Description-Discrimination Collaborative Tracking

Dapeng Chen[1], Zejian Yuan[1], Gang Hua[2], Yang Wu[3], and Nanning Zheng[1]

[1] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, China
[2] Department of Computer Science, Stevens Institute of Technology, USA
[3] Academic Center for Computing and Media Studies, Kyoto University, Japan

Abstract. Appearance model is one of the most important components for online visual tracking. An effective appearance model needs to strike the right balance between being adaptive, to account for appearance change, and being conservative, to re-track the object after it loses tracking (*e.g.*, due to occlusion). Most conventional appearance models focus on one aspect out of the two, and hence are not able to achieve the right balance. In this paper, we approach this problem by a max-margin learning framework collaborating a descriptive component and a discriminative component. Particularly, the two components are for different purposes and with different lifespans. One forms a robust object model, and the other tries to distinguish the object from the current background. Taking advantages of their complementary roles, the components improve each other and collaboratively contribute to a shared score function. Besides, for realtime implementation, we also propose a series of optimization and sample-management strategies. Experiments over 30 challenging videos demonstrate the effectiveness and robustness of the proposed tracker. Our method generally outperforms the existing state-of-the-art methods.

Keywords: Descriptive model, discriminative model, collaborative tracking, SVDD, structural prediction, long-term and short-term memory.

1 Introduction

Visual tracking is a fundamental research problem in computer vision and is important for a large variety of applications. Although significant progress has been made, challenges still remain due to numerous factors such as partial occlusion, illumination change, pose variation, and background clutter, etc. To handle the challenges, it is important to adopt an appropriate appearance model.

An appearance model can be built descriptively, to form a robust object model; or be built discriminatively, to separate the object from surrounding background. Both have their strengths and weaknesses in visual tracking. The former directly models the object appearance [13,17,11,20], but easily drifts to similar distractors, the latter one distinguishes the target from the background [2,10,14], but is not robust enough as the background may change dramatically.

D. Fleet et al. (Eds.): ECCV 2014, Part I, LNCS 8689, pp. 345–360, 2014.

Although several collaborative models have been proposed to take the best of both [26,30], they usually learn the two kinds of appearance models separately, which hinders them seeking the right level of balance between these two types of models.

Another factor needs to be considered for appearance model is adaption. In order to capture the dynamically changing appearance, it is also required that the appearance model should be adaptively updated. Some models adjust an ad-hoc learning rate to update the appearance model with most recent observations [2,11,6], which makes the tracker be prone to drift in case of erroneous updates. Some other models learn from a subset of historically observed samples [10,12], which is not sufficiently adaptive to handle fast appearance change. To cope with the well known "stability-plasticity" dilemma, Santner et. al [21] combine complementary models operated at different timescales, Xing et al[28] collect samples at different time for online dictionary learning. Their success suggest that utilizing different lifespan information is important for adaption, but how to balance this these information remains to be an open problem.

We propose a novel way to collaborate the descriptive component with the discriminative component in a unified max-margin framework for appearance modeling. The two components are with different lifespans to better exploit their complementary modeling power, leading to a more data-dependent adaption of appearance model. The main contributions of the paper lies in three aspects:

Components: We employ a descriptive component and a discriminative component to composite the appearance model. The descriptive component is based on Support Vector Data Description (SVDD) [23]. It describes the global properties of the target from all the tracked frames, using representative samples to capture their essential characteristics. Meanwhile, the discriminative component is based on Structured Output SVM (SSVM) [24]. It differentiates the targets from its surrounding background in recent frames, focusing on the most violated background samples to guide the accurate localization.

Collaboration: We cast the two relevant but distinctive components in a unified max-margin learning framework, where they are combined in a mutually beneficial way. The descriptive component uses discriminative information to modify its descriptive boundary, and the discriminative component recalls relevant descriptive samples to increase its discriminative ability. More meaningfully, as the two components have different lifespans. The adaption of the appearance model is influenced by current discriminative samples, but at the same time seeks for a consistence with previous descriptive samples.

Computation: To reduce the computational burden, two kinds of strategies are taken. The first is the learning strategy. We optimize the collaborative model in its dual form to make use of optimized solution from previous time instance, and only select the most informative samples for fast approaching the optimum. The second is the implementation strategy. As the training data increase linearly during tracking, there is a need to control the size of sample set. We adopt a series of set management operations, which boost the tracking speed without impacting much of the tracking accuracy.

2 Related Work

We compare our description-discrimination collaborative tracker with the previous methods based on generative, discriminative and collaborative models.

Generative models estimate the distribution of object appearance directly, they usually form a robust object representation in a particular feature space, including superpixel [25], and feature histograms [1,11,6], etc. Recently, subspace based generative models attract a lot of attention [20,15], and the trackers making use of sparse representation become quite popular [17,13,30]. Different from generative models, the descriptive component in our method is based on the idea of SVDD [23], which estimates the support of the target distribution rather than the full density. As shown in [7], the decision function of SVDD can well capture the density and modality of the feature distribution by using kernel techniques [7,16], which is effective to capture the changing appearance of the target.

Discriminative models aim to distinguish the target from the background. They usually train a dynamic target classifier with the most prevalent algorithms, such as boosting [9,2], random forest [21,14] and SVM [12,10]. The discriminative component in our model is inspired by a state-of-the-art discriminative tracker [27], termed "Struck" [10]. Struck predicts the change in object location using structured output SVM(SSVM) [24], which alleviates the "label jitter" and turns out to be more suitable than binary classifier for prediction. Compared with Struck, our discriminative component regards the temporal inequality between target and background. Specifically, we only utilize recent background samples, which is more suitable for tracking in the dynamic environment.

Collaborative models have already attracted a lot of attention. They collaborate different models to explore their complementary strength to enhance the tracking robustness. For example, Wen et. al [26] and Zhong et. al [30] employ the different models in parallel, and predict the targets by fusing their separate results. Meanwhile, Kalal et. al [14] integrate different models in a cascade, successively selects the best sample from the candidates. Both kinds of collaboration do not build mutual beneficial connections between different models, therefore lack a unified and consistent treatment to explore the complementary strength.

3 Description-Discrimination Collaboration

An object is represented by a bounding box. Let $\mathcal{Y}$ stand for the set of possible bounding boxes, whose element $\mathbf{y} = \{x, y, s\}$ is a three dimensional vector describing position and scale. The features extracted from image $\mathbf{x}_t$ that correspond to the area inside the bounding box $\mathbf{y}$ are denoted as $\phi(\mathbf{x}_t, \mathbf{y})$. In this paper, $\phi(\mathbf{x}_t, \mathbf{y})$ is a high dimensional normalized vector, whose L_2 norm is required to be a constant.

