Using discriminative motion context for on-line visual object tracking

Stefan Duffner, Christophe Garcia

Using discriminative motion context for on-line visual object tracking

Stefan Duffner and Christophe Garcia

***Abstract*—In this paper, we propose an algorithm for on-line, real-time tracking of arbitrary objects in videos from unconstrained environments. The method is based on a particle filter framework using different visual features and motion prediction models. We effectively integrate a discriminative on-line learning classifier into the model and propose a new method to collect negative training examples for updating the classifier at each video frame. Instead of taking negative examples only from the surroundings of the object region, or from specific background regions, our algorithm samples the negatives from a contextual motion density function in order to learn to discriminate the target as early as possible from potential distracting image regions. We experimentally show that this learning scheme improves the overall performance of the tracking algorithm. Moreover, we present quantitative and qualitative results on four challenging public datasets that show the robustness of the tracking algorithm with respect to appearance and view changes, lighting variations, partial occlusions as well as object deformations. Finally, we compare the results with more than 30 state-of-the-art methods using two public benchmarks, showing very competitive results.**

***Index Terms*—Image sequence analysis, Image motion analysis, Object detection**

I. Introduction

We consider the problem of automatically tracking a single arbitrary object in a video, where the algorithm is initialised in the first frame from a bounding box around the object that is to be tracked. No prior knowledge about appearance, shape, or motion of the objects or the environment is used. Also, we focus here on *on-line* tracking, where at each time step, only past and present but no future information is used. Applications for on-line visual object tracking are numerous, including, for example, video indexing, Human-Computer or Human-Robot Interaction, video-surveillance, traffic monitoring, or autonomous driving.

In real-world scenarios, this problem is challenging as the object to track may change considerably its appearance, shape, size, and pose in the image (like the articulated human body for example). Furthermore, the object can be partially occluded by itself, other objects, or the environment. The object may also move abruptly or in unpredictable ways. Finally, the environment, *i.e*. the image background, may change considerably and rapidly in videos from moving cameras and be affected by varying illumination.

This weakly constrained setting requires a tracking algorithm that is able, with few data, to build an object (and possibly a scene) model that can well discriminate the object from the background, that copes with complex scene and object motion, and that is able to adapt to large changes of the object's appearance, size, and shape. But this adaptation also holds the risk of "drift" (the so-called stability-plasticity dilemma [1]), when the object model gradually includes information not belonging to it, *i.e*. from the background. As a consequence, the tracking algorithm will not precisely track the object any more or even lose it completely at some point in time.

S. Duffner and C. Garcia are with the Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France

A. Related Work

Numerous methods for on-line tracking of arbitrary objects have been published over the last years [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. Many recent works [16], [2], [4], [5], [3], [6], [9], [10], [11], [12] propose a tracking-by-detection framework, where a discriminative detector is trained with object and background image samples. At each frame of the video, this detector is applied inside a search window to estimate the current position of the object, and then the model is updated using this estimate and the current image. The advantage of this approach is that no specific motion model needs to be designed and parameterised, and the output is deterministic. Also the discriminative machine learning methods that are used are rather well studied in static settings, *e.g*. their performance on object detection in still images.

Another approach is to detect local feature points and match them from one frame to the next, in order to track an object [17], [7], [15], [18]. The problem here is to select and match prominent and discriminative feature points, taking into consideration the fact that some of them might disappear and reappear during tracking. Other works, *e.g*. [19], [20], [21], [22], [11], use some type of foreground-background segmentation to track the object. This can be in form of a parametric or active contour [23], [19], or a pixel-wise foreground mask [22], [24], [25] of the object, for example. Naturally, this alleviates the problem of drift, especially with highly deformable objects.

Classically, the tracking problem has been tackled in a probabilistic way with recursive Bayesian filters like Kalman filters or particle filters [26], [27], [28], [21], [29], [30], [31], [14]. These methods are able to estimate the posterior state distribution of the tracked object and allow for maintaining several state hypotheses. Usually, they explicitly integrate motion models used to predict the next object state by defining a probabilistic transition function independent from the image observations. Some particle filter techniques use some more

advanced motion models, like in Odobez *et al*. [27]; a dense parametric motion estimator with an affine model is applied to propose new state values, as we propose in this paper. Also similar to this paper, parametric motion models have been used to estimate background (*i.e*. camera) motion [32] and segment the object region from the background [33].

Other recently proposed approaches have also included this type of contextual motion information. For example, Yang *et al*. [34] introduced a method that, throughout a video, continuously discovers objects that move in the same direction as the tracked object by performing a motion correlation analysis. These auxiliary objects help to support and improve tracking by performing inference in a star-structured graphical model that includes their state.

Spatial context has also been exploited by using supporters, *i.e*. other objects or feature points around the target in the image. Grabner *et al*. [35], for example, extended the well-known Implicit Shape Model by detecting feature points in the image that have a correlated motion with the target. These supporters are matched from frame to frame and their relative displacement vectors are updated on-line. Wen *et al*. [36] also proposed a method that detects supporters (here called contributors) which are interest points within a local neighbourhood around the target, in order to improve the tracking performance. In addition, their method makes use of a longer-term temporal context using an on-line sub-space learning method that groups together observations from several frames. Similarly, the approach proposed by Sun *et al*. [37] tracks "helper" objects using an on-line Adaboost detector, initialised manually at the first frame. Their relative position is learnt on-line and used to predict the target object's position.

Dinh *et al*. [38] proposed a method using supporters as well as distractors, which are objects with similar appearance to the target. The distractors help to avoid confusion of the tracker with other similar objects in the scene, and they can possibly be used to reason about the objects' mutual occlusion. Supporters are not used directly for the target's state estimation but only to disambiguate between the target and its distractors. Hong *et al*. [39] recently proposed an approach based on the ℓ_1 tracker [9] that deals with distractors by automatically learning a metric not only between positive and negative examples but also within the collected negative examples, effectively replacing the originally proposed Euclidean distance.

Finally, Supančič and Ramanan [40] presented a self-paced learning tracker that also selects training examples from video frames in the past to perform long-term tracking, an idea that has also been used in the recent work of Hua *et al*. [41].

B. Motivation

The disadvantage with using supporting and distracting objects is that several objects need to be detected and tracked, which can be computationally expensive especially with a larger number of objects. Moreover, the success or failure of data association or, in some methods, matching local features points in successive video frames, heavily depends on the type of object to track and the surrounding background. This process can be error-prone and, in some situations, may rather harm the overall tracking performance. Finally, modelling the spatial, temporal, or appearance-based pairwise relationships between objects and/or interest points can lead to a combinatorial explosion and make the inference on the state space difficult.

