Tracking Aspects of the Foreground against the Background

Hieu T. Nguyen and Arnold Smeulders

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

Abstract. In object tracking, change of object aspect is a cause of failure due to significant changes of object appearances. The paper proposes an approach to this problem without a priori learning object views. The object identification relies on a discriminative model using both object and background appearances. The background is represented as a set of texture patterns. The tracking algorithm then maintains a set of discriminant functions each recognizing a pattern in the object region against the background patterns that are currently relevant. Object matching is then performed efficiently by maximization of the sum of the discriminant functions over all object patterns. As a result, the tracker searches for the region that matches the target object and it also avoids background patterns seen before. The results of the experiment show that the proposed tracker is robust to even severe aspect changes when unseen views of the object come into view.

1 Introduction

In visual object tracking, handling severe changes of viewpoint or object aspect has always been challenging. Change of aspect may be the result either of a self rotation of the tracked object or of a change of camera position. In either case, it is difficult to follow changing appearances of the object due to self-occlusion and disclosure at some parts of the object, and due to the lack of a reliable way for recovering the 3D motion parameters [1].

Current tracking methods handle viewpoint changes in two approaches: invariant-based and view-based. In the invariant-based approach, object matching is performed using appearance features invariant to viewpoint. The mean-shift tracking method [5], for example, uses histograms which are invariant to some degree of viewpoint change. Methods using a temporally smoothed and adaptive template also achieve some resistance to slight changes of object orientation [13,11]. The invariant-based methods, however, likely fail in case of severe changes of viewpoint, when a completely unseen side of the object moves into view.

View-based methods use considerably more a priori knowledge on the object. Many methods record a complete set of object views in advance [2,6,14]. An appearance model is then learned from this set to recognize any possible view of the object. The eigentracker by Black and Jepson [2], for example, extracts a few eigenimages from a set of object views. During tracking, the object region is localized simply by minimizing

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 446–456, 2004.

the distance to the subspace spanned by the eigenimages. The disadvantage of the view-based methods is that they need an a priori trained appearance model which is not always available in practice. Some other methods construct the view set online [12]. They store the key frames of the tracking results so as to recognize any previously seen object view when it appears again. There is no guarantee, however, that an unseen view can be identified. A fusion of offline and online learning of view information is proposed in [15].

This paper aims for robust tracking under severe changes of viewpoints in the absence of an a priori model. We achieve this using background information. This is based on the observation that even an unseen view of the object can still be identified if one can recognize the background and surrounding objects. It also conforms to a similar behavior of the human vision system where surrounding information is very important in localizing an object. Background has been used in tracking mainly via background subtraction, the well-known approach which works only for sequences with a stationary background. In case of a moving background, most current methods use the appearance information of the object only. Recent work by Collin and Liu [4] emphasizes the importance of the background appearance. The paper proposes to switch the mean-shift tracking algorithm between different linear combinations of the three color channels so as to select the features that distinguish the object most from the background. The features are ranked based on a variance test for the separability of the histograms of object and background. Improved performance compared to the standard mean-shift has been reported. Even so, color histograms have a limited identification power and the method appears to work only in the condition that the object appearance does not change drastically over the sequence. For high dimensional features like textures, the large number of combinations will be a problem for achieving real time performance.

In the presented approach, robustness to viewpoint change is attained by the discrimination of object textures from background textures. The algorithm should be working under a moving background.

Section 2 presents our discriminative approach for the target detection. The section discusses the representation of object appearance and how object matching is performed. Section 3 describes the tracking algorithm, the online training of object / background texture discriminant functions, and the updating of object and background texture templates. Section 4 shows the tracking results.

2 Discriminative Target Detection Using Texture Features

In the presented algorithm, the target object is detected by matching texture features. The locality and high discriminative power of theses features makes it easier to classify individual image patches as object or background.