Instead of training a binary classifier over $\phi(\mathbf{x}_t, \mathbf{y})$, we learn a score function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ that measures the compatibility between $(\mathbf{x}_t, \mathbf{y})$ pairs. Considering its efficiency at predictive stage, F is assumed to be linear that can yield higher scores to those more similar to the target. The optimal state $\hat{\mathbf{y}}_t$ is predicted by:

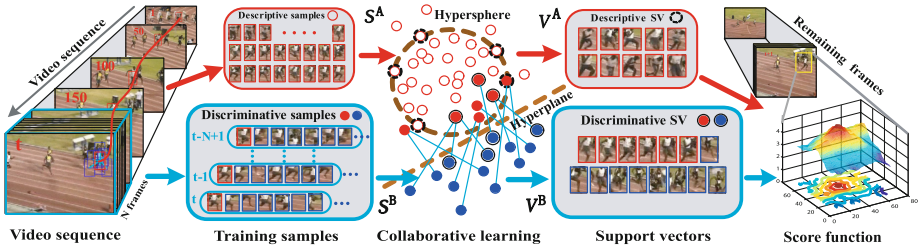


Fig. 1. Overview of our Description-Discrimination Collaborative Tracking algorithm. We crop the long-term target samples and short-term target-background samples into $\mathcal{S}^A$ and $\mathcal{S}^B$(red dot is the target while blue dot is the background). In order to keep both robustness and adaptiveness, the score the target samples are highlighted by different samples from different views. We put the learned support vectors (dots with loops) into $\mathcal{V}^A$ and $\mathcal{V}^B$. They together contribute to the score function.

$$\hat{\mathbf{y}}_t = \arg\max_{\mathbf{y}} F(\mathbf{x}_t, \mathbf{y}) = \langle \mathbf{w}, \phi(\mathbf{x}_t, \mathbf{y}) \rangle. \tag{1}$$

Parameters $\mathbf{w}$ encode the object's appearance, which is collaboratively learned from two components through a single objective function:

$$\min_{\mathbf{w}} \quad \mathcal{R}(\mathbf{w}) + C^{des} \cdot \mathcal{L}^{des}(\mathbf{w}) + C^{dis} \cdot \mathcal{L}^{dis}(\mathbf{w}), \tag{2}$$

where $\mathcal{L}^{des}$, $\mathcal{L}^{dis}$ represent the loss terms on the descriptive component and discriminative component. C^{des} and C^{dis} are scalar parameters to trade-off the impact between the two components. $\mathcal{R}(\mathbf{w}) = \|\mathbf{w}\|_2^2$ is the regularization term.

3.1 Descriptive Component

As the object appearance continuously changes in the feature space, neither off-line trained detector nor the appearance template from the first frame is able to capture its variations. To built an effective prior for tracking, we focus on describing the dynamical target set $\mathcal{S}^A$, which ideally contains the features of all tracked targets until the current time instance, *i.e.* $\mathcal{S}^A = \{\phi(\mathbf{x}_i, \hat{\mathbf{y}}_i) | i = 1...t\}$.

We describe the set $\mathcal{S}^A$ using SVDD. The basic idea of SVDD is to employ a hypersphere to enclose the target set and minimize the sphere's volume to exclude outliers. Given the hypersphere's center $\mathbf{c}$, the descriptive loss term is :

$$\mathcal{L}^{des}(\mathbf{c}) = \min_{R} R^2 + \bar{C} \sum_i H(\|\phi(\mathbf{x}_i, \hat{\mathbf{y}}_i) - \mathbf{c}\|^2 - R^2), \tag{3}$$

where R is the radius of the hypersphere, and $H(z) = max(0, z)$ is the hinge loss. As mentioned above, all the features are constrained to have a constant norm a, *i.e.*, $\|\phi(\mathbf{x}_t, \mathbf{y})\|_2 = a$. Let $\mathbf{w} = 2\mathbf{c}$ and $\rho = \frac{1}{4}\|\mathbf{w}\|_2^2 + a^2 - R^2$, $\mathcal{L}^{des}(\mathbf{c})$ can be transformed to $\mathcal{L}^{des}(\mathbf{w})$:

$$\mathcal{L}^{des}(\mathbf{w}) = \min_{\rho} \frac{1}{4}\|\mathbf{w}\|_2^2 - \rho + \bar{C} \sum_i H(\rho - \mathbf{w} \cdot \phi(\mathbf{x}_i, \hat{\mathbf{y}}_i)) + const. \tag{4}$$

where $\mathcal{L}^{des}(\mathbf{w})$ is in the form of 1-class svm [22], which is convenient to be optimized and is good at handling high dimensional data. The proposed descriptive component have two advantages for object tracking. It captures the global support of the samples in $\mathcal{S}^A$, hence is robust to outlier target samples. In addition, the learning of descriptive component needs less prior knowledge but depends more on the tracked samples, enabling the tracker to adapt to the complex changes of the object.

3.2 Discriminative Component

However, if the tracker merely relies on the descriptive component, it tends to fail when the object's appearance changes rapidly. This is because the descriptive component attempts to describe the whole distribution of target samples and may not well capture the current object appearance. Opposite to the target object, the background contains important contextual cues and is effective for accurate localization. To achieve tracking adaptivity, we only focus on the most recent N frames, where the both target and background samples are cropped into a set $\mathcal{S}^B$, *i.e.* $\mathcal{S}^B = \{\phi(\mathbf{x}_j, \mathbf{y}) | \mathbf{y} \in \mathcal{Y}, j = t-N+1, ..., t\}$.

Inspired by the Struck tracker [10], we discriminate the target and the background samples in $\mathcal{S}^B$ using SSVM [24]. The basic idea of SSVM is that the scores of the target should be larger than those of the background samples in the same frame at least by a margin $\Delta(\hat{\mathbf{y}}_j, \mathbf{y})$. Therefore, $\mathcal{L}^{dis}(\mathbf{w})$ is

$$\mathcal{L}^{dis}(\mathbf{w}) = \sum_{j, \mathbf{y} \neq \hat{\mathbf{y}}_\mathbf{j}} H(\Delta(\hat{\mathbf{y}}_j, \mathbf{y}) - \mathbf{w} \cdot \delta\phi_j(\mathbf{x}_j, \mathbf{y})), \tag{5}$$

where $\delta\phi_j(\mathbf{x}_j, \mathbf{y}) = \phi(\mathbf{x}_j, \hat{\mathbf{y}}_j) - \phi(\mathbf{x}_j, \mathbf{y})$, and $\Delta(\hat{\mathbf{y}}_i, \mathbf{y})$ is the structural loss that rescales the margin of each sample differently based on the bounding box overlap ratio, defined as $\Delta(\hat{\mathbf{y}}_i, \mathbf{y}) = 1 - \frac{\text{Area}(\hat{\mathbf{y}}_i \bigcap \mathbf{y})}{\text{Area}(\hat{\mathbf{y}}_i \bigcup \mathbf{y})}$. Different from binary classifiers, SSVM explores the structural relationship among samples that each target sample is associated with the background samples in the same frame. In this way, the contextual information contained in background samples is well oriented to the specific target instance and can be updated along with the target instance as well.

