Multi-timescale Collaborative Tracking

Dapeng Chen, Zejian Yuan, Gang Hua, Jingdong Wang, Nanning Zheng

Abstract—We present the multi-timescale collaborative tracker for single object tracking. The tracker simultaneously utilizes different types of "forces", namely *attraction*, *repulsion* and *support*, to take advantage of their complementary strengths. We model the three forces via three components that are learned from the sample sets with different timescales. The long-term descriptive component attracts the target sample, while the medium-term discriminative component repulses the target from the background. They are collaborated in the appearance model that can benefit each other. The short-term regressive component combines the votes of the auxiliary samples to predict the target's position, forming the context-aware motion model. The appearance model and the motion model collaboratively determine the target state, and the optimal state is estimated by a novel coarse-to-fine search strategy. We have conducted an extensive set of experiments on the standard 50 video benchmark. The results confirm the effectiveness of each component and their collaboration, outperforming current state-of-the-art methods.

Index Terms—Visual tracking, multi-timescale, descriptive, discriminative, regressive, context, collaboration.

1 Introduction

Visual tracking is a fundamental problem in computer vision. It serves as a core module of video analytic systems for action/activity/social behavior analysis [1], [2], [3], and has a wide range of applications, e.g., surveillance, human-computer interaction and augmented reality.

Recently, more and more researchers try to address tracking in real-word environments rather than in a certain particular scenario. For this purpose, online tracking remains very challenging because of the considerable variability of the target and the unpredictable influences from the scenes. The variability of the target requires a tracker to be online updated in order to capture the changing appearance, while the influences from the scenes (e.g., illumination change, severe occlusion) often contaminate the target's appearance, leading the tracker to drift away.

We argue that at least three types of "forces" should be utilized to deal with these challenges: the attraction from the historical targets, which is because of similarities among target samples; the repulsion from the background, which tries to discriminate between the target and background samples; the support from the context, which utilizes the spatial dependency on surrounding samples. The three forces focus on different relations within observed samples, hence it is essential to make a collaborative tracker that can take advantage of their complementary strengths.

The key idea of our collaboration strategy is that different forces should have their own *timescales*. The attraction spans over a long time because even the target samples tracked long before can constrain the current target appearance. The repulsion is related to more recent frames so that it can suppress the potentially distractive background samples. The support is transient, because the spatial dependency between the target and surrounding samples can change quickly. Considering the multi-timescale property not only preserves the effectiveness of different forces to improve the tracking reliability, but also excludes most unrelated samples from online learning to reduce the computational burden.

We propose to use the descriptive, the discriminative and the regressive components to model the three forces. In particular, the long-term descriptive component utilizes Support Vector Data Description (SVDD) [4]. It describes the set of tracked target samples by estimating the support of the target distribution. The medium-term discriminative component makes use of Structured SVM (S-SVM) [5]. It tries to discriminate the target samples from other samples in the same frame. The short-term regressive component is related to Support Vector Regression (SVR) [6]. It exploits the spatial correlations between the target and contextual samples, and in turn utilizes the correlations to help localize the target. Samples for all the three components are with a same kind of feature representation.

The components are cast into the appearance model and the motion model. For appearance model, the long-term descriptive component and the medium-term discriminative component are collaborated in a unified max-margin framework, where the descriptive component utilizes the discriminative information for adaption, while the discriminative component retrospects previous target samples to keep robustness. For motion model, the short-term regressive component is used to vote the target's position from a set of contextual samples. The appearance and motion models determine the probability of the target state together, and the optimal state is estimated via a novel coarse-to-fine search strategy. We term the proposed method the Multi-timescale Collaborative Tracker (MCT). The overview of MCT is illustrated in Fig. 1.

On a standard 50 video benchmark [7], MCT makes substantial performance improvement over previous methods. It performs continuous tracking even when the target is with severe variation or in critical conditions. By the attribute

• *D. Chen, Z. Yuan and N. Zheng are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049 China. Z. Yuan is the corresponding author. E-mail: dapengchenxj-tu@foxmail.com, {yuan.ze.jian,nnzheng}@mail.xjtu.edu.cn. G. Hua is with Stevens Institute of Technology, Hoboken, NJ, USA 07030. E-mail: ghua@stevens.edu. J. Wang is with the Microsoft Research, Beijing, P.R. China. E-mail: jingdw@microsoft.com.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2539956, IEEE Transactions on Pattern Analysis and Machine Intelligence

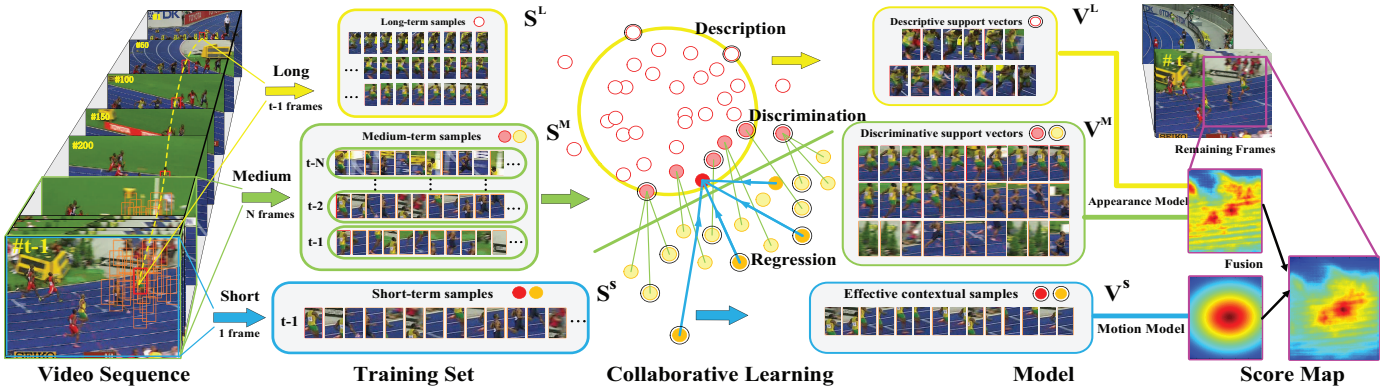


Fig. 1: The overview of our Multi-timescale Collaborative Tracking. We crop the long-term target samples, medium-term target-background samples and short-term contextual samples into sets $\mathcal{S}^L$, $\mathcal{S}^M$ and $\mathcal{S}^S$. The target samples are in **red**, while the samples from background are in **orange**. After collaborative learning, the samples with non-zeros dual variables are sent to the sets $\mathcal{V}^L$, $\mathcal{V}^M$ and $\mathcal{V}^S$. Samples in $\mathcal{V}^L$ and $\mathcal{V}^M$ contribute to the appearance model while samples in $\mathcal{V}^S$ contribute to the motion model. They determine the probability of target state together.

analysis, we also display the complementarity of the three components as well as the effectiveness of our collaboration strategy.

In summary, our contributions are: (1) Three components with multiple timescales are employed to exploit different tracking cues. (2) A collaboration strategy is proposed to mediate the components. It seeks model consistency between recent observations and past estimations, and localizes the target according to its appearance and surrounding context. (3) We design efficient implementations on both optimization steps and set management operations, in order to improve the processing speed.

Parts of this work have been appeared in [8], [9]. Compared with prior papers, this paper describes a complete tracking framework, which includes a newly added context-aware motion model formed by the short-term regressive component. It emphasizes on "multi-timescale", and shows its importance for model collaboration. We also include a substantial number of explanations, discussions and experiments to investigate various aspects of our tracker.

2 Related Work

Much progress has been witnessed in tracking. A comprehensive survey can be found in [10]. In this section, we briefly review the most relevant methods by categorizing them into different "forces".

Methods that focus on the "attraction" usually model the target distribution or the reconstruction error. To model the target distribution, the target samples are put in a specific feature space based on pixels [11], image patches [12], feature histograms [8], [13], [14], etc. Their distributions are characterized by various statistical methods such as mixture of Gaussian [15], non-parameteric Bayesian [16] and tree-structured graphical model [17]. Recently, methods based on subspace or sparse representation become quite popular, they estimate the likelihood of target by measuring the reconstruction error. In greater detail, methods based on subspace employ a set of learned subspaces to represent the target such as PCA [18], SPCA [19], IPPCA [20], while the trackers based on sparse representation represent the target by a sparse combination of target templates or the words in a learned dictionary [21], [12], [22].

Distinguished from generative models, the proposed long-term descriptive component is based on the idea of SVDD [4], which describes the target appearance by finding the support of the target distribution rather than by directly modeling the full density. With proper kernel techniques, the long-term descriptive component can well characterize the varying target distribution via representative samples.

Methods modeling the "repulsion" try to separate the target samples from background samples. They train classifiers with the most prevalent algorithms, e.g., boosting [23], random forest [24] and SVM [25]. Among them, Babenko et al. [23] propose an online MIL boosting algorithm for tracking. They learn the classifier from positive and negative "bags" rather than the selected positive and negative samples, allowing some flexility in finding a decision boundary. Saffari et al. [24] build a tracker based on online random forest. The tracker is more stable and more resistant to occlusion due to higher noise tolerance and the possibility to forget information by controlled discarding of trees.

Our medium-term discriminative component is inspired by a state-of-the-art discriminative tracker, named "Struck" [25]. Struck utilizes structured SVM to output the structured label of target, which turns out to be more suitable than binary classifiers for tracking. Compared with Struck, MCT regards the different temporal effectiveness of the target and background samples, therefore its medium-term discriminative component only needs to consider more recent frames, which is effective in dynamic scenes.

Trackers exploiting the "support" often mine the regions in the context with stable motion correlations to the target. These contextual regions are represented in various forms. Yang et al. [26] segment the regions over the image to find the potential auxiliary objects. Grabner et al. [27] and Dinh et al. [28] extract the key points from the image to help localize the target. All these methods maintain a database that stores these auxiliary contextual patterns.

However, selecting and updating these patterns are computational expensive, furthermore, the sparsity nature of these patterns prohibits those trackers to fully make use of the context. MCT gets rid of this limitation by assuming that the spatial correlations between the target and context are stable in a very short-term. It encodes the correlations by the short-term regressive component, and in turn uses the component to predict the target position.

Collaborative trackers have already achieved great effectiveness. Zhong et al. [29] collaborate generative

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2539956, IEEE Transactions on Pattern Analysis and Machine Intelligence

and discriminative models in parallel, and predict the targets by fusing their results together. Kalal et al. [30] integrate different models in a cascade, successively select the target sample from the candidates. MCT proposes a different collaboration strategy. It builds mutual beneficial connections between the long-term descriptive component and the medium-term discriminative component, forming an appearance model. The appearance model is loosely coupled with a motion model. They together determine the target position from different tracking cues.

The term "multi-timescale" has been mentioned in tracking. Xing et al. [22] collect training samples from multiple lifespans, then use the samples to learn a common dictionary to sparsely represent the target. Different from their approach that only considers the "multi-timescale" training samples, MCT presents "multi-timescale" in both training samples and tracking models.

3 MCT TRACKER

3.1 Representation

Given a frame $\mathbf{x}_i$, the state of a sample is represented by a bounding box $\mathbf{y}$. In this work, $\mathbf{y}$ is a 3 dimensional vector, i.e., $\mathbf{y} = [\mathbf{y}^u, \mathbf{y}^v, \mathbf{y}^s]^\top$. The sub-vector $[\mathbf{y}^u, \mathbf{y}^v]^\top$ represents the position, and $\mathbf{y}^s$ represents the scale. In frame $\mathbf{x}_i$, the feature map of sample $\mathbf{y}$ is denoted by $\phi(\mathbf{x}_i, \mathbf{y})$, and the training samples cropped from the frame are included in a set $\mathcal{Y}_i$. We introduce the $\phi(\mathbf{x}_i, \mathbf{y}_i)$ and $\mathcal{Y}_i$ as follows.

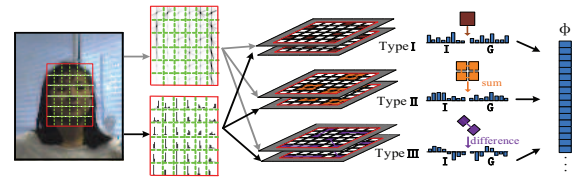
3.1.1 Feature Map $\phi(\mathbf{x}_i, \mathbf{y})$

In our framework, $\phi(\mathbf{x}_i, \mathbf{y})$ can be in various forms. We employ a kind of feature that is robust to local deformation and can capture the global structure of the target.

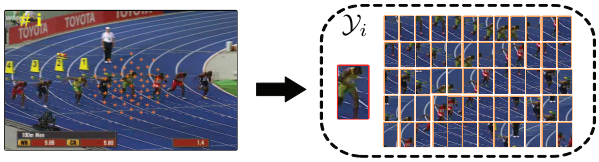
We begin by dividing the region within $\mathbf{y}$ into $m \times n$ disjoint rectangular patches, where m and n are determined by the shape of initial bounding box. Each patch is described by histogram features encoding the information of intensity (I) and oriented gradient (G). The intensities are gray scale values with gamma normalization, and the gradients are computed similarly to those in HOG [31]. Both information are split into 8 bins according to the intensity level and the orientation angle, resulting in 16 channels. We blur these channels with a $\sigma = 2$ pixel Gaussian and compute the descriptors for all the patches via the integral images [32].

