Weighted Part Context Learning for Visual Tracking

Guibo Zhu, Jinqiao Wang*, *Member, IEEE,* Chaoyang Zhao, and Hanqing Lu, *Senior Member, IEEE*

***Abstract*—Context information is widely used in computer vision for tracking arbitrary objects. Most existing works focus on how to distinguish the object of interest from background or how to use keypoint-based supporters as their auxiliary information to assist them in tracking. However, in most cases, how to discover and represent both the intrinsic property inside the object and the surrounding context is still an open problem. In this paper, we propose a unified context learning framework that can effectively capture spatio-temporal relations, prior knowledge and motion consistency to enhance the tracker's performance. The proposed Weighted Part Context Tracker (WPCT) consists of an appearance model, an internal relation model and a context relation model. The appearance model represents the appearances of the object and parts. The internal relation model utilizes the parts inside the object to describe the spatio-temporal structure property directly, while the context relation model takes advantage of the latent intersection between the object and background regions. Then the three models are embedded in a max-margin structured learning framework. Furthermore, prior label distribution is added, which can effectively exploit the spatial prior knowledge for learning the classifier and inferring the object state in the tracking process. Meanwhile, we define online update functions to decide when to update WPCT as well as how to reweight the parts. Extensive experiments and comparisons with the state-of-the-arts demonstrate the effectiveness of the proposed method.**

***Index Terms*—Visual Tracking, Part Context model, Structure Leaning**

I. Introduction

VISUAL tracking is a fundamental problem in computer vision and has wide-ranging applications including activity recognition, surveillance, augmented reality, and human-computer interaction [1]–[5]. For a visual tracking approach, it should be designed to cope with the inevitable appearance changes due to occlusion, rotation, illumination, etc. Recent progresses in object tracking [6]–[15] have yielded a steady increase in performance, but designing a robust approach to track generic objects in presence of occluded and deformable targets is still a major challenge. To overcome these difficulties, numerous models have been designed, most of which focus on building a strong appearance model to encode the variations of the object appearance.

Meanwhile, there is additional information (e.g., context information) which can be exploited instead of using only the object region. Context information has been applied actively in object detection [16], object classification [17], object recognition [18]. Since the spatio-temporal context information is important and necessary for tracking, it has been employed recently in several tracking methods [19]–[28], where it was still underestimated and under-utilized because these methods mainly paid attention to the supporting roles of the external or internal context patches, rather than considered the internal and external relations together in the spatio-temporal space.

Most existing works focus on how to distinguish the tracked object from background (i.e., we treat it as global context) or how to use inter-frame object similarity information (e.g., fragment-based template matching) or key-points supporters in the object (i.e., it can be treated as internal context) as auxiliary information in tracking. However, global context cannot deal with the object deformation problem, while internal context with key points ignore the background context. We observe that the local part context interactions are relatively stable. In other words, when the target appearance changes gradually, the intrinsic property of internal interaction between the parts inside object and context interaction between object and background are relatively stable while the global context provides an effective representation. Therefore, effectively exploiting the rich context information around the tracked object could improve the tracking performance. In this paper, we propose a novel Weighted Part Context Tracker (WPCT). It consists of an appearance model, an internal relation model, a context relation model and online update functions. The appearance model depicts the whole variation of the decision boundary between the object and its surrounding background. The internal relation model formulates the temporal relations of the object itself or the intra-object parts themselves as well as the spatio-temporal relations between the object and intra-object parts to preserve the internal structure. The context relation model constructs the spatio-temporal relations between the intra-object parts as well as the context parts and the temporal relations of the context parts themselves to preserve the external structure. The online update functions not only decide when to update the model, but also consider the importance of different parts based on the prior knowledge of occlusion, motion or spatial distribution. Hence the physical properties and the appearance information are considered in the optimization process through parts and relations.

In summary, our main contributions are four-fold: (1) We first propose a unified context framework which formulates the single object tracking as a part context learning problem; (2) The parts of the intra-object and context region are selected so that we not only pay attention to the appearance of object, but also consider the stable relations among the object, the intra-object parts and the context parts; (3) Prior label distribution is added in the processes of model learning and inference.

