracker of Jepson et al (2001)

Figure 9. Tracking results under occlusion.

in Fig. 12. The small drifts are accumulated slowly over a period of time, and eventually ruin the templates. It is hard to solve this problem in the condition of adaptive models used for both foreground and background. A thorough solution, however, can be achieved when the model of one layer is stable.

We remark that the mentioned problems remain relevant for other standard tracking algorithms as well

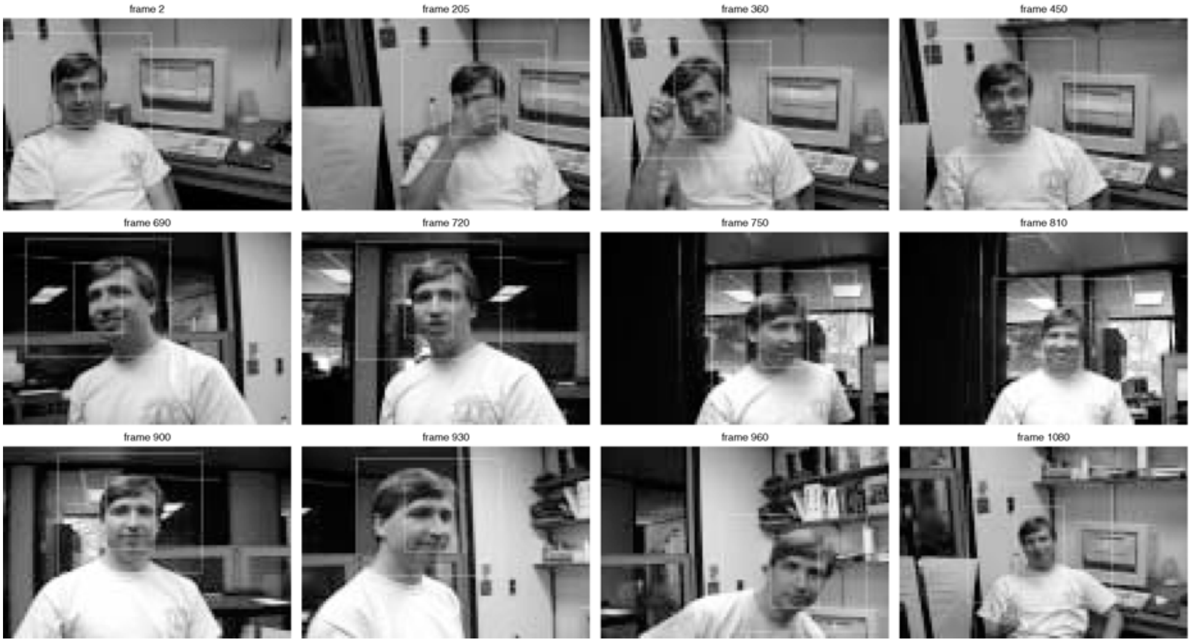


Figure 10. The results of the proposed tracker for the Dudek face sequence used in Jepson *et al.* (2001).

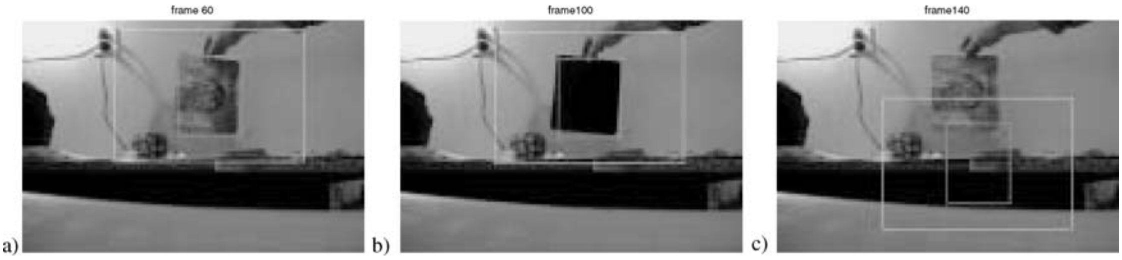


Figure 11. A failure example of the proposed algorithm.

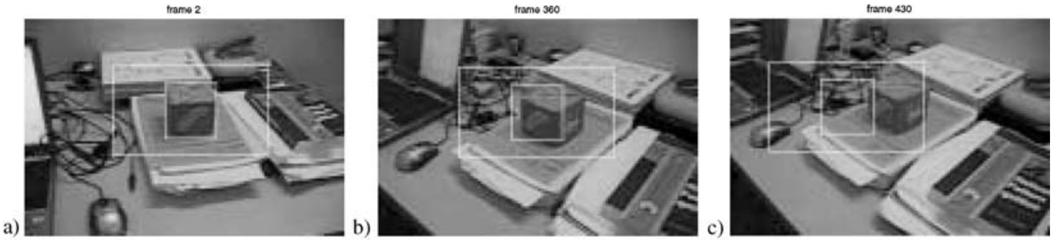


Figure 12. A failure example of the proposed algorithm.

including the SSD trackers. Note also that while the last two conditions are very hard for any updating scheme, they may be repaired in case of tracking in a confined environment as discussed in Section 5 and for experiments in the next paragraph.

