Beyond Semi-Supervised Tracking: Tracking Should Be as Simple as Detection, but not Simpler than Recognition

Severin Stalder[1] Helmut Grabner[1] Luc van Gool[1,2]

[1]Computer Vision Laboratory
ETH Zurich
{sstalder, grabner, vangool}@vision.ee.ethz.ch

[2]ESAT - PSI / IBBT
K.U. Leuven
luc.vangool@esat.kuleuven.be

Abstract

We present a multiple classifier system for model-free tracking. The tasks of detection (finding the object of interest), recognition (distinguishing similar objects in a scene), and tracking (retrieving the object to be tracked) are split into separate classifiers in the spirit of simplifying each classification task. The supervised and semi-supervised classifiers are carefully trained on-line in order to increase adaptivity while limiting accumulation of errors, i.e. *drifting. In the experiments, we demonstrate real-time tracking on several challenging sequences, including multi-object tracking of faces, humans, and other objects. We outperform other on-line tracking methods especially in case of occlusions and presence of similar objects.*

1. Introduction

Robust visual tracking under real-world conditions is still an unsolved problem and limits the use of state-of-the-art methods in commercial systems (*e.g.*, video surveillance [5]). Recently, tracking formulated as a binary classification problem received a lot of attention due to its promising results. The basic idea of this approach is to learn a classifier which distinguishes the tracked object from the local background. Classifiers are trained either (i) off-line, mainly for speeding up the matching process (*e.g.*, [14]) or (ii) on-line, in order to cope with variations of the object that are not known a priori. This paper focuses on the second option.

Many methods using different object representations and learning methods have been proposed for adaptive classifiers (*e.g.*, [2, 4]). Grabner *et al*. [8] have designed an on-line boosting framework that adaptively selects features to discriminate the object from the background. The classifier is updated using a self-learning policy, *i.e.*, the tracker relies on its own predictions. Therefore the tracker is able to adapt to any appearance changes, but unfortunately also suffers from the drifting problem, *i.e.*, amplifying small errors and adapting to other objects. The underlying assumption is that the updates are correct and furthermore that those belonging to the object are also correctly aligned with the object (no label jitter). The fundamental problem is to robustly integrate data derived during tracking into the model without drifting. In general, model-free tracking has basically to cope with a trade off between adaptivity and stability [12]. Matthews *et al*. [16] have coined this problem the "template update problem". Additional knowledge might be used, *e.g.*, geometric verification [11], combination of generative and discriminative models [21], co-learning using different types of features [23], or constrained updates [13]. Another principled approach has been proposed by formulating tracking as a semi-supervised learning problem [10].

Figure 1. Continuum of approaches from a fixed detector (no updates) to a fully adaptive tracker. Our proposed tracker is balancing between semi-supervised and the fully adaptive tracking.

In semi-supervised machine learning, unlabeled data can be included in addition to labeled data. In fact, an initial model is assumed to be given (*e.g.* built during initialization), while all further tracking examples are then only included as unlabeled data. Hence no label noise or label jitter is integrated as the initial model is being refined. This approach can also be interpreted as combined detection and tracking [15] where the detector serves as prior model for the semi-supervised learning. The initial information does not get lost and one can recover from drifting while still being adaptive to appearance changes of the object. Besides this advantage, there are mainly two drawbacks. Firstly, the influence of the prior might not be optimal, especially in the case of partial occlusions. Secondly, the prior does not specialize to a specific object. Thus, tracking multiple, sim-

