Real-Time Tracking via On-line Boosting

Helmut Grabner, Michael Grabner, Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology
{hgrabner, mgrabner, bischof}@icg.tu-graz.ac.at

Abstract

Very recently tracking was approached using classification techniques such as support vector machines. The object to be tracked is discriminated by a classifier from the background. In a similar spirit we propose a novel on-line AdaBoost feature selection algorithm for tracking. The distinct advantage of our method is its capability of on-line training. This allows to adapt the classifier while tracking the object. Therefore appearance changes of the object (e.g. out of plane rotations, illumination changes) are handled quite naturally. Moreover, depending on the background the algorithm selects the most discriminating features for tracking resulting in stable tracking results. By using fast computable features (e.g. Haar-like wavelets, orientation histograms, local binary patterns) the algorithm runs in real-time. We demonstrate the performance of the algorithm on several (publically available) video sequences.

1 Introduction

The efficient and robust tracking of objects in complex environments is important for a variety of applications including video surveillance [25], autonomous driving [1] or human-computer interaction [5, 4]. Thus it is a great challenge to design robust visual tracking methods which can cope with the inevitable variations that can occur in natural scenes such as changes in the illumination, changes of the shape, changes in the viewpoint, reflectance of the object or partial occlusion of the target. Moreover tracking success or failure may also depend on how distinguishable an object is from its background. Stated differently, if the object is very distinctive, a simple tracker may already fulfill the requirements. However, having objects similar to the background requires more sophisticated features. As a result there is the need for trackers which can handle on the one hand all possible variations of appearance changes of the target object and on the other hand are able to reliably cope with background clutter.

Several approaches have been proposed to fulfill these two main requirements for tracking. To cope with appearance variations of the target object during tracking, existing tracking approaches (e.g. [3, 8, 10]) are enhanced by adaptivness to be able to incrementally adjust to the changes in the specific tracking environment (e.g. [14, 23, 17, 22, 13, 2, 15, 26]). In other words, invariance against the different variations is obtained by adaptive methods or representations. Many classical algorithms have been modified in order to be able to adjust the tracking algorithm to the tracking environment. The classical subspace tracking approach of Black et al. [3] was enhanced by incremental subspace updating in [14, 22]. In [14] it is proposed to express the general adaption problem as a subspace adaption problem, where the visual appearance variations at a short time scale are represented as a linear subspace. In contrast, [26] suggests an on-line selection of local Haar-like features in order to handle possible variations in appearance.

In addition, to the on-line adaption problem, recently many techniques have addressed the idea of using information about the background in order to increase the robustness of tracking [1, 19, 2, 26, 6]. Especially the work of Collins and Liu [6] emphasizes the importance of the background appearance. They postulate that the feature space that best distinguishes between object and background is the best feature space to use for tracking. The idea of considering the tracking problem as a classification problem between object and background has lead to further works [1, 2, 27] applying well known classifiers to the tracking problem. In [1] a tracker is realized by using a Support Vector Machine which learns off-line to distinguish between the object and the background. The work most closely related to ours is that of [2]. Again tracking is considered as a binary classification problem, where an ensemble of weak classifiers is combined into a strong classifier capturing the idea of AdaBoost for selecting discriminative features for tracking. To achieve robustness against appearance variations novel weak classifiers can be added to the ensemble. However, this is done in a batch processing mode using off-line boosting in a batch manner after new training examples have been collected. Moreover the features are quite restricted.

The novelty of this paper is to present a real-time object tracking method which is based on a novel on-line version of the AdaBoost algorithm. Our algorithm performs on-line updating of the ensemble of features for the target object during tracking and thus is able to cope with appearance changes of the object. Furthermore, the on-line trained classifier uses the surrounding background as negative examples in the update and becomes therefore very robust against the drifting problem [18]. In addition this negative update allows the algorithm to choose the most discriminative features between the object and the background. Therefore the method can deal with both appearance variations of the object and different backgrounds. The algorithm, which uses only grayvalue information (but can also be extended to color), is able to run in real-time since training is simply done by updating the model with the positive and negative examples of the current frame.

