On-line Semi-supervised Multiple-Instance Boosting*

Bernhard Zeisl[1,2], Christian Leistner[1], Amir Saffari[1], Horst Bischof[1]

[1] Institute for Computer Graphics and Vision, TU Graz

{leistner,saffari,bischof@}icg.tugraz.at

[2] Computer Vision and Geometry Group, ETH Zurich

zeisl@inf.ethz.ch

Abstract

A recent dominating trend in tracking called tracking-by-detection uses on-line classifiers in order to redetect objects over succeeding frames. Although these methods usually deliver excellent results and run in real-time they also tend to drift in case of wrong updates during the self-learning process. Recent approaches tackled this problem by formulating tracking-by-detection as either one-shot semi-supervised learning or multiple instance learning. Semi-supervised learning allows for incorporating priors and is more robust in case of occlusions while multiple-instance learning resolves the uncertainties where to take positive updates during tracking. In this work, we propose an on-line semi-supervised learning algorithm which is able to combine both of these approaches into a coherent framework. This leads to more robust results than applying both approaches separately. Additionally, we introduce a combined loss that simultaneously uses labeled and unlabeled samples, which makes our tracker more adaptive compared to previous on-line semi-supervised methods. Experimentally, we demonstrate that by using our semi-supervised multiple-instance approach and utilizing robust learning methods, we are able to outperform state-of-the-art methods on various benchmark tracking videos.

*This work has been supported by the Austrian FFG project EVis (813399) and Outlier (820923) under the FIT-IT program and the Austrian Science Fund (FWF) under the doctoral program Confluence of Vision and Graphics W1209.

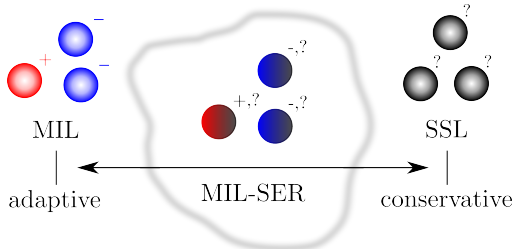

Figure 1. Our proposed algorithm combines the benefits of multiple instance learning (MIL) and semi-supervised learning (SSL). Thus in tracking it utilizes robustness and adaptivity from MILBoost and the regularization behavior and dependence on a prior for SemiBoost.

1. Introduction

Visual object tracking is one of the biggest challenges in computer vision. Despite the huge amount of research spent on this task it is still hard to design robust tracking systems that can perform similar to humans. Visual trackers have to cope with all variations that occur in natural scenes such as shape and appearance changes, different illuminations as well as varying poses or partial occlusions. Numerous tracking methods have been proposed, such as global template-based trackers, *e.g.*, [1], shape-based methods [3], probabilistic models using mean-shift [5], particle filtering [12], local key-point based trackers [16], or flow-based trackers [18]. See also [21] for a more detailed review.

One dominating trend in object tracking is to apply appearance-based classifiers. Such tracking-by-detection systems [1] usually train a classifier at the first frame versus its local background and perform re-detection in succeeding frames. In order to handle rapid appearance changes, recent works, *e.g.*, [9]

use on-line classifiers that perform self-updating on the target object. Such on-line classifiers are usually highly accurate and fast since they only have to discriminate the object from its current local background. However, these classifiers perform self-learning and it is difficult to decide autonomously where exactly to take the positive and negative updates, respectively. Even if the object is tracked correctly, the alignment may not be perfect which can lead to slightly wrong updates of the tracker (*a.k.a* label jitter). If these errors accumulate over time, they can finally lead to drifting of the tracker [14].

Recent approaches try to tackle the drifting problem by formulating the tracking-by-detection as one-shot semi-supervised learning. Grabner *et al*. [10] proposed an on-line semi-supervised boosting algorithm (Online SemiBoost) and update the learner with a supervised loss only at the beginning, *i.e*., the first frame, and then regularize this learned classifier in subsequent frames on the unlabeled data by using an unsupervised loss function. Although this method has shown to be less error prone to drifting and simultaneously more adaptive than a static classifier, it still cannot reach the performance of a self-learning classifier in case of a rapid appearance changes [2]. Also highlighting this problem of Online SemiBoost, recently Babenko *et al*. [2] formulated the tracking task as a multiple-instance learning (MIL) problem. Using MIL, the classifier in principle is still performing self-learning; however, the allowed positive update area around the current tracker can be increased and the classifier resolves the ambiguities by itself, yielding more robust tracking results.