2.1 Object Appearance Representation

Let us first 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 θ. Object textures are then

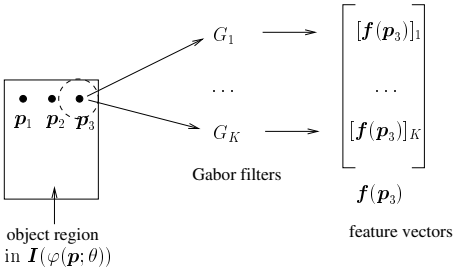

Fig. 1. *Illustration for the representation of object appearance.*

analyzed for the transformation compensated image $I(\varphi(\boldsymbol{p};\theta))$ using Gabor filters [10]. These filters have been used in various applications for visual recognition [7,9] and tracking [3]. Each pair of Gabor filters has the form:

$$G_{symm}(\boldsymbol{p}) = \cos\left(\frac{\boldsymbol{p}}{r}\cdot\boldsymbol{n}_\nu\right)\exp\left(-\frac{\|\boldsymbol{p}\|^2}{2\sigma^2}\right)$$
$$G_{asymm}(\boldsymbol{p}) = \sin\left(\frac{\boldsymbol{p}}{r}\cdot\boldsymbol{n}_\nu\right)\exp\left(-\frac{\|\boldsymbol{p}\|^2}{2\sigma^2}\right) \quad (1)$$

where σ, r and ν denote the scale, the central frequency and orientation respectively, and $\boldsymbol{n}_\nu = \{\cos(\nu), \sin(\nu)\}$. Setting these parameters to different values creates a bank of filters. Denote them $G_1, \ldots, G_K$. 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 $I(\varphi(\boldsymbol{q};\theta))$ to the Gabor filters:

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

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

$$\mathcal{F} = \{\boldsymbol{f}(\boldsymbol{p}_1), \ldots, \boldsymbol{f}(\boldsymbol{p}_n)\} \quad (3)$$

As $\boldsymbol{f}(\boldsymbol{p})$ governs the information of an entire neighborhood of $\boldsymbol{p}$, there is no need to compute the texture vector for all pixels in Ω. Instead, $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$ are sampled with a spacing.

2.2 Object Matching

The target detection amounts to finding the parameters θ that give the optimal $\mathcal{F}$. This is based on the two criteria:

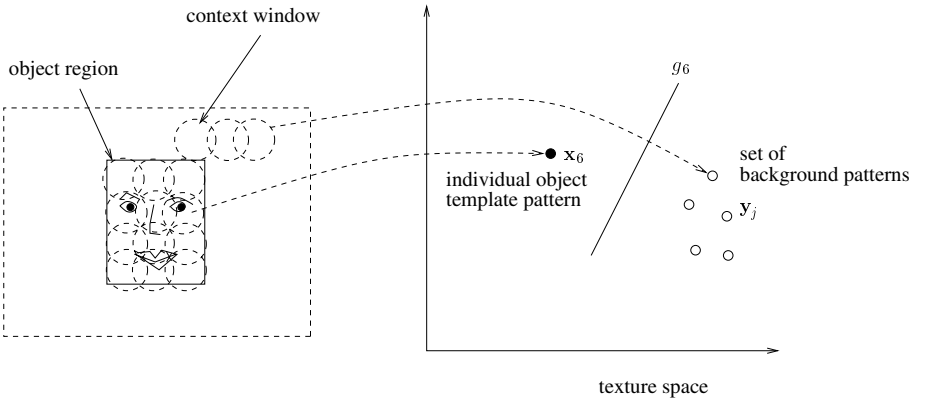

Fig. 2. *Illustration of the target detection using object/background texture discrimination.*

1. The similarity between $\mathcal{F}$ and a set of object template features:

$$\mathcal{O} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\},\ \mathbf{x}_i \in \mathbb{R}^K \tag{4}$$

   There is a correspondence between the vectors in $\mathcal{F}$ and $\mathcal{O}$, that is, $\boldsymbol{f}(\boldsymbol{p}_i)$ should match $\mathbf{x}_i$, since both of them represent the texture at pixel $\boldsymbol{p}_i$. This valuable information is ignored in the related approach [4], as it is based on histogram matching. The object templates are updated during tracking to reflect the most recent object appearance.
2. The contrast between $\mathcal{F}$ and a set of background template features:

$$\mathcal{B} = \{\mathbf{y}_1, \ldots, \mathbf{y}_M\},\ \mathbf{y}_j \in \mathbb{R}^K \tag{5}$$