3.3 Collaborative Model

We take Eq. 4 and Eg. 5 into Eq. 2. After arranging the coefficients, the original objective function is rewritten as:

$$\min_{\mathbf{w}, \rho} \frac{1}{2}\|\mathbf{w}\|_2^2 - C_1\rho + C_2 \sum_i H(\rho - \mathbf{w} \cdot \phi(\mathbf{x}_i, \hat{\mathbf{y}}_i)) + C_3 \sum_{j, \mathbf{y} \neq \hat{\mathbf{y}}_\mathbf{j}} H(\Delta(\hat{\mathbf{y}}_i, \mathbf{y}) - \mathbf{w} \cdot \delta\phi_j(\mathbf{x}_j, \mathbf{y})), \tag{6}$$

Using standard Lagrangian duality and reparametrizing techniques [4], we introduce multipliers $\alpha_i, \beta_j^{\mathbf{y}}$ for each feature $\phi(\mathbf{x}_i, \hat{\mathbf{y}}_i)$ in $\mathcal{S}^A$ and $\phi(\mathbf{x}_j, \mathbf{y})$ in $\mathcal{S}^B$. Then the dual form of Eq. 6 is[1]:

[1] We leave the derivation in the supplementary materials.

$$\begin{aligned} &\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad -\frac{1}{2}\boldsymbol{\alpha}^\top K^A \boldsymbol{\alpha} - \boldsymbol{\alpha}^\top K^{AB} \boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\beta}^\top K^B \boldsymbol{\beta} - \boldsymbol{\beta}^\top \Delta, \\ &\mathbf{s.t.} \quad \forall i \quad \sum_i \alpha_i = C_1, 0 \le \alpha_i \le C_2; \quad \forall j, \mathbf{y} \quad \sum_{\mathbf{y}} \beta_j^{\mathbf{y}} = 0, \beta_j^{\mathbf{y}} \le C_3 \delta(\mathbf{y}, \hat{\mathbf{y}}), \end{aligned} \tag{7}$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and Δ are column vectors that concatenate the α_i, $\beta_j^{\mathbf{y}}$ and $\Delta(\hat{\mathbf{y}}_j; \mathbf{y})$; K^A and K^B are the kernel matrices for $\mathcal{S}^A$ and $\mathcal{S}^B$; and K^{AB} measures the inter affinities between the two sets. The entries of the three matrices are all calculated based on a linear kernel function: $k(\mathbf{x}, \mathbf{y}, \bar{\mathbf{x}}, \bar{\mathbf{y}}) = \langle \phi(\mathbf{x}, \mathbf{y}), \phi(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \rangle$. With the multipliers α and $\beta_j^{\mathbf{y}}$, the parameters $\mathbf{w}$ in Eq.7 is represented as: $\mathbf{w} = \boldsymbol{\alpha}^\top \Phi_A + \boldsymbol{\beta}^\top \Phi_B$, where Φ_A and Φ_B are feature matrices that concatenate the features in $\mathcal{S}^A$ and $\mathcal{S}^B$ along the column. $\boldsymbol{\alpha}^\top \Phi_A$ corresponds to the descriptive component, which is a nonnegative linear combination of features in $\mathcal{S}^A$, while $\boldsymbol{\beta}^\top \Phi_B$ corresponds to the discriminative component, which is a linear combination of features in $\mathcal{S}^B$ highlighting the difference between target and background samples. All the features with non-zero multipliers are called *Support Vectors*.

Discussion. The proposed collaborative model intends to better exploit the different properties of the target samples and background samples to build more robust appearance model for visual tracking. Generally, there are a small number of target samples, while background samples surround the target are abundant. Furthermore, the appearances of target samples from different frames are relatively similar, while the appearances of background samples vary a lot especially in dynamic scenes. The collaborative model is well oriented to the two properties.

Firstly, the discriminative component takes advantage of SSVM [24] like Struck. SSVM makes use of structured samples, it does not need to sample around the target to obtain positive samples that may cause "label jitter", but only stresses that the score of target sample should be larger than the scores of background samples in the same frame. Secondly, the descriptive component utilizes SVDD [23]. SVDD explicitly puts a prior on the target samples to capture the major characteristic of the object. It is robust to outlier and alleviates the learning burden of SSVM by avoiding using obsolete background samples.

The collaborative strategy is superior to Struck - the tracker using SSVM, which, along with its variants has been regarded as the state-of-the-art during recent evaluations and challenges [18,27]. Under the framework of SSVM, in order to retrieve historical target samples for learning the appearance model, Struck has to use the obsolete background samples in the same frame with the target sample. However, for tracking in dynamic scenes, the obsolete background samples can hardly help the current tracking, instead it would actually contaminate the appearance model and increase the computational cost. Our collaborative model gets rid of this limitation. The descriptive component summarizes the previous target samples to be robust, while the discriminative component adopts most effective background samples to be accurate. The two components build natural connections between each other, and the samples in $\mathcal{S}^A$ and $\mathcal{S}^B$ together decide the learning of each component.

More interesting, as the two components have different lifespans, their collaboration corresponds to the theory of the long-term and short-term memory in

human brain, where the long-term memory (descriptive component)recalls more about the object itself rather than the background, while the short-term memory (discriminative component) utilize the contextual information to influence the forming of the long-term memory.

4 Online Optimization

Eq.7 is a typical quadratic optimization problem for both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Considering the tracking efficiency, we decompose the original problem into a sequence of subproblems inspired by the SMO algorithm [19]. Each subproblem first selects coefficient pairs (α_+, α_-) and $(\beta_j^{\mathbf{y}+}, \beta_j^{\mathbf{y}-})$, then optimizes the coefficients using an *elementary step*. In this section, we first discuss the elementary step, then explain the online selection.

4.1 Elementary Step

As constrained by $\sum_i \alpha_i = C_1$ and $\sum_{\mathbf{y}} \beta_j^{\mathbf{y}} = 0$, the elementary step modifies the coefficient pairs by opposite amounts:

$$\begin{cases} \alpha_+ \leftarrow \alpha_+ + \lambda^\alpha \\ \alpha_- \leftarrow \alpha_- - \lambda^\alpha \end{cases} \qquad \begin{cases} \beta_j^{\mathbf{y}+} \leftarrow \beta_j^{\mathbf{y}+} + \lambda^\beta \\ \beta_j^{\mathbf{y}-} \leftarrow \beta_j^{\mathbf{y}-} - \lambda^\beta \end{cases}, \tag{8}$$

where $\lambda^\alpha, \lambda^\beta \geq 0$, leading to an one-step maximization subject to the constraints in Eq. 7. In order to obtain $\lambda^\alpha, \lambda^\beta$, we first introduce $g(\alpha_i)$ and $g(\beta_j^{\mathbf{y}})$, which are the gradients of Eq.7 w.r.t. the multipliers α_i and $\beta_j^{\mathbf{y}}$, respectively:

$$g(\alpha_i) = -\langle \mathbf{w}, \phi(\mathbf{x}_i, \hat{\mathbf{y}}_i) \rangle; \qquad g(\beta_j^{\mathbf{y}}) = -\langle \mathbf{w}, \phi(\mathbf{x}_j, \mathbf{y}) \rangle - \Delta(\hat{\mathbf{y}}_j, \mathbf{y}). \tag{9}$$