In this work, we propose to combine the ideas from semi-supervised learning and multiple-instance learning (see Fig. 1). Our approach inherently combines the benefits of these two learning trends and hence is more robust than applying them separately. In order to achieve this goal, we first extend the semi-supervised boosting approach of SERBoost [17] to on-line learning case. Based on online SERBoot, we build our on-line semi-supervised multiple-instance algorithm which is able to use both labeled and unlabeled bags. We call our algorithm MIL-SERBoost.

Furthermore, we revisit the on-line semi-supervised learning formulation for tracking. In contrast to the previous formulation [10], we highlight that one cannot directly impose the same assumptions from off-line semi-supervised learning tasks to sequential data analyses, in our case tracking, because the underlying data distribution is continuously changing and usually non *i.i.d.* [7]. This explains why, in a tracking scenario, a prior classifier may become invalid even after a short time (depending on the sequence). Additionally, in [10] during tracking the data is only used to *regularize* the classifier learned at the first frame rather than learning new samples with a powerful loss. Thus, we emphasize that exclusive regularization prevents the system from being adaptive enough in order to handle rapid appearance changes.

In the following Sec. 2 we review the on-line semi-supervised and multiple instance learning and tracking. Based ont this we outline regularization algorithms for instance and multiple instance based learning in Sec. 3. Sec. 4 illustrates the application of our algorithm to the problem of visual tracking. In Sec. 5 we cover experiments which compare our approach to other state of the art tracking algorithms.

2. On-line Boosting

Boosting additively combines several weak classifiers to a strong one in the form

$$F(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i f_i(\mathbf{x}), \tag{1}$$

where α_i determines the influence of the i^{th} weak learner. During learning, boosting keeps a weight distribution over the training samples. Since, in the on-line case each training sample is provided only once to the learner, Oza and Russell [15] proposed to model the sequential incoming of the samples using a Poisson distribution and compute the importance w of a sample by propagating it through the set of weak classifiers. Later, Grabner *et al*. [8] introduced selectors in order to allow for on-line feature selection. On-line boosting is performed on these selectors and not directly on the hypothesis space.

2.1. Semi-supervised On-line Boosting

Semi-supervised learning [4] deals with using unlabeled samples in order to improve the performance of a classifier. Many semi-supervised learning algorithms use the unlabeled samples to regularize a loss function which is designed for the labeled samples. Let $\mathcal{X}$ to be a set of samples $\mathbf{x} \in \mathbb{R}^D$. Additionally, let $\mathcal{X}_l$ be the labeled samples $(\mathbf{x}, y)$ and $\mathcal{X}_u$ be the set of unlabeled samples. The following loss function

form is usually used for semi-supervised algorithms

$$\mathcal{L}(\mathcal{X}) = \mathcal{L}_l(\mathcal{X}_l) + \beta\mathcal{L}_u(\mathcal{X}_u) \quad (2)$$
$$= \sum_{(\mathbf{x},y)\in\mathcal{X}_l} l_l\left(yF(\mathbf{x})\right) + \beta \sum_{\mathbf{x}\in\mathcal{X}_u} l_u(F(\mathbf{x})),$$

where $l(yF(\mathbf{x}))$ and $l_u(F(\mathbf{x}))$ are the loss functions for labeled and unlabeled samples, respectively. β steers the importance of the unlabeled data.

We address a binary decision problem with classes $y \in \{-1, +1\}$. Thus, $yF(x)$ describes the margin of the classifier for sample $\mathbf{x}$. $F(x)$ itself is a large margin classifier yielding confidence-rated predictions. Minimizing this loss off-line, the learner has access to labeled and unlabeled samples simultaneously. For on-line minimization, especially for the task of tracking, [10] proposed the Online SemiBoost (OSB) to pass labeled samples to the supervised loss only at the first frame. In all subsequent frames, samples are only passed through the unsupervised loss. This principle results in a very conservative tracker thus successfully reducing drifting; however, adaptation to rapid appearance changes of the object is hard to achieve.

On the other hand, fully supervised methods are adaptive and thus are superior in learning appearance changes but tend to drift. However, it has been shown that drifting can be reduces, if robust loss functions [11] are applied, or multiple instance learning [2] is used, because both approaches are capable of handling label noise in their updates. Our algorithm will combine the benefits of both approaches.