6.3. *Quantitative Evaluation*

We have also evaluated the methods quantitatively. The tracking accuracy is measured as the ratio of the overlap between the groundtruth object region and the target

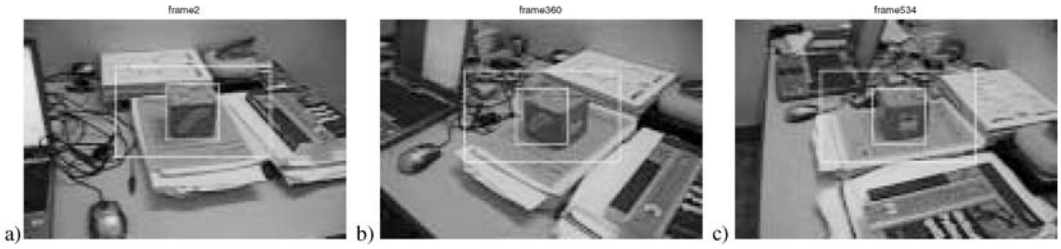


Figure 13. The result of the version of the proposed algorithm that uses a prior background model for the sequence in Fig. 12.

Figure 14. Some of the background training images used for the experiment in Fig. 13.

window to the average area of these two regions:

$$\omega = 2\frac{|R_{\text{object}} \cap R_{\text{window}}|}{|R_{\text{object}}| + |R_{\text{window}}|} \tag{23}$$

where R_{object} denotes the groundtruth region of the object, R_{window} denotes the target window provided by the tracker, and $||$ denotes the area. Groundtruth data were created by manually selecting the image region considered by human as the best match of the object.

We have collected a test set of 22 videos each has at least one difficult condition like change of view point, change of illumination or partial occlusion. The value of ω is evaluated for each frame. For each sequence we then calculate the following statistics: $\omega_{\min}$: the minimum value of ω, $\bar{\omega}$: the average of ω, and τ: the fraction of time where ω exceeds 50%. These quantities are then averaged over the test sequences and shown in Table 1. Note that the numbers shown should not be considered as accurate statistics for the algorithms due to the small size of the test set. However, they do provide an accurate comparison of performance of the methods evaluated.

Table 1. Quantitative performance measures for the different trackers. $\bar{\omega}$: the average overlap of the detected target and the groundtruth, $\omega_{\min}$: the minimum overlap, τ the fraction of time where ω exceeds 50%. These quantities are averaged over 22 test sequences.

	$\bar{\omega}$	$\omega_{\min}$	τ
Proposed tracker	80%	62%	13%
SSD tracker	65%	38%	26%
Collins and Liu tracker	50%	16%	42%

6.4. *Tracking in a Confined Environment*

Figure 13 shows the results of the version of the algorithm that uses a complete background model. We used five training images of the background taken from five different view angles as shown in Fig. 14. From each image, 48 reference contexts were extracted, giving 240 contexts in total. Due to the low dimensionality of the feature space, the computational expense of the selection of the appropriate context is just marginal compared to the target search. On the other hand, the stable model obtained yields better tracking results. As observed in Fig. 13(b), the target frame is still shifted a bit into the background at the aspect transition, but it did not loose the track and locked back to the right object in Fig.13(c).

7. Conclusion

The paper has shown the advantage of discriminating the object information from the background information for object tracking under severe changes in target appearance, especially changes in viewpoint and changes in illumination. We propose a new approach to tracking based on discrimination of object textures from background textures. The texture features allow for a good separation between the foreground and the background. While the representation of the background by a set of patterns is robust to background motion, weighting the patterns in a time-decaying manner allows to get rid of outdated patterns. The algorithm keeps track of a set of discriminant functions each separating one pattern in the object region from the background patterns. The target is detected by the maximization of the sum of those discriminant functions, taking into account the spatial distribution of object textures. The discriminative approach prevents the tracker from accepting background patterns, and therefore enables the tracker to identify the correct object region even in case of substantial changes in object appearance.

Acknowledgments

The authors would like to thank anonymous reviewers for their thorough reading and analysis of the paper and the improvements it has brought to the experiments and the text. We are also grateful to Dr. Thomas El-Maraghi and Prof. Allan Jepson for providing the results of their WSL tracker for our test video sequences.

References

Avidan, S. 2004. Support vector tracking. *IEEE Trans. on PAMI*, 26(8):1064–1072.

Black, M.J. and Jepson, A.D. 1996. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *Proc. of European Conf. on Computer Vision*, pp. 329–342.

Chen, H.T., Liu, T.L., and Fuh, C.S. 2004. Probabilistic tracking with adaptive feature selection. In *Proc. of Int. Conf. on Pattern Recogn. ICPR04*, pp. II: 736–739.