This work was supported by 863 Program 2014AA015104, and National Natural Science Foundation of China (61273034 and 61332016).

G. Zhu, J. Wang, C. Zhao and H. Lu are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing,100190, China.
E-mail: {gbzhu, jqwang, chaoyang.zhao, luhq}@nlpr.ia.ac.cn.
* is the corresponding author.

Manuscript received April 19, 2005; revised September 17, 2014.

(4) Online update functions are presented to decide when to update WPCT as well as how to reweight the parts.

A preliminary conference version of this paper can be referred to Zhu *et al.* [29] and [30]. Compared to the prior papers, this study contains (1) a substantial number of additional explanations and analysis, (2) prior label distribution as context information to improve the tracker performance, and (3) online update functions to decide when to update the model and how to reweight the parts, and (4) various additional experiments to investigate the impact of spatio-temporal part context in the tracking process.

II. Related Work

In recent decades, numerous tracking methods have been proposed in literatures. In what follows, we only briefly make a representative selection of recent trackers and categorize them into generative trackers and discriminative approaches roughly.

A. Generative Trackers

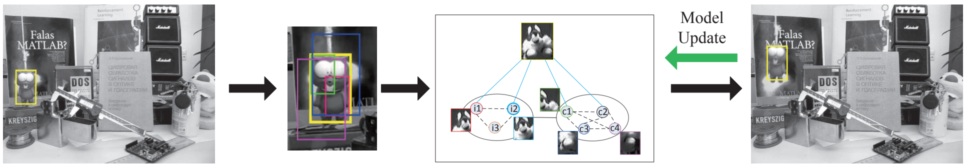
These methods learn an appearance model to represent only the object and search for the most similar image region as the predicted object. Examples of generative approaches are FT [6], IVT [31], L1 [32], VTD [33], MTT [12], ASLA [11] and LSHT [34] . FT [6] represented the target with histogram of local patches, which took fixed spatial structural information of the target itself and handle partial occlusion very well. However, its template is not updated over time and the correlation of target and surrounding is not constructed. IVT [31] incrementally learned a low-dimensional subspace representation, and efficiently adapted to online changes in target appearance, where the lack of spatial information resulted in drift problem. L1 tracker [32] was to represent the candidates sparsely using $\ell 1$ norm minimization. VTD [33] effectively extended the conventional particle filter framework [35] with multiple motion and observation models to account for appearance variation. MTT [12] mined the self-similarities between particles via multi-task learning to improve the tracking performance. ALSA [11] proposed a structural local sparse appearance model to exploit both partial information and spatial information of the target for visual tracking. LSHT [34] adopted a locality sensitive histogram which exploited the spatial weight for every pixel. Generally, generative trackers are robust to the object occlusion and tend to obtain more accurate performance in a small searching region, but sensitive to similar distracters in the surrounding area of the object.

B. Discriminative Trackers

These methods formulate visual object tracking as a classification or structure prediction problem, which seeks the object location that can best separate the object from its background. Examples of discriminative methods are OAB [36], co-training tracking [37], MIL [38], TLD [22], PROST [39], Struck [9], CSK [40], SPOT [13], PT [41], AOGTracker [42] and CNTracker [43]. AOB [36] was adopted to select useful features using boosting for object tracking. Its performance was affected by background clutter, and the tracker can easily drift. TLD [10] decomposed the long-term tracking task into tracking, learning and detection, which utilized the P-N learning to guarantee the online detector's estimated error. PROST [39] merged the template correlation, mean shift optical flow and random forests in a cascade way which can alleviate the drift problem. Struck [9] first introduced the structure output learning for visual tracking which avoided the label prediction problem existing in common online classifiers and got good performance. CSK [40] exploited the circulant structure to get fast tracking through the Fourier analysis, and worked by evaluating a classifier trained using kernel regularised least squares quickly at all sub-windows around the estimated object location and maximising the response. SPOT [13] incorporated spatial constraints between the objects using a pictorial-structures framework [44] and trained a structured SVM online, which was effective for occlusion and deformation. PT [41] modeled the unknown parts of a part-based target model using latent variables into a structure prediction case for tracking. Song [28] explored the most informative features from random projections by maximising entropy energy for object tracking. AOGTracker [42] simultaneously combined with tracking, learning and parsing objects with a hierarchical and compositional And-Or graph (AOG) representation so as to handle occlusion and background clutter. CNTracker [43] builded on correlation filters by introducing colour attributes to achieve superior performance on colour sequences. In general, discriminative trackers are relatively more robust in suppressing background clutters than generative trackers.