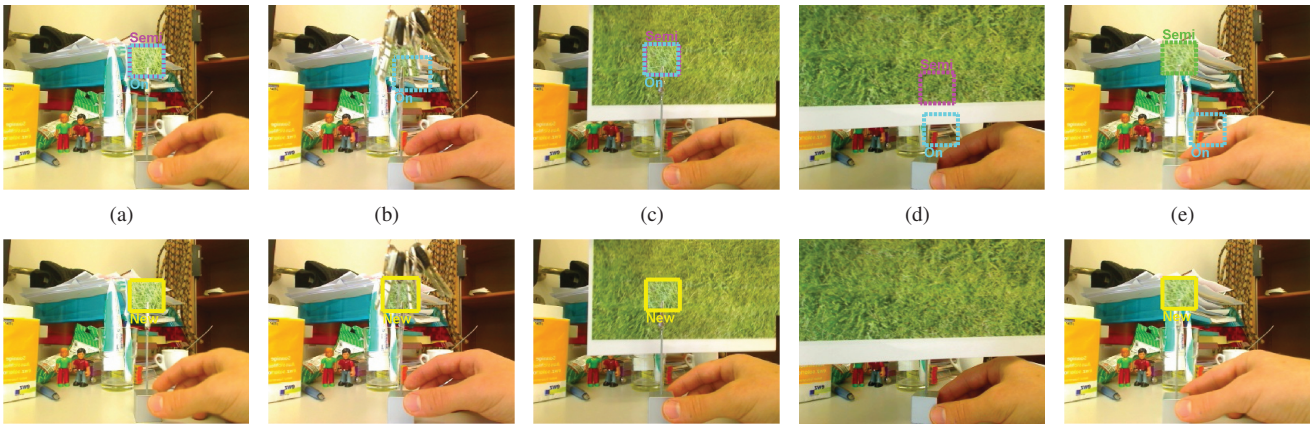


Figure 2. Tracking of a texture patch against a similar and cluttered background. State-of-the-art adaptive trackers (first row; dotted cyan: supervised [8]; dotted pink: semi-supervised tracker [10]) might suffer from problems concerning (i) drifting, (ii) distinguishing similar objects, and (iii) robustness to partial and full occlusions. Our proposed approach (second row) is both more stable and more robust.

ilar objects (e.g. several faces) is not directly feasible and the tracker can wrongly switch to those other objects. We overcome this limitation by extending the semi-supervised learning approach with adaptive priors as depicted in Fig.1.

As shown in Fig. 2 our proposed approach (bottom row) is able to be more adaptive but at the same time also more stable than the adaptive trackers (top row). The on-line, adaptive tracker [8] (dotted cyan) is still tracking the initial object under partial occlusion, however it may adapt wrongly to the occluder and start tracking it after a while (b). In contrast, the semi-supervised tracker [10] (dotted pink) loses the object since the changed appearance can not be explained by the prior until the object gets fully visible again. Due to their discriminative power, all trackers are able to track the object in front of a very similar background (c). However, if a sheet with similar texture appears in front of the initial object, the adaptive trackers will not notice the difference and start tracking a piece of the similar object (d). The semi-supervised tracker has the ability to return afterward to the object whereas the on-line tracker does not recover from drift (e). We propose an approach which has no difficulties with this scenario.

The reminder of the paper is organized as follows. Sec. 2.1 briefly reviews and discusses limitations of former on-line boosting based tracking. Our approach, which extends semi-supervised tracking, is described in Sec. 3. Sec. 4 presents detailed experiments and compares them to the existing approaches. Finally, conclusions are presented in Sec. 5.

2. Preliminaries and Discussions

Before introducing our novel tracking approach, we review and discuss the two basic types of classifiers which are used for tracking.

2.1. On-line Boosting for Feature Selection

The goal boosting [7] is to minimize the error by selecting and combining a set of N "weak" classification algorithms $\{h_n(\mathbf{x})|h_n(\mathbf{x}) : \mathcal{X} \rightarrow \{+1, -1\}\}$ into a strong classifier

$$H(\mathbf{x}) = \text{sign}(f(\mathbf{x})) \quad \text{where} \quad f(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n h_n(\mathbf{x}) \ . \quad (1)$$

The absolute value of $f(\mathbf{x})$ (which is related to the margin) can be interpreted as a confidence measure. For on-line training the strong classifier is initialized at the beginning and is updated by each training sample. The individual weak classifiers are updated according to an importance weight λ (*i.e.*, samples which are misclassified are given more importance), which is propagated through all of them.

Supervised: Boosting can be used for feature selection [22], where the features correspond to weak classifiers. Here we describe the on-line [8] variant, where the main idea is to perform on-line boosting on *selectors* rather than on the weak classifiers directly. A selector holds a set of M weak classifiers and selects the one with the lowest estimated error. A strong classifier is (randomly) initialized with a fixed number of N selectors $h_1^{sel}, .., h_N^{sel}$. Firstly, the weak classifiers in each selector are updated, as soon as a new training sample $\langle \mathbf{x}, y \rangle$, $\mathbf{x} \in \mathcal{X}$, $y \in \{+1, -1\}$ is available. Secondly, the importance weight λ_0 of the sample is initialized to 1, which is used to update the weak classifier (any on-line learning algorithm is applicable).