Sample descriptors are constructed upon patch descriptors. There are three types of descriptors with different "receptive fields":

- *Type I* descriptor is just the patch descriptor, which describes the local appearance. The total number of *Type I* descriptors is $m \times n$. Concatenating the descriptors of all patches together can implicitly encode the spatial configuration of the patches, thus is robust to partial occlusion and partial deformation.
- *Type II* descriptor integrates the descriptors of the neighbouring 2×2 patch descriptors. The total number of *Type II* descriptors is $(m-1) \times (n-1)$. With larger "receptive fields", these descriptors can capture larger object structures and are suited to describe the object in low resolution.



(a) The generation of the feature representation.



(b) The generation of the training sample in frame i.

Fig. 2: Illustration of the representation: (a) shows the generation process of feature for one samples; and (b) shows the spatial configuration of training samples in one frame.

- *Type III* descriptor is related to any two patches. For efficiency, we randomly sample $\lfloor 0.1\binom{m\times n}{2} \rfloor$ patch pairs, and use the difference vector to encode the dependency within each pair. The obtained descriptors are sensitive to position and scale change, thus can localize the target with high accuracy.

All the descriptors belonging to the three types are first normalized within each type of information (I and G), and then are concatenated to form a vector $\phi(\mathbf{x}_i, \mathbf{y}) \in \mathbb{R}^d$, which is finally re-scaled to have a unit norm. The feature generation process is shown in Fig. 2a.

3.1.2 Set $\mathcal{Y}_i$

Set $\mathcal{Y}_i$ includes the training samples cropped from frame $\mathbf{x}_i$. To accelerate the learning process, we collect a small number of samples around the target, which are more helpful for tracker learning. Letting $\hat{\mathbf{y}}_i$ stand for the estimated target state, we obtain 61 samples based on a polar grid centered on $\hat{\mathbf{y}}_i$. An example is shown in Fig. 2b.

3.2 Multi-timescale Components

Three forces mentioned above are modeled by linear models. According to their timescales and attributes, we name them the long-term descriptive component, the medium-term discriminative component and the short-term regressive component, respectively.

3.2.1 Long-term Descriptive Component

As target appearance continuously changes in the feature space, neither an offline trained detector nor an appearance template formed in one frame is sufficient to capture the variations. To build a robust target model, we describe a long-term set $\mathcal{S}^L$, which contains the features of historical target samples, i.e., $\mathcal{S}^L = \{\phi(\mathbf{x}_i, \hat{\mathbf{y}}_i) | i = 1, ..., t-1\}$.

The long-term descriptive component describes the set $\mathcal{S}^L$ using SVDD, which employs a hypersphere to enclose the target set, at the same time, minimizes the sphere's volume to exclude outliers. Given the hypersphere's center $\mathbf{c}$, the loss term for the long-term component is:

$$\mathcal{L}^L(\mathbf{c}) = \min_R R^2 + C \sum_i [\|\phi(\mathbf{x}_i, \hat{\mathbf{y}}_i) - \mathbf{c}\|_2^2 - R^2]_+, \quad (1)$$

where R is the radius of the hypersphere, i is the frame index for the elements in $\mathcal{S}^L$, and $[\cdot]_+$ denotes the hinge loss. According to Sec. 3.1.1, all the features are re-scaled to have a constant norm, i.e., $\|\phi\|_2 = a$. Let $\mathbf{w}^L = 2\mathbf{c}$ and $\rho = \frac{1}{4}\|\mathbf{w}^L\|_2^2 + a^2 - R^2$, $\mathcal{L}^L(\mathbf{c})$ can be transformed to $\mathcal{L}^L(\mathbf{w}^L)$ by discarding the constant term, where

$$\mathcal{L}^L(\mathbf{w}^L) = \min_{\rho} \frac{1}{4}\|\mathbf{w}^L\|_2^2 - \rho + C\sum_i [\rho - \langle \mathbf{w}^L, \phi(\mathbf{x}_i, \hat{\mathbf{y}}_i)\rangle]_+. \quad (2)$$

The long-term component models the attraction within target samples. It describes the global modalities of the elements in $\mathcal{S}^L$, hence it is less affected by the influence from outliers such as occluded target samples and incorrect tracking results. Compared with generative models, the descriptive model evades the typical problems such as the limited expressive power of parametric models and the danger of overfitting for non-parametric ones, which is suitable for tracking with less prior knowledge.

3.2.2 Medium-term Discriminative Component

If our tracker merely relies on the long-term component, it tends to fail when the appearance of target changes rapidly. This is because the component attempts to describe the entire distribution of target and can not accurately characterize the current target appearance. Moreover, background samples may be similar to target, which easily leads the tracker to drift away. To improve tracking adaptivity and localization accuracy, we utilize background information to learn a discriminative component. In particular, we focus on the samples in most recent N frames ($N = 15$), and form a medium-term set, i.e., $\mathcal{S}^M = \{\phi(\mathbf{x}_j, \mathbf{y}) | \mathbf{y} \in \mathcal{Y}_j\}_{j=t-N,\ldots,t-1}$.

The medium-term discriminative component employs structured SVM [5] to discriminate the target samples from other samples. Given score function $\langle \mathbf{w}^M, \phi(\mathbf{x}_j, \mathbf{y})\rangle$, the score of estimated target $\hat{\mathbf{y}}_j$ should be higher than the score of any other sample in the same frame, i.e., $\forall \mathbf{y} \in \mathcal{Y}_j/\hat{\mathbf{y}}_j$, at least by a margin $\Delta(\hat{\mathbf{y}}_j, \mathbf{y})$. The loss term for the medium-term component is:

$$\mathcal{L}^M(\mathbf{w}^M) = \sum_j [\max_{\mathbf{y} \in \mathcal{Y}_j/\hat{\mathbf{y}}_j} \Delta(\hat{\mathbf{y}}_j, \mathbf{y}) - \langle \mathbf{w}^M, \delta\phi_j(\mathbf{x}_j, \mathbf{y})\rangle]_+, \quad (3)$$

where $\delta\phi_j(\mathbf{x}_j, \mathbf{y}) = \phi(\mathbf{x}_j, \hat{\mathbf{y}}_j) - \phi(\mathbf{x}_j, \mathbf{y})$ and $\Delta(\hat{\mathbf{y}}_j, \mathbf{y}) = 1 - \text{Area}(\hat{\mathbf{y}}_j \bigcap \mathbf{y})/\text{Area}(\hat{\mathbf{y}}_j \bigcup \mathbf{y})$. $\Delta(\hat{\mathbf{y}}_j, \mathbf{y})$ re-scales the margin between the target and background samples. It assigns smaller margins to the background samples that have large overlaps with the estimated target, thus will find better decision boundary than the classifiers that treat background samples equally.

3.2.3 Short-term Regressive Component

Context has proven helpful for visual tracking [27] [26] [28], since many temporary but very informative cues in surrounding environment imply the location of target. Such cues can be very strong, e.g., the samples containing certain part of the target always move along with the target; or be quite weak, e.g., a sample including complete background can only help to localize the target instantaneously. Due to the continuous interactions between the target and its surroundings, at least in a short-term, most contextual samples can precisely or coarsely infer the position of target. To keep effectiveness of the contextual information, we construct a short-term set $\mathcal{S}^S$, which includes the features of samples in the last frame $\mathbf{x}_{t-1}$, i.e., $\mathcal{S}^S = \{\phi(\mathbf{x}_{t-1}, \mathbf{y}) | \mathbf{y} \in \mathcal{Y}_{t-1}\}$.

The short-term regressive component includes two linear regressors along the horizontal and vertical directions, which map a feature $\phi(\mathbf{x}_{t-1}, \mathbf{y})$ to the displacement from $\mathbf{y}$ to $\hat{\mathbf{y}}_{t-1}$ as shown in Fig. 3. The mapping function is $\langle \mathbf{w}_o^S, \phi(\mathbf{x}_{t-1}, \mathbf{y})\rangle + b_o^S$, where $o \in \{u, v\}$ indicates the direction. The loss term for this component is to penalize the regression errors along the direction o:

$$\mathcal{L}^S(\mathbf{w}_o^S, b_o^S) = \sum_{\mathbf{y} \in \mathcal{Y}_{t-1}} |\langle \mathbf{w}_o^S, \phi(\mathbf{x}_{t-1}, \mathbf{y})\rangle + b_o^S - d_o(\mathbf{y})|, \quad (4)$$

where $d_o(\mathbf{y}) = \hat{\mathbf{y}}_{t-1}^o - \mathbf{y}^o$ is the displacement along o. The short-term component is a special case of ϵ-Support Vector Regression (ϵ-SVR) [6], but the parameter ϵ is zero. This setting will make the component over-fit the training samples heavier than the general ϵ-SVR. Such over-fitting is allowable, because we only need the regressor to be effective in a short term during which the context usually does not change very much.

3.3 Learning the Appearance and Motion Models

Considering the linear function $\langle \mathbf{w}, \phi(\mathbf{x}_t, \mathbf{y})\rangle$, both the long-term and the medium-term components tend to yield high score for the sample of target, meanwhile the short-term component predicts the target's position from contextual samples. As they rely on disjoint visual cues, we cast them into the appearance model and the motion model separately.

3.3.1 Learning the Collaborative Appearance Model

To model the target's appearance, we aim to learn a linear coefficient vector $\mathbf{w}^{app}$, which collaborates the effectiveness of the long-term and medium-term components. Specifically, we cast $\mathcal{L}^L$ and $\mathcal{L}^M$ into a unified learning framework by adding a regularization term, that is:

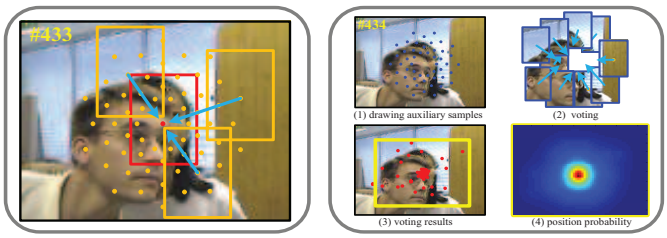
$$\begin{aligned} \min_{\mathbf{w}^{app}, \rho} \frac{1}{2}\|\mathbf{w}^{app}\|_2^2 - C_1\rho + C_2\sum_i [\rho - \langle \mathbf{w}^{app}, \phi(\mathbf{x}_i, \hat{\mathbf{y}}_i)\rangle]_+ \\ + C_3\sum_j [\max_{\mathbf{y} \in \mathcal{Y}_j/\hat{\mathbf{y}}_j} \Delta(\hat{\mathbf{y}}_j, \mathbf{y}) - \langle \mathbf{w}^{app}, \delta\phi_j(\mathbf{x}_j, \mathbf{y})\rangle]_+, \end{aligned} \quad (5)$$

where C_1, C_2 and C_3 are scalar parameters that balance $\mathcal{L}^L$, $\mathcal{L}^M$ and the regularization term. By adopting reparameterizing techniques [33], we transform Eq. 5 into its dual form:

$$\begin{aligned} \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \; & -\frac{1}{2}\boldsymbol{\alpha}^\top K^L \boldsymbol{\alpha} - \boldsymbol{\alpha}^\top K^{LM} \boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\beta}^\top K^M \boldsymbol{\beta} - \boldsymbol{\beta}^\top \Delta \\ \mathbf{s.t.} \quad & \sum_i \alpha_i = C_1, \quad 0 \le \alpha_i \le C_2, \quad \forall i, \\ & \sum_{\mathbf{y} \in \mathcal{Y}_j} \beta_j^{\mathbf{y}} = 0, \quad \beta_j^{\mathbf{y}} \le C_3\delta(\hat{\mathbf{y}}_j, \mathbf{y}), \quad \forall j, \mathbf{y}. \end{aligned} \quad (6)$$

In Eq. 6, $\delta(\hat{\mathbf{y}}_j, \mathbf{y})$ is Kronecker delta, α_i and $\beta_j^{\mathbf{y}}$ are dual variables associated with $\phi(\mathbf{x}_i, \hat{\mathbf{y}}_i)$ in $\mathcal{S}^L$ and $\phi(\mathbf{x}_j, \mathbf{y})$ in $\mathcal{S}^M$, respectively. $\boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{S}^L|}$, $\boldsymbol{\beta} \in \mathbb{R}^{|\mathcal{S}^M|}$ and $\Delta \in \mathbb{R}^{|\mathcal{S}^M|}$ concatenate α_i, $\beta_j^{\mathbf{y}}$ and $\Delta(\hat{\mathbf{y}}_j, \mathbf{y})$. $K^L \in \mathbb{R}^{|\mathcal{S}^L| \times |\mathcal{S}^L|}$, $K^M \in \mathbb{R}^{|\mathcal{S}^M| \times |\mathcal{S}^M|}$ and $K^{LM} \in \mathbb{R}^{|\mathcal{S}^L| \times |\mathcal{S}^M|}$ are similarity matrices for elements in $\mathcal{S}^L$, elements in $\mathcal{S}^M$ and elements between $\mathcal{S}^L$ and $\mathcal{S}^M$. The entries of K^L, K^M, K^{LM} are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2539956, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) Learning the motion model (b) Predicting the target's position

Fig. 3: The motion model: (a) we learn the shot-term component in the last frame; (b) we predict the target position via auxiliary samples.