2.2. Multiple Instance Learning

Traditional supervised learning deals with data samples which are presented to the algorithm in a pair $(\mathbf{x}, y)$. In multiple-instance learning, the data is presented in form of labeled *bags* $(\mathcal{B}, y)$ where $\mathcal{B} = \{\mathbf{x}_1, \cdots, \mathbf{x}_{N_B}\}$ is a collection of data instances. The relation between the label of a bag and the labels of its instances is ambiguous for the positive bags: if a bag label is positive $y = 1$, it is only guaranteed that there exists at least one instance in the bag belonging to the positive class. For the negative bags $y = -1$, there is no ambiguity and we know that all the instances belong to the negative class.

Multiple instance approaches try to resolve the inherent ambiguities given in many practical learning problems. For example, when training an object detector, access might only be available to weakly supervised labels, *i.e.*, only the presence of an object in an image is known and not its location. In this setting, an image can be represented as a bag containing all or a subset of its sub-windows. If it is known that there is a target object in this image, at least one of these sub-windows should represent the object; however, it is not known which one exactly. Similarly, if an image does not contain the target class, all of its sub-windows can be considered belonging to the negative class as well. Therefore, multiple instance settings exactly define the problem of training a visual object detector from weakly supervised labels. When dealing with object tracking by using a detector, multiple instance learning can be used in a similar way in order to solve the problem of where to take positive updates [20, 2].

3. Regularization for Instance and Multiple Instance Learning

Based on the previous discussion, we propose to combine a robust loss function or multiple instance learning with a regularization part over unlabeled data. The general formulation of our algorithm is equivalent to Eq. (2); however, we introduce an online learning formulation. Moreover, for the application of tracking we significantly differ from previous approaches in the way we apply samples to the algorithm. This will be explained in detail in Sec.4. In the following, we keep the algorithmic derivations generic, so that it can be applied to other application domains than visual tracking.

To achieve on-line learning we adapt the functional gradient descent view of boosting [6]. According to the gradient descent principle in optimization, at each step of boosting the algorithm is looking for the m^{th} weak learner $f_m(\mathbf{x})$ (chosen from a set $\mathcal{F}$ of weak learners), which if added to the current strong learner $F(\mathbf{x})$ results in an overall classification improvement. This optimization problem is written as

$$f_m(\mathcal{X}) = \arg\max_{f\in\mathcal{F}} \langle -\nabla\mathcal{L}, f(\cdot)\rangle . \quad (3)$$

In each iteration boosting is specializing on samples which have been misclassified so far by introducing a weight w. In the gradient descent framework, the magnitude of the gradient for a given sample directly relates to its weighting for the next weak learner. Eq (3) applied with the combination of loss

functions, thus yields

$$\begin{aligned}\arg\min_{f\in\mathcal{F}} & \sum_{(\mathbf{x},y)\in\mathcal{X}_l} \underbrace{\frac{\partial l_l\left(yF_m(\mathbf{x})\right)}{\partial F_m(\mathbf{x})}}_{yw_l}\cdot f(\mathbf{x}) \\ & + \beta \sum_{\mathbf{x}\in\mathcal{X}_u} \underbrace{\frac{\partial l_u\left(F_m(\mathbf{x})\right)}{\partial F_m(\mathbf{x})}}_{\hat{y}w_u}\cdot f(\mathbf{x}),\end{aligned} \tag{4}$$

with w_l being the weight for a labeled and w_u and $\hat{y}$ denoting the pseudo weight and label for unlabeled samples.

In the following sections, we first introduce the learning mechanism for a single instance and then extend it to the case of multiple-instances.

3.1. Instance Regularization: SERBoost

For the instance based loss on labeled data any arbitrary loss function can be used. Due to its robustness, we employ the log-likelihood for the loss over labeled samples in form of

$$\begin{aligned}\mathcal{L}_l(\mathcal{X}_l) &= \sum_{(\mathbf{x},y)\in\mathcal{X}_l} \log\left(1+e^{-2yF(\mathbf{x})}\right) \\ &= \sum_{(\mathbf{x},y)\in\mathcal{X}_l} -yF(\mathbf{x}) + \log\left(e^{F(\mathbf{x})}+e^{-F(\mathbf{x})}\right).\end{aligned} \tag{5}$$

To incorporate unlabeled samples in the loss function we will take a similar approach as Saffari *et al*. [17] and extend their off-line expectation regularization formulation for on-line boosting. In expectation regularization the probability $P(y|\mathbf{x})$ for a sample belonging to one class is compared with a given prior probability $P_p(y|\mathbf{x})$. The loss function is extended by a term for unlabeled data, which incorporates this prior information. Saffari *et al*. define the unlabeled loss as the Kulbach-Leibler-divergence between prior probability and the current model and show that the optimization reduces to minimizing the cross entropy

$$H(P_p,P) = \sum_{z\in\{-1,1\}} -P_p(z|\mathbf{x})\log P(z|\mathbf{x}) \tag{6}$$