Semi-Supervised: Recently, the supervised approach was extended in order to include unlabeled data [10], based on the idea of considering pairs of samples which are connected via a similarity measure. Similar samples (*i.e.*, a labeled and an unlabeled sample) should share the same label. In fact, a prior classifier $H^P(\mathbf{x})$ is used for that pur-

pose. Similar to the on-line supervised version, the importance weights encode information from one weak classifier for the next. For labeled examples the supervised boosting approach is used directly. For unlabeled examples, not only the importance is adapted, but also the label is re-estimated. More formally, in each selector n, a pseudo-label y_n and pseudo-importance λ_n are set according to

$$y_n = \text{sign}(\tilde{z}_n(\mathbf{x})) \text{ and } \lambda_n = |\tilde{z}_n(\mathbf{x})|, \text{where} \quad (2)$$

$$\tilde{z}_n(\mathbf{x}) = \tanh(H^P(\mathbf{x})) - \tanh(H_{n-1}(\mathbf{x})). \quad (3)$$

is calculated depending on the prior classifier $H^P(\mathbf{x})$ and the current on-line classifier $H(\mathbf{x})$.

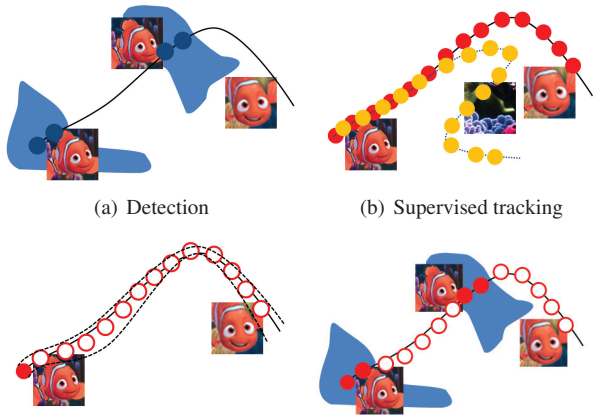
Influence of the prior: Using supervised boosting, *i.e.*, the true label is given, the importance of the example is adapted according to the misclassification error. In semi-supervised boosting, unlabeled samples can be included whereas a prior classifier determines their pseudo-label and their importance weight. If the prior is very confident, it dictates the label. A label switch can happen, *i.e.*, $H(\mathbf{x})$ can overrule $H^P(\mathbf{x})$, if $\tilde{z}_n(\mathbf{x})$ has a different label than the prior $H^P(\mathbf{x})$. As can be easily seen from Eq. (3), this is the case if $|H_n(\mathbf{x})| > |H^P(\mathbf{x})|$. Therefore, the more confident the prior is, the longer (with respect to n) the label is not allowed to change. Another interpretation can be made by rewriting the right hand side of Eq. (3) as

$$\cosh(H_n(\mathbf{x}))\sinh(H^P(\mathbf{x})) - \cosh(H^P(\mathbf{x}))\sinh(H_n(\mathbf{x})). \quad (4)$$

Since $\cosh(\cdot) \geq 1$ weighs the decision of the corresponding classifier (sign of the asymmetric $\sinh(\cdot)$ function), unlabeled data is used for regularization. Similar to the co-training assumption [3] the prior classifier should be never "confident but wrong". In order to put more emphasis on the prior or decrease its influence an additional factor might be used to scale it. In extreme cases the prior may vanish (drifting might happen) or may dominate too much (no adaptation is possible at all). This mainly addresses the assumptions of semi-supervised learning [24, 19].

2.2. On-line Boosting for Tracking

On-line boosting used for tracking generally works as follows. The tracking loop is initialized with a detection. Then, an initial classifier H is built by taking positive samples from the object and negative ones from the surrounding background. The classifier is evaluated exhaustively on the image (or a provided local search region) at time $t+1$. The resulting confidence distribution is analyzed and in the simplest case the local maximum is considered to be the new object position. In order to adapt to appearance changes of the object (*e.g.* different illumination) or changed background, the classifier gets updated and the loop repeats. In contrast to the supervised on-line boosting tracker [8] which uses fixed self-labeled updates, the semi-supervised on-line boosting tracker [10] takes a detection to initialize the tracking classifier which is then only refined with unlabeled data (*i.e.*, the detector result serves as prior). Drifting is limited since the tracker cannot get too far away from the prior. The tracking principles are summarized in Fig. 3.



(a) Detection (b) Supervised tracking

(c) Semi-supervised tracking (d) Our Approach

Figure 3. Drifting vs. adaptation: A fixed object detector does not suffer from the drifting problem. However, to be adaptive, *e.g.*, to appearance changes, on-line updates are performed in a supervised manner (filled circles, b). By formulating tracking as a semi-supervised learning problem, only unlabeled data is used during tracking (empty circles, c). This limits drifting but may restrict the adaptivity too much. Our proposed approach (d) makes use of additional information and extends the semi-supervised learning by an *adaptive* and object specific prior.