C. Most Related Approaches

Many approaches utilize the context information or structure property in some sense. CAT [45] tracked random field around the target instead of the target. The tracker in [21] utilized strong motion coupling constraints to locate the target even when the target was invisible, with the help of some available related context information. However, detecting and matching all of the local features are expensive and the motion of the object is not easily predicted. CXT [23] developed a new context framework based on distracters and supporters. STT [25] proposed a spatio-temporal context method in which temporal context captured the historical appearance information while spatial context model integrated key-points based contributors. Generally, since these trackers in [21], [23], [25] worked with the key points as auxiliary information, the main differences are how to utilize supporters or distracters. Although the introduction of context in these trackers expanded the available information which can be obtained from the scene, it may collapse when motion blur occurs due to the utilization of key-points descriptors. STC [27] utilized the spatiao-temporal context in the Bayesian framework to interpret correlation filter based tracking. CKST [24] proposed a SVM tracker by designing a graph mode-based contextual kernel, where the high-order contexts between the training samples could be discovered in their similarity matrix. PNTracker [22] proposed P-N learning (i.e., P-constraints and N-constraints) to restrict the binary labeling of the unlabeled set for training the long-term detector and extended it to TLD tracker [10]. Different from



Input Frame Part Model Initialization Weighted Part Model Prediction

Fig. 1: The pipeline of the proposed tracker.

them, we not only explore the similarity context information between training samples, but also model the spatio-temporal context between their correlative intra-object parts and context parts in structure learning framework. Moreover, our labeling was that suppose the label of the estimated object state without occlusion was 1, the labels of other samples were set as the overlapping rates between the samples states and the estimated object state so that the labels of these samples fitted a Gaussian distribution for training. In addition, the prior label distribution could be treated as motion prior to assist in inferring the final object state.

Part-based visual models have been investigated by many researchers. In general, there are two types of part-based appearance representation: local patch-based visual representation and global-local coupled visual representation. The foreground shape was modeled as a small number of rectangular blocks which were selected with non-empty intersection with the interior region of the target defined by the contour [46]. An object by a set of local patches with topological structure was represented in [19]. However, these methods only consider the local information. LGT [26] modeled a target's global and local appearance by a coupled-layer with coupled constraints.

An object detection approach with structured SVM [47] was proposed in [48]. Motivated by this success, structured learning was applied to online visual tracking [9], [49]. Inspired by deformable part-based appearance models [50]–[52], Zhang and van der Maaten [53] proposed a structure preserving model and Yao *et al.* [41] presented part-based with latent structural learning for tracking. Although the two approaches paid attention to the parts of the object and their deformation cost, there are still many intrinsic properties in object tracking (e.g., temporal constraints, context information) which have not been considered. Actually they only considered one aspect about the appearance model and the internal relations that our model proposed, but not considered the context relations in our model, e.g., the context parts in the surrounding environment that have motion consensus with the object. In addition, they did not treat all of the auxiliary information as context.

III. Weighted Part Context Tracking

In this section, we will give an overview of the proposed weighted part context learning model in a unified framework shown in Fig. 1. We first introduce the part context formulation and then describe the model training problem with a structured learning approach. After the learning mechanism, we develop an online learning strategy to update the model parameters efficiently. Then how to select the parts and give them weights are discussed. Finally, prior label distribution is adopted to maximally preserve the spatial structure of the object and assist for inferring the final object state.