   These are the texture vectors of background patterns observed so far in a context window surrounding the object, see Figure 2. The modelling of the background through a set of local patterns is mainly to deal with the difficulty in the construction of a background image. It is desired that every $\boldsymbol{f}(\boldsymbol{p}_i)$ is distinguished from all $\mathbf{y}_j$. As the background moves, $\mathcal{B}$ is constantly expanded to include new appearing patterns. On the other hand, a time-decaying weighting coefficient α_j is associated with every pattern $\mathbf{y}_j$. The coefficient enables the tracker to forget patterns that have left the context window.

We optimize $\mathcal{F}$ by maximizing the sum of a set of local similarity measures each computed for one vector in $\mathcal{F}$:

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

Here, $g_i(\boldsymbol{f}(\boldsymbol{p}_i; \theta))$ is the local similarity measure for object texture at pixel $\boldsymbol{p}_i$. We choose g_i to be a linear function:

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

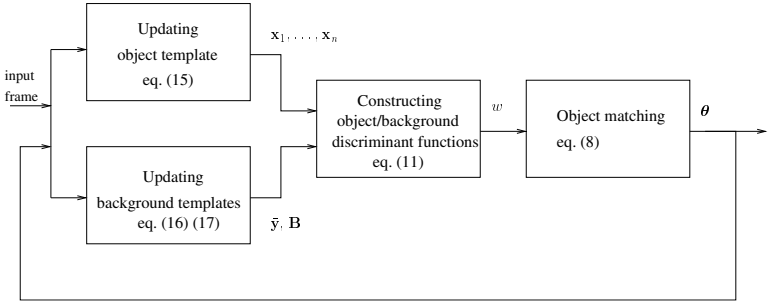

Fig. 3. *The illustration of the tracking algorithm.*

where $\boldsymbol{a}_i \in \mathbb{R}^K, b_i \in \mathbb{R}$ are the parameters. Furthermore, to satisfy the two mentioned criteria, g_i is chosen to be a discriminant function. Specifically, g_i trained to respond positively when $\boldsymbol{f} = \mathbf{x}_i$ and negatively when $\boldsymbol{f} \in \mathcal{B}$, see Figure 2. Note that in case where $\boldsymbol{f}(\boldsymbol{p}_i; \theta)$ represents an unseen object pattern, it may not match $\mathbf{x}_i$ but does not belong to $\mathcal{B}$ either. In this case, $g_i(\boldsymbol{f}(\boldsymbol{p}_i; \theta))$ likely has a value around zero, which is still higher than a mismatch. As such, by avoiding the background patterns the tracker is still able to find the correct object even in case of an aspect change.

In eq. (6), only the directions $\boldsymbol{a}_i$ matter. The value of b_i does not affect the maximization result. Using eq. (2), eq. (6) is rewritten as:

$$\max_{\theta} \sum_{\boldsymbol{q}} I(\varphi(\boldsymbol{q}; \theta)) w(\boldsymbol{q}) \tag{8}$$

where

$$w(\boldsymbol{q}) = \sum_{i=1}^{n} \sum_{k=1}^{K} a_{ik} G_k(\boldsymbol{p}_i - \boldsymbol{q}) \tag{9}$$

and a_{ik} denotes the k^{th} component of $\boldsymbol{a}_i$. As observed, (8) is the inner product of image $I(\varphi(\boldsymbol{q}; \theta))$ and function w. In particular, if only the translational motion is considered, $\varphi(\boldsymbol{q}; \theta) = \boldsymbol{q} + \theta$, and hence, object matching boils down to the maximization of the convolution of the current frame $I(\boldsymbol{q})$ with function w which is regarded as the target detection kernel.

3 Algorithm Description

Based on the matching method described above, we propose a tracking algorithm whose data flow diagram is given in Figure 3. This section addresses the issues that remain, including the construction of discriminant functions g_i and the updating of object and background templates.