Beside the advantages of semi-supervised tracking exposed before, typically two problems arise in practice:

Limited appearance changes and partial occlusions: Possible appearance changes or dominant partial occlusions are also interpreted as some sort of drift. Thus, they are limited by the prior as well, *i.e.*, the prior might be too strong, too weak, or even certainly wrong. Without any external knowledge, this is summarized by the stability-plasticity dilemma [12]. Thus, really informative examples are those which are not in consonance with the prior model[1].

No discrimination between different objects from one class: The prior might be too generic in the sense that it covers variations between different objects of the same class. For instance, having a face detector as prior, it is not possible to distinguish different persons. Thus, recognition is not taken into account and the tracker is likely to jump to similar objects.

The tracking approach so far can also be seen as tracking by detection. However, in practice, the tasks of detection, tracking and recognition are highly coupled.

[1]There are also relations to active learning and bootstrapping used for training of high performance detectors [22].

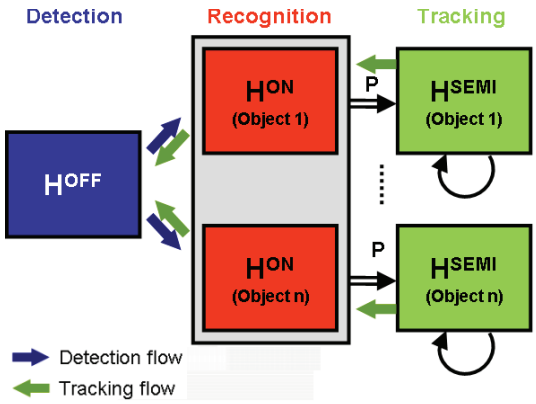


Figure 4. The core classifier system consists of an off-line, an on-line supervised and an on-line semi-supervised classifier which are interpreted as detector, recognizer and tracker. The classifiers interact via information flows thus avoiding direct feedback loops which may cause drifting.

3. Beyond Semi-Supervised Tracking

Based on the previous discussions, we propose a multiple classifier framework in which detection, recognition, and tracking are highly coupled, see Fig. 4. We assign to each individual classifier a specific sub-task interpreted as 'detection' (finding the object of interest), 'recognition' (distinguishing similar objects in a scene) and 'tracking' (retrieving the object to be tracked). We make use of temporal and spatial assumptions which can be made in visual object tracking[2] to train the classifiers on different training sets in the spirit of simplifying each classification task. However, the desired benefit of being more adaptive comes with a restricted temporal and spatial applicability.

Summarizing, the challenge is to include training data during tracking in a robust manner, *e.g.*, without drifting, in order to increase adaptivity.

3.1. Multiple Classifier System

Two types of on-line classifiers are used in the proposed system. There are supervised classifiers for including trustworthy information and semi-supervised classifiers for including less reliable information derived during tracking. In the following we describe each classifier in detail. The information exchange between the classifiers is emphasized in Sec. 3.2.

Detector (off-line classifier): The goal of the detector is to reliably find the object of interest. The detector classifier is not updated during tracking to guarantee a fixed false positive and detection rate. Any kind of object detector can be integrated in the system, see Sec. 4. The detector is generic and should be applicable on any scene.

Recognizer (supervised on-line classifier): The recognizer is object specific and serves as an adaptive prior for tracking. Updates are only performed conservatively. The positive training set consists of tracked samples which are validated by the detector. The negative training set is composed of hard examples collected in the background image at the time of a detection. This allows to distinguish similar objects in a scene. Additional negative updates are performed at the tracked position in the background image. Thereby it is assured that static occluders will be present in both training classes and implicitly ignored by the classifier. The recognizer is only valid in the specific scene during one track.

Tracker (semi-supervised on-line classifier): The tracker is essential to retrieve the object in the next frame. The confidence map is analyzed via semi-supervised updates to retrieve a stable maximum, see Sec. 3.4. The tracked object samples are then given to the detector to eventually update the recognizer. The tracker is only valid in the search region of the tracked object during one track.

The novel recognizer is balancing between semi-supervised tracking [10] (without adaptation, recognizer = detector) and on-line tracking [8] (full adaptation, recognizer = tracker). The recognizer is an on-line supervised classifier as it can be reliably updated by the detector. On the other hand it is serving as a prior for the tracker which is a semi-supervised classifier to prevent drifting. The recognizer can also be interpreted as an object instance detector trained in case of a generic detection at the tracked position. Therefore, with more training data, the recognizer is able to aggregate information during tracking to progressively specialize and improve the model. Moreover, multiple object tracking becomes feasible as the recognizer distinguishes similar objects in the scene.