The reminder of the paper is organized as follows. In Section 2 we introduce the tracking algorithm and the novel on-line version of AdaBoost for feature selection which forms the bases of the approach. In Section 3 we show experimental results illustrating the adaptivity, the robustness and the generality of the proposed tracker.

2 Tracking

The main idea is to formulate the tracking problem as a binary classification task and to achieve robustness by continuously updating the current classifier of the target object. The principle of the tracking approach is depicted in Figure 1.

Since we are interested in tracking, we assume that the target object has already been detected. This image region is assumed to be a positive image sample for the tracker. At the same time negative examples are extracted by taking regions of the same size as the target window from the surrounding background. These samples are used to make several iterations of the on-line boosting algorithm in order to obtain a first model which is already stable. Note that these iterations are only necessary for initialization of the tracker. The

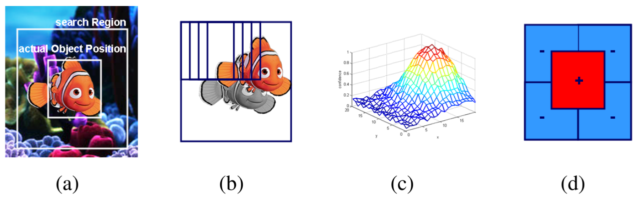


Figure 1: The four main steps of tracking by a classisifer. Given an initial position of the object (a) in time t, the classifier is evaluated at many possible positions in a surrounding search region in frame $t+1$. The achieved confidence map (c) is analyzed in order to estimate the most probable position and finally the tracker (classifier) is updated (d).

tracking step is based on the classical approach of template tracking [12]. We evaluate the current classifier at a region of interest and obtain for each position a confidence value. We analyze the confidence map and shift the target window to the new location of the maxima. For maximum detection also a mean shift procedure [7] can be used. Using a motion model for the target object would allow a reduction of the search window. Once the objects has been detected the classifier has to be updated in order to adjust to possible changes in appearance of the target object and to become discriminative to a different background. The current target region is used as a positive update of the classifier while again the surrounding regions represent the negative samples. As new frames arrive, the whole procedure is repeated and the classifier is therefore able to adapt to possible appearance changes and in addition becomes robust against background clutter. Note that the classifier focuses on the current target object while at the same time tries to distinguish the target from its surrounding. Apart from this, tracking of multiple objects is feasible by initializing a separate classifier for each target object.

2.1 On-line AdaBoost

In this section we briefly review the on-line boosting algorithm (for more details see [11]) which allows to generate classifiers that can be efficiently updated by incrementally applying samples. For better understanding of this approach we define the following terms:

Weak classifier: A weak classifier has only to perform slightly better than random guessing (i.e., for a binary decision problem, the error rate must be less than 50%). The hypothesis h^{weak} generated by a weak classifier corresponds to a feature and is obtained by applying a defined learning algorithm.

Selector: Given a set of M weak classifiers with hypothesis $\mathscr{H}^{weak} = \{h_1^{weak}, \ldots, h_M^{weak}\}$, a selector selects exactly one of those.

$$h^{sel}(\mathbf{x}) = h_m^{weak}(\mathbf{x}) \tag{1}$$

where m is chosen according to a optimization criterion. In fact we use the estimated error e_i of each weak classifier $h_i^{weak} \in \mathscr{H}^{weak}$ such that $m = \arg\min_i e_i$.

Strong classifier: Given a set of N weak classifiers, a strong classifier is computed by a linear combination of selectors. Moreover, the value $conf(\cdot)$ (which is related to