3.1 Construction of Object / Background Discriminant Functions

In principle, any linear classifier from pattern recognition can be used for training g_i. However, in view of the continuous growth of the set of background patterns, the selected classifier should allow for the training in the incremental mode, and should be computationally tractable in real time tracking. To this end, we adapt the LDA (Linear Discriminant Analysis) [8]. Function g_i minimizes the cost function:

$$\min_{\boldsymbol{a}_i, b_i} (\boldsymbol{a}_i^T \mathbf{x}_i + b_i - 1)^2 + \sum_{j=1}^{M} \alpha_j (\boldsymbol{a}_i^T \mathbf{y}_j + b_i + 1)^2 + \frac{\lambda}{2} \|\boldsymbol{a}_i\|^2 \tag{10}$$

over $\boldsymbol{a}_i$ and b_i. The weighting coefficients α_j are normalized so that $\sum_{j=1}^{M} \alpha_j = 1$. 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. (10) is obtained in closed form:

$$\boldsymbol{a}_i = \kappa_i [\lambda \mathbf{I} + \mathbf{B}]^{-1} [\mathbf{x}_i - \bar{\mathbf{y}}] \tag{11}$$

where

$$\bar{\mathbf{y}} = \sum_{j=1}^{m} \alpha_j \mathbf{y}_j \tag{12}$$

$$\mathbf{B} = \sum_{j=1}^{m} \alpha_j [\mathbf{y}_j - \bar{\mathbf{y}}][\mathbf{y}_j - \bar{\mathbf{y}}]^T \tag{13}$$

$$\kappa_i = \frac{1}{1 + \frac{1}{2}[\mathbf{x}_i - \bar{\mathbf{y}}]^T [\lambda \mathbf{I} + \mathbf{B}]^{-1} [\mathbf{x}_i - \bar{\mathbf{y}}]} \tag{14}$$

As observed, the discriminant functions depend only on the object templates $\mathbf{x}_i$, the mean vector of background textures $\bar{\mathbf{y}}$ and the covariance matrix $\mathbf{B}$. These quantities can efficiently be updated during tracking.

Note that the background is usually non-uniform and therefore its textures are hardly represented just by one mean pattern $\bar{\mathbf{y}}$. Instead, the diversity of background patterns is encoded in the covariance matrix $\mathbf{B}$.

3.2 Updating Object and Background Templates

As we are dealing with sequences with severe viewpoint changes, the object templates need to be updated constantly to follow up varying appearances. On the other hand, a hasty updating is sensitive to sudden tracking failure and stimulates template drift. So, the updated template should be a compromise between the latest template and the new data. For this purpose, sophisticated temporal smoothing filters have been proposed [13]. In this work, however, for the implementation simplicity, we use the simple averaging filter:

$$\mathbf{x}_i^{(t)} = (1-\gamma)\mathbf{x}_i^{(t-1)} + \gamma \boldsymbol{f}(\boldsymbol{p}_i;\theta) \tag{15}$$

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

During object motion, constantly new patterns enter the context window and some other patterns leave the window. The background representation should be updated accordingly. It would be difficult to track reliably a background pattern from the moment of entering until its leaving. So, we keep all the observed patterns and gradually decrease the coefficients α_j that control the influence of the patterns in eq. (10). In this way, the tracker can forget the outdated patterns.

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 $\mathbf{y}_{M+1}, \cdots, \mathbf{y}_{M+m}$. The weighting coefficients are then distributed over the new and old elements in $\mathcal{B}$ 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 $\mathcal{B}$ is re-scaled with the factor $1-\gamma$. Let $\bar{\boldsymbol{y}}_{new} = \frac{1}{m}\sum_{j=M+1}^{M+m} \mathbf{y}_j$. The update equations for $\bar{\mathbf{y}}$ and $\boldsymbol{B}$ are:

$$\bar{\mathbf{y}}^{(t)} = (1-\gamma)\bar{\mathbf{y}}^{(t-1)} + \gamma \bar{\boldsymbol{y}}_{new} \tag{16}$$

$$\boldsymbol{B}^{(t)} = (1-\gamma)\boldsymbol{B}^{(t-1)} + (1-\gamma)\bar{\mathbf{y}}^{(t-1)}\bar{\mathbf{y}}^{(t-1)\,T} - \bar{\mathbf{y}}^{(t)}\bar{\mathbf{y}}^{(t)\,T} + \frac{\gamma}{m}\sum_{j=M+1}^{M+m} \mathbf{y}_j\mathbf{y}_j^T \tag{17}$$