3.2. Information Exchange

In the proposed system feedback *loops* are strictly avoided to prevent accumulation of small errors, *i.e.* drifting. However, there are two simultaneous information *flows* which are validating the current track in order to be adaptive. The information flows separate the tasks of determining the position of the object (tracking flow) and updating the classifiers at the given position (detection flow).

Detection flow: A new recognizer and a new tracker are initialized if a detection has no overlap with an existing tracked object. The recognizer is updated only if a detection has high overlap with an existing tracked object.

Tracking flow: Once a track is started, the tracker determines the position of the object with the recognizer as prior.

[2]Contrary to a pure machine learning approach in which detection, recognition, and tracking might be coupled differently, *e.g.*, by transfer learning (see [17] for a survey).

Hence, the tracker is the dominant element in our approach since it is sampling the positive training data by determining the position of a possible re-detection. In other words, the tracker is exploring the data to be included in its prior. Our experiments show that this approach is suitable for robust tracking in case of multiple similar objects which are partially or fully occluded. However, we regret that no theoretical underpinning could have been made up to now.

3.3. Extensions

Re-identification to bridge short temporal gaps is done in separated supervised classifiers. Additionally, all tracking results of the specific scene are aggregated to train supplementary local detectors in case of static cameras.

Identifier (supervised on-line classifier): The identifier is used for re-identification of a tracked object whose track has been lost. Although we propose to train a separate classifier for identification, in principle, any identification system can be used. Positive updates are taken from the current tracker and negative ones from all the other identities. Once a tracker has lost its object, the identity is only stored for a limited time in order to bridge short time gaps. Hence it is valid only at the location of tracked objects for a limited time. This allows to handle the data association problem.

Local Grid Detectors (supervised on-line classifiers): The local grid detectors are used to aggregate tracking results. A local detector is created on a grid at each tracked position (similar to [18]). The classifiers are trained each time any tracked object passes. The positive update is made with the object, the negative update is performed with the background image at the same position. So, each local detector has the very simple task to distinguish foreground from background at one location only. The local detectors are generic and only valid at a specific location in the scene. The output of the classifier is not used to trigger tracking in order to prevent feedback loops. Indeed, the local detections can help to substitute tracking when a track is lost. In contrast to the classifier grid of [18], we are able to gather positive samples of the particular scene. This idea is further exploited in [20].

3.4. Analysis of the Confidence Map

The search for the most likely position of the object in the next frame is modified in order to stabilize the tracked position. In former approaches (further described in Sec. 2.2), the classifiers are evaluated on the image and shifted directly to the maximum of the confidence map[3]. We propose a different method using semi-supervised classifiers. In fact, we first perform *unlabeled* updates at the position of the supposed maximum in the image and at the same position in the background image. Only if that position remains the maximum, we continue tracking. Experimentally, this approach shows superior results in case of partial or full occlusions.

[3] Or using more sophisticated methods, *e.g.*, by first smoothing the confidence map or by applying non-maxima suppression.

4. Experiments and Discussion

The proposed tracking approach is able to track a variety of objects in challenging situations. We performed experiments in order to demonstrate its abilities in comparison to [8] and [10]. During the experiments, we explored three different kinds of fixed detectors: (i) the face detector taken from *OpenCV 1.0*[4] [22] (ii) a state-of-the-art person detector [6], and (iii) self-trained detectors similar to one-shot learning in [10].

Each classifier in our system is a boosted strong classifier which consists of 50 selectors having access to a dynamic pool of 100 weak classifiers, the actual image features. We use Haar-like features, histograms of oriented gradients, and color histograms. The latter are only used for classifiers which are object specific and only valid for a certain period of time (*e.g.* the recognizer and the tracker). As background image we always take the previous frame excluding the tracked region from being changed.

The performance depends on the size of the search regions, maximum number of objects to be tracked, and minimum displacement of the tracking region. In our experiments we do not use a motion model to estimate a scaled search window, however, it could be incorporated quite easily. All experiments are performed on a common 3.0 GHz PC Dual Core with 2 GB RAM, where we achieve typically 10 fps with our non optimized C++ implementation[5].

Quantitative comparison (Tab. 1): We evaluated three trackers based on on-line boosting using recall, precision and f-measure similar to object detection [1]. We manually marked the object in each frame to obtain the ground truth. If the overlap between the tracker and the ground truth bounding box (with a fixed scale) is greater than 75%, we count the frame as true-positive. False-negatives are counted if nothing or something else is tracked but the object is still visible. False-positives typically indicate drifting since the tracker loses the object. Summarizing, the recall (tracking success) of our approach is very similar to the semi-supervised tracker, whereas it shows a superior precision.