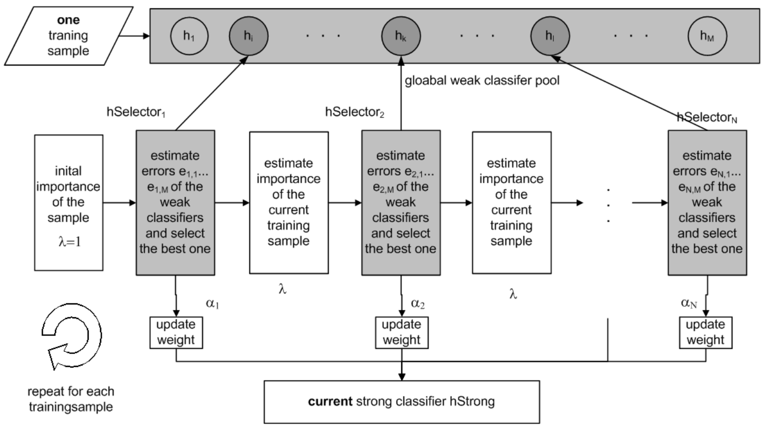

one traning sample
h1
hi
hk
hj
hM
gloabal weak classifer pool
hSelector1
hSelector2
hSelectorN
inital importance of the sample λ=1
estimate errors e1,1... e1,M of the weak classifiers and select the best one
estimate importance of the current training sample
estimate errors e2,1... e2,M of the weak classifiers and select the best one
estimate importance of the current training sample
estimate errors eN,1... eN,M of the weak classifiers and select the best one
λ
λ
α1
α2
αN
update weight
update weight
update weight
repeat for each trainingsample
current strong classifier hStrong

Figure 2: Principle of on-line boosting for feature selection.

the margin) can be interpreted as a confidence measure of the strong classifier.

$$hStrong(\mathbf{x}) = \operatorname{sign}(conf(\mathbf{x})) \tag{2}$$

$$conf(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n \cdot h_n^{sel}(\mathbf{x}) \tag{3}$$

The main idea of on-line boosting is the introduction of the so called *selectors*. They are randomly initialized and each of them holds a seperate feature pool of weak classifiers. When a new training sample arrives the weak classifiers of each selector are updated. The best weak classifier (having the lowest error) is selected by the selector where the error of the weak classifier is estimated from samples seen so far. The complexity is determined by the number of selectors.

The part which requires most of the processing time is the updating of weak classifiers. In order to speed up this process, we propose as a modification (similar to [28]) to use a single "global weak classifier" pool (see Figure 2) which is shared by all selectors instead of single pools for each of them. The advantage of this modification is that now for each sample that arrives, all weak classifiers need to be updated only once. Then the selectors sequentially switch to the best weak classifier with respect to the current estimated λ and the importance weight is passed on to the next selector. This procedure is repeated until all selectors are updated. Finally, at each time step an updated strong classifier is available. In order to increase the diversity of the weak classifiers and to allow changes in the environment, the worst weak classifier of the shared feature pool is replaced with a new randomly chosen one.

2.2 Features

We use three different types of features for generating weak hypotheses. Haar-like features like Viola and Jones [24], orientation histograms [16, 21, 9] and a simple[1] version of local binary patterns (LBPs) [20]. Note, that the computation of all feature types can be

[1]Binary patterns using a four-neighborhood (i.e. $2^4 = 16$ patterns) as a 16 bin histogram feature.

done very efficiently using integral images and integral histograms as data structures [21]. This allows to do exhaustive template matching while tracking is still real-time.

To obtain a hypothesis from these features we model the probability distribution for positive samples and negative samples. Probability density estimation is done by a Kalman filtering technique. For the classic Haar-like wavelets we use a simple threshold and a Bayesian decision criterion as learning algorithm. For the histogram based feature types (orientation histograms and LBPs), we use nearest neighbor learning. Of course, other types and other learning algorithms can be used to obtain a weak hypotheses. (For more details see [11])

3 Experiments and Discussion

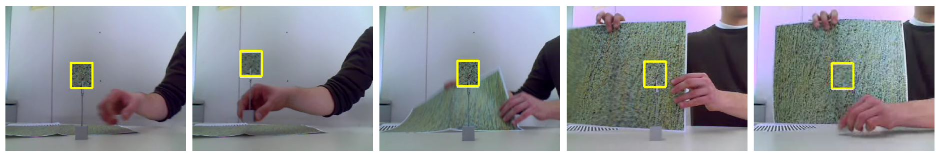
The experimental section is divided into two parts. First, we perform experiments demonstrating three specific properties of our tracking approach and second we present results on public available sequences for comparison to other tracking approaches. Each tracking task has been initialized by manually marking the target object in the first frame. Tracking has been applied to sequences consisting of hundreds of frames. The performance (speed) depends on the size of the search region which we have defined by enlarging the target region by one third in each direction (for this region the integral representations are computed). In our experiments no motion model has been used. We achieve a frame rate of about 20 fps. The strong classifier consists of 50 selectors and the shared feature pool provides 250 weak classifiers. All experiments have been done on a standard 1.6 GHz PC with 256 MB RAM.