Coarsely search Densely search Results

Fig. 4: Search Strategy. We utilize SMC Sampler to coarsely localize the target, then exhaustively search in a small region considering scales.

calculated by a joint inner product kernel $k(\mathbf{x},\mathbf{y},\bar{\mathbf{x}},\bar{\mathbf{y}}) = \langle\phi(\mathbf{x},\mathbf{y}),\phi(\bar{\mathbf{x}},\bar{\mathbf{y}})\rangle$. The coefficient vector $\mathbf{w}^{app}$ equals:

$$\mathbf{w}^{app} = \sum_i \alpha_i \phi(\mathbf{x}_i,\hat{\mathbf{y}}_i) + \sum_{j,\mathbf{y}\in\mathcal{Y}_j} \beta_j^{\mathbf{y}} \phi(\mathbf{x}_j,\mathbf{y}). \tag{7}$$

Interestingly, as $\sum_i \alpha_i\phi(\mathbf{x}_i,\hat{\mathbf{y}}_i)$ is related to the long-term component and $\sum_{j,\mathbf{y}\in\mathcal{Y}_j}\beta_j^{\mathbf{y}}\phi(\mathbf{x}_j,\mathbf{y})$ is related to the medium-term component, $\mathbf{w}^{app}$ can be regarded as the direct collaboration of $\mathbf{w}^L$ and $\mathbf{w}^M$. Most of dual variables will be 0 [34], only the elements with nonzero dual variables, known as *support vectors*, contribute to the appearance model.

3.3.2 Learning the Context-aware Motion Model

The motion model contains the short-term regressive components along the horizontal and vertical directions, and its coefficients is represented by: $\mathbf{w}^{mo} = \{\mathbf{w}_u^S, b_u^S, \mathbf{w}_v^S, b_v^S\}$. Given direction o, we learn $\mathbf{w}_o^S$ and b_o^S by minimizing $\mathcal{L}^S(\mathbf{w}_o^S, b_o^S)$ and a regularizer $\frac{1}{2}\|\mathbf{w}_o^S\|_2^2$:

$$\min_{\mathbf{w}_o^S, b_o^S} \frac{1}{2}\|\mathbf{w}_o^S\|_2^2 + C_4 \sum_{\mathbf{y}\in\mathcal{Y}_{t-1}} |\langle\mathbf{w}_o^S, \phi(\mathbf{x}_{t-1},\mathbf{y})\rangle + b_o^S - d_o(\mathbf{y})|. \tag{8}$$

Eq. 8 can be transformed into its dual form:

$$\begin{aligned} &\max_{\boldsymbol{\gamma}_o} -\frac{1}{2}\boldsymbol{\gamma}_o^\top K^S \boldsymbol{\gamma}_o + \mathbf{d}_o^\top \boldsymbol{\gamma}_o \\ &\mathbf{s.t.} \quad \sum_{\mathbf{y}\in\mathcal{Y}_{t-1}} \gamma_o^{\mathbf{y}} = 0, \quad -C_4 \le \gamma_o^{\mathbf{y}} \le C_4, \quad \forall \mathbf{y}, \end{aligned} \tag{9}$$

where $\gamma_o^{\mathbf{y}}$ is the dual variable for $\phi(\mathbf{x}_{t-1},\mathbf{y})$; vectors $\boldsymbol{\gamma}_o$ and $\mathbf{d}_o$ concatenate $\gamma_o^{\mathbf{y}}$ and $d_o(\mathbf{y})$; K^S is the similarity matrix for $\mathcal{S}^S$. Coefficients $\mathbf{w}_o^S$ and b_o^S are determined by:

$$\mathbf{w}_o^S = \sum_{\mathbf{y}\in\mathcal{Y}_{t-1}} \gamma_o^{\mathbf{y}} \phi(\mathbf{x}_{t-1},\mathbf{y}), \tag{10}$$

$$b_o^S = \underset{\mathbf{y}\in\mathcal{Y}_{t-1}}{\text{median}}\{d_o(\mathbf{y}) - \langle\mathbf{w}_o^S, \phi(\mathbf{x}_{t-1},\mathbf{y})\rangle\}. \tag{11}$$

3.4 Collaborative Tracker

3.4.1 Probabilistic Model

As the appearance model and the motion model rely on different tracking cues, we assume they are conditional independent, and collaborate them in product form to build the probability of target state:

$$p(\mathbf{y}_t|\mathbf{x}_t,\hat{\mathbf{y}}_{t-1}) \propto p_{app}(\mathbf{y}_t|\mathbf{x}_t;\mathbf{w}^{app}) \cdot p_{mo}(\mathbf{y}_t|\mathbf{x}_t,\hat{\mathbf{y}}_{t-1};\mathbf{w}^{mo}). \tag{12}$$

$p_{app}(\mathbf{y}_t|\mathbf{x}_t;\mathbf{w}^{app})$ is corresponding to the appearance model, which is built in a log-linear fashion, i.e.,

$$p_{app}(\mathbf{y}_t|\mathbf{x}_t;\mathbf{w}^{app}) \propto \exp\left(\frac{1}{Z}\langle\mathbf{w}^{app}, \phi(\mathbf{x}_t,\mathbf{y}_t)\rangle\right), \tag{13}$$

where $Z = \|\mathbf{w}^{app}\|_2 \cdot \|\phi(\mathbf{x}_t,\mathbf{y}_t)\|_2$. The probability corresponding to the motion model is given by:

$$\begin{aligned} &p_{mo}(\mathbf{y}_t|\mathbf{x}_t,\hat{\mathbf{y}}_{t-1};\mathbf{w}^{mo}) \\ &\quad\propto \sum_{\mathbf{z}_t\in\mathcal{Z}_t} p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{z}_t;\mathbf{w}^{mo}) p(\mathbf{z}_t|\mathbf{x}_t,\hat{\mathbf{y}}_{t-1}), \end{aligned} \tag{14}$$

where $\mathcal{Z}_t$ is a set of auxiliary samples. As shown in Fig. 3, we draw samples in the neighbourhood where the target lastly appeared. The samples in $\mathcal{Z}_t$ have equal weights, therefore $p_{mo}(\mathbf{y}_t|\mathbf{x}_t,\hat{\mathbf{y}}_{t-1};\mathbf{w}^{mo})$ is proportional to $\sum_{\mathbf{z}_t\in\mathcal{Z}_t} p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{z}_t;\mathbf{w}^{mo})$. In our case, $p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{z}_t;\mathbf{w}^{mo})$ is decomposed along the horizontal and vertical directions:

$$p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{z}_t;\mathbf{w}^{mo}) \propto \prod_{o\in\{u,v\}} p(\mathbf{y}_t^o|\mathbf{x}_t,\mathbf{z}_t;\mathbf{w}_o^S,b_o^S),$$

$$p(\mathbf{y}_t^o|\mathbf{x}_t,\mathbf{z}_t;\mathbf{w}_o^S,b_o^S) \propto \exp\left(-\frac{|\langle\phi(\mathbf{x}_t,\mathbf{z}_t),\mathbf{w}_o^S\rangle + b_o^S - d_o(\mathbf{z}_t)|}{\sigma_t}\right),$$

where $d_o(\mathbf{z}_t) = \mathbf{y}_t^o - \mathbf{z}_t^o$ and σ_t is set to be $\frac{1}{2}$ diagonal length of lastly estimated bounding box $\hat{\mathbf{y}}_{t-1}$.

3.4.2 Search Strategy

The optimal state in current frame $\mathbf{x}_t$ is estimated by maximizing the conditional probability of Eq. 12:

$$\hat{\mathbf{y}}_t = \arg\max_{\mathbf{y}_t} p(\mathbf{y}_t|\mathbf{x}_t,\hat{\mathbf{y}}_{t-1}). \tag{15}$$

Obtaining an exact solution for Eq.15 needs to search over all the possible states of $\mathbf{y}_t$. However, this task is prohibitively expensive as the search space is extremely large considering both position and scale. In fact, the position between two consecutive frames of a target may have an abrupt change while the scale usually varies slightly. We accordingly propose a two-stage search strategy.

Coarse Search. The first stage coarsely estimates the position of target. Inspired by simulated annealing, we construct a sequence of distributions $\{\pi_l(\mathbf{y}_t)\}_{l=1,\ldots,L}$, where $\pi_l(\mathbf{y}_t)$ is proportional to the lth power of $p(\mathbf{y}_t|\mathbf{x}_t,\hat{\mathbf{y}}_{t-1})$, i.e.,

$$\pi_l(\mathbf{y}_t) \propto p(\mathbf{y}_t|\mathbf{x}_t,\hat{\mathbf{y}}_{t-1})^l. \tag{16}$$

We adopt SMC sampler [35] to draw samples from $\pi_1(\mathbf{y}_t)$ to $\pi_L(\mathbf{y}_t)$, which can gradually obtain the samples with high probability. The details are presented in Alg. 1.

Let $\mathbf{y}_t^{l,i}$ be the ith particle sampled at the lth iteration, $\mathbf{y}_t^{1,i}$ is initialized from the importance distribution $\eta_1(\mathbf{y}_t^1) = \mathcal{N}(\mathbf{y}_t^1|\hat{\mathbf{y}}_{t-1},\boldsymbol{\Sigma}_t^1)$, where $\boldsymbol{\Sigma}_t^1 = diag(\sigma_{t,u}^2,\sigma_{t,v}^2,0)$ and $\sigma_{t,u},\sigma_{t,v}$ is $1/3$ diagonal length of $\hat{\mathbf{y}}_{t-1}$. We iteratively draw $\mathbf{y}_t^{l,i}$ from $\mathbf{y}_t^{l-1,i}$ by a transition kernel $K_l(\mathbf{y}_t^{l-1,i},\mathbf{y}_t^{l,i})$, which is chosen to be an MCMC kernel of $\pi_l(\mathbf{y}_t)$ with a Gaussian

Algorithm 1 SMC Sampler for Coarse Search

Input: $\hat{\mathbf{y}}_{t-1}, \mathbf{x}_t, \mathbf{w}^{app}, \mathbf{w}^S$

Initialization

1: **for** $i = 1$ to P **do**
2: draw $\mathbf{y}_t^{1,i} \sim \eta_1(\mathbf{y}_t^1)$;
3: evaluate $w_t^{1,i} \propto \pi_1(\mathbf{y}_t^{1,i})/\eta_1(\mathbf{y}_t^{1,i})$;
4: **end for**

Resampling and Sampling

5: **for** $l = 2$ to L **do**
6: resample $\{w_t^{l-1,i}, \mathbf{y}_t^{l-1,i}\}$ to obtain $\{\frac{1}{P}, \mathbf{y}_t^{l-1,i}\}$;
7: **for** $i = 1$ to P **do**
8: draw $\mathbf{y}_t^{l,i} \sim K_l(\mathbf{y}_t^{l-1,i}, \cdot)$;
9: evaluate $w_t^{l,i} \propto w_t^{l-1,i} \frac{\pi_l(\mathbf{y}_t^{l,i}) B_{l-1}(\mathbf{y}_t^{l,i}, \mathbf{y}_t^{l-1,i})}{\pi_{l-1}(\mathbf{y}_t^{l-1,i}) K_l(\mathbf{y}_t^{l-1,i}, \mathbf{y}_t^{l,i})}$;
10: **end for**
11: **end for**

Output: $\mathcal{Y}_t^C = \{\mathbf{y}_t^{l,i}\}_{l=1,\ldots,L, i=1,\ldots,P}$

proposal distribution $\mathcal{N}(\mathbf{y}_t^{l,i}|\mathbf{y}_t^{l-1,i}, 0.1\mathbf{\Sigma}_t^1)$. Particle weights are evaluated via line 9 of Alg. 1, where $B_{l-1}(\mathbf{y}_t^{l,i}, \mathbf{y}_t^{l-1,i})$ is the artificial backward kernel in the form of:

$$B_{l-1}(\mathbf{y}_t^{l,i}, \mathbf{y}_t^{l-1,i}) = \frac{\pi_l(\mathbf{y}_t^{l-1,i}) K_l(\mathbf{y}_t^{l-1,i}, \mathbf{y}_t^{l,i})}{\pi_l(\mathbf{y}_t^{l,i})}. \quad (17)$$

Eq. 17 is a practical approximation to optimal backward kernel when $K_l(\mathbf{y}_t^{l-1,i}, \mathbf{y}_t^{l,i})$ is an MCMC kernel [35]. Due to the re-sampling we adopted, $w_t^{l-1,i}$ becomes a equal weight for all i, and $w_t^{l,i}$ is finally evaluated by:

$$w_t^{l,i} \propto \frac{\pi_l(\mathbf{y}_t^{l-1,i})}{\pi_{l-1}(\mathbf{y}_t^{l-1,i})} = p(\mathbf{y}_t^{l-1,i}|\mathbf{x}_t, \hat{\mathbf{y}}_{t-1}). \quad (18)$$

All the generated samples are collected into set $\mathcal{Y}_t^C$. The optimal state at this stage is estimated by $\tilde{\mathbf{y}}_t = \arg\max_{\mathbf{y}_t \in \mathcal{Y}_t^C} p(\mathbf{y}_t|\mathbf{x}_t, \hat{\mathbf{y}}_{t-1})$.