Implicit recognition through a background model (Fig. 5, 1st row): This experiment shows that the recognizer is able to distinguish very similar objects. In fact, the on-line and the semi-boosting trackers prefer to jump

[4] `http://sourceforge.net/projects/opencvlibrary/`, 03/16/2008

[5] Precompiled Win32 binaries as well as the complete source code is available at `http://www.vision.ee.ethz.ch/boostingTrackers/`

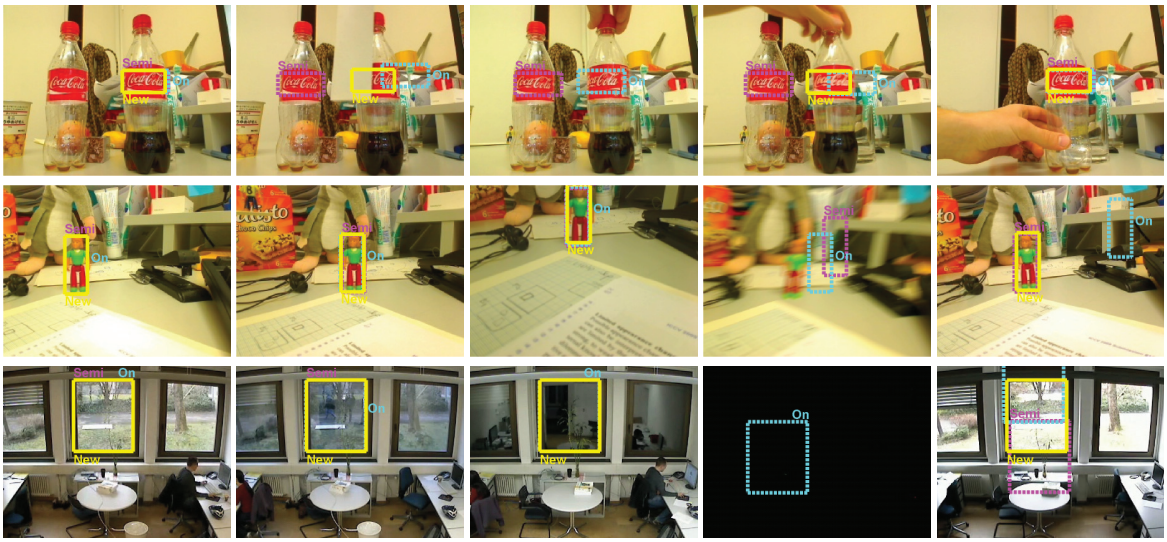


Figure 5. Comparisons of our proposed tracker (yellow) and the related on-line tracker (supervised: dotted cyan; semi-supervised: dotted pink). Our approach is more adaptive while still being robust to occlusions, identity changes and drift.

	On-line [9]	Semi [10]	our approach
recall	0.15	0.76	0.76
precision	0.89	0.32	0.99
f-measure	0.26	0.45	0.86

Table 1. Recall, precision and f-measure of boosting based trackers for the sequence shown in Fig. 1.

to a similar object instead of tracking the object changing its appearance. The proposed tracker, however, is trained negatively on the background image and will not confuse the initial object with the similar ones. In case of a large appearance change (3rd image) it looses the track and re-detects it afterward (4th image).

Tracking with a moving camera (Fig. 5, 2nd row): Even though we are using a background model we are not restricted to a static camera. As the background image is built from the previous frame, we quickly adapt to changed conditions like different illumination or a moving camera. Our approach performs similarly to the compared trackers. The proposed approach lost the track in case of too much motion blur instead of tracking something else (4th image) and successfully re-detects it shortly after (5th image).

Long-term tracking (Fig. 5, 3rd row): This experiment shows tracking performance in the long-term. A static object with significant appearance changes has been tracked for 24h. The on-line tracker is slowly drifting, whereas the semi-supervised tracker can not handle large appearance which are limited by the fixed prior. The proposed tracker is more adaptive to appearance changes than the semi-supervised approach (3rd and 5th image) without drifting.

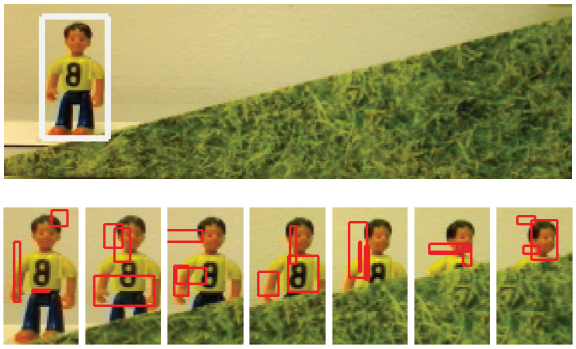


Figure 6. Three most important features selected by the local detector at the corresponding tracked position trained while tracking the object moving from the left to the right behind a static occluder.