3.1 Illustrations

The goal of this section is to illustrate three properties of the proposed tracker - adaptivity, robustness and generality. Therefore multiple challenging sequences have been captured with a static camera having a resolution of 640×480.

Adaptivity

To illustrate the adaptivity of the tracker and its capability to select the best features depending on the background, we process a scene where the target object is a small textured patch, see Figure 3. The goal of the scenario is to show how the proposed on-line feature selection method can adapt its model to the current tracking problem. Since our approach looks for suitable features which can best discriminate the object from the background, we change the background from a homogeneous to a textured one (same texture as the patch). For evaluation we consider the distribution of the selected feature types (for this experiment we simply used Haar-like features and LBPs). As we can see in the second row of Figure 3, the initial choice mainly uses Haar-like features (green) while LBPs (red) are rather rarely used for handling this tracking task. However, after putting texture to the background we can see from the plot in the second row, that the features for tracking the target immediately change. Note that the exchange of features occurs within a quite short period. As a result LBP features have become much more important for the tracking task because the Haar-like features are no longer distriminative. Furthermore the target object is still successful tracked even though the texture of the target is the same as in the background.



(a) Frame 7 (b) Frame 55 (c) Frame 121 (d) Frame 152 (e) Frame 245

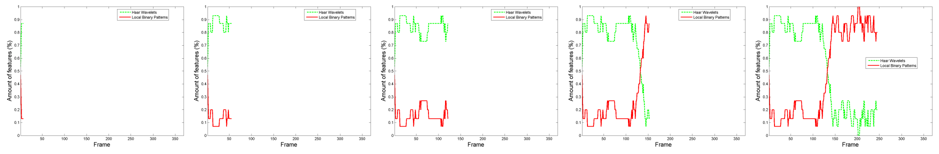


Figure 3: First row shows the tracked object which is marked with a yellow rectangle. On-line feature selection for tracking is analyzed by considering the ratio of two selected feature types - Haar-like features (dashed green) and local binary patterns (solid red).

This experiment shows the importance of on-line learning because we cannot train for all different backgrounds prior tracking starts. In addition, it shows that there is the need of different types of features.

Robustness

For successful real-world tracking a tracker must be able to handle various appearance changes (i.e.: illumination changes, occlusions, out-of-plane rotations, movement) which can occur in natural scenes. Figure 4 illustrates the behavior of our proposed method in case of such interferences of the target object. The sequence shows a glass which initially gets occluded (more than 50%) by a paper, afterwards it is moved behind it with additional illumination changes which are caused by the occlusion and finally view-point changes of the target object. The continuous increase of the confidence maximum value in the initial phase (see row 3, image 1) implies the adapting of the tracker to the target object with respect to its background. However if the target object changes its appearance or the surrounding background of the target becomes different, the tracker needs to update his features which is reflected in oscillations of the confidence maximum (see row 3) and a flattened confidence map (see row 2). Movement of the tracking target is represented by a shifted peak in the confidence map. To summarize, the tracker is able to handle all kinds of appearance variations of the target object and always aims at finding the best discriminative features for discriminating the target object from the surrounding background. Therefore, the adaptivity is strongly related to the robustness of a tracker.