Thus, the loss over unlabeled samples results in

$$\begin{aligned}\mathcal{L}_u(\mathcal{X}_u) &= \sum_{\mathbf{x}\in\mathcal{X}_u} H(P_p,\hat{P}) \\ &= \sum_{\mathbf{x}\in\mathcal{X}_u} -y_p(\mathbf{x})F(\mathbf{x}) + \log\left(e^{F(\mathbf{x})}+e^{-F(\mathbf{x})}\right).\end{aligned} \tag{7}$$

Here, $y_p(\mathbf{x}) = 2P_p(y=1|\mathbf{x}) - 1 \in [-1,1]$ is the prior soft label. For the prior probability $P_p(y=1|\mathbf{x})$ any available prior information can be used.

Note that the formulation of Eq. (7) is the logit loss of Eq (5) in case the prior soft label $y_p(\mathbf{x})$ is casted to a hard label $\in\{-1,1\}$. As a result, labeled and unlabeled loss are coherent.

3.1.1 Online Learning

In order to use gradient boost, we need to compute the negative gradients a_{ij} with respect to the current classifier of the logit loss function:

$$\begin{aligned}a_i(z) &= \frac{\partial\, zF(\mathbf{x}_i) - \log\left(e^{F(\mathbf{x}_i)}+e^{-F(\mathbf{x}_i)}\right)}{\partial F(\mathbf{x}_i)} \\ &= z - \tanh(F(\mathbf{x}_i)),\end{aligned} \tag{8}$$

where z is either the hard label y for labeled samples, or the soft prior label $y_p(\mathbf{x}_i)$ for unlabeled samples. Employing the negative gradients the overall objective to optimize becomes

$$\arg\max_{f\in\mathcal{F}} \sum_{(\mathbf{x},y)\in\mathcal{X}_l} a_i(y) + \beta \sum_{\mathbf{x}\in\mathcal{X}_u} a_i(y_p(\mathbf{x})). \tag{9}$$

The weight w_i for a given sample $\mathbf{x}_i$ is defined as

$$\begin{aligned}&\forall\,(\mathbf{x}_i,y_i)\in\mathcal{X}_l: \\ &\quad w_i = |a_i(y_i)| = |y_i - \tanh(F(\mathbf{x}_i))|\,,\end{aligned} \tag{10}$$

and for unlabeled samples, the weights and pseudo-labels are

$$\begin{aligned}&\forall\,(\mathbf{x}_i)\in\mathcal{X}_u: \\ &\quad w_i = \beta\,|a_i(y_p(\mathbf{x}_i))| = \beta\,|y_p(\mathbf{x}_i) - \tanh(F(\mathbf{x}_i))| \\ &\quad y_i = \mathbb{I}\left(\beta y_p(\mathbf{x}_i) > \tanh(F(\mathbf{x}_i))\right)\end{aligned} \tag{11}$$

3.2. Bag Regularization: MIL-SERBoost

Based on the on-line SERBoost algorithm, we now proceed to introduce our on-line semi-supervised multiple instance learning model.

Let $\mathcal{X}_l^{\mathcal{B}} = \{(\mathcal{B}_1^l, y_1), \ldots, (\mathcal{B}_{N_l}^l, y_{N_l})\}$ and $\mathcal{X}_u^{\mathcal{B}} = \{\mathcal{B}_1^u, \ldots, \mathcal{B}_{N_u}^u\}$ denote the set of labeled and unlabeled bags, where $\mathcal{B} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_B}\}, \forall\mathbf{x}\in\mathbb{R}^D$ is a bag containing $N_{\mathcal{B}}$ samples and $y\in\mathcal{Y}=\{0,1\}$.

Note that in tracking we will train the classifier in an on-line manner and provide one positive and one negative bag per frame; however, the following derivations are kept general and the algorithm is by no means limited to tracking, but applicable in any other machine learning task.