TABLE I: Important notation and terms

M_A	The appearance model
M_I	The internal relation model
M_C	The context relation model
B_0	The object
$B_1, .., B_K$	The intra-object parts
$B_{K+1}, ..., B_{K+M}$	The context parts
$\Phi_{\cdot}(\cdot)$	The transformed feature representation
$\Phi_{\cdot,\cdot}(\cdot,\cdot)$	The transformed relation feature representation
$\mathbf{w}_{\cdot}$	The weights of the feature represenation
$\mathbf{w}_{\cdot,\cdot}$	The weights of the relation feature

A. Model Definition

Our framework not only models the object with intra-object parts, but also incorporates the interaction between the object and background with context parts. The deformable configurations [44], [54] together with the temporal structure of these parts are also considered in. Please refer to Table I for important terms used throughout the rest of this section.

In Fig. 2, with the object bounding box as the root R, the intra-object parts I are defined as the parts selected inside R, which covers part of the object appearance. The context parts C are selected from the overlapping area between the object and the background. For a target with K intra-object parts and M context parts, the configuration is denoted as $B = (B_0, B_1, ..., B_K, B_{K+1}, ..., B_{K+M})$, where B_0 stands for the target bounding box R, $(B_1, ..., B_K) \in I$ are the K intra-object part boxes, and $(B_{K+1}, ..., B_{K+M}) \in C$ are the M context part boxes. The corresponding features of the root and parts are represented as $X = (\mathbf{x}_0, ..., \mathbf{x}_K, \mathbf{x}_{K+1}, ..., \mathbf{x}_{K+M})$. In a word, our framework models the object with three components:

$$M = M_A + M_I + M_C, \tag{1}$$

where M_A, M_I and M_C are the appearance model, the internal relation model and the context relation model respectively.

For online tracking, an appearance model is essential. It represents the intrinsic property of one object or the discriminative information between the object and background. To better mine the information, we factorize the appearance model M_A into Eq. (1):

$$\begin{aligned} M_A &= A_R + A_I + A_C \\ &= \mathbf{w}_R^T \Phi_R(\mathbf{x}_0) + \sum_{i=1}^{K} \mathbf{w}_I^T \Phi_I(\mathbf{x}_i) + \sum_{i=K+1}^{K+M} \mathbf{w}_C^T \Phi_C(\mathbf{x}_i). \end{aligned} \tag{2}$$

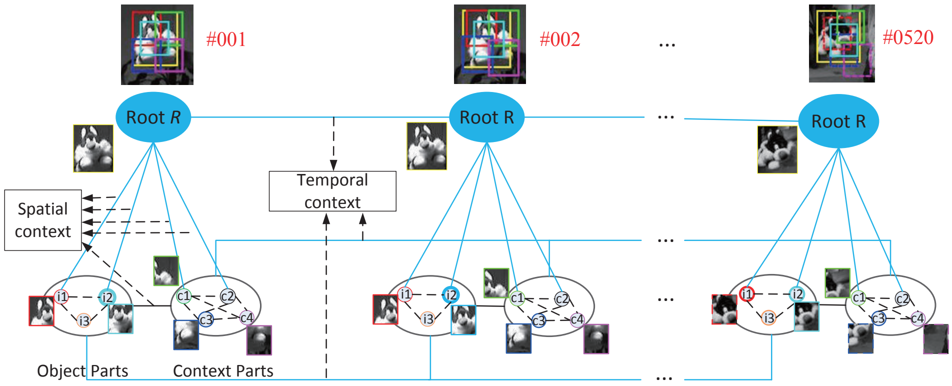


Fig. 2: Illustration of the proposed weighted part context tracking framework using the“sylvester” video. Bold circles denote larger weights, e.g., cyan circle in the first frame, and red circle in the 520^{th} frame.

where A_R, A_I and A_C are the global root appearance model, intra-object parts appearance model and context parts model separately. Φ_R, Φ_I and Φ_C denote the root appearance feature, the intra-object parts appearance feature and the context parts appearance feature. $\mathbf{w}_R$, $\mathbf{w}_I$ and $\mathbf{w}_C$ are the weights of appearance features correspondingly. $\mathbf{x}_i$ is the i^{th} part corresponding to bounding box $B_i = (c_i, r_i, w_i, h_i)$ with center location $B_{i,c} = (c_i, r_i)$, width w_i and height h_i.