Dense Search. The second stage densely searches for the optimal state considering both position and scale. The dense sample set is created based on the optimal state estimated at the coarse search stage: $\mathcal{Y}_t^D = \{\mathbf{y}_t | \mathbf{y}_t^s = k\tilde{\mathbf{y}}_t^s, k = 0.98, 0.99, 1, 1.01, 1.02, |\mathbf{y}_t^u - \tilde{\mathbf{y}}_t^u| \leq 4, |\mathbf{y}_t^v - \tilde{\mathbf{y}}_t^v| \leq 4\}$. Using the given scale change ratios is sufficient to capture the scale change between consecutive frames. The final optimal state is obtained by $\hat{\mathbf{y}}_t = \arg\max_{\mathbf{y}_t \in \mathcal{Y}_t^D} p(\mathbf{y}_t|\mathbf{x}_t, \hat{\mathbf{y}}_{t-1})$.

3.5 Discussion

The appearance model, the motion model and the search strategy compose a complete tracking system.

Our appearance model is inspired by the one of "Struck" [25]. Struck assigns individualized margins between target sample and background samples in one frame. The margin is small for a sample that is heavily overlapped with the estimated target, which makes Struck more mild than the trackers based on binary classifiers as it allows a sample to achieve a relative high score when it is geographically close to the estimated target sample. However, Struck only utilizes the discriminative power of the samples. In order to recall a historical target sample for learning, it has to utilize the background samples in the same frame. In dynamic scenes, the obsolete background samples can hardly help the current tracking, instead, it would actually contaminate the appearance model and increase the memory cost.

We collaborate the long-term descriptive component and the medium-term discriminative component to get rid of the mentioned limitations. The descriptive component can relate the tracker to the target samples tracked long ago, which is quite stable; while the discriminative component adopts the most relevant background samples, which is rather flexible. The two components build natural connections, and the support vectors in them determine the appearance model together.

Our motion model directly estimates the target's position in a data-dependent fashion, which is different from previous motion models defined by standard distributions (e.g., Gaussian, Epanechikov) [36], [12], [25]. It learns a regressor via the contextual samples in the previous frame, and predicts the target position according to the votes of auxiliary samples cropped from the current frame. The model is robust to camera motion. When the camera shakes, all the samples will move roughly in a same direction, while their relative positions hardly change. Moreover, our model is also resistent to samples with distractive appearance, because the score of target is not only determined by its appearance but also dependent on the surrounding context.

The coarse search strategy employing SMC sampler is distinguished from traditional tracking systems adopting the standard SMC framework [37]. The tracking systems based on the standard SMC (e.g., [18], [22], [21]) sequentially estimate the target state in a dynamical system, treating tracking as a filtering problem across different frames; while the coarse search strategy draws samples from a sequence of distributions in one frame, gradually obtaining the samples with high probability. The dense search stage is operated in a small region to estimate both position and scale. As the scale change between two frames is usually slight, it is reasonable and practical to first coarsely estimate the position, then densely search the position and scale.

4 Online Optimization

Eq. 6 and Eq. 9 are typical quadratic optimization problems w.r.t. $\boldsymbol{\alpha}, \boldsymbol{\beta}$ and $\boldsymbol{\gamma}_o$, which can be solved by a SMO-like method. The method iteratively reduces the dual problem to a subproblem w.r.t. only two dual variables and solves the subproblem by an elementary step. In this section, we introduce the elementary step and the online selection of the two dual variables. Besides, we also discuss set management strategies to ensure optimization efficiency.

4.1 Elementary Step

The elementary steps for $\boldsymbol{\alpha}, \boldsymbol{\beta}$ and $\boldsymbol{\gamma}_o$ are quite similar. Constraints in Eq. 6 and Eq. 9 require $\sum_i \alpha_i = C_1$, $\sum_{\mathbf{y} \in \mathcal{Y}_j} \beta_j^{\mathbf{y}} = 0$ and $\sum_{\mathbf{y} \in \mathcal{Y}_{t-1}} \gamma_o^{\mathbf{y}} = 0$, thus these steps can only modify the dual variable pairs by opposite amounts:

$$\begin{cases} \alpha_+ \leftarrow \alpha_+ + \lambda^\alpha \\ \alpha_- \leftarrow \alpha_- - \lambda^\alpha \end{cases}, \quad \begin{cases} \beta_j^{\mathbf{y}+} \leftarrow \beta_j^{\mathbf{y}+} + \lambda^\beta \\ \beta_j^{\mathbf{y}-} \leftarrow \beta_j^{\mathbf{y}-} - \lambda^\beta \end{cases}, \quad \begin{cases} \gamma_o^{\mathbf{y}+} \leftarrow \gamma_o^{\mathbf{y}+} + \lambda^{\gamma_o} \\ \gamma_o^{\mathbf{y}-} \leftarrow \gamma_o^{\mathbf{y}-} - \lambda^{\gamma_o} \end{cases},$$

where $(\alpha_+, \alpha_-), (\beta_j^{\mathbf{y}+}, \beta_j^{\mathbf{y}-}), (\gamma_o^{\mathbf{y}+}, \gamma_o^{\mathbf{y}-})$ are the selected dual variable pairs associated with $\{\phi(\mathbf{x}_+, \hat{\mathbf{y}}_+), \phi(\mathbf{x}_-, \hat{\mathbf{y}}_-)\}$ in $\mathcal{S}^L$, $\{\phi(\mathbf{x}_j, \mathbf{y}+), \phi(\mathbf{x}_j, \mathbf{y}-)\}$ in $\mathcal{S}^M$ and $\{\phi(\mathbf{x}_{t-1}, \mathbf{y}+), \phi(\mathbf{x}_{t-1}, \mathbf{y}-)\}$ in $\mathcal{S}^S$, respectively. $\lambda^\alpha, \lambda^\beta, \lambda^{\gamma_o}$ are non-negative adjustments, which lead one-step maximization of the objective function.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2539956, IEEE Transactions on Pattern Analysis and Machine Intelligence

	$\alpha_+ \quad (\alpha_+ < C_2)$	$\alpha_- \quad (\alpha_- > 0)$	frame index	$\beta_j^{\mathbf{y}+} \quad (\beta_j^{\mathbf{y}+} < C_3\delta(\mathbf{y}+, \hat{\mathbf{y}}_j))$	$\beta_j^{\mathbf{y}-}$
UPDATE	α_{t-1}	$\arg\min_{\alpha \in \mathcal{C}^{VL}} g(\alpha)$	$t-1$	$\beta_{t-1}^{\hat{\mathbf{y}}_{t-1}}$	$\arg\min_{\beta_{t-1}^{\mathbf{y}} \in \mathcal{C}^{SM}} g(\beta_{t-1}^{\mathbf{y}})$
RETRIEVE	$\arg\max_{\alpha \in \mathcal{C}^{SL}} g(\alpha)$	$\arg\min_{\alpha \in \mathcal{C}^{VL}} g(\alpha)$	a SV pattern k	$\arg\max_{\beta_k^{\mathbf{y}} \in \mathcal{C}^{VM}} g(\beta_k^{\mathbf{y}})$	$\arg\min_{\beta_k^{\mathbf{y}} \in \mathcal{C}^{SM}} g(\beta_k^{\mathbf{y}})$
ADJUST	$\arg\max_{\alpha \in \mathcal{C}^{VL}} g(\alpha)$	$\arg\min_{\alpha \in \mathcal{C}^{VL}} g(\alpha)$	a SV pattern k	$\arg\max_{\beta_k^{\mathbf{y}} \in \mathcal{C}^{VM}} g(\beta_k^{\mathbf{y}})$	$\arg\min_{\beta_k^{\mathbf{y}} \in \mathcal{C}^{VM}} g(\beta_k^{\mathbf{y}})$

TABLE 1: The three basic blocks for selecting the dual variable pairs from $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Specifically, for $\boldsymbol{\beta}$, we first determine frame containing support vectors, then select the dual variable pair from that frame. The constraints in the parentheses of the first line ensure effective elementary steps.

To obtain $\lambda^{\alpha}, \lambda^{\beta}$ and λ^{γ_o}, we first calculate the unconstrained adjustments $\tilde{\lambda}^{\alpha}, \tilde{\lambda}^{\beta}$ and $\tilde{\lambda}^{\gamma_o}$ by $\tilde{\lambda}^{\alpha} = (g(\alpha_+) - g(\alpha_-))/Z_{\alpha_+\alpha_-}$, $\tilde{\lambda}^{\beta} = (g(\beta_j^{\mathbf{y}+}) - g(\beta_j^{\mathbf{y}-}))/Z_{\beta_j^{\mathbf{y}+}\beta_j^{\mathbf{y}-}}$ and $\tilde{\lambda}^{\gamma_o} = (g(\gamma_o^{\mathbf{y}+}) - g(\gamma_o^{\mathbf{y}-}))/Z_{\gamma_o^{\mathbf{y}+}\gamma_o^{\mathbf{y}-}}$, where $g(\alpha_i)$, $g(\beta_j^{\mathbf{y}})$ are derivatives of Eq. 6 w.r.t. α_i, $\beta_j^{\mathbf{y}}$, while $g(\gamma_o^{\mathbf{y}})$ is a derivative of Eq. 9 w.r.t. $\gamma_o^{\mathbf{y}}$. Denominator $Z_{\alpha_+\alpha_-}$ equals $\|\phi(\mathbf{x}_+, \hat{\mathbf{y}}_+) - \phi(\mathbf{x}_-, \hat{\mathbf{y}}_-)\|_2^2$, and $Z_{\beta_j^{\mathbf{y}+}\beta_j^{\mathbf{y}-}}$,$Z_{\gamma_o^{\mathbf{y}+}\gamma_o^{\mathbf{y}-}}$ are similarly calculated. We enforce the constraints in Eq. 6 and Eq. 9, obtaining:

$$\begin{aligned} \lambda^{\alpha} &= \max(0, \min(\tilde{\lambda}^{\alpha}, \alpha_-, C_2 - \alpha_+)), \\ \lambda^{\beta} &= \max(0, \min(\tilde{\lambda}^{\beta}, C_3\delta(\mathbf{y}+, \hat{\mathbf{y}}_j) - \beta_j^{\mathbf{y}+})), \\ \lambda^{\gamma_o} &= \max(0, \min(\tilde{\lambda}^{\gamma_o}, C_4 - \gamma_o^{\mathbf{y}+}, C_4 + \gamma_o^{\mathbf{y}-})). \end{aligned} \tag{19}$$

The coefficient vector $\mathbf{w}^{app}, \mathbf{w}_o^S$ are updated by:

$$\begin{aligned} \mathbf{w}^{app} &= \mathbf{w}^{app} + \lambda^{\alpha}(\phi(\mathbf{x}_+, \hat{\mathbf{y}}_+) - \phi(\mathbf{x}_-, \hat{\mathbf{y}}_-)), \\ \mathbf{w}^{app} &= \mathbf{w}^{app} + \lambda^{\beta}(\phi(\mathbf{x}_j, \mathbf{y}+) - \phi(\mathbf{x}_j, \mathbf{y}-)), \\ \mathbf{w}_o^S &= \mathbf{w}_o^S + \lambda^{\gamma_o}(\phi(\mathbf{x}_{t-1}, \mathbf{y}+) - \phi(\mathbf{x}_{t-1}, \mathbf{y}-)). \end{aligned} \tag{20}$$

4.2 Online selection

Online selection hinges on how to choose proper dual variable pairs. We apply different selection strategies for the appearance model and the motion model, respectively.

4.2.1 Variable Selection for Appearance Model

The variable selection for the appearance model is related to Eq. 6. Intuitively, the dual variable pairs should be selected along the feasible search direction with highest gradient, however, keeping searching such variable pairs considering all the elements in $\mathcal{S}^L$ and $\mathcal{S}^M$ is computationally infeasible. Boredes and Botou [38] have observed that the support vectors are not updated very often during the optimization. They suggested to alternatively update the dual variable corresponding to a fresh sample and the dual variable corresponding to a randomly selected support vector.