4 Experiment

We have performed several experiments to verify the ability of the proposed tracking algorithm in handling severe viewpoint changes. In the current implementation, only translational motion is considered. For the extraction of texture features, the algorithm uses a set of twelve Gabor filters created for scale $\sigma = 4$ and $r = 2.0$ and six directions of ν equally spaced by 30^o. The target region is set to a rectangle. Object pixels $\boldsymbol{p}_1, \cdots, \boldsymbol{p}_n$ are sampled with a spacing of 4 pixels between each other in both horizontal and vertical axes. The same spacing is applied for the background pixels in the context window. For the updating of the object and background texture templates, we have set the weighting coefficient $\gamma = 0.2$.

For comparison, we also applied an intensity SSD tracker using an adaptive template. In every frame this algorithm recalculates the template as a weighted average between the latest template and new intensity data, where the weight of the new data is $\gamma = 0.2$. This averaging results in a smoothed template which is also resilient to viewpoint changes in some degree. Unlike the proposed approach, this algorithm does not use background information.

Figure 4 shows an example of head tracking. Initially the head is at the frontal view pose. The background is non-uniform, and the camera is panning back and forth, keeping the head in the center. The guy turns to the sides and even to the back, showing completely

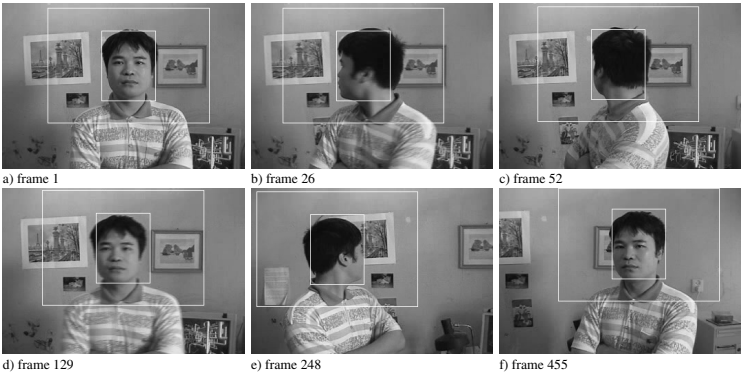

Fig. 4. *Head tracking results under severe viewpoint changes by the proposed algorithm. The outer rectangle indicates the context window.*

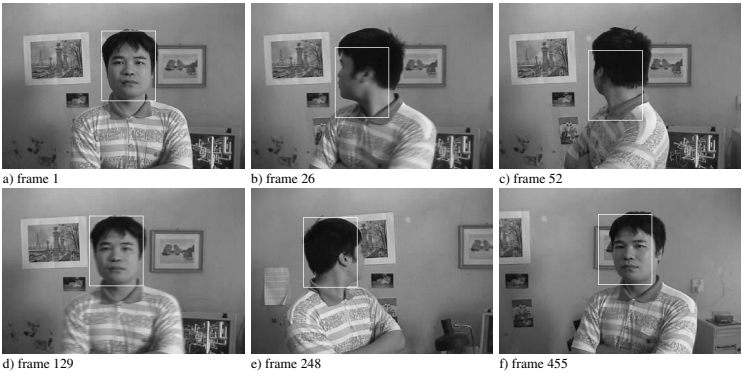

Fig. 5. *Tracking results for the same sequence in Figure 4 by the SSD tracker using an adaptive template.*

different views of the head. As observed, the proposed tracker could capture even the back view of the head which is unseen previously and is rather different from the initial frontal view. Figure 5 shows the tracking results for the same sequence but with the SSD tracker. This tracker also exhibited a robust performance under slight pose changes of the head, but it gave wrong results when the head pose changed severely as in Figure 5b and c. 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.