We first calculate the unconstrained $\tilde{\lambda}^\alpha$ and $\tilde{\lambda}^\beta$ as:

$$\begin{aligned} \tilde{\lambda}^\alpha &= \frac{g(\alpha_+) - g(\alpha_-)}{Z_{\alpha_+\alpha_-}}, & \tilde{\lambda}^\beta &= \frac{g(\beta_j^{\mathbf{y}+}) - g(\beta_j^{\mathbf{y}-})}{Z_{\beta_j^{\mathbf{y}+}\beta_j^{\mathbf{y}-}}} \\ Z_{\alpha_+\alpha_-} &= k_{\alpha_+\alpha_+} + k_{\alpha_-\alpha_-} - 2k_{\alpha_+\alpha_-}, & Z_{\beta_j^{\mathbf{y}+}\beta_j^{\mathbf{y}-}} &= k_{\beta_j^{\mathbf{y}+}\beta_j^{\mathbf{y}+}} + k_{\beta_j^{\mathbf{y}-}\beta_j^{\mathbf{y}-}} - 2k_{\beta_j^{\mathbf{y}+}\beta_j^{\mathbf{y}-}} \end{aligned} \tag{10}$$

where $k_{\alpha_+\alpha_+}$,$k_{\beta_j^{\mathbf{y}+}\beta_j^{\mathbf{y}+}}$, ... are kernel values for the corresponding feature pairs. We enforce the constraints in Eq.7 to get the exact adjustment of λ^α and λ^β, *i.e.*:

$$\lambda^\alpha = \max(\min(\tilde{\lambda}^\alpha, \alpha_-, C_2 - \alpha_+), 0), \quad \lambda^\beta = \max(\min(\tilde{\lambda}^\beta, C_3\delta(\mathbf{y}, \hat{\mathbf{y}}_j) - \beta_j^{\mathbf{y}+}), 0). \tag{11}$$

Finally, the parameter $\mathbf{w}$ is updated according to

$$\mathbf{w} \leftarrow \mathbf{w} + \lambda^\beta(\phi(\mathbf{x}_j, \mathbf{y}+) - \phi(\mathbf{x}_j, \mathbf{y}-)) + \lambda^\alpha(\phi(\mathbf{x}_+, \hat{\mathbf{y}}_+) - \phi(\mathbf{x}_-, \hat{\mathbf{y}}_-)). \tag{12}$$

The entire elementary step is summarized in Alg. 1.

Algorithm 1. Elementary step

Compute the gradients $g(\alpha_+), g(\alpha_-), g(\beta_j^{\mathbf{y}+}), g(\beta_j^{\mathbf{y}-})$	Eq. 9
Compute the unconstrained $\tilde{\lambda}^\alpha, \tilde{\lambda}^\beta$	Eq. 10
Enforce the constrains to obtain $\lambda^\alpha, \lambda^\beta$	Eq. 11
Update the coefficients $\alpha_+, \alpha_-, \beta_j^{\mathbf{y}+}, \beta_j^{\mathbf{y}-}$	Eq. 8
Update the parameter $\mathbf{w}$	Eq. 12

4.2 Online Selection

Online selection hinges on how to choose proper coefficient pairs that should be optimized by the elementary step. Intuitively, the pair of coefficients should define the feasible search direction with highest gradient. Even by doing so, searching such coefficients from all the samples still need large storage and expensive computation, which hinders online tracking. As it has been observed that support vectors are not updated frequently [3], it is indeed effective to select coefficients focusing on support vectors. Inspired by OLaRank [5], we design three blocks for selection, which can *update*, *retrieve*, and *adjust* the support vectors respectively:

- **UPDATE** selects the coefficients from newly incoming frame $\mathbf{x}_t$ to improve the model with new information.
- **RETRIEVE** selects the coefficients from past frames to retrieve past data to assure the model's generalization ability.
- **ADJUST** selects the coefficients of the current support vectors , and adjust them to better adapt the model.

For convenience, we define $\mathcal{V}^A$ and $\mathcal{V}^B$ as the support vectors in $\mathcal{S}^A$ and $\mathcal{S}^B$, and we also define $\mathcal{C}^{SA}$, $\mathcal{C}^{SB}$, $\mathcal{C}^{VA}$, $\mathcal{C}^{VB}$ as the coefficient sets for $\mathcal{S}^A$, $\mathcal{S}^B$, $\mathcal{V}^A$, $\mathcal{V}^B$, respectively. Each block simultaneously selects the coefficients from $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, and the process is summarized in Tab.1. All the coefficients associated with a new frame are initialized to be zeros except α_1. We initialized $\alpha_1 = C_1$ to satisfy the constraint $\sum_i \alpha_i = C_1$, and α_1 will gradually decrease to be within $[0, C_2]$ as the online optimization proceeds. As a result, the appearance model stresses more on the first frame at the primary stage of tracking, which is reasonable before forming a stable appearance model. We schedule the three blocks as suggested by Bordes et al. [5], which is a simple scheme that considers both the computation time and the progress of the objective function.

5 Implementation

We now explain some important implementation details of our algorithm.

Features. We use intensity histograms and gradient orientation histograms to represent $\phi(\mathbf{x}, \mathbf{y})$. The bounding box region is divided into 5×5 cells, and then the intensity value and gradient orientation in a cell are quantized into 8 bins. Therefore, each cell is represented by a 16 dimensional vector. Besides, for every

Table 1. The three basic blocks for selecting the coefficient pairs to be optimized. Specifically, for $\boldsymbol{\beta}$, we first determine frame j , then select the coefficient pair from the frame.

	UPDATE	**RETRIEVE**	**ADJUST**
α_+	α_t	$\arg\max_{\alpha\in\mathcal{C}^{SA}} g(\alpha)$	$\arg\max_{\alpha\in\mathcal{C}^{VA}} g(\alpha)$
α_-	$\arg\min_{\alpha\in\mathcal{C}^{SA}} g(\alpha)$	$\arg\min_{\alpha\in\mathcal{C}^{SA}} g(\alpha)$	$\arg\min_{\alpha\in\mathcal{C}^{VA}} g(\alpha)$
frame j	t	a random k in $\mathcal{S}^B$	a random k in $\mathcal{V}^B$
$\beta_j^{\mathbf{y}+}$	$\beta_t^{\hat{\mathbf{y}}_t}$	$\arg\max_{\beta_k^{\mathbf{y}}\in\mathcal{C}^{SB}} g(\beta_k^{\mathbf{y}})$	$\arg\max_{\beta_k^{\mathbf{y}}\in\mathcal{C}^{VB}} g(\beta_k^{\mathbf{y}})$
$\beta_j^{\mathbf{y}-}$	$\arg\min_{\beta_t^{\mathbf{y}}\in\mathcal{C}^{SB}} g(\beta_t^{\mathbf{y}})$	$\arg\min_{\beta_k^{\mathbf{y}}\in\mathcal{C}^{SB}} g(\beta_k^{\mathbf{y}})$	$\arg\min_{\beta_k^{\mathbf{y}}\in\mathcal{C}^{VB}} g(\beta_k^{\mathbf{y}})$

neighbouring 2×2 cells, we calculate the histogram sum to represent the appearance of a larger region; for a set of randomly selected 30 cell pairs, we calculate the histogram difference of each pair to capture the inter-cell dependency. All these 16 dim histograms are L_2-normalized within their separate channels, and then they are concatenated together to form a 1136 dim vector. Note that the norm of the feature is made to be a constant. By using integral histogram [1], the features can be computed efficiently.