Implicit static occlusion handling (Fig. 6): This experiment shows that static occluders are implicitly ignored by the classifiers. A one-shot detector is trained in the first frame to track the toy. Local detectors are trained during tracking. In Fig. 6 the location of the three most important Haar-like features of the local detector at the tracked position are shown. The static occluder (green cotter) is present in the image and the background image which are to be distinguished by the local detector. Thus, no discriminant features are selected on the occluder (green cotter).

Local detectors (Fig. 7): In this experiment we show the qualitative characteristics of the local detectors in a challenging surveillance sequence[6]. As input we applied the person detector to the single images. We manually estimated the ground plane to reject detections at wrong scales

[6] *i-Lids* medium sequence, ftp://motinas.elec.qmul.ac.uk/pub/iLids/, 03/06/2009.

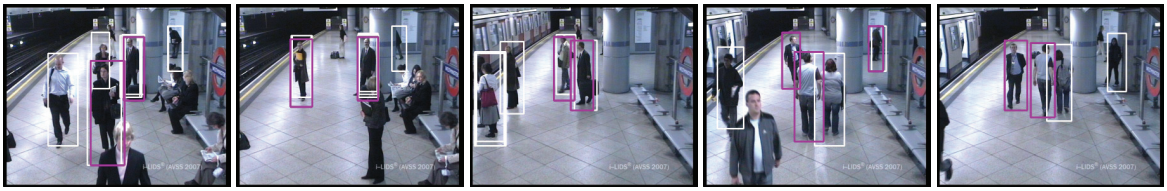


Figure 7. Local detectors (white) trained through information aggregation in specific image regions, compared to the input detector (pink).

and to align the local detectors. The typical application of local detectors is 24/7 surveillance, where it is helpful to aggregate scene knowledge over time to gain in performance. However, only qualitative results are presented as the local detectors are not the primary scope of this paper.

Multiple object tracking with re-identification (Fig. 8): In this experiment, we take a face detector and track two persons[7]. Temporal gaps are bridged by re-identification. The emphasis is put on the matching of the identities when they reappear in a similar pose. The longer the track, the higher the likelihood of a successfull re-identification as more appearance changes can be integrated. The identification matching is done very conservatively, *i.e.*, only if an identifier has significantly higher response than all others. Sometimes a re-identification may fail because the appearance is too different (*e.g.*, id2 and id3). However, both actors can be successfully matched to their initial tracks without confusing the identities.

Typical updates of each classifier (Fig. 9): Here, the previous experiment is described in detail to give some insight into the typical updates of each classifier. The face detector typically finds frontal faces which are used as positive updates of the recognizer. The tracked objects contain clearly more appearance changes than the detected objects. Indeed, the tracker which is sampling the patches is more adaptive than the recognizer. Note, that the confidence maps are only meaningful in certain locations, *e.g.*, the identifier confidences are only used on the current tracked faces.

5. Conclusion

We presented a multiple object tracking approach extending semi-supervised tracking by object specific and adaptive priors. Valuable information which would be ignored in a pure semi-supervised approach is safely included in the prior using a detector for validation and a tracker for sampling. The prior is interpreted as recognizer of the object as similar objects are distinguished. If a track is lost, we can re-identify the object by separately trained re-identification classifiers. The tracked objects are used to train local detectors to simplify detection in the specific scene.

[7]Dev Patel and Freida Pinto in a talk about their movie *Slumdog Millionaire*, `http://www.youtube.com/watch?v=cwzwhHB_L4Q`, 03/06/2009.

The novel classifier framework is able to track various objects, even under appearance changes and partial occlusions, in challenging environments. Drifting is limited due to careful use of supervised updates and preventing feedback loops. Our experiments show superior performance compared to previously proposed adaptive trackers.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for objet detection. In *Proc. ECCV*, 2002.

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

[3] M.-F. Balcan, A.Blum, and K. Yang. Co-training and expansion: Towards bridging theory and practice. In *NIPS*. 2004.

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

[5] H. Dee and S. Velastin. How close are we to solving the problem of automated visual surveillance? *Machine Vision and Applications*, 19(5-6):329–343, 2008.

[6] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. CVPR*, 2008.

[7] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

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

[9] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *Proc. BMVC*, volume 1, pages 47–56, 2006.

[10] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *Proc. ECCV*, 2008.

[11] M. Grabner, H. Grabner, and H. Bischof. Learning features for tracking. In *Proc. CVPR*, 2007.

[12] S. Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Neural networks and natural intelligence*, pages 213–250, 1998.