To alleviate this problem, in this work, we propose a probabilistic method that dynamically updates the foreground and background model depending on distracting objects or image regions in the scene background. This contextual appearance information is extracted from moving image regions and used to train on-line a discriminative binary classifier that, in each video frame, detects the image region corresponding to the object to track.

Traditionally, these discriminative on-line classifiers used in tracking-by-detection approaches [16], [6], [10], [11], [12], [9] learn negative examples extracted from the image region surrounding the current target object region. This choice is motivated by the fact that the object will move only slightly from one frame to the other w.r.t. the background or other objects, and by computational speed. In contrast, our method uses a stochastic sampling process to extract negative examples from image regions that move. We call these: *contextual motion cues* (see Fig. 1). In that way, regions that correspond to possibly distracting objects are detected efficiently and early, *i.e*. without them having to be inside a search window and without scanning the whole image at each point in time. More precisely, the contributions of this paper are the following:

- a method for on-line learning of a discriminative classifier using stochastic sampling of negative examples from contextual motion cues in videos,
- the integration of this incremental discriminative model in an efficient adaptive particle filter framework combining effectively several visual cues,
- a thorough evaluation on difficult public benchmarks experimentally showing the performance increase from this type of on-line learning as well as an improvement over state-of-the-art tracking methods.

Compared to our previous work [42], we performed more extensive experiments, including the recent tracking benchmark VOT 2014 [43], and we validated our approach by evaluating it with different discriminative on-line tracking algorithms: Multiple Instance Learning (MIL) Tracker in addition to On-line Adaboost.

The paper is organised as follows. In Section II, we describe the overall tracking algorithm. Section III explains how the motion context is used to update the appearance models in the tracker. Experimental results are presented in Section IV, and conclusions are drawn in the last section.

II. Tracking algorithm

In order to be able to handle more complex, multi-modal state distributions in a computationally efficient way, we propose a tracking algorithm based on a recursive Bayesian framework. Assuming we have the observations $\mathbf{Y}_{1:t}$ from time 1 to t, we want to estimate the posterior probability

distribution over the state $\mathbf{X}_t$ at time t:

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) = \frac{1}{C}\, p(\mathbf{Y}_t|\mathbf{X}_t) \times \int_{\mathbf{X}_{t-1}} p(\mathbf{X}_t|\mathbf{X}_{t-1}) p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1})\, \mathrm{d}\mathbf{X}_{t-1}\ , \quad (1)$$

where C is a normalisation constant. As closed-form solutions are usually not available in practice, this estimation is implemented using a particle filter, *i.e.* sampling importance resampling (SIR) or bootstrapping. We refer to [44], [45] for more details on particle filters and only explain the main elements in the following sections.

A. Object state representation and inference

The state $\mathbf{X} = (x, y, v_x, v_y, s, e) \in \mathbb{R}^6$ of the object to track is described by an upright bounding box defined by the object's centre position (x, y) in the image, its 2D speed (v_x, v_y) in the image plane, scale (s), and eccentricity (e), *i.e.* the ratio of height and width. The state $\mathbf{X}_0$ is initialised manually by providing a bounding box around the object in the first frame. Then, for each video frame, the particle filter performs its classical steps of *predicting* particles $\mathbf{X}^{(i)}$ sampled from the proposal distribution $q(\mathbf{X}_t|\mathbf{X}_{t-1})$ and *updating* their weights according to the observation likelihood $p(\mathbf{Y}_t|\mathbf{X}_t)$, state dynamics $p_m(\mathbf{X}_t|\mathbf{X}_{t-1})$ and proposal (see Section II-B): $w_i = p(\mathbf{Y}_t|\mathbf{X}_t)\frac{p_m(\mathbf{X}_t|\mathbf{X}_{t-1})}{q(\mathbf{X}_t|\mathbf{X}_{t-1})}$, for each particle $i \in 1..N$. At the end of each iteration, the observation likelihood model parameters are updated using the mean of the posterior distribution $p(\mathbf{X}_x|\mathbf{Y}_{1:t})$. And finally, systematic resampling is performed.

B. State dynamics and proposal function

In order to cope with fairly complex motion of arbitrary objects in videos from a possibly moving camera, we use a proposal function composed of a mixture of three distributions:

$$\begin{aligned} q(\mathbf{X}_t|\mathbf{X}_{t-1}) = {} & \beta_m p_m(\mathbf{X}_t|\mathbf{X}_{t-1}) \\ & + \beta_f p_f(\mathbf{X}_t|\mathbf{X}_{t-1}) \\ & + \beta_d p_d(\mathbf{X}_t|\mathbf{X}_{t-1})\ , \end{aligned} \quad (2)$$

where β_m, β_f and β_d define the mixture weights, and $p_m(\mathbf{X}_t|\mathbf{X}_{t-1})$ is the state dynamics model defined by Eq. 3 and 4, $p_f(\mathbf{X}_t|\mathbf{X}_{t-1})$ is an optical flow-like motion-based proposal function, and $p_d(\mathbf{X}_t|\mathbf{X}_{t-1})$ proposes states coming from a discriminative on-line trained detector. In the following, we will describe each term in more detail.

The state dynamic model $p_m(\mathbf{X}_t|\mathbf{X}_{t-1})$ is defined for each individual component of $\mathbf{X}$. The position and speed components of the object are described by a mixture of a first-order auto-regressive model p_a with additive Gaussian noise and a uniform distribution p_u. If $\hat{x} = (x, y, v_x, v_y)$ denotes a position and speed component vector, we have:

$$p(\hat{x}_t|\hat{x}_{t-1}) = \alpha p_a(\hat{x}_t|\hat{x}_{t-1}) + (1-\alpha) p_u(\hat{x}_t|\hat{x}_{t-1})\ , \quad (3)$$

with $p_a(\hat{x}_t|\hat{x}_{t-1}) = \mathcal{N}(\hat{x}_{t-1}; 0, \hat{\mathbf{\Sigma}})$, and $p_u(\hat{x}_t|\hat{x}_{t-1}) = c$ with c being a constant (defined empirically) allowing for small "jumps" coming from the proposal function (Eq. 2). A simple first order model is used for the scale and eccentricity parameters, s and e. Let $\bar{x} = (s, e)$. Then:

$$p(\bar{x}_t|\bar{x}_{t-1}) = \mathcal{N}(\bar{x}_{t-1}; 0, \bar{\mathbf{\Sigma}})\,. \quad (4)$$

The second term of the proposal function:

$$p_f(\mathbf{X}_t|\mathbf{X}_{t-1}) = \mathcal{N}(\mathbf{d}(\mathbf{X_{t-1}}); 0, \mathbf{\Sigma}^f) \quad (5)$$

predicts the new state by performing a parametric robust motion estimation between the previous and current image of the region defined by $\mathbf{X}_t$ like in [46] and similar to optical flow computation. The output of this estimation is a set of parameters $\{d_x, d_y, d_s\}$ defining a affine transformation $\mathbf{d}(\cdot)$ that translates and scales the state vector $\mathbf{X}_{t-1}$ of the previous frame to best match the current image. As the motion estimation is performed on a pyramid of image scales, this term is very useful to compensate for large camera motion or abrupt object accelerations.