In addition, all relatively stable spatio-temporal relations between the object and its corresponding parts frame-to-frame provide rich information for tracking. Therefore we design an internal relation model to formulate the interactions between root and the intra-object parts, which includes the spatial constraints and the temporal constraints between them, as:

$$\begin{aligned} M_I &= S_I + E_R + E_I \\ &= \sum_{i=1}^{K} \mathbf{w}_{R,I}^T \Phi_{R,I}(\mathbf{x}_0, \mathbf{x}_i) + \sum_{t=-H}^{-1} \mathbf{w}_{t,R}^T \Phi(\mathbf{x}_0^t, \mathbf{x}_0) \\ &+ \sum_{t=-H}^{-1} \sum_{i=1}^{K} \mathbf{w}_{t,I}^T \Phi(\mathbf{x}_i^t, \mathbf{x}_i) \end{aligned} \tag{3}$$

where S_I, E_R, and E_I are spatial relation between root and intra-object parts, temporal relation between root and their historical roots, and temporal relation between intra-object parts and their historical information respectively. $\Phi_{R,I}(\mathbf{x}_0, \mathbf{x}_i)$ denotes the spatial interaction function between the root B_0 and intra-object part B_i. $\Phi(\mathbf{x}_0^t, \mathbf{x}_0)$ is the temporal relation function of the bounding box B_0 in the last t^{th} frame and the current frame. H is the upper bound of last frames. Likely, $\Phi(\mathbf{x}_i^t, \mathbf{x}_i)$ is the bounding box B_i's temporal relation function. $\mathbf{w}_{R,I}$, $\mathbf{w}_{t,R}$ and $\mathbf{w}_{t,I}$ are the weights correspondingly. Similar to [51], the spatial interaction between $\mathbf{x}_i$ and $\mathbf{x}_j$ is $\mathbf{f}_c = (c_j - c_i, r_j - r_i)$ and:

$$\Phi(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{f}_c, \mathbf{f}_c^2). \tag{4}$$

Herein, $\mathbf{f}_c$ and $\mathbf{f}_c^2$ can preserve the relative and absolute information between $\mathbf{x}_i$ and $\mathbf{x}_j$. For detail, the temporal relation function $\Phi_{\mathbf{x}_i^t, \mathbf{x}_i}$ can be represented as:

$$\Phi(\mathbf{x}_i^t, \mathbf{x}_i) = exp(-(||B_{i,c}^t - B_{i,c}||^2/\delta^2)) \tag{5}$$

where δ is a constant value.

Except internal relations inside the object, some information in latent intersection area between the object and background is neglected by previous works, such as the partial contour and the object are consensus in motion. To make full use of this information, we formulate the context relation model to express the interactions between root and the context parts, which also includes the spatial and temporal constraints between them. Similar to Eq. (3), we describe the context relation model mathematically as:

$$\begin{aligned} M_C &= S_C + S_{C,I} + E_C \\ &= \sum_{j=1}^{M} \mathbf{w}_{R,C}^T \Phi_{R,C}(\mathbf{x}_0, \mathbf{x}_i) + \sum_{i=1}^{K} \sum_{j=1}^{M} \mathbf{w}_{C,I}^T \Phi_{C,I}(\mathbf{x}_i, \mathbf{x}_j) \\ &+ \sum_{t=-H}^{-1} \sum_{j=1}^{M} \mathbf{w}_{t,C}^T \Phi(\mathbf{x}_j^t, \mathbf{x}_j) \end{aligned} \tag{6}$$