MILBoost [20, 13] tries to minimize the negative log-likelihood as

$$\mathcal{L}_l(\mathcal{X}_l^{\mathcal{B}}) = -\sum_{i=1}^{N_l}\sum_{z\in\mathcal{Y}} \mathbb{I}(z=y_i)\log(P(y=z|\mathcal{B}_i^l)), \tag{12}$$

where $P(y|\mathcal{B}_i)$ is the estimated posterior by the model. In order to incorporate also unlabeled bags, we take the same approach as for unlabeled instances in Sec. 3.1 and define the loss over the unlabeled bags as the deviation of the model from the prior. We impose the *prior* conditional probability in the form of $P_p(y|\mathcal{B})$, which is calculated via the posterior model for bags. The objective is to minimize the negative log-likelihood over both the labeled and unlabeled bags. The cross entropy between the prior and the model is again derived using the Kullback-Leibler (KL) divergence between these two distributions as

$$\mathcal{L}_u(\mathcal{X}_u^{\mathcal{B}}) = -\sum_{i=1}^{N_u}\sum_{z\in\mathcal{Y}} P_p(z|\mathcal{B}_i^u)\log(P(z|\mathcal{B}_i^u)). \tag{13}$$

Using the logistic regression as the probabilistic model of boosting [6], we apply the model posterior for an instance as

$$P(y=1|\mathbf{x}) = \frac{e^{F(\mathbf{x})}}{e^{F(\mathbf{x})}+e^{-F(\mathbf{x})}}. \tag{14}$$

To model the probability of a bag being positive given the probabilities of its instances, we follow the approach of Lin *et al*. [13] who define the posterior for a bag by the *geometric mean* function as

$$P(y=1|\mathcal{B}_i) = 1-\Big[\prod_{j=1}^{N_{\mathcal{B}_i}}(1-P(y=1|\mathbf{x}_{ij}))\Big]^{1/N_{\mathcal{B}_i}}. \tag{15}$$

This is suited better than the original *Noisy OR* definition introduced by [20], where the posterior probability converges towards 1 for a larger number of samples independent of their instance prior probability. The prior probability for a bag $P_p(y|\mathcal{B}_i)$ is computed the same way as the posterior from instance probabilities, where any prior instance information can be used in the algorithm.

3.2.1 Online Learning

In order to use gradient boost [6], we need to compute the negative derivatives a_{ij} of the log-likelihood of the bags with respect to the response of the classifier to their instances:

$$a_{ij}(z) = \frac{\partial\log(P(y=z|\mathcal{B}_i))}{\partial F(\mathbf{x}_{ij})} \tag{16}$$

For the geometric mean model explained above the gradients result in

$$a_{ij}(z) = \frac{2}{N_{\mathcal{B}_i}}\frac{z-P(y=1|\mathcal{B}_i)}{P(y=1|\mathcal{B}_i)}P(y=1|\mathbf{x}_{ij}). \tag{17}$$

Using the gradients, the optimization over combined labeled and unlabeled loss is written as

$$\begin{aligned}\arg\max_{f\in\mathcal{F}} &\sum_{\mathcal{B}_i^l\in\mathcal{X}_l^{\mathcal{B}}}\sum_{z\in\mathcal{Y}}\mathbb{I}(y_i=z)\sum_{j=1}^{N_{\mathcal{B}_i}}a_{ij}(z)f(\mathbf{x}_{ij}) + \\ &\beta\sum_{\mathcal{B}_i^u\in\mathcal{X}_u^{\mathcal{B}}}\sum_{z\in\mathcal{Y}}P_p(y=z|\mathcal{B}_i)\sum_{j=1}^{N_{\mathcal{B}_i}}a_{ij}(z)f(\mathbf{x}_{ij}).\end{aligned} \tag{18}$$

Derived from the above formulation, following weights for labeled samples and weights and pseudo labels for unlabeled samples are defined:

$$\forall\,(\mathcal{B}_i,y_i)\in\mathcal{X}_l^{\mathcal{B}},\forall\mathbf{x}_{ij}\in\mathcal{B}_i : w_{ij} = |a_{ij}(y_i)| \tag{19}$$

$$\begin{aligned}\forall\,\mathcal{B}_i\in\mathcal{X}_u^{\mathcal{B}},\forall\mathbf{x}_{ij}\in\mathcal{B}_i :& \\ w_{ij} =& \beta\Big|\sum_{z\in\mathcal{Y}}P_p(z|\mathcal{B}_i^u)a_{ij}(z)\Big| \\ y_{ij} =& \mathbb{I}\Big(\beta\sum_{z\in\mathcal{Y}}P_p(z|\mathcal{B}_i^u)a_{ij}(z)>0\Big)\end{aligned} \tag{20}$$

4. MIL-SERBoost in Tracking

For the purpose of visual tracking we propose to combine both, supervised and unsupervised boosting by a convex combination

$$\mathcal{L}(\mathcal{X}) = \lambda\mathcal{L}_l(\mathcal{X}_l) + (1-\lambda)\mathcal{L}_u(\mathcal{X}_u) \tag{21}$$

Thereby, we can cover the whole range from fully supervised to fully unsupervised learning with an appropriate choice of λ. The fundamental difference to On-line SemiBoost is that the supervised loss is kept over all frames. In doing so we combine the benefits of robust loss functions and semi-supervised learning by allowing the unsupervised regularization term to amplify or weaken the supervised decision.