The last term:

$$p_d(\mathbf{X}_t|\mathbf{X}_{t-1}) = \mathcal{N}(\mathbf{X}^d; 0, \mathbf{\Sigma}^d) \quad (6)$$

uses the output $\mathbf{X}^d$ of a detector (see Section III) that has been trained on-line and that is applied in the neighbourhood around $\mathbf{X}_t$ on the current frame to predict the new object position and scale (as in [47], [29] for example). The variances $\mathbf{\Sigma}^f$ and $\mathbf{\Sigma}^d$ are relatively small compared to the ones in the auto-regressive model $\hat{\mathbf{\Sigma}}$ and $\bar{\mathbf{\Sigma}}$. Note that, to be coherent and to strictly preserve the probabilistic independence, the detector's output for the *previous* frame could be used as well but represents a less accurate proposal leading to a higher variance.

See Section IV for a summary of parameter values.

C. Observation likelihood

The observation likelihood function $p(\mathbf{Y}|\mathbf{X})$ that we propose is designed to be robust against object deformations, pose and illumination changes as well as partial occlusions. It is a geometric mean of three distributions corresponding to different visual cues:

$$p(\mathbf{Y}_t|\mathbf{X}_t) = \left(p_H(\mathbf{Y}_t|\mathbf{X}_t)\, p_S(\mathbf{Y}_t|\mathbf{X}_t)\, p_T(\mathbf{Y}_t|\mathbf{X}_t)\right)^{1/3}\ , \quad (7)$$

where p_H computes a local colour histogram likelihood ratio, p_S measures the global colour distribution similarity, and p_T is a texture likelihood. Taking the cube root of the product ensures that the overall likelihood distribution does not become too peaked. In the following, we explain each of the likelihood function in more detail.

1) Histogram likelihood ratio: The histogram likelihood function is defined as a ratio of foreground and background likelihoods:

$$p_H(\mathbf{Y}_t|\mathbf{X}_t) = \frac{p_{FG}(\mathbf{Y}_t|\mathbf{X}_t)}{p_{BG}(\mathbf{Y}_t|\mathbf{X}_t)}\,, \quad (8)$$

where

$$p_{FG}(\mathbf{Y}_t|\mathbf{X}_t) = \exp\left(-\lambda_{FG} \sum_{r=1}^{9} (D^2[h_t^*(r), h(r, \mathbf{X}_t)])\right)\ , \quad (9)$$

is the foreground likelihood defined over a grid of 3×3 regions r. D computes the Bhattacharyya distance between the

HSV histograms h_t extracted from state $\mathbf{X}_t$ and the respective reference histograms h_t^* initialised from the first frame, and λ_{FG} is a constant. Similarly, the background likelihood:

$$p_{BG}(\mathbf{Y}_t|\mathbf{X}_t) = \exp\left(-\lambda_{BG}(D^2[\hat{h}_t^*, \hat{h}(\mathbf{X}_t)])\right) , \qquad (10)$$

is defined by a reference histogram $\hat{h}_t^*$ from the first frame and one depending on the object's current state: $\hat{h}(\mathbf{X}_t)$. Here, $\hat{h}^*$ and $\hat{h}(\mathbf{X}_t)$ are computed over the image region twice as large as the object and *surrounding* it. All histograms contain two different quantisation levels, 4 and 8 bins, in the HSV colour space, using 4×4, respectively 8×8, bins for the H and S channels and 4/8 separate bins for the V channel [26]. The reference models h^* and $\hat{h}^*$ are updated linearly at each iteration using the object's current bounding box.

2) Global colour segmentation likelihood: In addition to the more local colour models with one histogram per object part, we also use a global colour histogram model based on a pixel-wise colour segmentation of foreground and background. This further helps to delimit the object boundaries. As with p_H, HSV colour histograms with separate colour and greyscale bins are extracted – one inside the current bounding box of the object, and one around it. Then, a probabilistic soft-segmentation is performed (similar to [24]) computing the probability $p(c_i|z_i)$ of each pixel i inside a search window belonging to the foreground $c = 1$ or background $c = 0$ given its colour z_i.

Then, the likelihood function is defined as:

$$p_S(\mathbf{Y}_t|\mathbf{X}_t) = \frac{\exp(-\lambda_S S_{FG}(\mathbf{X}_t)^2)}{\exp(-\lambda_S S_{BG}(\mathbf{X}_t)^2)} , \qquad (11)$$

where λ_S is a constant, S_{FG} is the proportion of foreground pixels, *i.e.* for which $p(c = 1|z) > 0.5$, *inside* the object's bounding box and S_{BG} is the proportion of foreground pixels *outside* the bounding box. Clearly, the better the bounding box delimits foreground and background of the segmentation the higher is this likelihood. The foreground and background histograms used for the segmentation are updated linearly at each iteration using the current bounding box.

3) Texture likelihood: The likelihood $p_T(\mathbf{Y}|\mathbf{X})$ is based on the (greyscale) texture of the object to track. This visual cue helps to track objects that have little discriminative colour information (for example in very dark environments) or in greyscale videos. A discriminative classifier is trained at the first frame using the object region as positive and the background regions as negative examples. Then, the classifier is updated at each iteration collecting positive and negative examples from the foreground and background respectively (see Section III). Here, we use the On-line Adaboost classifier presented by Grabner *et al.* [16] that uses Haar-like features, but any other on-line classifier could be used as well.

The likelihood is based on the detector's confidence $c_D \in [0, 1]$ for the image patch defined by $\mathbf{X}_t$:

$$p_D(\mathbf{Y}_t|\mathbf{X}_t) = \exp(-\lambda_D(1 - c_D)^2) , \qquad (12)$$

with λ_D being a constant.

III. Model adaptation with contextual cues

In this section, we will describe the main contribution of the proposed approach: a method to exploit motion context effectively for visual object tracking using a discriminative classifier that is trained on-line on specific parts of the input video. Our approach is different from previous work, where motion context or background motion has been integrated tightly in the tracking process, *e.g.* in the state dynamics, or where specific appearance models are used to avoid distractions in the background.

As mentioned earlier, in the particle filter, we use a binary discriminative classifier based on the On-line Adaboost (OAB) algorithm [16] for proposing new particles (Eq. 6) as well as for evaluating the observation likelihood (Eq. 12). The classifier is trained with the first video frame using the image patch inside the object's bounding box as a positive example and surrounding patches within a search window as negative examples. Then, the classifier is updated at each tracking iteration using the same strategy for extracting positive and negative examples. We refer to [16] for details on the model and how it is trained.