Generality

In order to demonstrate the generality we use four different sequences with diverse target objects, see Figure 5. The first sequence, which is depicted in row 1, illustrates the tracking of a tiny Webcam in a cluttered scene. Even though the target object contains few texture and changes in pose occur it is robustly tracked. The second sequence (row 2)

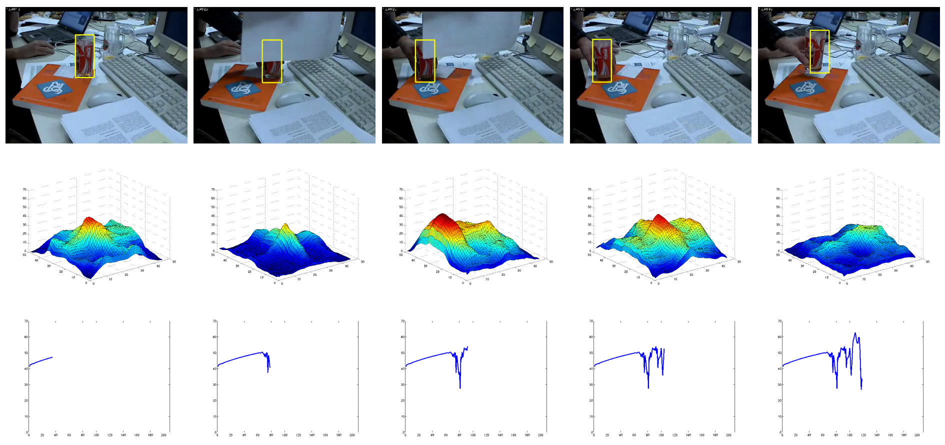


Figure 4: Tracking results on a sequence (row 1) containing a combination of appearance changes (i.e. illumination, occlusion, movement, out of plane rotation). The behaviour of the proposed tracker is analyzed considering the confidence map (row 2) and the maximum confidence value over time (row 3).

shows the results of tracking a face. As we can see from the depicted frames, the proposed approach can cope very well with occlusions which again is the effect of the large number of local features for tracking. Moreover even large pose variations of the head do not confuse the tracker showing that the tracker adapts to novel appearances of the target object. Row 3 illustrates that only little texture of the object is sufficient for tracking. A glass, having almost no texture, is tracked and again shows the reliability and adaptivity of the proposed tracker. Row 4 demonstrates the behavior in case of multiple very similar target objects. As can be seen, even though the objects significantly overlap the trackers get not confused demonstrating that the classifiers have really learned to distinguish the specific object from its background. Of course, an overlap over a long duration can cause adaption to the foreground object and finally leading to failure to track the occluded object because of the bad updates. However, this can be prevented by using some higher level control logic.

To summarize, the proposed tracker has the ability to adapt to the appearance of all kinds of objects by learning a good selection of features. Therefore the tracker's property to adapt is useful for both the handling of appearance variations and for selecting features in order to adapt to any target object.

3.2 Evaluation on public available sequences

Unfortunately up to now there is no public available framework for comparing tracking techniques. Therefore we decided to process public available sequences, see Figure 6 and 7, which have already been used in other publications for illustrating tracking results ([17, 15]). These sequences contain changes in brightness, view-point and further appearance variations. The results show that the proposed tracker can cope with all these variations and results are at least as good as those presented in the according publications.