We provide a bag as both labeled and unlabeld ($\mathcal{X}^{\mathcal{B}} = \mathcal{X}_l^{\mathcal{B}} = \mathcal{X}_u^{\mathcal{B}}$) to the algorithm. The weight for a given sample then is calculated as the sum over both gradients, resulting in a reduction of Eq. (19) and Eq. (20) to

$$\forall\,(\mathcal{B}_i, y_i) \in \mathcal{X}^{\mathcal{B}}, \forall \mathbf{x}_{ij} \in \mathcal{B}_i : \tag{22}$$
$$w_{ij} = \Big|\lambda a_{ij}(y_i) + (1-\lambda)\sum_{z\in\mathcal{Y}} P_p(z|\mathcal{B}_i^u)a_{ij}(z)\Big|$$
$$y_{ij} = \mathbb{I}\Big(\lambda a_{ij}(y_i) + (1-\lambda)\sum_{z\in\mathcal{Y}} P_p(z|\mathcal{B}_i^u)a_{ij}(z) > 0\Big).$$

Eq.(22) nicely illustrates the effect of the regularization term. In case it coincides with the given supervisory decision, it enforces learning the sample with the given label by increasing its weight. Contrary, the regularizer has the ability to down weight the sample influence and also can provide a label switch, depending on the choice of the regularization parameter λ. The algorithmic work-flow, as it is applied for tracking, is illustrated in Alg. 1.

Algorithm 1 MIL-SERBoost for Tracking

Require: Initial object location
Require: Classifier $F(\mathbf{x})$ and prior classifier $F_p(\mathbf{x})$
Train prior classifier $F_p(\mathbf{x})$ and classifier $F(\mathbf{x})$ in supervised manner on first frame
while next frame exists **do**
  Load next frame
  Detect object in frame: $position = \arg\max F(\mathbf{x}) \;\; s.t. \;\; F(\mathbf{x}) > 0$
  if object was detected **then**
    Generate bags $\mathcal{B}_i$ containing samples $\mathbf{x}_{ij}$
    for all $(\mathcal{B}_i, y_i) \in \mathcal{X}^{\mathcal{B}}$ **do**
      Evaluate prior instance probabilities $P_p(y_{ij} = 1|\mathbf{x}_{ij})$ using $F_p(\mathbf{x})$ and Eq. 14
      Calculate the prior bag probability $P_p(y_i = 1|\mathcal{B}_i)$ based on instance probabilities via Eq. 15
      Calculate instance weights w_{ij} and pseudo labels y_{ij} with Eq. 22
      Update the classifier $F(\mathbf{x})$ with all samples $\mathbf{x}_{ij}$ and their related weights and labels (w_{ij}, y_{ij})
    end for
  end if
end while

For sampling the bags, we follow a simple strategy: we use the detected location of the object as a center point and sample bags which are very close to the center forming the positive bags or do not overlap considerably with the target object which provides the negative bags. This way, we could make sure that the multiple instance conditions are satisfied for the sampled bags.

The regularization performance strongly depends on the accuracy of provided prior information. For tracking we utilize the prior classifier trained on the very first frame; during tracking the prior is not updated. SemiBoost works better if an adaptive prior is provided [19], such that the classifier is not kept too tight to the initial prior. Contrary, we do not require an adaptive prior as we explicitly encode the adaptivity in our algorithm via the self learning part.

In case no prior information is available, that is bag instances can not be classified and $P_p(y = 1|\mathbf{x}) = 0.5$, the prior bag probability also encodes maximum entropy with $P_p(y = 1|\mathcal{B}^u) = 0.5$. The weight for instances in an unlabeled bag thus evaluates to 0 according to Eq. 20, ensuring that unlabeled data does not have any influence in the learning process.

5. Experiments

In tracking-by-detection appearance changes and (partly) occlusions constitute the main two problems where algorithms fail. Thus, we want to evaluate our algorithms for both situations. In case of appearance changes an algorithm mostly lead by a supervised loss is outperforming an unlabeled approach, simply because it guarantees enough adaptivity to object changes. On the other hand in case of occlusions the presence of a prior via regularization will give better results. We want to underline this fact by applying our algorithm with various settings of λ.