A clear example where the proposed algorithm outperforms the SSD tracker is shown in Figure 6 and Figure 7. The figures show the tracking results by the two trackers respectively for a sequence where a mousepad is rotated around its vertical axis, switching between the blue front side and the completely black back side. As we expected, the SSD tracker drifted off at the first transition of view, see Figure 7b. This is easily explained by the similarity between the color of the front side of the mousepad and the color of the

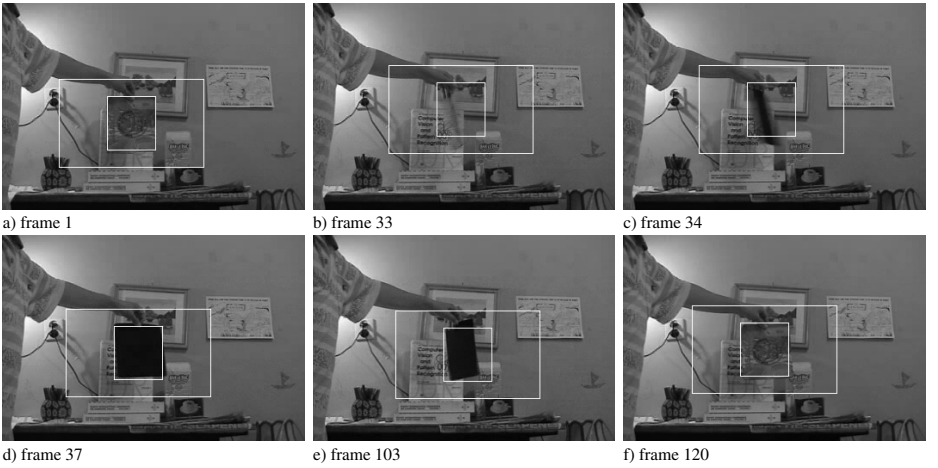

Fig. 6. *Tracking results by the proposed algorithm.*

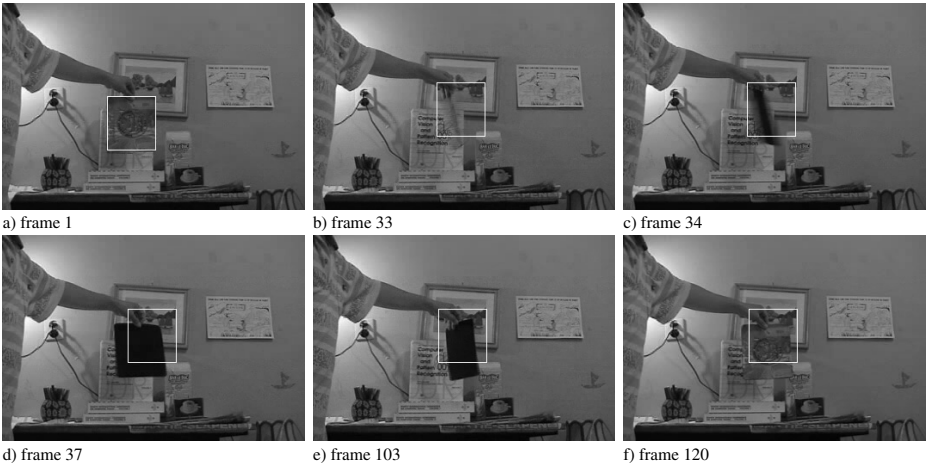

Fig. 7. *Tracking results for the sequence in Figure 6 by the SSD tracker using an adaptive template.*

wall. In contrast, the proposed algorithm recovered perfectly when the unseen dark side comes into view, see Figure 6d. It could also successfully lock back on the front side as in Figure 6f. The results prove that the proposed tracker rather chooses an unseen object region instead of a background region.

Figure 8 shows another head tracking result by the proposed algorithm for a movie clip. In this sequence, the camera pans fast to the left. The background is cluttered and contains several other moving objects. The results show the success of the proposed algorithm in tracking the head through several severe pose changes, as well as the robustness to the background motion and clutter.