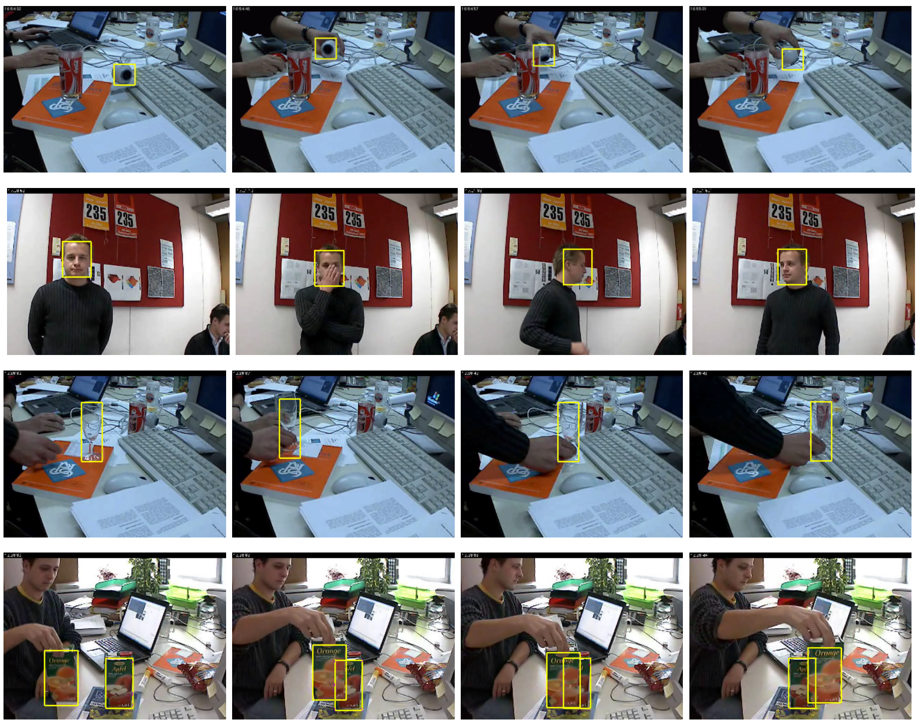
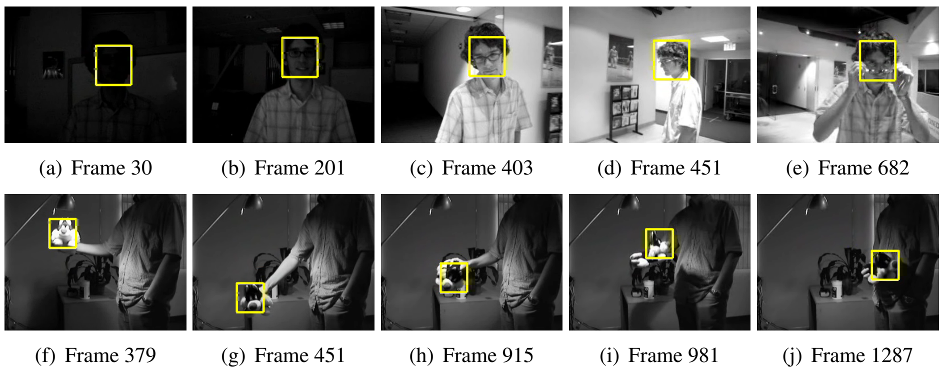


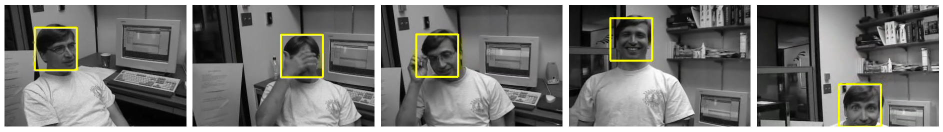
Figure 5: To illustrate the generality of the proposed method, sequences of four different objects have been captured. The tracking results show that even objects with almost no texture (see row 3) can be successfully tracked. Moreover the tracking algorithm can cope with multiple initialized objects even if they have similar appearance (see row 4).



(a) Frame 30 (b) Frame 201 (c) Frame 403 (d) Frame 451 (e) Frame 682

(f) Frame 379 (g) Frame 451 (h) Frame 915 (i) Frame 981 (j) Frame 1287

Figure 6: These sequences have been provided by Lim and Ross ([17]). The first sequence shows a person moving from dark towards bright area while making changes in pose and and partial occlusion of the target region can be seen. In the second row, an animal doll is moving with large pose, light variations in a cluttered background.



(a) Frame 1 (b) Frame 127 (c) Frame 269 (d) Frame 509 (e) Frame 885

Figure 7: Experimental results on the sequence provided by Jepson [15]. Again the object of interest is a face which moves in front of cluttered background and contains variations in appearance.

4 Conclusion

In this paper we have proposed a robust and generic real-time tracking technique (about 20 fps using a standard 1.6 GHz PC with 512 MB RAM) which considers the tracking problem as a binary classification problem between object and background. Most existing approaches construct a representation of the target object before the tracking task starts and therefore utilize a fixed representation to handle appearance changes during tracking. However, our proposed method does both - adjusting to the variations in appearance during tracking and selecting suitable features which can learn any object and can discriminate it from the surrounding background. The basis is an on-line AdaBoost algorithm which allows to update features of the classifier during tracking. Furthermore the efficient computation of the features allows to use this tracker within real-time applications. Finally, since the tracker is based on a classifier approach now there are several new venues of research like how we can construct a more generic model (like a detector) of the target object during tracking.