We define $\mathcal{V}^L, \mathcal{V}^M$ as the sets that maintain the support vectors in $\mathcal{S}^L, \mathcal{S}^M$, and define $\mathcal{C}^{SL}, \mathcal{C}^{SM}, \mathcal{C}^{VL}, \mathcal{C}^{VM}$ as the sets that contain corresponding dual variables. Inspired by OLarank [39], we design three building blocks, namely *update*, *retrieve* and *adjust*. Each of them alternatively selects the variable pairs (α_+, α_-) and $(\beta_j^{\mathbf{y}+}, \beta_j^{\mathbf{y}-})$. In particular,

- **UPDATE** selects two dual variables regarding the samples from the newly coming frame to improve the model's adaptivity.
- **RETRIEVE** selects one dual variable associated with a support vector and selects the other dual variable associated with a non-support vector, retrieving previous data to ensure model's generalization ability.
- **ADJUST** selects two dual variables regarding support vectors, adjusting them to achieve better consistency.

The details are described in the Tab. 1. In UPDATE, variables α_{t-1} and $\beta_{t-1}^{\mathbf{y}}$ ($\forall \mathbf{y} \in \mathcal{Y}_{t-1}$) are initialized to be zero except α_1, which is initialized to be C_1 to satisfy $\sum_i \alpha_i = C_1$. In RETRIEVE and ADJUST, the optimization of $\boldsymbol{\beta}$ first needs to randomly pick a frame that includes support vectors, then selects a dual variable pair from that frame. All the three blocks are scheduled according to OLarank as shown in Alg. 2, which takes into account both the computing time and the progress of the objective function.

4.2.2 Variable Selection for Motion Model

The variable selection for the motion model is related to Eq.9 . As $\mathcal{S}^S$ contains the features of training samples in one frame, it is possible for us to select dual variable pairs regarding all the samples. Let $\mathcal{C}_o^{SS}$ be the set that contains corresponding dual variables of $\mathcal{S}^S$ along direction o, all the dual variables in $\mathcal{S}^S$ are initialized to be 0. The dual variable pair $(\gamma_o^{\mathbf{y}+}, \gamma_o^{\mathbf{y}-})$ is selected by:

$$\begin{aligned} \gamma_o^{\mathbf{y}+} &\leftarrow \arg\max_{\gamma_o^{\mathbf{y}} \in \mathcal{C}_o^{SS}} g(\gamma_o^{\mathbf{y}}), \quad \text{s.t.} \quad \gamma_o^{\mathbf{y}} < C_4, \\ \gamma_o^{\mathbf{y}-} &\leftarrow \arg\min_{\gamma_o^{\mathbf{y}} \in \mathcal{C}_o^{SS}} g(\gamma_o^{\mathbf{y}}), \quad \text{s.t.} \quad \gamma_o^{\mathbf{y}} > -C_4. \end{aligned} \tag{21}$$

We require $\gamma_o^{\mathbf{y}+} < C_4$ and $\gamma_o^{\mathbf{y}-} > -C_4$ in order to ensure effective elementary steps.

4.3 Set Management

We maintain the training data in sets $\mathcal{S}^L$, $\mathcal{S}^M$, $\mathcal{S}^S$, $\mathcal{V}^L$, $\mathcal{V}^M$ and $\mathcal{V}^S$. As tracking goes on, these sets need to be managed, in order to keep both effectiveness and efficiency. In practice, we fix the maximum set size as $N_{SL}, N_{SM}, N_{SS}, N_{VL}, N_{VM}$ and N_{VS}, then operate a set of management strategies:

1) For $\mathcal{S}^L$, each time we include the feature of the last estimated the optimal state. When $|\mathcal{S}^L| > N_{SL}$, we randomly cast one element away from set $\mathcal{S}^L$.
2) For $\mathcal{S}^M$, each time we include the features of the samples in $\mathcal{Y}_{t-1}$. When $|\mathcal{S}^N| > N_{SM}$, we drop the samples in the oldest frame away from $\mathcal{S}^M$.
3) For $\mathcal{S}^S$, we always keep the set $\mathcal{Y}_{t-1}$, and drop older samples.
4) $\mathcal{V}^L$ maintains the support vectors in $\mathcal{S}^L$. When $|\mathcal{V}^L| > N_{VL}$, we delete the sample with smallest dual variable, and transit the amount of its dual variable to the second smallest dual variable, keeping $\sum_i \alpha_i = C_1$.
5) $\mathcal{V}^M$ maintains the support vectors in $\mathcal{S}^M$. When $|\mathcal{V}^M| > N_{VM}$, we delete all the support vectors from the oldest frames.
6) $\mathcal{V}^S$ maintains the samples with nonzero dual variable in $\mathcal{S}^S$.

The entire optimization procedures are displayed in Alg. 2. To increase the robustness of the motion model, we further utilize an ad-hoc updating rule for $\mathbf{w}_o^S$ with a learning rate $\lambda = 0.8$. The rule implicitly exploits stronger contextual correlations during tracking.

Algorithm 2 Multi-timescale Collaborative Tracker

Input: $\mathbf{x}_t, \hat{\mathbf{y}}_{t-1}, \mathbf{w}^{app}, \mathbf{w}^S, \mathcal{S}^L, \mathcal{S}^M, \mathcal{S}^S, \mathcal{V}^L, \mathcal{V}^M, \mathcal{V}^S$

Update the Appearance Model

manage $\mathcal{S}^L$, $\mathcal{S}^M$;
$(\alpha_+, \alpha_-) \leftarrow$ UPDATE; manage $\mathcal{V}^L$; Ele-Step(α_+, α_-);
$(\beta_j^{\mathbf{y}+}, \beta_j^{\mathbf{y}-}) \leftarrow$ UPDATE; manage $\mathcal{V}^M$; Ele-Step$(\beta_j^{\mathbf{y}+}, \beta_j^{\mathbf{y}-})$;
for $l_1 = 1$ to n_1 **do**
  $(\alpha_+, \alpha_-) \leftarrow$ RETRIEVE; manage $\mathcal{V}^L$; Ele-Step(α_+, α_-);
  $(\beta_j^{\mathbf{y}+}, \beta_j^{\mathbf{y}-}) \leftarrow$ RETRIEVE; manage $\mathcal{V}^M$; Ele-Step$(\beta_j^{\mathbf{y}+}, \beta_j^{\mathbf{y}-})$;
  for $l_2 = 1$ to n_2 **do**
    $(\alpha_+, \alpha_-) \leftarrow$ ADJUST; manage $\mathcal{V}^L$; Ele-Step(α_+, α_-);
    $(\beta_j^{\mathbf{y}+}, \beta_j^{\mathbf{y}-}) \leftarrow$ ADJUST; manage $\mathcal{V}^M$; Ele-Step$(\beta_j^{\mathbf{y}+}, \beta_j^{\mathbf{y}-})$;
  end for
end for

Update the Motion Model

manage $\mathcal{S}^S$;
for each $o \in \{u, v\}$ **do**
  $(\mathbf{w}_o^S)' \leftarrow \mathbf{w}_o^S, \quad \mathbf{w}_o^S \leftarrow 0$;
  for $l_3 = 1$ to n_3 **do**
    $(\gamma_o^{\mathbf{y}+}, \gamma_o^{\mathbf{y}-}) \leftarrow$ Eq. 21; manage $\mathcal{V}^S$; Ele-Step $(\gamma_o^{\mathbf{y}+}, \gamma_o^{\mathbf{y}-})$;
  end for
  $\mathbf{w}_o^S \leftarrow (1-\lambda)(\mathbf{w}_o^S)' + \lambda \mathbf{w}_o^S, \quad b_o^S \leftarrow$ Eq. 11;
end for

Estimate the Target State

coarse search stage: $\tilde{\mathbf{y}}_t = \arg\max_{\mathbf{y}_t \in \mathcal{Y}_t^C} p(\mathbf{y}_t | \mathbf{x}_t, \hat{\mathbf{y}}_{t-1})$ by Alg. 1;
dense search stage : $\hat{\mathbf{y}}_t = \arg\max_{\mathbf{y}_t \in \mathcal{Y}_t^D} p(\mathbf{y}_t | \mathbf{x}_t, \hat{\mathbf{y}}_{t-1})$.

Output: $\hat{\mathbf{y}}_t, \mathbf{w}^{app}, \mathbf{w}^S, \mathcal{S}^L, \mathcal{S}^M, \mathcal{S}^S, \mathcal{V}^L, \mathcal{V}^M, \mathcal{V}^S$

5 Experiments

5.1 Benchmark and Tracker Setting

Benchmark. We have tested our method on the CVPR 2013 benchmark [7]. The benchmark consists of 50 fully annotated sequences and covers a variety of challenging scenarios, which is a comprehensive benchmark specifically designed for evaluation of tracking performance.

To evaluate the robustness of the trackers, three protocols are provided by the dataset [7]: one pass evaluation (OPE), temporal robustness evaluation (TRE) and spatial robustness evaluation (SPE). In particular, TRE selects different start frames from the sequence and runs the trackers through the rest of the sequence; SRE perturbs the initialization bounding box by scaling and shifting.

Given the presence and the position of the target object, two metrics are employed to demonstrate the overall performance [7]: success plot and precision plot. Both plots show the percentage of successfully tracked frames according to the threshold values: the success plot thresholds the overlap ratio [8], and the precision plot thresholds the center location error [29]. The score for the overall success plot is evaluated by the area under curve(AUC), and the score for the overall precision plot is given by the precision at 20 pixel threshold.

Tracker Setting. The parameters of the MCT tracker were *fixed for all the sequences*. We empirically fixed the maximum set size as $N_{SL} = 60$, $N_{SM} = 15 \times 61$, $N_{SS} = 61$, $N_{VL} = 25$, $N_{VM} = 40$ and $N_{VS} = 61$, where $\mathcal{S}^M$ and $\mathcal{S}^S$ include 15 frames and 1 frame, and each frame further contains 61 samples. We set $C_1 = 16$, $C_2 = 1$, $C_3 = 1$ and $C_4 = 1000$. The iteration numbers n_1, n_2, n_3 in Alg. 2 are determined as $n_1 = 10$, $n_2 = 5$ and $n_3 = 20$. Given the above parameter settings, the MATLAB/MEX implementation of MCT runs about 15 frames per second with a 3.07Ghz, 2 Cores CPU.

5.2 Empirically Analysis of Our Approach

5.2.1 Representation Analysis

In order to study the effectiveness of the feature map $\phi(\mathbf{x}_i, \mathbf{y})$, we built variant trackers I-T, II-T and III-T based on the three types of descriptors introduced in Sec. 3.1.1. We compare the three trackers and MCT according to the OPE protocol, and display the success plots and the precision plots in Fig. 5a and Fig. 5b. It can be seen that the performance of the MCT is significantly better than the trackers using a single type of descriptors.

In Fig. 5c, we compare the AUC scores of success plots over different challenging attributes among the four trackers. The results not only confirm the effectiveness of MCT but also reveal the characteristics of different descriptors. *Type I* descriptor performs more robust and accurate than the other two over most of the attributes. For example, it performs well on occlusion. If occlusion happens, some descriptors that are in charge of occluded region will lose effectiveness while others still remain effective. As *Type I* descriptor utilizes more local information than the other two, it has a better chance to stick on the target when occlusion happens. *Type II* and *Type III* descriptors are complementary. *Type II* descriptor is more suitable to handle large deformation and low resolution, because its large "receptive field" is invariant to motion variation and can integrate more information. *Type III* descriptor handles better the sequences that have motion blur and fast motion, which is because the difference of patch descriptors can exploit more details about the target and is sensitive to position change.

In addition, we further constructed three trackers (II+III)-T, (I+III)-T and (I+II)-T, which cast *Type I*, *Type II* and *Type III* descriptors away from MCT. Their AUC scores regarding the OPE protocol are 0.583 , 0.591 and 0.562, respectively. Using two descriptors generally performs better than using only one descriptor, but performs less better than taking all the three descriptors, validating the necessity of each type of descriptors.

5.2.2 Component Analysis

In order to investigate the properties of each component, we constructed three reduced versions of MCT, namely L*-T, M*-T and S*-T, which cast the long-term, the medium-term and the short-term components away from MCT, respectively. In greater detail, we set $C_2 = 0$ for L*-T to exclude the influence of the descriptive support vectors, and set $C_3 = 0$ for M*-T to exclude the influence of the discriminative support vectors. For S*-T, we only utilized the appearance model to estimate the target state. It is notable that S* corresponds to the DDCT tracker in our previous conference paper [9].

Three reduced trackers are compared with MCT according to OPE protocol, and the success plots, the precision plots and the AUC scores over different attributes are displayed in Figs. 6a, 6b and 6c. All of them show the superiority of MCT against its reduced versions. Here, we summarize the performance of each component one by one.