A. Background sampling

We propose to sample negative examples from image regions that contain motion and thus likely correspond to moving objects (see Fig. 1). The idea is that these regions may distract the tracker at some point in time. Therefore it is preferable to learn these negative examples as early as possible, *i.e.* as soon as they appear in the scene. One can see this as a kind of long-term prediction of possible negative samples, in contrast to the much shorter (frame-by-frame) time scale of the proposal function. To perform this negative sampling, we first compensate for camera motion between two consecutive frames using a classical parametric motion estimation approach [46]. We apply a three-parameter model to estimate the translation and scale of the scene, and then compute the intensity differences for each pixel with its corresponding pixel in the previous frame. This gives an image $M(x, y)$ approximating the amount of motion present at each position (x, y) of the current frame of the video. We then transform this image into a probability density function (PDF) $m(x, y)$ over the 2-dimensional image space:

$$m(x, y) = Z^{-1} \sum_{(u,v)\in\Omega(x,y)} M(u, v) , \qquad (13)$$

where $\Omega(x, y)$ defines an image region of the size of the bounding box of the object being tracked, centred at (x, y), and Z is a constant normalising the density function to sum up to 1. Thus, $m(x, y)$ represents the relative amount of motion inside the region centred at (x, y). Finally, N^- image positions (x, y) are sampled from this PDF corresponding to rectangles centred at (x, y), where, statistically, regions with high amount of motion are sampled more often than static image regions. This process is illustrated in Fig. 1.

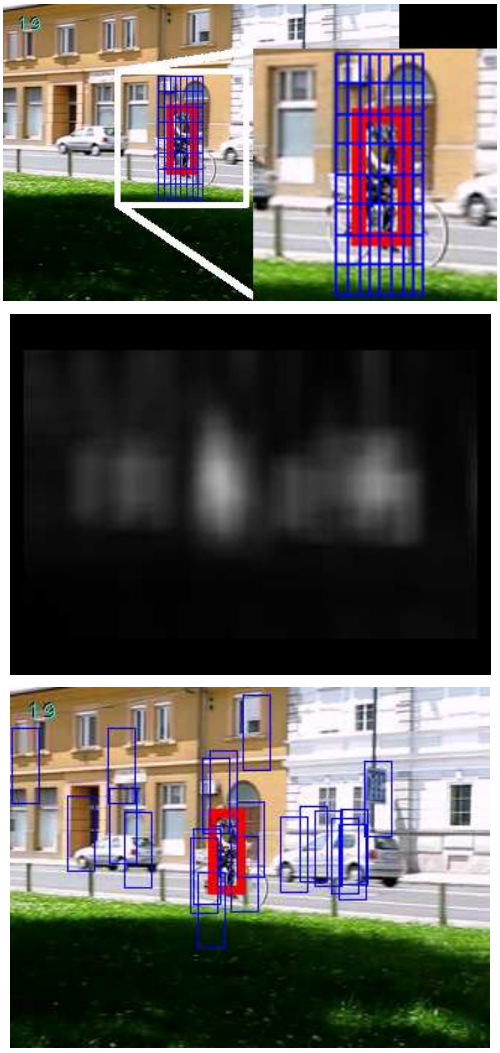

Fig. 1. Illustration of different sampling strategies of negative examples (blue). *Top:* traditional sampling at fixed positions within a search window around the object (red). *Middle:* the motion probability density function m (Eq. 13). *Bottom:* the proposed negative sampling from m.

B. Classifier update

The N^- image patches corresponding to the sampled regions as well as the positive example coming from the mean particle of the tracker are then used to update the classifier. In this case, the OAB method needs a balanced number of positives and negatives, thus the positive example is used N^- times, alternating positive and negative updates.

The advantage of sampling positions from these motion cues is that we do not need to care about explicitly detecting, initialising, tracking, and eventually removing a certain number of distracting objects at each point in time. Note that we could also sample regions of different scales but as scale does not change rapidly in most videos the benefit of this would be relatively small. Note also that the PDF could as well include appearance similarity with the tracked target. However, this would considerably increase the computational complexity.

IV. Experiments

A. Parameters

The following table summarises the tracking parameters that have been used for all the experiments:

α	c	$\hat{\Sigma}$	$\bar{\Sigma}$	$\Sigma^{f/p}$
0.5	$\frac{1}{15}$	$(7,7)$	$(0.001, 0.001)$	$(1, 1, 10^{-4}, 10^{-4})$

β_m	β_f	β_d	λ_{FG}	λ_{BG}	λ_S	λ_D
0.7	0.2	0.1	120	36	0.1	10

The variances for x and y values are scaled by $\frac{w}{200}$, w being the current width of the bounding box. The variances for s and e are relatively small, thus more rapid scale and eccentricity changes cannot be accommodated easily, but on the other hand the overall tracking robustness is increased. We should highlight that only 100 particles have been used throughout all experiments. This turns out to be sufficient due to our design of effective proposal and discriminative likelihood functions.

B. Datasets

We performed a quantitative evaluation on four challenging public tracking datasets that are described below.

1) Babenko: The first dataset[1] has been constructed by Babenko *et al.* [6] from various other publications, and it has been used by many others afterwards. It contains 8 videos (with more than 5 000 frames) of objects or faces that undergo mostly rigid deformations and some rather large lighting variations as well as partial occlusions. Most of these sequences are actually in grey-scale format (except "David", "Girl", and "Face Occlusions 1").

2) Non-rigid objects: The second, more challenging dataset[2] is composed of 11 videos (around 2 500 frames) showing moving objects that undergo considerable rigid and non-rigid deformations. This dataset has also been used by [11] and partially by [8] among others.

3) VOT2013: The third dataset[3] has been used for the Visual Object Tracking (VOT) Challenge 2013 [48]. It contains 16 videos that have been automatically selected from a larger set by maximising the variability in terms of certain criteria, such as camera motion, illumination change, occlusion, size change, or motion. Four of these sequences ("David", "diving", "face", "jump") are also part of the first or second dataset.

4) VOT2014: This is the 2014 version of the VOT2013 dataset available from the same web site. The dataset contains 25 challenging videos.

Note that similar benchmarks ([49], [50]) are available but due to limited space, we cannot report all the results here.