Chomat, O. and Crowley, J.L. 1999. Probabilistic recognition of activity using local appearance. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pp. II: 104–109.

Collins, R. and Liu, Y. 2003. On-line selection of discriminative tracking features. In *Proc. IEEE Conf. on Computer Vision*, pp. 346–352.

Comaniciu, D., Ramesh, V., and Meer, P. 2000. Real-time tracking of non-rigid objects using mean shift. In *CVPR00*, pp. II: 142–149.

Cootes, T.F., Wheeler, G.V., Walker, K.N., and Taylor, C.J. 2002. View-based active appearance models. *Image and Vision Computing*, 20(9–10):657–664.

Daugman, J.G. 1993. High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans. on PAMI*, 15(11):1148–1161.

Davon, D. 1977. Forest before the trees: The precedence of global features in visual perception. *Cognitive Psychology*, 9:353–383.

Duda, P. O., Hart, P. E., and Stork, D. G. 2001. *Pattern Classification*. Wiley: New York.

Friedman, N. and Russell, S. 1997. Image segmentation in video sequences: A probabilistic approach. In *Proc. of 13th Conf. on Uncertainty in Artificial Intelligence*, pp. 175–181.

Gong, S., McKenna, S.J., and Collins J.J. 1996. An investigation into face pose distributions. In *Proc. of 2nd Inter. Conf. on Automated Face and Gesture Recognition*, pp. 265. Killington, Vermont.

Hayman, E. and Eklundh, J.O. 2003. Statistical background subtraction for a mobile observer. In *Proc. IEEE Conf. on Computer Vision*, pp. 67–74.

Isard, M., and MacCormick, J.P. 2001. BraMBLe: A Bayesian multiple-blob tracker. In *Proc. IEEE Conf. on Computer Vision*, pp. II: 34–41.

Jain, A.K. and Farrokhnia, F. 1991. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12): 1167–1186.

Jepson, A.D., Fleet, D.J., and El-Maraghi, T.F. 2001. Robust online appearance models for visual tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recogn., CVPR01*, pp. I: 415–422.

Jojic, N., Frey, B.J., and Kannan, A. 2003. Epitomic analysis of appearance and shape. In *Proc. IEEE Conf. on Computer Vision*, pp. 34–41.

Lucas, B.D. and Kanade, T. 1981. An iterative image registration technquie with an application to stereo vision. In *Proc. DARPA Imaging Understanding Workshop*, pp. 121–130.

Matthews, I., Ishikawa, T., and Baker, S. 2004. The template update problem. *IEEE Trans. on PAMI*, 26(6):810–815.

Nguyen, H.T. and Smeulders, A.W.M. 2004. Fast occluded object tracking by a robust appearance filter. *IEEE Trans. on PAMI*, 26(8):1099–1104.

Nguyen, H.T. and Smeulders, A.W.M. 2004. Tracking aspects of the foreground against the background. In *Proc. of European Conf. on Computer Vision*, pp. Vol II: 446–456.

Ravela, S., Draper, B.A., Lim, J., and Weiss, R. 1996. Tracking object motion across aspect changes for augmented reality. In *ARPA Image Understanding Workshop*, pp. 1345–1352.

Rittscher, J., Kato, J., Joga, S., and Blake, A. 2000. A probabilistic background model for tracking. In *Proc. of European Conf. on Computer Vision*, pp. II: 336–350.

Ross, D., Lim, J., and Yang, M.H. 2004. Adaptive probabilistic visual tracking with incremental subspace update. In *Proc. of European Conf. on Computer Vision*, pp. Vol II: 470–482.

Roth, S., Sigal, L., and Black, M.J. 2004. Gibbs likelihoods for bayesian tracking. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pp. I: 886–893.

Sidenbladh, H. and Black, M.J. 2003. Learning the statistics of people in images and video. *Int. J. Computer Vision*, 54(1):181–207.

Stauffer, C. and Grimson, W.E.L. 1999. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pp. II: 246–252.

Sullivan, J., Blake, A., and Rittscher, J. 2000. Statistical foreground modelling for object localisation. In *Proc. of European Conf. on Computer Vision*, pp. II: 307–323.

Tao, H., Sawhney, H.S., and Kumar, R. 2000. Dynamic layer representation with applications to tracking. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn., CVPR00*, pp. II:134–141.

Torralba, A. and Sinha, P. 2001. Statistical context priming for object detection. In *Proc. IEEE Conf. on Computer Vision*.

Wren, C.R., Azarbayejani, A., Darrell, T.J., and Pentland, A.P. 1997. Pfinder: Real-time tracking of the human body. *IEEE Trans. on PAMI*, 19(7):780–785.

Wu, Y. and Huang, Th. S. 2000. Color tracking by transductive learning. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pp. I:133–138, 2000.