Searching Strategy. Based on the histogram features, the distribution of score values is usually smooth in the state space $\mathcal{Y}$. Hence, we employ a coarse-to-fine search strategy similar to that presented in [6]. This method iteratively samples the candidates based on SMC [8], which gradually approches the high score region without the need of hand-tuning the motion parameters for different video sequences.

Set Management. As tracking proceeds, the sizes of all the sets $\mathcal{V}^A, \mathcal{V}^B, \mathcal{S}^A, \mathcal{S}^B$ will increase incrementally, making the optimization more and more expensive. Considering efficiency, we keep these sets with fixed size $N_{VA}, N_{VB}, N_{SA}, N_{SB}$, and therefore an appropriate set management is necessary.

1. For $\mathcal{S}^A$, each time we add the feature of the optimal state. When the number of its elements exceed N_{SA}, we condense the set by sampling its elements. Specifically, we reserve the existing support vectors, then uniformly sample half of the rest features, finally combine them to form the new S^A.
2. For $\mathcal{S}^B$, each time we add the features of both target and background samples. We only consider the samples in the neighborhood of the target, hence we produce these samples by sampling around the target state on a polar grid centered on the target, which gives 81 different locations. These samples are produced with same scale as the current target state. Only the features from the last N frames are kept in $\mathcal{S}^B$.
3. We maintain the features with coefficient $\alpha > 0$ in $\mathcal{V}^A$. When $|\mathcal{V}^A| > N_{VA}$, we delete the support vector with smallest α, and transit its coefficient to the one with second smallest α.
4. We maintain features with coefficient $\beta \neq 0$ in $\mathcal{V}^B$. When $\mathcal{V}^B > N_{VB}$, we delete all the support vectors from the oldest frames.

The entire algorithm of our proposed Description-Discrimination Collaborative Tracker (DDCT) is summarized in Alg. 2.

Algorithm 2. Description-Discrimination Collaboration Tracker

Input: $\hat{\mathbf{y}}_{t-1}$, $\mathbf{w}, \mathcal{S}^A, \mathcal{S}^B, \mathcal{V}^A, \mathcal{V}^B$

1. $\hat{\mathbf{y}}_t = \arg\max_{\mathbf{y}} \langle \mathbf{w}, \phi(\mathbf{x}_t, \mathbf{y}) \rangle$ according to Searching strategy
2. manage $\mathcal{S}^A$, $\mathcal{S}^B$ according to Set management
3. UPDATE $\longrightarrow$ elementary step
4. manage $\mathcal{V}^A$, $\mathcal{V}^B$ according to Set management
5. **for** $j = 1$ to n_R **do**
6. RETRIEVE $\longrightarrow$ elementary step
7. manage $\mathcal{V}^A$, $\mathcal{V}^B$ according to Set management
8. **for** $k = 1$ to n_A **do**
9. ADJUST $\longrightarrow$ elementary step
10. **end for**
11. **end for**

Output: $\hat{\mathbf{y}}_t$, $\mathbf{w}, \mathcal{S}^A, \mathcal{S}^B, \mathcal{V}^A, \mathcal{V}^B$

6 Experiments

Datasets and Metric. Experiments are conducted over 30 publicly available video sequences, which include the full MIL dataset [2] (*tiger1*, *tiger2*, *coke*, *cliffbar*, *david*, *dollar*, *face1*, *face2*, *girl*, *surfer*, *sylv*, *twinnings*), the full PROST dataset [21] (*lemming*, *board*, *box*, *liquor*), the full VTD dataset [15] (*animal*, *basketball*, *football*, *skating1*, *skating2*, *singer1*, *singer2*, *soccer*, *shaking*) and other 5 frequently used sequences (*woman*, *bolt*, *car4*, *trellis*, *jump*). The challenges of the data are summarized in Tab. 2. We use two widely accepted evaluation metrics during our experiments: the center location error (CLE) [29] and the Pascal VOC overlap ratio (VOR) [30]. Based on CLE and VOR, we employ the precision plot and success plot to demonstrate the trackers' performance. The definition of the two plots can be found in [27].

Experiment Settings. The proposed Description Discrimination Collaborative Tracker (DDCT) is implemented in MATLAB/C and runs about 12 FPS with a 3.07GHZ CPU. We empirically set the parameters as $C_1 = 8, C_2 = 0.75, C_3 = 0.75$, where C_1 is the sum for coefficients of descriptive support vectors, and C_2, C_3 restrict the influence of a single support vector in the descriptive and the discriminative component, respectively. We fix the set sizes as $N_{VA} = 20, N_{VB} = 50, N_{SA} = 50, N_{SB} = 20 \times 81$. N_{VA}, N_{VB} define the maximum number of support vectors in each component, and N_{SA}, N_{SB} are the sizes of $\mathcal{S}^A$ and $\mathcal{S}^B$. For $\mathcal{S}^B$, we only keep the last $N = 20$ frames and extract features for 81 samples in each frame. The iteration times in Alg. 2 are: $n_A = 12$ and $n_R = 10$. **All the parameters of DDCT are fixed in the experiments**.

6.1 Analysis of the Proposed Method

Component analysis. In order to investigate the properties of the descriptive component and the discriminative component, we construct two trackers using

Table 2. The challenges of experimental sequences

Main Challenges	Sequences
Distractor	*dollar, basketball, liquor, football, bolt*
Rotation	*cliffbar, face2, girl, surfer, board, shaking*
Illumination	*tiger1, tiger2, coke, david, trellis, basketball,sylv, car4 , singer1, singer2, skating1, shaking*
Occlusion	*box, football, coke, face1, face2, girl, skating2, tiger1, tiger2, basketball, lemming, liquor, woman*
Scaling	*cliffbar, david, girl, singer1, singer2, car4, lemming, board, liquor, skating1, skating2, trellis*