C. Evaluation

We performed several experiments with different evaluation protocols. For the first two datasets we evaluated the robustness of the proposed algorithm by measuring the proportion of correctly tracked frames. A frame is counted as correct, if the

[1] http://vision.ucsd.edu/∼bbabenko/project_miltrack.html
[2] http://lrs.icg.tugraz.at/research/houghtrack/
[3] http://votchallenge.net/

	fixed	fixed+rand.	motion	fixed+mot.
David	99.9	98.9	99.9	**100.0**
Sylvester	64.9	60.9	74.3	**96.2**
Girl	45.7	32.1	**59.2**	51.9
Face Occlusions 1	69.5	94.6	92.9	**95.5**
Face Occlusions 2	67.2	73.2	78.1	**93.6**
Coke	94.7	90.3	**94.9**	90.5
Tiger 1	51.2	45.0	**63.2**	56.6
Tiger 2	93.0	97.4	95.9	**97.7**
average	73.3	74.1	82.3	**85.3**

TABLE I
BABENKO SEQUENCES: PERCENTAGE OF CORRECTLY TRACKED FRAMES WITH FIXED NEGATIVE SAMPLING, SAMPLING FROM MOTION, COMBINED FIXED+RANDOM SAMPLING, AND FIXED+MOTION SAMPLING.

	fixed	fixed+rand.	motion	fixed+mot.
Cliff-dive 1	82.9	89.3	83.2	**96.8**
Motocross 1	72.4	68.5	**88.9**	85.7
Skiing	72.2	64.1	75.6	**79.2**
Mountain-bike	**100.0**	**100.0**	**100.0**	**100.0**
Cliff-dive 2	**76.7**	67.7	71.6	51.6
Volleyball	22.5	39.4	**81.9**	78.1
Motocross 2	80.0	83.3	71.3	**98.7**
Transformer	84.8	**90.2**	85.6	89.7
Diving	26.7	30.2	32.2	**60.8**
High Jump	40.6	43.5	38.3	**49.9**
Gymnastics	97.0	97.0	88.5	**99.1**
average	68.7	70.3	74.3	**80.9**

TABLE II
NON-RIGID OBJECT SEQUENCES: PERCENTAGE OF CORRECTLY TRACKED FRAMES WITH FIXED NEGATIVE SAMPLING, SAMPLING FROM MOTION, COMBINED FIXED+RANDOM, AND FIXED+MOTION SAMPLING.

proposal	likelihood	babenko	non-rigid
p_m	p_H	57.7	69.0
p_m	p_H, p_S	55.1	71.9
p_m	p_H, p_S, p_T	83.7	78.1
p_m, p_f	p_H, p_S, p_T	77.9	**81.1**
p_m, p_d	p_H, p_S, p_T	84.7	79.7
p_m, p_f, p_d	p_H, p_S, p_T	**85.3**	80.9

TABLE III
PERCENTAGE OF CORRECTLY TRACKED FRAMES FOR THE BABENKO AND NON-RIGID SEQUENCES WITH DIFFERENT COMBINATIONS OF PROPOSAL AND LIKELIHOOD TERMS (*c.f.* EQ. 2 AND 7).

tracking accuracy $A = \frac{R_T \cap R_{GT}}{R_T \cup R_{GT}}$ is greater than a threshold, where R_T is the rectangle corresponding to the mean particle from the tracking algorithm, and R_{GT} is the ground truth rectangle surrounding the object. We set the threshold to 0.2 in order not to penalise fixed-size, fixed-ratio trackers in our comparison. For every experiment and sequence, the proposed algorithm has been run 5 times and the average result is reported.

For the VOT datasets, we used the evaluation protocol of the VOT2013/2014 challenges, which measures accuracy and robustness. For evaluating the accuracy, the measure A, defined above, is used. The robustness is measured in terms of number of tracking failures, where trackers are re-initialised after failures. Every sequence is evaluated 15 times and the average results are reported. In addition to this "baseline" experiment, there are two other experiments using the same data. In the "region-noise" experiment the initial bounding box is slightly shifted randomly for each run, and in the "greyscale" experiment (only in VOT2013), each video is transformed into greyscale format. See [48] and [43] for more details.

D. Results

In the first experiments, we evaluated different strategies for the collection of negative examples of the discriminative OAB classifier, as explained in Section III. We compared four different strategies:

- **fixed:** N^- negatives are taken from fixed positions around the positive example inside the search window, which is twice the size of the object's bounding box.
- **fixed+random:** $N^-/2$ examples are taken from fixed position (as for "fixed"), and $N^-/2$ examples are sampled from random image positions.
- **motion:** N^- negative examples are sampled from the contextual motion distribution m (Eq. 13).
- **fixed+motion:** $N^-/2$ examples are taken from fixed positions, and $N^-/2$ examples are sampled from the contextual motion distribution.

In any case, the negative examples do not overlap more than 70% with the positive ones in the image.

Table I and II show the results for the first two datasets in terms of the percentage of correctly tracked frames. In most cases, the sampling of negative examples from the contextual motion PDF, *i.e.* "motion" and "fixed+motion", improves the tracking performance. For the Babenko sequences, the improvement is smaller because there are not many other moving objects that can distract the tracker. On average, the best strategy is "fixed+motion", with a relative improvement of around 7.5%. In another experiment, we studied the influence of each proposal and likelihood term (Eq. 2 and 7) on the overall tracking performance. Table III summarises the results. Some terms seem to be complementary, like the motion-based proposal p_f and the detector-based one p_d. On average, the combination of *all* terms gives the best performance. We further replaced OAB in our algorithm with the MIL Online Boosting classifier [6] in order to see if our proposed method for sampling negatives from motion context depends on the underlying classifier. The results are summarised in Table IV, and we can see that for both classifiers OAB and MIL, the use of motion context outperforms the other sampling strategies. We use this strategy in combination with OAB for the following experiments and call the overall tracking algorithm "Motion Context Tracker" (MCT).

We compared the proposed Motion Context Tracker (MCT), with other state-of-the-art trackers on the two datasets: Hough-

		fixed	fixed+rand.	motion	fixed+mot.
Babenko	OAB	73.28	74.06	82.30	**85.25**
	MIL	70.24	68.19	66.05	**70.94**
Non-rigid objects	OAB	68.71	70.30	74.29	**80.87**
	MIL	74.04	73.78	**77.49**	75.95

TABLE IV
BABENKO SEQUENCES: AVERAGE PERCENTAGE OF CORRECTLY TRACKED FRAMES WITH THE PROPOSED METHOD USING DIFFERENT ONLINE CLASSIFICATION ALGORITHMS.

	HT [11]	TLD [7]	PixelTrack [25]	MIL [6]	STRUCK [12]	PF	MCT
David	44.0	95.3	**100.0**	76.2	58.0	58.7	**100.0**
Sylvester	**100.0**	86.6	49.4	56.9	99.6	96.7	96.2
Girl	60.7	92.0	92.8	97.0	**98.6**	46.1	51.9
Face Occlusions 1	99.4	99.4	**100.0**	**100.0**	**100.0**	**100.0**	95.5
Face Occlusions 2	**100.0**	84.0	51.5	98.8	99.4	97.9	93.6
Coke	33.9	74.6	69.5	49.2	83.1	13.9	**90.5**
Tiger 1	19.7	76.1	39.4	62.0	**98.6**	34.9	56.6
Tiger 2	30.1	57.5	24.7	84.9	86.3	70.1	**97.7**
average	61.0	83.2	65.9	78.1	**90.4**	64.8	85.3

TABLE V
BABENKO SEQUENCES: PERCENTAGE OF CORRECTLY TRACKED FRAMES WITH VARIOUS TRACKING ALGORITHMS.