The long-term descriptive component can well handle the sequences with occlusion, deformation and out of view. It describes the global characteristics of the target set, hence

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2539956, IEEE Transactions on Pattern Analysis and Machine Intelligence

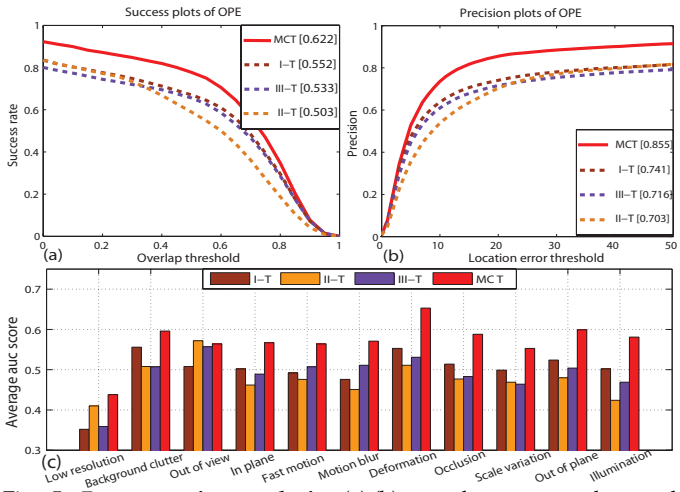


Fig. 5: **Representation analysis**. (a),(b) are the success plots and precision plots for different types of descriptors with the overall scores in the legend. (c) demonstrates the AUC scores over attributes.

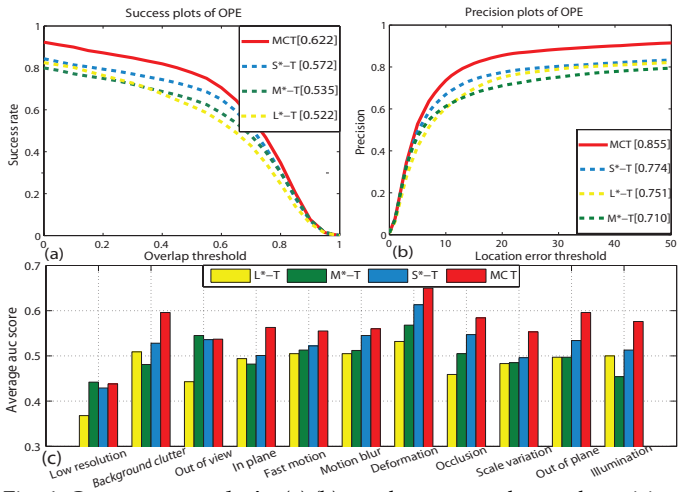


Fig. 6: **Component analysis.** (a),(b) are the success plots and precision plots for different component combination with the overall scores in the legend. (c) demonstrates the AUC scores over attributes.

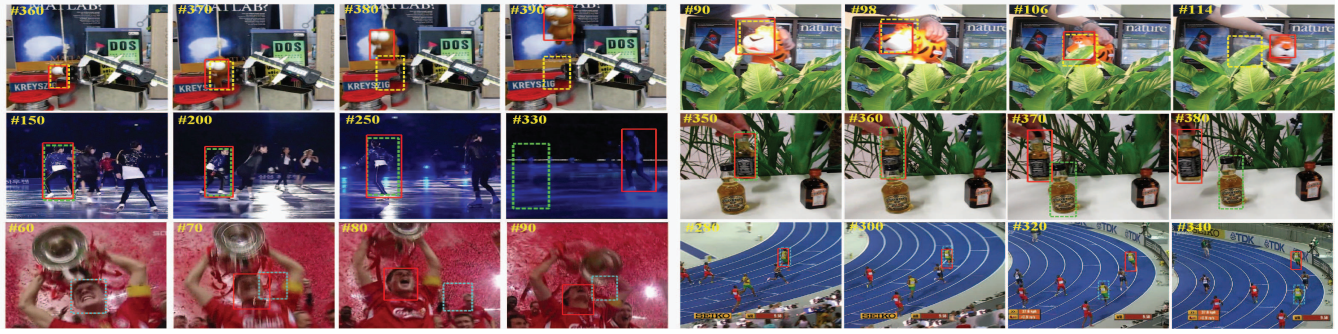


Fig. 7: **Qualitative analysis of components**. We compare MCT with L*-T in *lemming* and *tiger1* in the top row, compare MCT with M*-T in *skating1* and *liquor* in the middle row, compare MCT with S*-T in *soccer* and *bolt* in the bottom row.

is robust to deformation and is less vulnerable to outliers, including the situations in which the target is severely occluded or out of image. We compare MCT and L*-T in the upper row of Fig. 7. In *lemming*, the major part of the target is hidden behind the lighter, both MCT and L*-T are affected by the occluded samples. When the target comes out, MCT can re-track the target while L*-T stays on the lighter. Similar situation can also be observed in *tiger1*. When the target moves across the leaves, MCT is less affected by the interruption caused by occlusion, while L*-T updates the appearance model to the leaves.

The medium-term discriminative component can well handle the sequences having background clutters and illumination changes. This component can suppress the potential distractive background samples beforehand, thus largely reduces the chance of drifting away. The superiority of this component is demonstrated in the middle row of Fig. 7. In *skating1*, the target appearance changes significantly due to the illumination change. M*-T drifts away while MCT tracks well. In *liquor*, MCT recognizes the target liquor even when the target is partially occluded, while M*-T directly follows another similar liquor.

The short-term regressive component further improves the performance. It utilizes the co-existence relations between the target and contextual samples, in order to prevent the "sudden jump" of the bounding box from the target. Examples are shown in *soccer* and *bolt* in the bottom row of Fig. 7. In *soccer*, MCT can stick to the target despite the challenges of fast motion and severe blur, where the arm and T-shirt serve as stable contextual cues. In *bolt*, S*-T may be attracted by another player wearing similar clothes without using the contextual information.

5.2.3 Component Visualization

In Fig. 8, the components of MCT are visualized by the samples with non-zero dual variables in $\mathcal{S}^L$, $\mathcal{S}^M$ and $\mathcal{S}^S$.

MCT can well adapt to the data as shown in *walking2*. The long-term component changes with the distribution of tracked target samples: the descriptive support vectors are quite coherent till frame 170 as the target hardly varies, while they become diversified after frame 220 because of the gradually occluded appearance. The medium-term component updates according to the background: when the background is quite simple, no discriminative support vectors are needed; while when distractive object appears, the discriminative component starts to work. The short-term component continuously selects the effective samples from new frame. The selected samples are usually with significant edges, thus they are much more reliable to predict the position of the target in a short-term.

Two examples are displayed to show how MCT handles occlusion. In *coke*, the target is temporally and totally occluded by the leaves in frame 260. The long-term component excludes the erroneously estimated target sample (in the red box of the image) by regarding it as an outlier. In *woman*, the target person is frequently and partially occluded by the car. As the partially occluded target samples still share

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2539956, IEEE Transactions on Pattern Analysis and Machine Intelligence

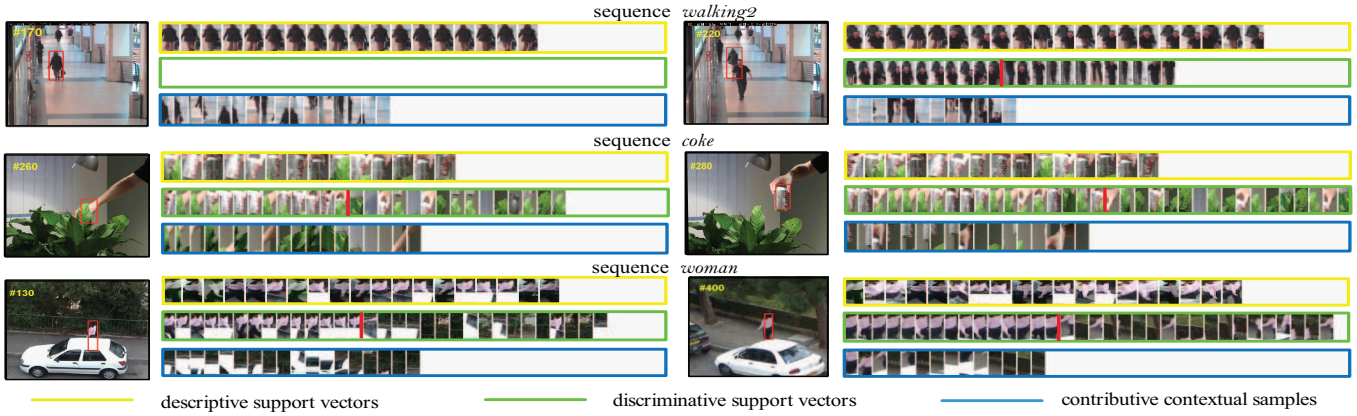


Fig. 8: **Component visualization**. We visualize the three components, where the descriptive, discriminative and regressive samples with non-zero dual variables are shown in the yellow, the green and blue boxes beside the corresponding frames. As discriminative support vectors include both positive samples and negative samples, we arrange the positive samples before the negative samples, and separate them by a red line.

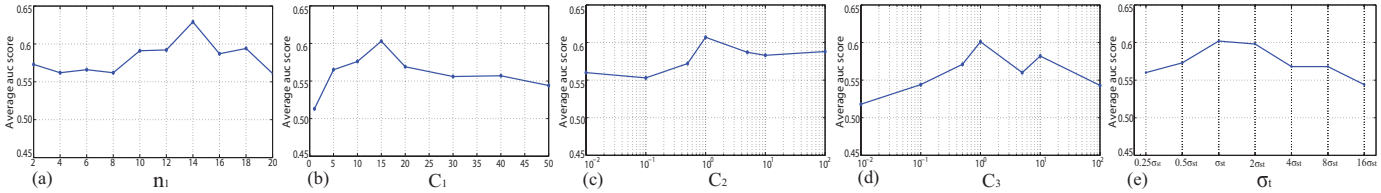


Fig. 9: **Parameter analysis**. The default parameters are set according to Sec. 5.1. We demonstrate the performance change of average AUC score w.r.t. (a) parameter n_1, (b) parameter C_1, (c) parameter C_2, (d) parameter C_3 and (e) parameter σ_t.

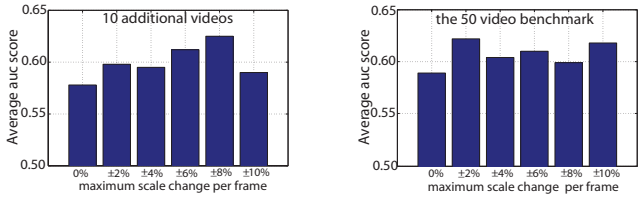


Fig. 10: Influence of changing maximum scale change per frame on 10 additional video sequences and the 50 video benchmark [7].

Fig. 11: The failure analysis of *skiing*. We display the gray image along with the color image, where the results are in red box while ground truth is in green. MCT failed without using the color information,

many commonalities with the previous tracked target, they are considered as prototypes by the long-term component, which makes MCT robust to similar occlusion patterns.

5.2.4 Parameter Analysis

To study the influences of parameters n_1, C_1, C_2, C_3 and σ_t, we first set them according to Sec. 5.1, then demonstrate how performance changes w.r.t. each parameter in Fig. 9. Besides, we also discuss the influence of scale change. All experiments here are tested over 10 additional video sequences (*biker*, *clifbar*, *coupon*, *dancer*, *dancer2*, *dragonbaby*, *human9*, *man*, *bird2*, *vase*), which can be found in [40] . Results are in terms of the AUC score of the success plot regarding the OPE protocol.

Parameter n_1 in Alg. 2 balances UPDATE and RETRIEVE, controlling the pace of appearance model update. n_1 influences the tracking more mildly as shown in Fig 9a, where the average AUC score ranges from 0.56 to 0.63. Nevertheless, the trade-off effect still exists, as both too large n_1 and too small n_1 will lead to inferior performance.

Parameter C_1 is the dual variable sum of descriptive support vectors. It determines the overall descriptive power of the long-term component, thus making a balance with the medium-term discriminative component. We observe how C_1 influences the performance in Fig 9b, and find the range from 10 to 20 is the optimal for C_1. Parameters C_2 and C_3 constrain the influence of a single sample in the descriptive component and the discriminative component, respectively. Experimental results in Figs. 9c and 9d verify their trade-off effects, and show that taking the value around 1 for both C_2 and C_3 can make good performance.

Parameter σ_t in Eq.3.4.1 determines the shape of motion probability, compromising with the appearance model. We denote the σ_t set in Eq.3.4.1 by σ_{st}, then investigate how performance changes with different σ_t. Fig.9 shows that the results are not very sensitive to σ_t, as the performance changes smoothly from $0.5\sigma_{st}$ to $4\sigma_{st}$.