where S_C, $S_{C,I}$ and E_C denote spatial relation between root and context parts, spatial relation between intra-object parts and context parts, and temporal relation between context parts and their historical information. $\Phi_{R,C}(\mathbf{x}_0, \mathbf{x}_j)$ denotes the spatial interaction function between the root B_0 and context part B_j, $\Phi_{C,I}(\mathbf{x}_i, \mathbf{x}_j)$ denotes the spatial interaction function between the intra-object part B_i and the context part B_j. $\Phi(\mathbf{x}_j^t, \mathbf{x}_j)$ denotes the bounding box B_j's temporal relation function. $\mathbf{w}_{R,C}$, $\mathbf{w}_{C,I}$ and $\mathbf{w}_{t,C}$ are the weights corresponding to $\Phi_{R,C}(\mathbf{x}_0, \mathbf{x}_j)$, $\Phi_{C,I}(\mathbf{x}_i, \mathbf{x}_j)$ and $\Phi(\mathbf{x}_j^t, \mathbf{x}_j)$ respectively.

For the linear property, the model of object and its configuration can be simplified as:

$$M = \mathbf{w}^T \Phi(X) \tag{7}$$

where

$$\mathbf{w} = [\mathbf{w}_R^T, \mathbf{w}_I^T, \mathbf{w}_C^T, \mathbf{w}_{R,I}^T, \mathbf{w}_{R,C}^T, \mathbf{w}_{I,C}^T, \mathbf{w}_{t,R}^T, \mathbf{w}_{t,I}^T, \mathbf{w}_{t,C}^T]^T, \tag{8}$$

$$\begin{aligned}\Phi(X) =&[\Phi_R^T(\mathbf{x}_0), \Phi_I^T(\mathbf{x}_i), \Phi_C^T(\mathbf{x}_i), \sum_{i=1}^{K} \Phi_{R,I}^T(\mathbf{x}_0, \mathbf{x}_i),\\ &\sum_{t=-H}^{-1} \Phi^T(\mathbf{x}_0^t, \mathbf{x}_0), \sum_{t=-H}^{-1}\sum_{i=1}^{K} \Phi^T(\mathbf{x}_i^t, \mathbf{x}_i),\\ &\sum_{j=1}^{M} \Phi_{R,C}^T(\mathbf{x}_0, \mathbf{x}_i), \sum_{i=1}^{K}\sum_{j=1}^{M} \Phi_{C,I}^T(\mathbf{x}_i, \mathbf{x}_j),\\ &\sum_{t=-H}^{-1}\sum_{j=1}^{M} \Phi^T(\mathbf{x}_j^t, \mathbf{x}_j)]^T\end{aligned} \tag{9}$$

where $\mathbf{w}$ is the model parameter we need to learn. Given a configuration B in a frame F, there needs a function to measure how well the configuration B matches object model M. We compute the similarity score as follows,

$$S(F, B, M) = S(F, B, M_A) + S(F, B, M_I) + S(F, B, M_C) \tag{10}$$

Eq. (7)-(9) are hard to solve so we need to relax it to be a convex optimization problem. Moreover, to reduce the time complexity of learning the model parameter $\mathbf{w}$, we utilize the tree structure with minimum spanning tree based on the parts' locations inspired by [13].

B. Optimization

In this section, we will describe the optimization of the proposed discriminative model from three aspects: inference, model learning and update strategy.

1) Inference: Although the object appearance varies frequently during the tracking process, there exist stable intrinsic relations between the object and intra-object parts or context parts across continuous frames. Given the definition of M, a model is constructed to constrain the deformation of parts by modeling their temporal and spatial relations with the root. To avoid the problem caused by high-order loopy graph, we not only keep the model to be tree-structured, but also introduce the temporal relations based on the historical information without increasing time complexity. Standard sliding window procedure is used to scan images around the previous object location with a fixed scale to determine the object location. For each scanning window in image F, we first fit the window to structure model M to get the part configuration on it, and then calculate the score of the window according to the inferred configuration by Eq. (1)-(10).