	HT [11]	TLD [7]	PixelTrack [25]	MIL [6]	STRUCK [12]	PF	MCT
Cliff-dive 1	**100.0**	94.1	**100.0**	**100.0**	97.1	88.4	96.8
Motocross 1	**100.0**	1.3	40.4	0.0	33.3	36.1	85.7
Skiing	95.9	11.0	**100.0**	9.6	4.1	84.2	79.2
Mountain-bike	**100.0**	13.6	38.6	0.5	36.8	**100.0**	**100.0**
Cliff-dive 2	**90.2**	4.9	26.2	13.1	9.8	51.5	51.6
Volleyball	43.1	35.0	**86.2**	86.0	37.2	85.4	78.1
Motocross 2	**100.0**	86.7	80.0	80.0	93.3	92.7	98.7
Transformer	36.3	8.1	84.7	33.9	43.5	88.5	**89.7**
Diving	7.8	14.7	55.0	44.6	46.8	52.3	**60.8**
High Jump	68.0	6.6	**93.4**	78.7	47.5	42.0	49.9
Gymnastics	87.9	65.3	98.7	46.3	97.9	98.3	**99.1**
average	75.4	31.0	73.0	44.8	49.8	74.5	**80.9**

TABLE VI
NON-RIGID OBJECT SEQUENCES: PERCENTAGE OF CORRECTLY TRACKED FRAMES WITH VARIOUS TRACKING ALGORITHMS.

Track [11], Tracking-Learning-Detection (TLD) [7], Pixel-Track [25], Multiple-Instance Learning (MIL) Tracker [6], STRUCK [12], and a pure Particle Filter (PF) method (MCT without the discriminative detector). For the Babenko sequences, STRUCK showed the best average performance which can be explained by the videos mostly being in grey-scale, whereas MCT relies on colour information. However, for the more difficult non-rigid dataset, the average performance of MCT is superior to the one of the other methods. Note that MCT also outperforms STRUCK in the two VOT benchmarks (see below). Table V and VI show the results.

We further evaluated MCT with the VOT2013 dataset using the protocol of the VOT challenge and comparing it with 27 other state-of-the-art tracking methods. Table VII lists the top 7 ranks for the experiments baseline, region-noise, and greyscale, combining accuracy and robustness. The results of MCT are very competitive, being the second-best method for baseline and region-noise and the third-best for greyscale. Only one method, the Pixel-based LUT Tracker (PLT), is consistently outperforming MCT on this dataset. It is an optimisation of the tracker called STRUCK [12], currently unpublished but some explanation can be found in [48]. Note that, PLT is a single-scale tracker and it uses different feature sets for greyscale and colour videos. As opposed to PLT, MCT fails for example in the "Hand" video (2.13 failures on average), where large appearance changes, motion, and difficult lighting occur at the same time (see Fig. 3). Other failures may happen in the "Torus" and "Bolt" videos with large object deformations and many similar distracting objects. We also added the method PF (MCT without the detector) to the VOT2013 evaluation. Its overall ranks for the baseline, region-noise, and greyscale experiments are 16.1, 14.5, and 14.4 respectively. This clearly shows that the benefit of the motion context-based discriminative classifier.

baseline		region-noise		greyscale	
PLT	4.96	PLT	3.58	PLT	3.96
MCT	**6.62**	**MCT**	**5.08**	FoT [51]	4.75
FoT [51]	8.25	CCMS	8.33	**MCT**	**6.25**
EDFT [52]	9.5	FoT [51]	9.04	EDFT [52]	7.5
CCMS	9.54	LGT++ [53]	9.04	GSDT [54]	9.5
LGT++ [53]	10.2	EDFT [52]	9.08	LGT++ [53]	9.58
DFT [55]	11.1	LGT [14]	10.5	Matrioska [56]	10.7

TABLE VII
OVERALL RANKING RESULT WITH THE VOT2013 DATASET. ONLY THE FIRST 7 OUT OF 28 RANKS ARE SHOWN. THE NUMBERS REPRESENT THE ACTUAL AVERAGE RANKING.

Figure 2 shows the accuracy-robustness ranking plots for the VOT2014 dataset as evaluated in the context of the Visual Object Tracking Challenge 2014 [43]. The plots show the results on the "baseline" and "region noise" experiments for 39 different state-of-the-art methods. It can be seen that MCT (yellow circle) is among the top-performing methods, its overall rank being four (counting PLT and its extension PLT_14 as one entry). Table VIII lists the 10 best methods for VOT2014 and the respective accuracy ranks, robustness ranks, and overall ranks. Taking the average of accuracy and robustness ranks, PLT and its extension PLT_14 are still slightly better, as well as the correlation filter-based method SAMF [57], and the method DGT [58] which relies on graph matching and super-pixel representations. The method PF, *i.e.* MCT without the discriminative classifier, is only slightly worse on average with this benchmark. This might be due to the more challenging type of videos with deformable objects for which the texture-based classifier is not powerful enough.

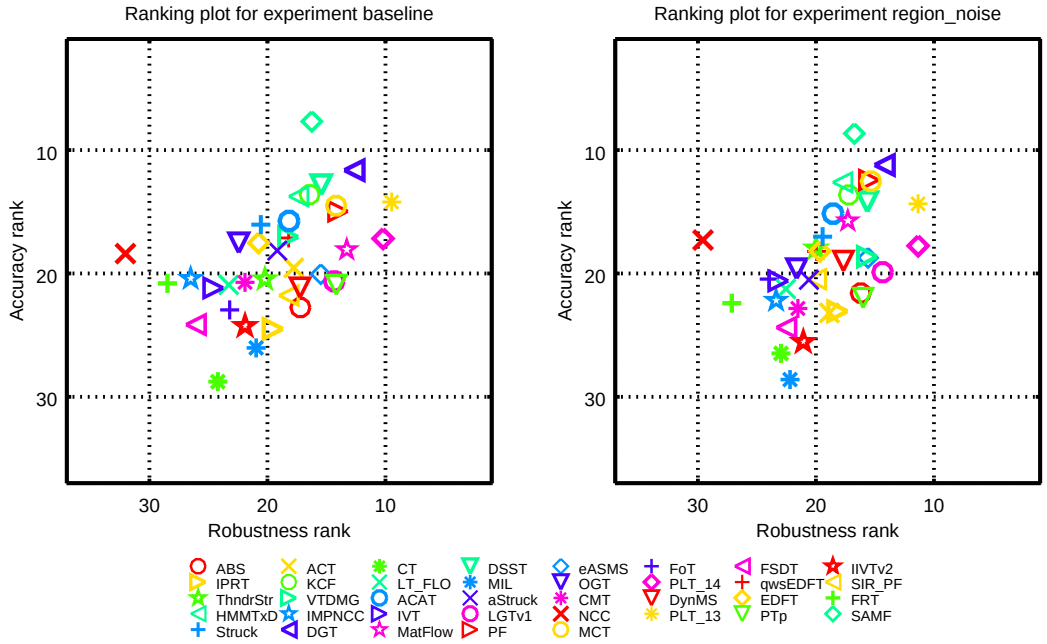

Fig. 2. Accuracy-robustness ranking plots for 39 state-of-the-art methods evaluated with the VOT2014 dataset [43]. The more on the top right, the better. The proposed method MCT (yellow circle) is among the top-performing methods.