In the dense search stage, we restrict MCT with maximum scale change of $\pm 2\%$ per frame to keep both robustness and efficiency. To study the influence of scale controlling, we conduct a set of the experiments allowing different extent of scale change per frame in Fig. 10. All the experiments are with 1% scale search precision. The results on the two datasets are diversified: for 10 additional videos, the performance gradually grows with the maximum scale change but falls at $\pm 10\%$; while for the 50 video benchmark, the performance fluctuates from $\pm 2\%$ to $\pm 10\%$. The effectiveness of controlling scale change is highly dependent on the data, but generally, considering scale change is better than using a fixed scale for MCT.

5.2.5 Failure Case

Currently we do not make use of color information, so MCT will fail when color is the most important tracking

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2539956, IEEE Transactions on Pattern Analysis and Machine Intelligence

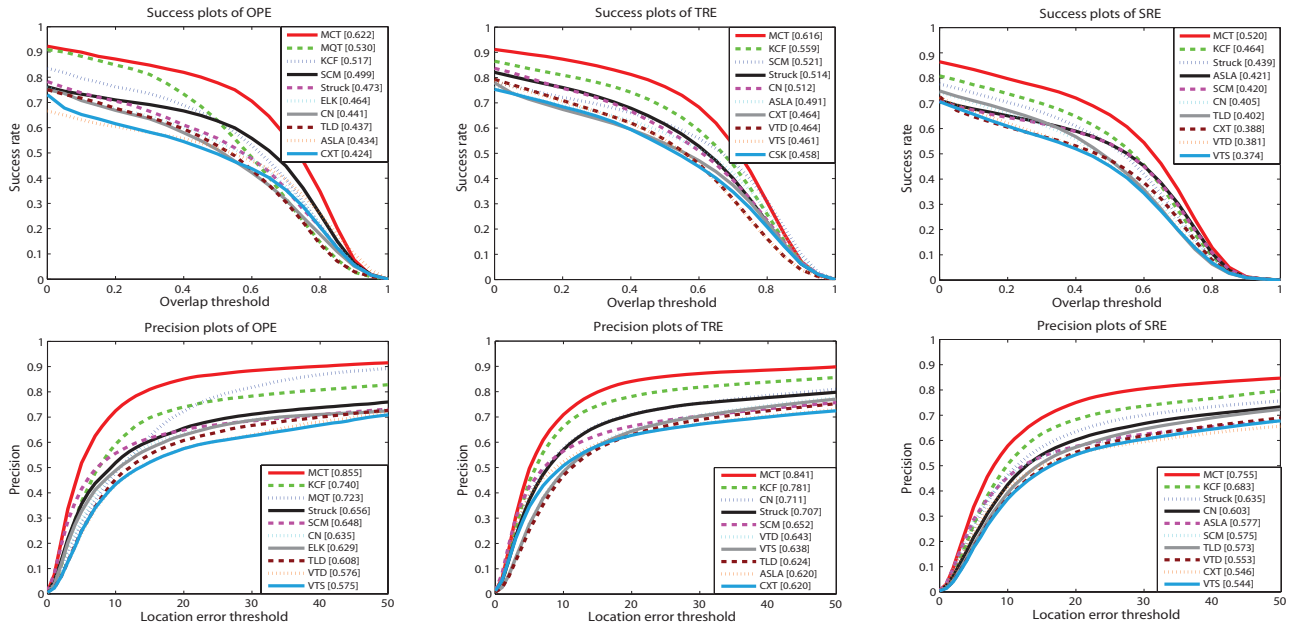


Fig. 12: **Comparison over the whole dataset**. We present success plots (top row) and precision plots (bottom row) according to the OPE, TRE and SRE protocols. The performance score for each tracker is shown in the legend. For each figure, only the top 10 plots are presented. More results are given in the supplementary materials.

Trackers	DFT	CXT	LSK	CSK	VTD	VTS	ASLA	ELK	TLD	CN	Struck	SCM	MQT	KCF	MCT
illumination	0.383	0.368	0.371	0.369	0.420	0.429	0.429	0.393	0.399	0.417	0.428	0.473	0.489	0.493	0.593
out of plane	0.387	0.418	0.400	0.386	0.434	0.425	0.422	0.448	0.420	0.443	0.432	0.470	0.521	0.495	0.606
scale variation	0.329	0.389	0.373	0.350	0.405	0.400	0.452	0.415	0.421	0.384	0.425	0.518	0.464	0.427	0.561
occlusion	0.381	0.372	0.409	0.365	0.403	0.398	0.376	0.428	0.402	0.428	0.413	0.487	0.519	0.514	0.608
deformation	0.439	0.324	0.377	0.343	0.377	0.368	0.372	0.476	0.378	0.438	0.393	0.448	0.589	0.534	0.649
motion blur	0.333	0.369	0.302	0.305	0.309	0.304	0.258	0.383	0.404	0.410	0.433	0.298	0.488	0.497	0.572
fast motion	0.320	0.388	0.328	0.316	0.302	0.300	0.247	0.440	0.417	0.373	0.462	0.296	0.494	0.459	0.573
in-plane rotation	0.365	0.452	0.411	0.399	0.430	0.416	0.425	0.404	0.416	0.469	0.444	0.458	0.482	0.497	0.573
out of view	0.351	0.427	0.430	0.349	0.446	0.443	0.312	0.402	0.457	0.410	0.459	0.361	0.502	0.550	0.627
background clutter	0.407	0.338	0.388	0.421	0.425	0.428	0.408	0.434	0.345	0.453	0.458	0.450	0.518	0.535	0.625
low resolution	0.200	0.312	0.235	0.350	0.177	0.168	0.157	0.336	0.309	0.311	0.372	0.279	0.326	0.312	0.456

TABLE 2: **Comparison over different attributes**. We present the average AUC score regarding the OPE protocol for top 15 trackers (DFT [13], CXT [28], LSK [41], CSK [42], VTD [19], VTS [43], ASLA [12], ELK [44], TLD [30], CN [45], Struck [25], SCM [29], MQT [46], KCF [47], MCT). The best 5 results of each attribute are denoted in **red**, **orange**, **yellow**, **green** and **cyan**.

cue. E.g., the target in *skiing* is a deformable, fast moving player as shown in Fig. 11, neither intensity nor the shape can distinguish the target from background. As our tracker is not restricted in a particular feature descriptor, we can integrate color descriptor into our method in the future.

5.3 Comparisons with the state-of-the-art Approaches

5.3.1 Comparisons over the Benchmark

Results of 29 trackers on the 50 video benchmark have been reported in [7]. For a more complete comparison, we add five recent trackers, including CN [45], KCF [47], STC [48], EKT [44], and MQT [46]. The results of CN, KCF and STC are obtained by running their publicly available source codes with default parameters, while the results of EKT and MQT are directly obtained from the authors, which are only evaluated using the OPE protocol.

Success and precision plots of OPE, TRE and SRE are displayed in Fig. 12. All these plots reveal the advantages of the proposed MCT, which outperforms the other trackers by significant large margins. The average results of SRE are lower than those of OPE for all the trackers, reflecting that all the trackers are sensitive to the initialization bounding boxes. Interestingly, other trackers usually achieve higher scores under TRE than those under OPE, while our tracker performs stably for both protocols. This phenomenon shows that our tracker is robust to different lengths of video sequences. Such temporal robustness can be owed to the employment of multi-timescale components, which not only keeps the high accuracy of the tracker, but also enables our tracker to capture the target even if the target had been lost for a while.

5.3.2 Comparisons over Different Attributes

We compare these trackers on difference groups of sequences, where each group corresponds to a kind of challenging attribute that a tracker will face. Tab. 2 shows the comparison results in terms of the AUC scores regarding the OPE protocol, where the results of top 5 trackers for each attribute are denoted in color. It can be seen that MCT outperforms the other state-of-the-art trackers on all the attributes. We select 6 typical attributes for qualitative evaluation. For each attribute, we just demonstrate the results of 5 best performed trackers.

Illumination Change. Fig. 13 presents the tracking results of *shaking* and *singer2*, where the target's appearance is heavily contaminated due to the dramatic changes of the stage lighting conditions. Our tracker successfully tracks both sequences and performs more stable than other 4 leading trackers of this attribute. The robustness to illumination change is mainly due to the medium-term component as analyzed in Fig. 6c. One reason is that the discriminative ability of this component highlights the edge information and off-sets some influences of illumination changes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2539956, IEEE Transactions on Pattern Analysis and Machine Intelligence

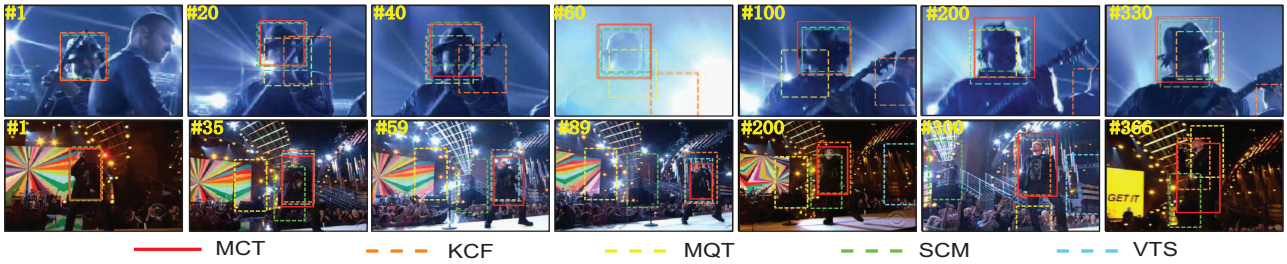


Fig. 13: Tracking results of *shaking* (top) and *singer2* (bottom), where there are severe illumination changes.

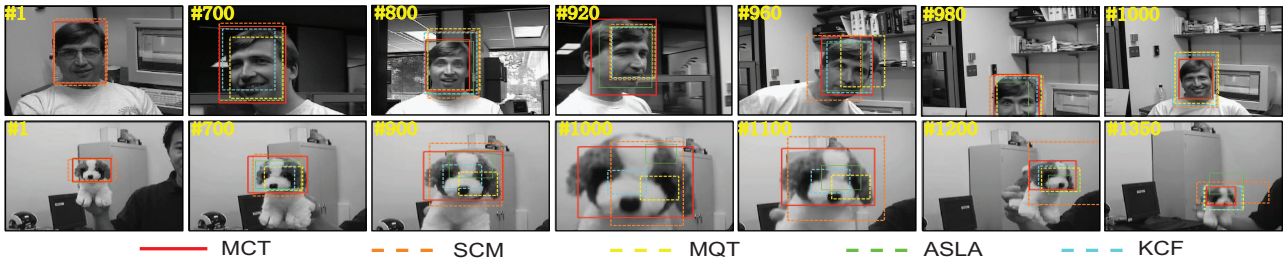


Fig. 14: Tracking results of *Dudek* (top) and *dog* (bottom), where there are large scale changes.

Fig. 15: Tracking results of *suv* (top) and *jogging1* (bottom), where there is heavy occlusion.

Scale Variation. Fig. 14 gives the results for the sequence with significant scale change. Among the top five leading trackers for this attribute, SCM, ASLA and MCT consider the scale change, while MQT and KCF utilize fixed scale. The results indicate the superiority of our coarse-to-fine search strategy, which enables MCT to consistently adapt to the changing scale.

Occlusion. Fig. 15 demonstrates how MCT handles occlusion. The car in sequence *suv* is partially occluded by the trees, while the runner in sequence *jogging* is completely occluded by the pole. MCT tracks *suv* quite well as it utilizes the medium-term component to distinguish the car from the complex background. It re-tracks the runner in *jogging1* because the long-term component can preserve the information of previous tracked target samples.

Deformation. Deformation is caused by target motion or by view point changes. Severe deformation is challenging as it leads very different appearance compared with the initialized one. MCT provides more moderate results than other leading trackers for this attributes as shown in Fig. 16. As the long-term component keeps historical target samples with different appearance, MCT is robust to multiple poses. The medium-term and short-term components utilize discriminative and contextual information to localize the targets, which is especially useful when the target's appearance is abruptly and severely deformed.

Motion Blur. Motion blur is challenging as it leads to the loss of target's details. KCF and MCT perform well on sequences *football1* and *soccer* (Fig. 17). Among them, KCF implicitly utilizes a large number of samples at different relative translations to compensate the lost information [47]. Meanwhile, MCT enriches the target information by feature composition. Especially, *Type-III* descriptor is shown to be the most effective one to handle motion blur in Fig. 5c.

Background Clutter. Sequence *couple* and sequence *football* have severe background clutters. We find that discriminative ability is the key to handle background clutter. This conclusion is not only obtained from the analysis in Fig. 6c, but also is confirmed by the results of other trackers. For example, Struck employs a discriminative model, so it performs better on background clutter than on most of the other attributes.

6 Conclusion

We have proposed an effective visual tracking framework based on the collaboration of three components with multiple timescales. In our framework, the components take advantage of different forces within observed samples, thus can account for various challenges in tracking. In particular, the long-term descriptive component and the medium-term discriminative component are collaborated in a unified max-margin framework, forming a collaborative appearance model that is not vulnerable to outliers and can track the target with high accuracy. The short-term regressive component collaborates the votes of auxiliary samples, forming the context-aware motion model to increase the tracking reliability.