For evaluation we chose several public available benchmark sequences and we use simple Haar features to allow for a comparison to other state-of-the-art algorithms, namely MILBoost [2], On-line SemiBoost [10] and On-line AdaBoost [15]; for these algorithms tracking results are provided by the authors. In addition we use the Overlap-Criterion of the VOC Challenge[1], which computes the overlap score as

$$(roi_T \cap roi_{GT})/(roi_T \cup roi_{GT}), \tag{23}$$

where roi_T is the tracker detection window and roi_{GT} the ground truth. We chose this kind of measurement, because it better illustrates the detection

[1] http://www.pascal-network.org/challenges/VOC/voc2009

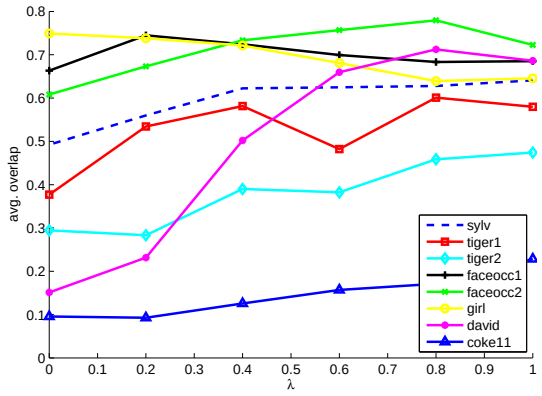

Figure 2. Tracking performance as average detection window and ground truth overlap in dependence of the regularization parameter λ.

Sequence	MILSER	MIL	OSB	OAB
sylv	**0.63**	0.61	0.46	0.50
david	**0.71**	0.54	0.31	0.32
faceocc2	**0.78**	0.65	0.63	0.64
coke11	0.18	**0.29**	0.12	0.20
tiger1	**0.60**	0.51	0.17	0.27
tiger2	0.46	**0.50**	0.08	0.25
faceocc1	0.68	0.63	**0.71**	0.47
girl	0.64	0.53	**0.69**	0.38

Table 1. Tracking results on the benchmark sequences measured as average detection window and ground truth overlap over 5 runs per sequence. For all sequences the regularization parameter λ was set to 0.8. Bold and underlined numbers indicates best and second best performance.

accuracy compared to raw pixel based distance measurements.

Since the regularization term is depending on a prior, λ expresses a trade-off between an adaptive algorithm able to follow appearance changes, and a more conservative approach where the regularizer ties the classifier to its prior. For an appropriate choice of λ, thus one also has to take into consideration the application domain of the tracker and weight between desired adaptive or conservative behavior.

If one knows the characteristics of the problem, λ can be chosen appropriately beforehand; however, if this is not the case a predefined value is required. Fig. 2 explicitly shows the expected result. As it can be seen, $\lambda = 0.8$ works reasonably well across all the sequences. Therefore, if no a priori knowledge is available we suggest to set $\lambda = 0.8$. On sequences *david*, *sylv*, and *coke* which are characterized by fast motion and rapid appearance changes, and *faceocc2* containing object modifications, the algorithm performs best with λ close to one; thus, mostly tracking and learning in a supervised manner. For sequences *girl* and *faceocc1*, λ equal to a small value yields best results. This means a setting similar to SemiBoost is appropriate, where the regularization term is the domination part. The remaining sequences *tiger1*, and *tiger2* are constituted by both appearance changes and (partly occlusions), thus a configuration, which allows both labeled and unlabeled loss to influence the learner is most suitable.

Table 1 reports our tracking results with the above choice of $\lambda = 0.8$ for all sequences. This clearly illustrates that our algorithm provides state-of the art tracking results compared to other methods. If we would have chosen λ according to the prior knowledge about the scene, tracking results improve (see Fig. 2). Fig. 3 shows various images from the tracking sequences together with the detection windows obtained by the different algorithms.

6. Conclusion

In this work, we have introduced a new on-line semi-supervised multiple-instance learning method and applied it to the task of visual tracking. Our method combines the robustness of semi-supervised updates towards occlusions and the flexibility of multiple instance learning on where to select positive updates. In order to increase the adaptivity of semi-supervised boosting, we use a combined loss during ongoing tracking which simultaneously performs both supervised and unsupervised updates. In the experiments, we showed that our approach is as good as or improves over state-of-the-art methods and on average performs better than applying MILBoost or SemiBoost separately. However, we also highlighted that different tracking application domains can require different settings for the convex combination of the loss functions. Hence, in future work we plan to investigate alternatives for finding more general convex combinations.