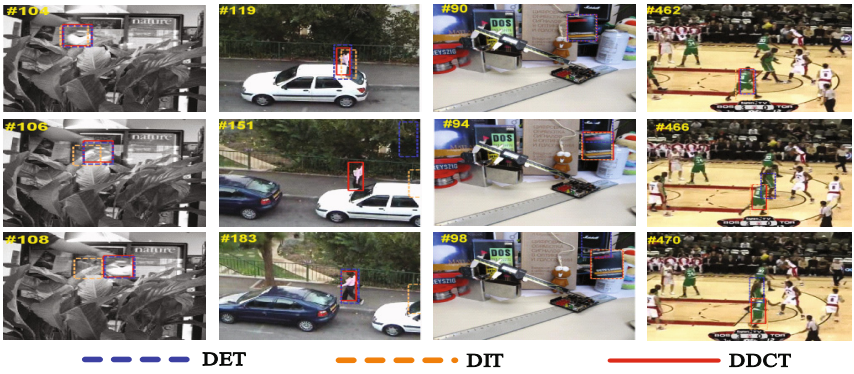
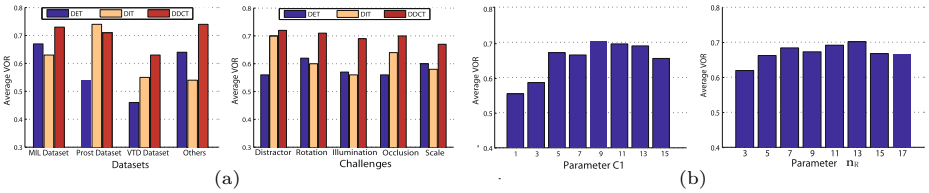


Fig. 2. Qualitative analysis of components. From left to right are *tiger2*, *woman*, *box* and *basketball*.

either the descriptive or the discriminative component, named as DET and DIT, respectively. For DET, we set $C_3 = 0$ excluding the influence from discriminative support vectors. For DIT, we set $C_1 = 3, C_2 = 0$ to preserve the initial object to collaborate with discriminative support vectors. Together with DDCT, we run the three trackers over all sequences. We analyze their average VORs over different datasets and challenges in Fig. 3(a). The overall performance and the detailed quantitative results are provided in Fig. 5 and Tab. 3, respectively.

DDCT significantly outperforms the other two on the overall performance, as shown in Fig. 3, 5 and Tab. 3. The results confirm the effectiveness of Description-Discrimination collaboration, which enables the two components to benefit from each other. DDCT performs most stably over different challenges, although the overall performance fluctuates a bit over different datasets. This demonstrates the robustness of our tracker, and at the same time verifies the degree of difficulty for each individual dataset.

DET performs better than DIT on sequences in the MIL dataset and the additional video sequences as shown in Fig. 3. In these Datasets, the change of the object's appearance is relatively small and mild, so that the major characteristic of objects can be well captured by the descriptive component. Typical examples are sequences *tiger2* and *woman* (Fig. 2). In *tiger2*, the target moves across the leaves, DET tolerates the interruption caused by occlusion, while DIT updates its appearance to the leaves. In *woman*, the pedestrian goes out of the occlusion by the car, and both DET and DIT fail on the abrupt appearance change, but when the target recovers its usual appearance, DET can re-track the object.



(a) (b)

Fig. 3. Quantitative analysis of (a) different components, (b) different parameters over 15 sequences

DIT performs better than DET on the sequences in PROST dataset and VTD dataset as shown in Fig. 3. The object's appearance changes drastically in VTD dataset, and there exists similar objects in PROST dataset. The superiorities of DIT are demonstrated in sequences *box* and *basketball* (Fig. 2). In *box*, DIT can distinguish the target from a similar black box. In *basketball*, the pose of the player frequently changes, and DIT can adapt to the changing appearance rather than drift to a distracter that is similar to its previous appearance.

Parameter analysis. We study the effect of two important trade-off parameters including the parameter C_1 and the iteration number n_R. The evaluations are implemented on randomly selected 15 sequences.

Parameter C_1 reflects the descriptive ability of descriptive support vectors. Fixing other parameters, we observe how C_1 influences the tracking performance as shown in Fig. 3(b). It can be seen that C_1 should be neither too small nor too large. If C_1 is too small, C_2 in Eq.7 can not effectively constrain the coefficients $\boldsymbol{\alpha}$. Therefore the descriptive support vectors tend to overfit the samples at the initial stage, and are reluctant to adapt to newly coming samples (Fig. 4, in black dash). On the contrary, if C_1 is too large, it will drive more samples to become support vectors, and the descriptive support vectors become redundant and increase the risk of incorporating outliers (Fig. 4, in blue dash). We set a moderate C_1 to balance between the two situations, which is flexible to capture distinctive poses without redundancy.

Parameter n_R is related to appearance model updating. It balances the two blocks UPDATE and RETRIEVE as described in Alg. 2. Different from the conventional learning rates that significantly influence the tracking performance, n_R is relative mild. However, we can still observe trade-off phenomena in Fig. 3 (b), a smaller n_R leads to an adaptive tracker where small error accumulate quickly and cause the track to drift, while a larger n_R generates a conservative tracker which may not be able to respond to the changes in the appearance.

Discussion. Through the above analysis, we find that the tracking performance of DDCT is not very sensitive to the parameters C_1, C_2 and C_3. Among them, C_1 easily achieves stable performance in its range (in Fig. 3(b)). C_2 and C_3 balance influence of the two components, even independent trackers DET and DIT perform reasonably well(Fig. 3(a)). The robustness to the parameters lies in the online optimization stage, where the updating the of model is more dependent

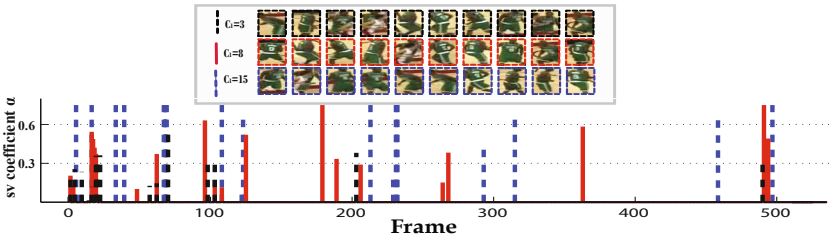


Fig. 4. The comparison of different C_1 configurations on sequence *basketball*. **Top:** The descriptive support vectors selected by different C_1, ranked in descending order of sv coefficients $\boldsymbol{\alpha}$. **Bottom:** The distribution of support vectors over time with different C_1. We display these support vectors when the tracker arrives at # 510.

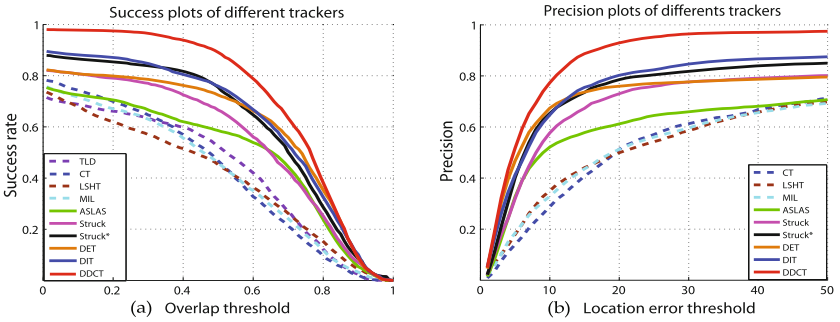


Fig. 5. The success plots and the precision plots for the comparison of different algorithms. We don't include the precision plot for TLD in (b), because some results of it are not available.

on the data – the gradients (in Eq.9). Meanwhile, parameters C_1, C_2 and C_3 play a gentle role in updating the appearance model. Specifically, C_1 controls the global updating property, and C_2 and C_3 prevent over-fitting (in Eq.11).