The appearance model and the motion model determine the probability of the target state togehter, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2539956, IEEE Transactions on Pattern Analysis and Machine Intelligence

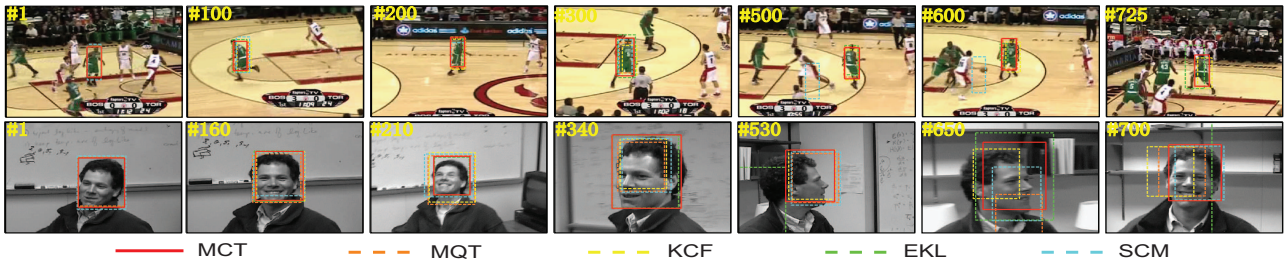


Fig. 16: Tracking results of *basketball* (top) and *Fleetface* (bottom), where there is large deformation.

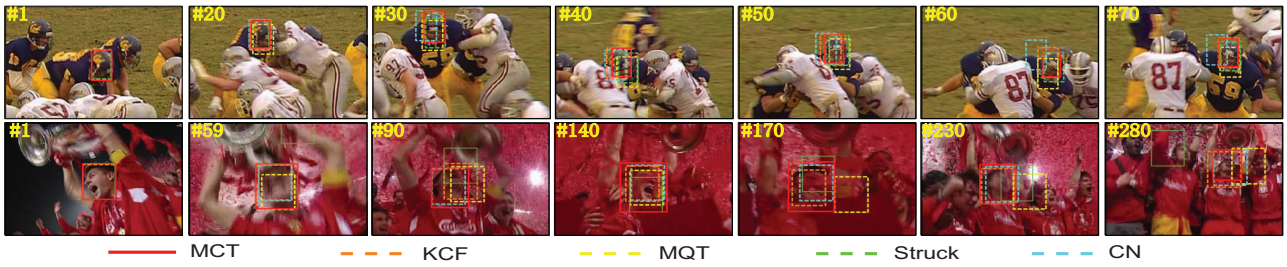


Fig. 17: Tracking results of *football1* (top) and *soccer* (bottom), where there is heavy motion blur.

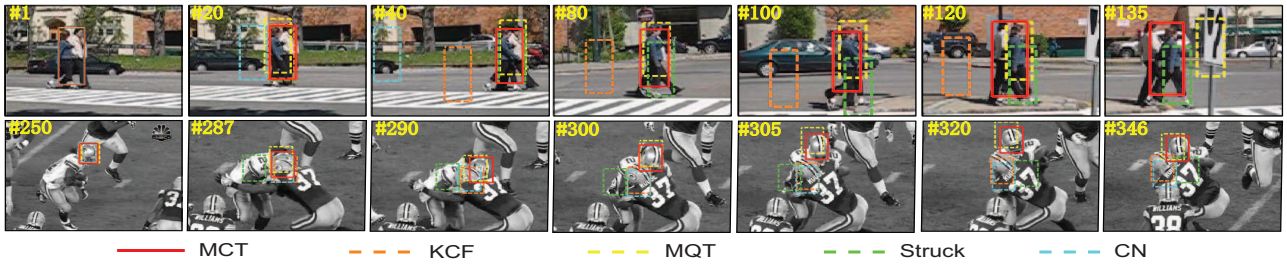


Fig. 18: Tracking results of *couple*(top) and *football*(bottom), where there is background clutter.

the optimal state is estimated via a novel coarse-to-fine search strategy. For online implementation, we devised a set of efficient online optimization rules as well as set management operations to accelerate the tracking process. Experiments over the 50 video tracking benchmark confirm that the collaboration of three components actually makes better performance, and our method generally outperforms existing trackers.

Acknowledgement This work was supported by National Basic Research Program of China (No.2015CB351703), National Natural Science Foundation of China (No.61573280, No.61231018), and 111 Project (No.B13043).

REFERENCES

[1] J. Gall, A. Yao, N. Razavi, L. J. V. Gool, and V. S. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2188–2202, 2011.

[2] N. Krahnstoever, J. Rittscher, P. H. Tu, K. Chean, and H. T. Tomlinson, "Activity recognition using visual tracking and RFID," in *IEEE Workshop on Applications of Computer Vision*, 2005.

[3] B. Zhou, X. Tang, and X. Wang, "Learning collective crowd behaviors with dynamic pedestrian-agents," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 50–68, 2015.

[4] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004.

[5] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, 2005.

[6] V. Vapnik, *Statistical learning theory*. Wiley, 1998.

[7] Y. Wu, J. Lim, and M. Yang, "Online object tracking: A benchmark," in *Computer Vision and Pattern Recognition*, 2013.

[8] D. Chen, Z. Yuan, Y. Wu, G. Zhang, and N. Zheng, "Constructing adaptive complex cells for robust visual tracking," in *Int'l Conf. on Computer Vision*, 2013.

[9] D. Chen, Z. Yuan, G. Hua, Y. Wu, and N. Zheng, "Description-discrimination collaborative tracking," in *European Conf. Computer Vision*, 2014.

[10] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, 2014.

[11] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–575, 2003.

[12] X. Jia, H. Lu, and M. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Conference on Computer Vision and Pattern Recognition*, 2012.

[13] L. Sevilla-Lara and E. G. Learned-Miller, "Distribution fields for tracking," in *CVPR*, 2012.

[14] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Conference on Computer Vision and Pattern Recognition*, 2006.

[15] H. Wang, D. Suter, K. Schindler, and C. Shen, "Adaptive object tracking based on an effective appearance filter," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1661–1667, 2007.

[16] D. W. Park, J. Kwon, and K. M. Lee, "Robust visual tracking using autoregressive hidden markov model," in *Conference on Computer Vision and Pattern Recognition*, 2012.

[17] S. Hong and B. Han, "Visual tracking by sampling tree-structured graphical models," in *European Conference on Computer Vision*, 2014.

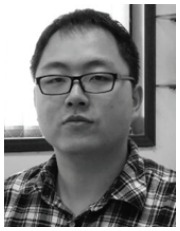
[18] D. A. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[19] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Conference on Computer Vision and Pattern Recognition*, 2010.

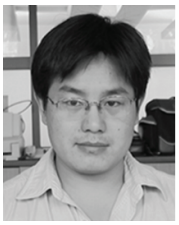
[20] H. T. Nguyen, Q. Ji, and A. W. M. Smeulders, "Spatio-temporal context for robust multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 52–64, 2007.

[21] X. Mei and H. Ling, "Robust visual tracking and vehicle

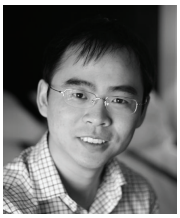
classification via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2259–2272, 2011.
[22] J. Xing, J. Gao, B. Li, W. Hu, and S. Yan, "Robust object tracking with online multi-lifespan dictionary learning," in *Int'l Conf. on Computer Vision*, 2013.
[23] B. Babenko, M. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, 2011.
[24] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *3rd IEEE - ICCV Workshop on On-line Learning for Computer Vision*, 2009.
[25] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Int'l Conf. on Computer Vision*, 2011.
[26] M. Yang, Y. Wu, and G. Hua, "Context-aware visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 7, pp. 1195–1209, 2009.
[27] H. Grabner, J. Matas, L. J. V. Gool, and P. C. Cattin, "Tracking the invisible: Learning where the object might be," in *Computer Vision and Pattern Recognition*, 2010.
[28] T. B. Dinh, N. Vo, and G. G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1177–1184.
[29] W. Zhong, H. Lu, and M. Yang, "Robust object tracking via sparsity-based collaborative model," in *CVPR*, 2012.
[30] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, 2012.
[31] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition*, 2005.
[32] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition*, 2001.
[33] A. Bordes, L. Bottou, P. Gallinari, and J. Weston, "Solving multiclass support vector machines with larank," in *International Conference on Machine Learning*, 2007.
[34] B. Schölkopf and A. J. Smola, "A short introduction to learning with kernels," in *Advanced Lectures on Machine Learning, Machine Learning Summer School*, 2002.
[35] P. Del Moral, A. Doucet, and A. Jasra, "Sequential monte carlo samplers," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 3, pp. 411–436, 2006.
[36] L. Wen, Z. Cai, Z. Lei, D. Yi, and S. Z. Li, "Robust online learned spatio-temporal context model for visual tracking," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 785–796, 2014.
[37] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
[38] A. Bordes and L. Bottou, "The huller: A simple and efficient online SVM," in *European Conference on Machine Learning*, 2005.
[39] A. Bordes, N. Usunier, and L. Bottou, "Sequence labelling svms trained in one pass," in *ECML/PKDD*, 2008.
[40] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2015.
[41] B. Liu, J. Huang, C. A. Kulikowski, and L. Yang, "Robust visual tracking using local sparse appearance model and k-selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2968–2981, 2013.
[42] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European Conference on Computer Vision*, 2012.
[43] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *International Conference on Computer Vision*, 2011.
[44] S. Oron, A. Bar-Hillel, and S. Avidan, "Extended lucas-kanade tracking," in *European Conference on Computer Vision*, 2014.
[45] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Conference on Computer Vision and Pattern Recognition*, 2014.
[46] Z. Hong, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Tracking using multilevel quantizations," in *European Conference on Computer Vision*, 2014.
[47] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2015.
[48] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *European Conference on Computer Vision*, 2014.



Dapeng Chen received his B.S. degree in Biomedical Engineering from Xi'an Jiaotong University, China, in 2010. He is currently a Ph.D. student at the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His current research interests include computer vision and machine learning. In particular, he worked on video data including tracking and person re-identification.



Zejian Yuan received the MS degree in electronic engineering from the Xian University of Technology in 1999 and the PhD degree (2003) in pattern recognition and intelligent system from Xian Jiaotong University. Dr. Yuan was a visiting scholar in the Advanced Robotics Lab of Chinese University of Hong Kong in 2008-2009. He is currently an associate professor at the Department of Automatic Engineering, Xian Jiaotong University, and a member of Chinese Association of Robotics. His research interests include image processing, pattern recognition, as well as machine learning methods in computer vision.



Gang Hua (M03-SM11) received the B.S. degree in automatic control engineering and M.S. degree in pattern recognition and intelligence system from Xian Jiaotong University in 1999 and 2002, respectively, and the Ph.D. degree in electrical and computer engineering from Northwestern University in 2006. He is an Associate Professor of Computer Science in Stevens Institute of Technology. Prior to that, he was a Research Staff Member at IBM Research T. J. Watson Center from 2010 to 2011, a Senior Researcher at Nokia Research Center, Hollywood from 2009 to 2010, and a Scientist at Microsoft Live Labs Research from 2006 to 2009. His primary research interests are in Computer Vision and Robotics, and Multimedia Content Analysis. Dr. Hua is a life member of the Association for Computing Machinery. He was the recipient of the Richter Fellowship and the Walter P. Murphy Fellowship from Northwestern University in 2005 and 2002, respectively.

Jingdong Wang is a Lead Researcher at the Visual Computing Group, Microsoft Research Asia. He received the M.Eng. and B.Eng. degrees in Automation from Tsinghua University in 2001 and 2004, respectively, and the PhD degree in Computer Science from the Hong Kong University of Science and Technology, Hong Kong, in 2007. His research interest include computer vision, machine learning, pattern recognition, and multimedia computing. In particular, he has worked on kernel methods, semi-supervised learning, data clustering, image segmentation, and image and video presentation, management and search. At present, he is mainly working on the Big Media project, including large-scale indexing and clustering, Web image search and mining, and visual understanding such as salient object detection, image recognition, face alignment and recognition.

Nanning Zheng (SM93-F06) graduated from the Department of Electrical Engineering, Xian Jiaotong University, Xian, China, in 1975, and received the M.S. degree in information and control engineering from Xian Jiaotong University in 1981 and the Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1985. He jointed Xian Jiaotong University in 1975, and he is currently a Professor and the Director of the Institute of Artificial Intelligence and Robotics, Xian Jiaotong University. His research interests include computer vision, pattern recognition and image processing, and hardware implementation of intelligent systems. Dr. Zheng became a member of the Chinese Academy of Engineering in 1999, and he is the Chinese Representative on the Governing Board of the International Association for Pattern Recognition.