References

[1] S. Avidan. Ensemble tracking. *PAMI*, 29(2):261–271, 2007.

[2] B. Babenko, M.-H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In *CVPR*, 2009.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002.

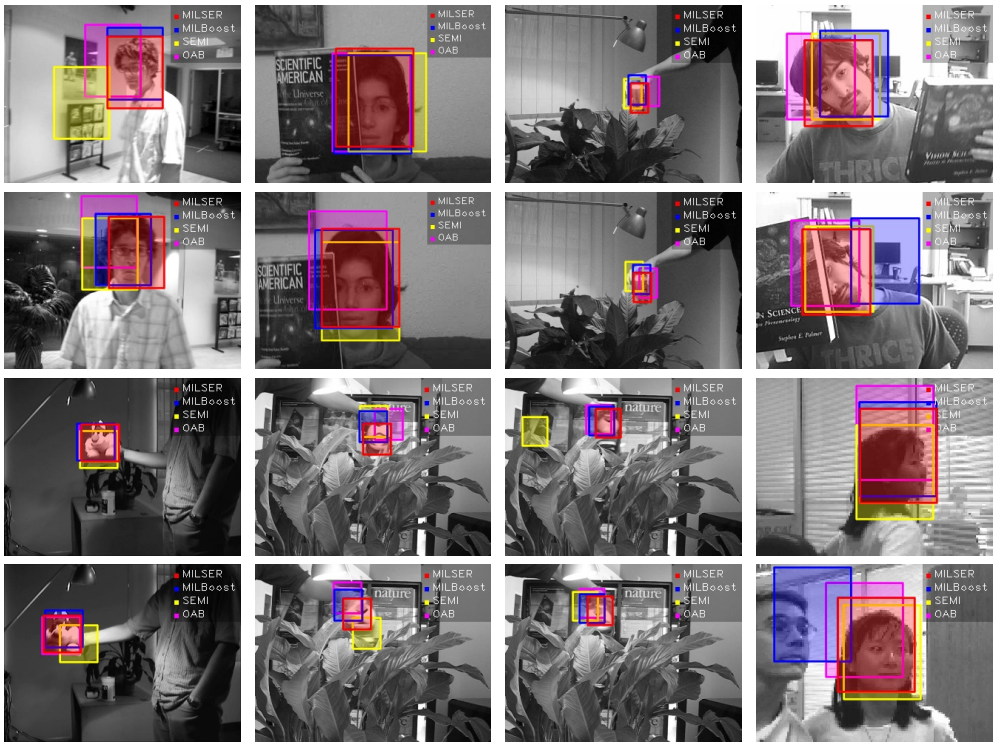

Figure 3. Exemplariy images from the tracking sequencs. (red) MILSERBoost, (blue) MILBoost, (yellow) SemiBoost, (magenta) AdaBoost.

[4] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. Cambridge, MA, 2006.

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

[6] J. Friedman. Greedy function approximation: A gradient boosting machine. *The annals of statistics*, pages 1189–1232, 2001.

[7] A. Goldberg, M. Li, and X. Zhu. Online manifold regularization: A new learning setting and empirical study. In *ECCV*, 2008.

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

[9] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, 2006.

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

[11] C. Leistner, A. Saffari, P. Roth, and H. Bischof. On robustness of on-line boosting - a competitive study. In *OLCV*, 2009.

[12] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. In *CVPR*, pages 1–8, 2007.

[13] Z. Lin, G. Hua, and L. Davis. Multiple instance feature for robust part-based object detection. In *CVPR*, 2009.

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

[15] N. C. Oza and S. Russell. Online bagging and boosting. In *Proc. Artificial Intelligence and Statistics*, pages 105–112, 2001.

[16] M. Özuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *CVPR*, 2007.

[17] A. Saffari, H. Grabner, and H. Bischof. Semi-supervised boosting via expectation regularization. In *ECCV*, 2008.

[18] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

[19] S. Stalder, H. Grabner, L. van Gool, E. Zurich, and K. Leuven. Beyond Semi-Supervised Tracking: Tracking Should Be as Simple as Detection, but not Simpler than Recognition. In *ICCV, WS on On-line Learning for Computer Vision*, 2009.

[20] P. Viola, J. Platt, and C. Zhang. Multiple instance boosting for object detection. In *NIPS*, volume 18, pages 1417–1424. MIT Press, 2006.

[21] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 2006.