The fitting step aims to find a candidate object's configuration B^* with the highest matching score according to the learned model M. Mathematically, the optimization problem is to find B^* that satisfies:

$$B_0^* = \arg\max_B S(F, B, M) = \arg\max_B \mathbf{w}^T\Phi(X) \tag{11}$$

The score of each part in the model is independent once the root is specified, so that maximizing B_i^* is transformed to find the optimal configuration B_i for part i:

$$B_i^* = \arg\max_{B_i} S(F, B_i, M) = \arg\max_{B_i} \mathbf{w}_i^T\Phi_i(\mathbf{x}_i) \tag{12}$$

where $\Phi(x_i)$ are the related items with x_i in $\Phi(X)$ and $\mathbf{w}_i$ is the weight vector corresponding to the i^{th} part. The complexity of maximizing a single sliding window is high, but benefiting from the generalized distance transform [55], the average complexity in simultaneously optimizing all the sliding windows is of linear complexity with the search radius.

2) Model Learning: Since we adopted the large-margin framework with structure SVM loss to learn our model, we follow the optimization method in structure SVM to solve our problem. Structure SVM was firstly used for visual tracking in [9]. By explicitly allowing the output space to accurately estimate of object position, the struck tracker can avoid the label prediction, which is an intermediate classification step in common tracking-by-detection methods. In addition, like other trackers [8], [9], [53], to enhance its adaptivity and robustness, we need to update the model online. In general, most of online trackers use the tracked object configuration in previous frames as positive examples to update. We argue this method and choose to update the proposed model while the last object configuration satisfies some conditions (e.g., an update threshold or occlusion detection). We apply an adaptive thresholding strategy to select samples to update.

The similarity S measures the compatibility between training pairs, and gives a higher score to well matched ones. By Eq. (7), it can be learned in a large-margin framework from a set of training sample pairs $\{(F, B_1), ..., (F, B_n)\}$ by minimizing the following convex optimization object function with structure SVM loss [47]:

$$\begin{aligned}&\min_{\mathbf{w},\eta\geq 0} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\eta_i\\ &\quad s.t. \forall i, \forall B \neq B_i : \langle \mathbf{w}, \delta\Phi_i(B)\rangle \geq \Delta(B_i, B) - \eta_i\end{aligned} \tag{13}$$

where $\delta\Phi_i(B) = \Phi(F, B_i) - \Phi(F, B)$. This optimization aims to ensure that the value of $S(F, B_i, M) = \langle \mathbf{w}, \Phi(F, B_i)\rangle$ is greater than $S(F, B, M)$ for $B \neq B_i$, by a margin which depends on a loss function Δ. Herein, $\Delta(B_i, B)$ measures dissimilarity between B_i and B, as in [9], [48], [53]:

$$\Delta(B_i, B) = 1 - \frac{B \cap B_i}{B \cup B_i}. \tag{14}$$

where the two bounding boxes B and B_i are both measured in pixels.

For training the structure SVM efficiently, we adopt the cutting plane algorithm [47] to select the most violated constraints to train. The most violated constraint can transform to structure SVM loss ℓ by configuration B:

$$\ell(w; F, B) = \max_B[S(F, B_i, M) - S(F, B, M) + \Delta(B, B_i)] \tag{15}$$

Then we apply online passive-aggressive algorithm [56]to perform the parameter update in the tracking process. Suppose learning the new weight vector $\mathbf{w}_{t+1}$ based on round t sets is treated as the solution to the following constrained optimization problem,

$$\mathbf{w}_{t+1} = \arg_{\mathbf{w}\in\mathbb{R}} \min \frac{1}{2}||\mathbf{w} - \mathbf{w}_t||^2 \quad s.t. \quad \ell(\mathbf{w}; (\mathbf{x}_t, y_t)) = 0. \tag{16}$$

where $\mathbf{w}_{t+1}$ is the projection of $\mathbf{w}_t$ into the half-space of vectors which attain a loss of zero on the current example.