[13] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *Proc. CVPR*, 2008.

[14] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Proc. CVPR*, volume 2, pages 775–781, 2005.

[15] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Tracking in low frame rate video: A cascade particle filter with dis-

Figure 8. Tracking of two faces and their successful re-identification after a track got lost. Typical patches and corresponding identifiers are illustrated in the lower part.

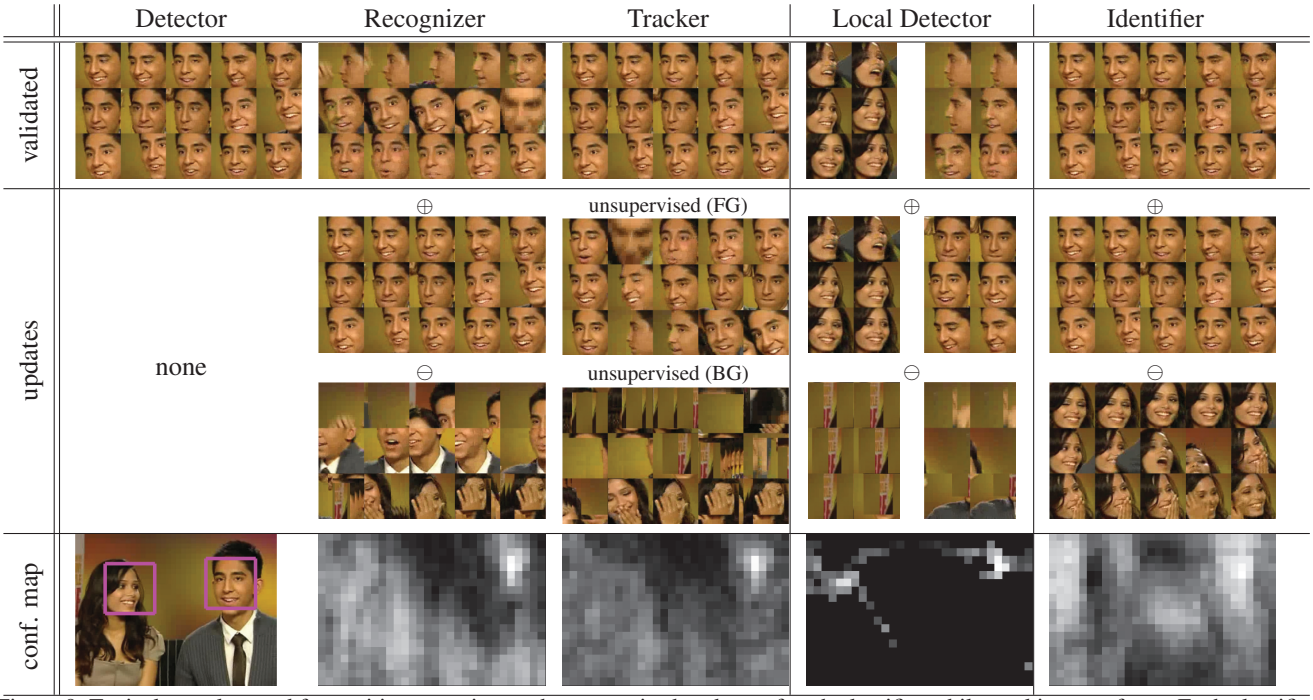


Figure 9. Typical samples used for positive, negative, and unsupervised updates of each classifier while tracking two faces. Each classifier is specializing to a particular task and is only valid for a certain time and region in the image as illustrated by the confidence maps (the brighter the patch, the stronger the classifier response. Only two typical local detectors are shown).

criminative observers of different lifespans. In *Proc. CVPR*, pages 1–8, 2007.

[16] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *PAMI*, 26:810 – 815, 2004.

[17] S. Pan and Q. Yang. A survey on transfer learning. Technical Report HKUST-CS08-08, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China, November 2008.

[18] P. Roth, H. Grabner, S. Sternig, and H. Bischof. Classifier grids for robust adaptive object detection. In *Proc. CVPR*, 2009.

[19] A. Singh, R. Nowak, and X. Zhu. Unlabeled data: Now it helps, now it doesn't. In *NIPS*. 2008.

[20] S. Stalder, H. Grabner, and L. V. Gool. Exploring context to learn scene specifc object detectors. In *Proc. PETS*, 2009.

[21] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-tracking using semi-supervised support vector machines. In *Proc. ICCV*, pages 1–8, 2007.

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

[23] Q. Yu, T. Dinh, and G. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *Proc. ECCV*, 2008.

[24] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Univ. of Wisconsin-Madison, 2005.