Acknowledgments

The project results have been developed in the MISTRAL Project which is financed by the Austrian Research Promotion Agency (www.ffg.at). In addition this work has been sponsored in part by the Austrian Federal Ministry of Transport, Innovation and Technology under P-Nr. I2-2-26p VITUS2, by the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04, and by the EU FP6-507752 NoE MUSCLE IST project.

References

[1] S. Avidan. Support vector tracking. *PAMI*, 26:1064–1072, 2004.

[2] S. Avidan. Ensemble tracking. In *Proc. CVPR*, volume 2, pages 494–501, 2005.

[3] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *IJCV*, 26(1):63–84, 1998.

[4] A. Bobick, S. Intille, J. Davis, F. Baird, C. Pinhanez, L. Campbell, Y. Ivanov, A. Schutte, and A.Wilson. The KidsRoom. *Communications of the ACM*, 39(3&4):438–455, 2000.

[5] G.R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 1998.

[6] R.T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *PAMI*, 27(10):1631–1643, Oct. 2005.

[7] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *Proc. ICCV*, volume 2, pages 1197–1203, 1999.

[8] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. CVPR*, volume 2, pages 142–149, 2000.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, volume 1, pages 886–893, 2005.

[10] A. Elgammal, R. Duraiswami, and L. S. Davis. Probabilistic tracking in joint feature-spatial spaces. In *Proc. CVPR*, volume 1, pages 781–788, 2003.

[11] H. Grabner and H. Bischof. On-line boosting and vision. In *Proc. CVPR*, volume 1, pages 260–267, 2006.

[12] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, 1998.

[13] B. Han and L. Davis. On-line density-based appearance modeling for object tracking. In *Proc. ICCV*, volume 2, pages 1492–1499, 2005.

[14] J. Ho, K. Lee, M. Yang, and D. Kriegman. Visual tracking using learned linear subspaces. In *Proc. CVPR*, volume 1, pages 782–789, 2004.

[15] A. D. Jepson, D. J. Fleet, and T.F. El-Maraghi. Robust online appearance models for visual tracking. In *Proc. CVPR*, volume 1, pages 415–422, 2001.

[16] K. Levi and Y. Weiss. Learning object detection from a small number of examples: The importance of good features. In *Proc. CVPR*, pages 53–60, 2004.

[17] J. Lim, D. Ross, R. Lin, and M. Yang. Incremental learning for visual tracking. In *Advances in Neural Information Processing Systems 17*, pages 793–800. MIT Press, 2005.

[18] Iain Matthews, Takahiro Ishikawa, and Simon Baker. The template update problem. *PAMI*, 26:810–815, 2004.

[19] H.T. Nguyen and A. Smeulders. Tracking aspects of the foreground against the background. In *Proc. ECCV*, volume 2, pages 446–456, 2004.

[20] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987, 2002.

[21] F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *Proc. CVPR*, volume 1, pages 829–836, 2005.

[22] D. Ross, J. Lim, and M. Yang. Adaptive proballistic visual tracking with incremental subspace update. In *Proc. ECCV*, volume 2, pages 470–482, 2004.

[23] J. Vermaak, P. Pérez, M. Gangnet, and A. Blake. Towards improved observation models for visual tracking: Selective adaption. In *Proc. ECCV*, pages 645–660, 2002.

[24] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, volume 1, pages 511–518, 2001.

[25] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. ICCV*, volume 2, pages 734–741, 2003.

[26] J. Wang, X. Chen, and W. Gao. Online selecting discriminative tracking features using particle filter. In *Proc. CVPR*, volume 2, pages 1037–1042, 2005.

[27] O. Williams, A. Blake, and R. Cipolla. Sparse bayesian learning for efficient visual tracking. *PAMI*, 27:1292–1304, 2005.

[28] J. Wu, J.M. Rehg, and M.D. Mullin. Learning a rare event detection cascade by direct feature selection. In *Advances in Neural Information Processing Systems 16*. MIT Press, 2003.