	accuracy rank	robustness rank	overall rank
SAMF [57]	8.16	16.49	12.33
PLT	14.28	10.41	12.35
DGT [58]	11.42	13.44	12.43
PLT_14	17.46	10.77	14.12
MCT	**13.52**	**14.76**	**14.14**
PF	13.70	14.74	14.22
DSST [59]	13.51	15.54	14.53
KCF [60]	13.62	16.82	15.22
HMMTxD [43]	13.18	17.57	15.38
MatFlow [56]	16.90	15.29	16.10

TABLE VIII
OVERALL RANKING RESULT WITH THE VOT2014 DATASET. ONLY THE FIRST 10 OUT OF 39 RANKS ARE SHOWN. THE NUMBERS REPRESENT THE SEQUENCE-NORMALISED AVERAGE RANKING.

	accuracy			**robustness**		
	base-line	region-noise	grey-scale	base-line	region-noise	grey-scale
VOT2013	0.60	0.58	0.59	0.46	0.42	0.87
VOT2014	0.54	0.51	–	0.99	1.19	–

TABLE IX
AVERAGE ACCURACY AND ROBUSTNESS OF THE PROPOSED METHOD FOR THE VOT2013 AND VOT2014 DATASETS.

Table IX summarises the average accuracy and robustness values for VOT2013 and VOT2014.

Finally, Fig. 3 shows some qualitative tracking results on some of the videos. One can see that the algorithm is very robust to changes in object appearance, illumination, pose as well as complex motion, and partial occlusions. The algorithm runs at around 20fps for a frame size of 320×240 on an Intel Xeon 3.4GHz.

V. CONCLUSIONS

We presented a new efficient particle filter-based approach for tracking arbitrary objects in videos. The method combines generative and discriminative models, by effectively integrating an on-line learning classifier. We propose a new method to train this classifier that samples the position of negative examples from contextual motion cues instead of a fixed region around the tracked object. The advantage of MCT compared to others is that it effectively combines different discriminant visual cues: colour, shape, texture, and motion. And it further takes advantage the motion context in the scene, by using a specific online learning scheme that is independent from the actual classification algorithm. Our extensive experimental results show that this procedure improves the overall tracking performance with different discriminative classification algorithms. Further, the proposed tracking algorithm gives state-of-the-art results on four different challenging tracking datasets, effectively dealing with large object shape and appearance changes, as well as complex motion, varying illumination conditions and partial occlusions.

Possible future extensions to improve the tracking robustness and precision would include the use of more scene context, for example not only related to motion but also appearance and the inference of higher-level scene information related to lighting, shape, and 3D positions.

REFERENCES

[1] S. Grossberg, "Competitive learning: From interactive activation to adaptive resonance," *Cognitive Science*, vol. 11, no. 1, pp. 23–63, 1987.
[2] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, Feb. 2007.
[3] D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, 2008.
[4] A. Saffari and C. Leistner, "On-line random forests," in *Proceedings of the International Conference on Computer Vision (Worksh. on Online Comp. Vis.)*, 2009.

Fig. 3. Tracking results of MCT on the sequences “David”, “Motocross1”, “Bolt”, “Sunshade”, “Woman”, “Gymnastics”, and “Hand” (VOT2013). MCT is very robust to partial occlusions, illumination changes, deformations, pose or other appearance changes. In the “Woman” video, the algorithm has some problems adapting to the scale change; in the “Gymnastics” example, the aspect ratio is not adapted fast enough although the track is not lost; and in the last example the algorithm loses track due to deformation, rotation, motion blur, and low lighting.

[5] S. Stalder, H. Grabner, and L. V. Gool, "Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition," in *Proceedings of the International Conference on Computer Vision (Worksh. on Online Comp. Vis.)*, 2009.

[6] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Dec. 2009.

[7] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2010.

[8] J. Kwon and K. Lee, "Visual tracking decomposition," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1269–1276.

[9] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–72, Nov. 2011.

[10] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2188–202, Nov. 2011.

[11] M. Godec and P. M. Roth, "Hough-based tracking of non-rigid objects," in *Proceedings of the International Conference on Computer Vision*, 2011.

[12] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proceedings of the International Conference on Computer Vision*, 2011.

[13] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp. 1940–1947.

[14] L. Čehovin, M. Kristan, and A. Leonardis, "Robust visual tracking using an adaptive coupled-layer visual model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 941–953, Apr. 2013.

[15] F. Pernici and A. Del Bimbo, "Object tracking by oversampling local features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Dec. 2013.

[16] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via online boosting," in *Proceedings of the British Machine Vision Conference*, 2006.

[17] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. IEEE, 1994, pp. 593–600.

[18] T. Vojíř and J. Matas, "The enhanced flock of trackers," in *Registration and Recognition in Images and Videos*, ser. Studies in Computational Intelligence, R. Cipolla, S. Battiato, and G. M. Farinella, Eds. Springer Berlin Heidelberg, 2014, vol. 532, pp. 113–136.

[19] D. Freedman and T. Zhang, "Active contours for tracking distributions." *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 518–526, Apr. 2004.

[20] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *ECCV worksh. on statist. learning in comp. vis.*, 2004.

[21] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Tracking deforming objects using particle filtering for geometric active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1470–1475, Aug. 2007.

[22] C. Bibby and I. Reid, "Robust real-time visual tracking using pixel-wise posteriors," in *Proceedings of the European Conference on Computer Vision*, 2008.

[23] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proceedings of the European Conference on Computer Vision*, vol. 2, 1996, pp. 343–356.

[24] C. Aeschliman, J. Park, and A. C. Kak, "A probabilistic framework for joint segmentation and tracking," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Jun. 2010, pp. 1371–1378.

[25] S. Duffner and C. Garcia, "Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects," in *Proceedings of the International Conference on Computer Vision*, Dec. 2013, pp. 2480–2487.