6.2 Empirical Comparison with Other Trackers

We compare DDCT with six competing trackers, named MIL [2], TLD [14], CT [29], ASLSA [13], LSHT [11] and Struck [10]. The tracking results are obtained by running their publicly available codes with default parameters. For a more intensive comparison with Struck, we equip the original version with the same feature as ours, named Struck*. Together with independent components DET and DIT, we quantitatively compare the 10 trackers on all the 30 sequences. Tab. 3 reports the average VOR and average CLE of each sequence respectively. Fig. 5 demonstrates their success plots and precision plots on all the frames to compare the overall performance. More quantitative and qualitative results are in the supplementary materials.

Table 3. Results in terms of CLE and VOR, the best three results are in **red**, **orange** and **green**

	CT		LSHT		MIL		TLD		ASLSA		Struck		Struck*		DET		DIT		DDCT	
	VOR	CLE	VOR	CLE	VOR.	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE.
tiger1	0.46	21.7	0.09	79.4	0.46	24.3	0.50	- -	0.23	36.4	0.63	7.2	**0.74**	**4.3**	**0.72**	**4.9**	0.65	7.4	**0.74**	**4.2**
tiger2	0.55	10.1	0.14	43.3	0.58	10.2	0.31	- -	0.30	31.6	0.65	6.5	**0.72**	**4.1**	**0.72**	**4.5**	0.24	31.5	**0.72**	**4.2**
coke	0.40	15.5	0.56	7.2	0.39	14.0	0.53	7.8	0.69	4.9	0.60	5.5	**0.77**	**3.0**	**0.74**	**3.8**	0.50	9.3	**0.75**	**3.4**
ciff.	0.43	18.1	0.11	66.3	0.51	**12.3**	0.35	- -	0.17	49.8	0.42	14.8	**0.60**	**6.7**	0.49	19.2	**0.53**	13.0	**0.52**	**12.2**
david	0.57	6.1	0.56	8.5	0.43	20.7	0.57	17.2	0.50	15.9	0.55	11.3	**0.58**	**4.6**	**0.83**	**2.8**	0.51	18.6	**0.73**	**4.2**
dollar	0.62	18.1	**0.87**	**2.4**	0.68	14.7	0.05	157.3	**0.86**	**2.4**	0.65	17.4	**0.86**	**2.6**	0.38	63.0	0.77	5.4	0.84	3.7
face1	0.64	25.4	0.66	28.7	0.63	24.6	0.53	27.5	0.61	43.6	**0.87**	**5.8**	**0.88**	**4.9**	0.83	8.0	**0.85**	**5.3**	0.78	8.5
face2	0.61	17.4	0.68	9.8	0.64	13.8	0.55	11.7	**0.74**	**8.6**	**0.72**	**8.0**	**0.73**	**7.3**	0.64	10.9	0.63	11.1	0.70	9.9
girl	0.54	19.4	0.26	74.5	0.49	26.1	0.51	- -	0.30	48.9	**0.64**	**5.9**	0.60	13.4	**0.74**	**8.7**	0.29	79.5	**0.76**	**7.1**
surf	0.15	28.4	0.19	28.9	0.49	9.8	**0.66**	**3.6**	0.41	23.6	0.58	6.5	0.59	5.9	0.40	40.6	**0.65**	**5.0**	**0.67**	**4.1**
sylv	0.57	12.7	0.63	15.1	0.55	15.7	0.64	12.4	0.71	7.7	0.72	6.9	**0.79**	**4.3**	**0.77**	**5.1**	0.71	8.3	**0.80**	**4.2**
twin.	0.53	12.8	0.46	20.8	0.58	9.7	0.33	- -	0.58	10.8	**0.64**	**7.3**	**0.59**	9.8	**0.61**	**7.5**	0.56	13.1	0.57	**9.2**
lem.	0.43	53.4	0.42	80.6	0.49	56.7	0.11	- -	0.14	200.7	0.52	75.4	**0.79**	**8.1**	0.59	30.3	**0.66**	**13.2**	**0.72**	**7.6**
board.	0.48	48.2	0.70	**16.7**	0.53	41.6	0.35	- -	**0.75**	19.0	0.71	18.9	0.71	18.0	**0.78**	**12.6**	0.73	19.0	**0.78**	**10.8**
box	0.47	**33.1**	0.31	109.4	0.15	111.3	**0.56**	- -	0.40	71.0	0.32	142.9	0.05	176.9	0.06	202.0	**0.73**	**9.6**	**0.72**	**9.5**
liquor	0.20	183.4	0.23	107.5	0.16	167.9	0.59	- -	0.21	238.5	0.61	51.4	0.62	56.4	**0.72**	**31.4**	**0.82**	**4.4**	**0.71**	**30.9**
animal	0.03	250.5	0.05	100.1	0.38	36.8	0.54	- -	0.62	19.7	**0.85**	**3.1**	0.79	6.4	**0.85**	**3.2**	**0.84**	3.6	**0.84**	**3.4**
basket.	0.19	136.1	**0.56**	**22.1**	0.22	97.0	0.06	- -	0.24	103.6	0.03	198.3	0.53	23.3	0.43	82.5	**0.63**	**18.3**	**0.69**	**9.8**
football	0.63	12.5	0.69	8.4	0.59	13.8	0.60	12.5	0.59	10.1	0.61	12.4	**0.83**	**3.2**	0.79	**2.6**	**0.80**	**2.4**	**0.80**	3.8
skate1	0.37	**50.7**	0.18	119.1	0.11	153.6	0.36	- -	**0.50**	**49.3**	0.29	83.9	0.38	64.6	**0.43**	60.8	0.27	61.0	**0.46**	**59.5**
skate2	0.06	120.8	**0.49**	**31.9**	0.11	109.1	0.04	- -	0.21	55.1	0.08	152.8	0.20	129.2	0.29	53.7	**0.58**	**17.4**	**0.55**	**18.9**
singer1	0.34	14.7	0.34	16.5	0.33	17.8	0.40	- -	**0.77**	**3.8**	0.34	15.2	0.34	25.2	0.76	6.8	**0.79**	**5.8**	**0.78**	**6.5**
singer2	0.09	124.9	**0.72**	**11.4**	0.39	33.2	0.02	- -	0.03	180.5	0.02	180.8	**0.69**	**13.3**	0.40	49.2	0.32	82.4	**0.57**	**16.9**
soccer	**0.35**	**52.6**	**0.31**	**40.6**	0.14	98.1	0.07	- -	0.12	131.2	0.18	111.4	0.19	113.0	0.17	164.6	0.21	120.2	**0.42**	**23.3**
shak.	0.65	8.2	0.53	17.7	0.58	14.5	0.51	- -	0.70	10.2	0.23	49.6	0.64	7.7	**0.81**	**4.6**	**0.77**	**6.5**	**0.79**	**6.3**
woman	0.13	112.7	0.14	123.8	0.14	120.0	0.62	- -	0.68	10.5	**0.73**	**3.5**	**0.75**	**4.5**	0.68	14.6	0.16	142.0	**0.75**	**4.0**
bolt	0.52	10.0	0.38	36.6	**0.58**	8.5	0.14	- -	**0.70**	**4.7**	0.17	80.7	0.51	**8.2**	0.05	250.7	0.06	126.0	**0.60**	**8.3**
car4	0.17	93.1	0.26	56.7	0.24	58.0	0.03	- -	**0.86**	**3.3**	0.55	**6.3**	0.25	80.8	**0.57**	16.6	0.37	77.6	**0.74**	**3.3**
trellis	0.32	38.5	0.37	52.5	0.23	56.9	0.33	- -	**0.68**	**8.7**	0.55	9.6	**0.57**	**7.9**	0.34	39.3	0.46	20.3	**0.72**	**8.7**
jump	0.77	3.5	0.23	30.8	0.78	2.8	0.80	- -	0.80	3.5	0.80	**2.2**	0.79	3	**0.86**	**2.1**	**0.86**	**2.2**	**0.84**	**1.8**
average	0.41	51.8	0.40	45.6	0.42	46.5	0.39	- -	0.50	46.9	0.51	43.4	**0.60**	**27.4**	**0.58**	40.1	0.57	**31.3**	**0.70**	**10.3**