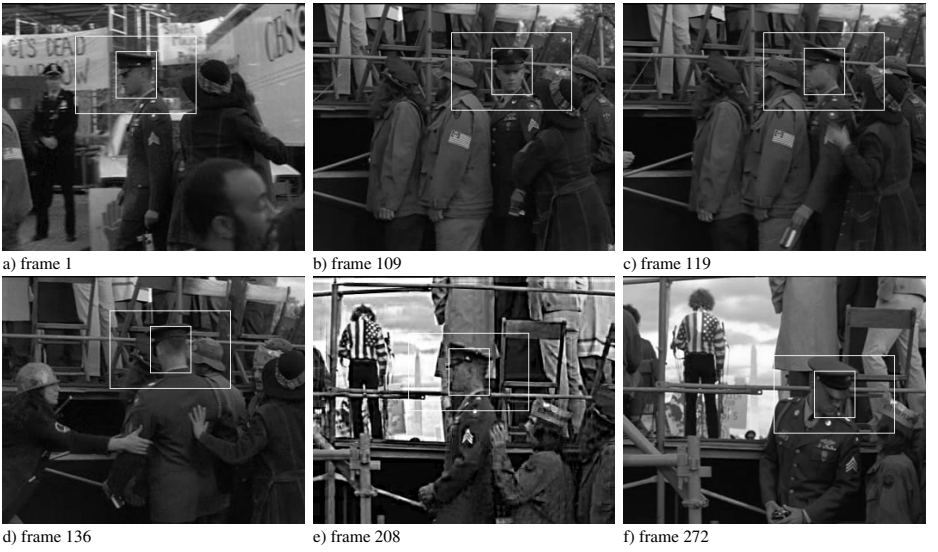

Fig. 8. *Tracking results by the proposed algorithm with a fast moving and cluttered background.*

5 Conclusion

The paper has shown the advantage of the background information for object tracking under severe viewpoint changes, especially when an unseen aspect of the object emerges. We have proposed a new tracking approach based on the discrimination of object textures from background textures. The high dimensionality of texture features allows for a good separation between the two scene layers. 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 a pattern in the object region from the background patterns. The target is detected by the maximization of the sum of the discriminant functions, taking into account the spatial distribution of object texture. 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.

For future work, we plan to improve several issues. We plan to test other more sophisticated classifiers to improve the accuracy of the target detection. The algorithm can also be extended to the multiscale mode with the propagation of the tracking result through the scales of the Gabor filters. Finally, more accurate models for the representation and updating of object and background template patterns will be considered.

References

1. G. Adiv. Inherent ambiguities in recovering 3-D motion and structures from noisy flow field. *IEEE Trans. on PAMI*, 11(5):477–489, 1989.

2. M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *Proc. of European Conf. on Computer Vision*, pages 329–342, 1996.
3. O. Chomat and J.L. Crowley. Probabilistic recognition of activity using local appearance. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pages II: 104–109, 1999.
4. R. Collins and Y. Liu. On-line selection of discriminative tracking features. In *Proc. IEEE Conf. on Computer Vision*, 2003.
5. D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR00*, pages II:142–149, 2000.
6. T.F. Cootes, G.V. Wheeler, K.N. Walker, and C.J. Taylor. View-based active appearance models. *Image and Vision Computing*, 20(9-10):657–664, 2002.
7. J.G. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans. on PAMI*, 15(11):1148–1161, 1993.
8. P. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, New York, 2001.
9. S. Gong, S. J. McKenna, and Collins J. J. An investigation into face pose distributions. In *Proc. of 2nd Inter. Conf. on Automated Face and Gesture Recognition*, Killington, Vermont, 1996.
10. A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
11. A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi. Robust online appearance models for visual tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recogn., CVPR01*, 2001.
12. L.P. Morency, A. Rahimi, and T. Darrell. Adaptive view-based appearance models. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pages I: 803–810, 2003.
13. H.T. Nguyen, M. Worring, and R. van den Boomgaard. Occlusion robust adaptive template tracking. In *Proc. IEEE Conf. on Computer Vision, ICCV'2001*, pages I: 678–683, 2001.
14. S. Ravela, B.A. Draper, J. Lim, and R. Weiss. Tracking object motion across aspect changes for augmented reality. In *ARPA Image Understanding Workshop*, pages 1345–1352, 1996.
15. L. Vacchetti, V. Lepetit, and P. Fua. Fusing online and offline information for stable 3D tracking in real-time. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pages II: 241–248, 2003.