[26] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proceedings of the European Conference on Computer Vision*, 2002.

[27] J.-M. Odobez, D. Gatica-Perez, and S. O. Ba, "Embedding motion in model-based stochastic tracking," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3514–3530, 2006.

[28] E. Maggio, "Adaptive multifeature tracking in a particle filtering framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 10, pp. 1348–1359, 2007.

[29] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *Proceedings of the International Conference on Computer Vision*, Oct. 2009.

[30] A. Del Bimbo and F. Dini, "Particle filter-based visual tracking with a first order dynamic model and uncertainty adaptation," *Computer Vision and Image Understanding*, vol. 115, no. 6, pp. 771–786, Jun. 2011.

[31] V. Belagiannis, F. Schubert, N. Navab, and S. Ilic, "Segmentation based particle filtering for real-time 2d object tracking," in *Proceedings of the European Conference on Computer Vision*, 2012, pp. 1–14.

[32] N. Gengembre and P. Pérez, "Probabilistic color-based multi-object tracking with application to team sports," INRIA, Tech. Rep. 6555, 2008.

[33] G. Zhang, J. Jia, W. Xiong, T. T. Wong, P. A. Heng, and H. Bao, "Moving object extraction with a hand-held camera," in *Proceedings of the International Conference on Computer Vision*, 2007, pp. 1–8.

[34] M. Yang, Y. Wu, and G. Hua, "Context-aware visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1195–1209, Jul. 2009.

[35] H. Grabner, J. Matas, L. Van Gool, and P. Cattin, "Tracking the invisible: Learning where the object might be," in *Proceedings of the Computer Vision and Pattern Recognition*, vol. 3, Jun. 2010, pp. 1285–1292.

[36] L. Wen, Z. Cai, Z. Lei, D. Yi, and S. Li, "Robust online learned spatio-temporal context model for visual tracking," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 785–796, 2013.

[37] Z. Sun, H. Yao, S. Zhang, and X. Sun, "Robust visual tracking via context objects computing," in *Proceedings of the International Conference Image Processing*, Sep. 2011, pp. 509–512.

[38] T. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proceedings of the Computer Vision and Pattern Recognition*, 2011.

[39] Z. Hong, X. Mei, and D. Tao, "Dual-force metric learning for robust distracter-resistant tracker," in *Proceedings of the European Conference on Computer Vision*, 2012, pp. 513–527.

[40] J. S. Supančič and D. Ramanan, "Self-paced learning for long-term tracking," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2013.

[41] Y. Hua, K. Alahari, and C. Schmid, "Occlusion and motion reasoning for long-term tracking," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 172–187.

[42] S. Duffner and C. Garcia, "Exploiting contextual motion cues for visual object tracking," in *Workshop on Visual Object Tracking Challenge (VOT2014) - ECCV*, Sep. 2014, pp. 1–12.

[43] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Cehovin, G. Nebehay, T. Vojir, and G. F. et al., "The Visual Object Tracking VOT2014 challenge results," in *Proceedings of the European Conference on Computer Vision (Workshops)*, 2014.

[44] M. Isard and A. Blake, "CONDENSATION – conditional density propagation for visual tracking," *Proceedings of the International Conference on Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[45] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[46] J.-M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of Visual Communication and Image Representation*, vol. 6, no. 4, pp. 348–365, Dec. 1995.

[47] K. Okuma, A. Taleghani, and N. D. Freitas, "A boosted particle filter: Multitarget detection and tracking," in *Proceedings of the European Conference on Computer Vision*, 2004, pp. 28–39.

[48] M. Kristan, L. Cehovin, R. Pflugfelder, G. Nebehay, G. Fernandez, J. Matas, and F. e. a. Porikli, "The Visual Object Tracking VOT2013 challenge results," in *Proceedings of the International Conference on Computer Vision (Workshops)*, 2013.

[49] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Jun. 2013, pp. 2411–2418.

[50] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.

[51] T. Vojíř and J. Matas, "Robustifying the flock of trackers," in *Computer Vision Winter Workshop*, 2011, pp. 91–97.

[52] M. Felsberg, "Enhanced distribution field tracking using channel representations," in *Visual Object Tracking Challenge (VOT2013), ICCV*, 2013.

[53] J. Xiao, R. Stolkin, and A. Leonardis, "An enhanced adaptive coupled-layer lgtracker++," in *Visual Object Tracking Challenge (VOT2013), ICCV*, 2013.
[54] J. Gao, J. Xing, W. Hu, and Z. X., "Graph embedding based semi-supervised discriminative tracker," in *Visual Object Tracking Challenge (VOT2013), ICCV*, 2013.
[55] L. Sevilla-Lara and E. G. Learned-Miller, "Distribution fields for tracking," in *Proceedings of the Computer Vision and Pattern Recognition*, 2012, pp. 1910–1917.
[56] M. E. Maresca and A. Petrosino, "Matrioska: A multi-level approach to fast tracking by learning," in *Proceedings of the International Conference on Image Analysis and Processing*, 2013, pp. 419–428.
[57] L. Yang and Z. Jianke, "A scale adaptive kernel correlation filter tracker with feature integration," in *Workshop on Visual Object Tracking Challenge (VOT2014) - ECCV*, 2014.
[58] Z. Cai, L. Wen, J. Yang, Z. Lei, and S. Z. Li, "Structured visual tracking with dynamic graph," in *Proceedings of the Asian Conference on Computer Vision*, 2012, pp. 86–97.
[59] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the British Machine Vision Conference*, 2014.
[60] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 3, pp. 125–141, 2014.

PLACE PHOTO HERE

Stefan Duffner received his PhD degree from University of Freiburg, Germany, in 2008, after doing his dissertation research at Orange Labs in Rennes, France, on face analysis with statistical machine learning methods. He then worked 4 years at Idiap Research Institute in Martigny, Switzerland, in the field of computer vision and multi-object tracking. As of today, Stefan Duffner is an associate professor in the IMAGINE team of the LIRIS research laboratory at the National Institute of Applied Sciences (INSA) of Lyon, France. His main research interests lie in machine learning for computer vision, and more specifically, on-line visual object tracking, and face image analysis.

PLACE PHOTO HERE

Christophe Garcia is Full Professor at INSA de Lyon, and head of the IMAGINE research team of the LIRIS laboratory. His current technical and research activities are in the areas of deep learning, pattern recognition and computer vision. He holds 17 industrial patents and has published more than 140 articles in international conferences and journals. He has served in more than 30 program committees of international conferences and is an active reviewers in 15 international journals where he co-organized several special issues. He has served as an associate editor of the International Journal of Visual Communication and Image Representation (Elsevier), Image and Video Processing (Hindawi) and Pattern Analysis and Application (Springer-Verlag).