Discussion. The results generally reveal the benefits of the proposed description discrimination collaborative tracker, which achieves robustness and high accuracy on diverse sequences. In the experiments, the enhanced Struck* is the major competitor (in Tab.3, Fig.5). During the 60 tests, Struck* achieved 14 bests, 9 seconds and 7 thirds, and DDCT achieved 19 bests, 22 seconds and 11 thirds. Struck* performs well on the sequences with rigid objects and static scenes such as *tiger1*, *tiger2*, *coke* and *face1*, etc. However, the use of obsolete background samples makes Struck* fail in dynamic scenes such as *skate1*, *skate2*, *singer1*, and *trellis*, etc. Compared with Struck*, the proposed DDCT uses historical target samples to be robust to outliers and uses current background samples to be distinctive to distractors, performing well in both static and dynamic scenes. Besides, the different performance between Struck* and Struck also suggests the superiority of employing a high-dimensional histogram feature, where the "high dimension" can better characterize an object and the "histogram" is less sensitive to the spatial alignment. In addition, from Tab. 3, we also observe the complementary property between DET and DIT, which again confirms the rationality of their collaboration.

7 Conclusion

In this paper, we have proposed a novel visual tracking method based on description discrimination collaboration. We integrate descriptive component and discriminative component in a unified max-margin learning framework, and take advantage of their complementary modeling power in both representation and

lifespans. The collaborative model is not vulnerable to outliers like occlusion, and it can track the target with high accuracy.

Furthermore, the adaptation of the model is more data-dependent, which strikes for a balance between past and current appearances. To solve the whole optimization problem, we devise a set of efficient and effective online selection rules, which significantly accelerate the tracking process. Experiments on 30 sequences verified our hypothesis that the collaboration between the descriptive and discriminative components would lead to better tracking performance. It is shown that our proposed tracker generally outperforms existing methods.

Acknowledgement. This work was supported by the National Basic Research Program of China under Grant No. 2012CB316402 and the National Natural Science Foundation of China under Grant No. 91120006.

References

1. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR (2006)
2. Babenko, B., Yang, M.H., Belongie, S.: Visual Tracking with Online Multiple Instance Learning. In: CVPR (2009)
3. Bordes, A., Bottou, L.: The huller: A simple and efficient online SVM. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 505–512. Springer, Heidelberg (2005)
4. Bordes, A., Bottou, L., Gallinari, P., Weston, J.: Solving multiclass support vector machines with larank. In: ICML (2007)
5. Bordes, A., Usunier, N., Bottou, L.: Sequence labelling SVMs trained in one pass. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 146–161. Springer, Heidelberg (2008)
6. Chen, D., Yuan, Z., Wu, Y., Zhang, G., Zheng, N.: Constructing adaptive complex cells for robust visual tracking. In: ICCV (2013)
7. Chen, Y., Zhou, X., Huang, T.S.: One-class svm for learning in image retrieval. In: ICIP, pp. 34–37 (2001)
8. Doucet, A., Johansen, A.M.: A tutorial on particle filtering and smoothing: fifteen years later (2011)
9. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 234–247. Springer, Heidelberg (2008)
10. Hare, S., Saffari, A., Torr, P.H.S.: Struck: Structured output tracking with kernels. In: ICCV (2011)
11. He, S., Yang, Q., Lau, R.W., Wang, J., Yang, M.H.: Visual tracking via locality sensitive histograms. In: CVPR (2013)
12. JamesSteven, S., Ramanan, D.: Self-paced learning for long term tracking. In: CVPR (2013)
13. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. In: CVPR (2012)
14. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE TPAMI 34(7), 1409–1422 (2012)

15. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR (2010)
16. Lampert, C.H., Blaschko, M.B.: Structured prediction by joint kernel support estimation. Machine Learning (2009)
17. Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: ICCV (2009)
18. Pang, Y., Ling, H.: Finding the best from the second bests - inhibiting subjective bias in evaluation of visual tracking algorithms. In: ICCV (2013)
19. Platt, J.C.: Sequential minimal optimization: A fast algorithm for training support vector machines. Tech. rep., Advances in Kernel Methods - Support Vector Learning (1998)
20. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. Int. J. Comput. Vision 77(1-3), 125–141 (2008), http://dx.doi.org/10.1007/s11263-007-0075-7
21. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: PROST Parallel Robust Online Simple Tracking. In: CVPR (2010)
22. Scholkopf, B., Williamson, R.C., Smola, A., Shawe-Taylor, J.: Sv estimation of a distribution's support. In: NIPS (1999)
23. Tax, D.M.J., Duin, R.P.W.: Support vector data description. Mach. Learn. (2004)
24. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. J. Mach. Learn. Res. (2005)
25. Wang, S., Lu, H., Yang, F., Yang, M.H.: Superpixel tracking. In: ICCV (2011)
26. Wen, L., Cai, Z., Lei, Z., Yi, D., Li, S.Z.: Online spatio-temporal structural context learning for visual tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 716–729. Springer, Heidelberg (2012)
27. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: CVPR (2013)
28. Xing, J., Gao, J., Li, B., Hu, W., Yan, S.: Robust object tracking with online multi-lifespan dictionary learning. In: ICCV (2013)
29. Zhang, K., Zhang, L., Yang, M.-H.: Real-time compressive tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012)
30. Zhong, W., Lu, H., Yang, M.H.: Robust object tracking via sparsity-based collaborative model. In: CVPR (2012)