The solution to the optimization problem in Eq. (16) has a simple closed form solution,

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t \quad where \quad \tau_t = \frac{\ell_t}{||\mathbf{x}_t||^2}. \tag{17}$$

where τ_t is the step size of learning. Please refer to [56] for more technical details. Then the parameter updates is calculated as follows,

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \frac{\ell(\mathbf{w}; F, B)}{||d||^2 + 0.5} d. \tag{18}$$

Herein, $d = \nabla_w S(\mathbf{w}; F, \widehat{B}) - \nabla_w S(\mathbf{w}; F, B)$ is gradient of the structured SVM loss, and $\widehat{B} = \arg\max_B (S(\mathbf{w}; F, B) + \Delta(B_i, B)$.

The construction of training set is important to decide whether a classifier is trained well. Inspired by [10], [51], we construct training data consisting of the positive samples and hard negative samples. An object model is trained by positive samples bootstrapped and negative samples randomly sampled around the object in the first frame. Then we collect the incorrectly classified samples whose configure matching scores are close to positive samples and have low overlapping rate as hard negative samples. Since there are many hard negative samples, we randomly discard some negative samples.

3) Prior Label Distribution for Parts: Prior label information can be used as an effective complement for a classifier, such as [57], [58]. To utilize the information better, we further add the prior label distribution for the tracking system. In one word, we not only add the label prior in the training step, but also in the decision step.

In the model learning of section III-B2, we can treat the dissimilarity between B_i and B as the label prior or label context, because it is used for guaranteeing the importance of training samples. It equals to that the higher the overlapping rate between the training sample and the labeled positive sample, the less the loss term is. If we sample every sample surrounding the labeled positive sample, assign the positive sample as one and other samples as the values equaling to the overlapping rate between the samples and positive sample, the label values centered the positive sample center can form one distribution which can be called as label distribution.

In the inference process, we can also treat the motion prior as prior label distribution for assisting the object prediction. That is to say, the higher the probability of the candidate sample which belongs to the positive object is, the closer the candidate sample from the previous object location. We adopt Gaussian distribution as our prior label distribution. In details, the prior label distribution is as follows:

$$p(y|B_{0,c}) = exp(-\frac{1}{2\sigma^2}\mathcal{D}(B_{c,c}, B_{p,c})), \tag{19}$$

where $\mathcal{D}(.,.)$ is the Euclidean distance function, σ is empirically set as $\sqrt{W \times H}$. Here, W, H are the width and height of the root object, respectively.

C. Online Update

1) Update function: The goal of updating the WPCT online is to account for both the structural and appearance variations of the target object, as well as handle hard negatives (distracters) in the background. If the object is occluded, the model doesn't need to be updated. But if the object is self-occluded (e.g., rotation) or appearance changes due to illumination, the model updating is necessary. However, evaluating whether and when the appearance changes (e.g., occlusion) is a difficult problem. Therefore, most of the tracking algorithms update the appearance model every frame.

Like [53], we only update the weight w_i corresponding to part bounding box B_i when the exponentiated score for that object exceeds a given threshold to avoid erroneous update. Different from [53], the threshold is adaptive and generated by an update criteria function for leveraging the adaptivity and stability of the object model. The update criteria function is as follows.

$$G(x) = \begin{cases} 1, & if \quad x >= T + \tau \\ 0.95 + rand(abs(x - T)), & if \quad |x - T| < \tau \\ exp(-\kappa(T - \tau - x)), & if \quad x < T - \tau \end{cases} \tag{20}$$

where T is the prior threshold, τ is the bandwidth for relaxing the limitation, and κ is the hyper-parameter to keep some small probability to update. In our paper, $T = 0.3$, $\tau = 0.05$, $\kappa = 30$. In particular, we only update the w_i when $G(x) >= rand(1)$, where $rand(1)$ is to generate uniform random variables ranging 0 to 1.

2) Reweighting the Parts: In the tracking process, parts are not contributed equally. For example, while a part is occluded, it should be less important than other parts.

Generally, three aspects are considered: (a) the similarity to the trained part model, O_i; (b) the degree of motion consistency between the tracked object and the historical parts, M_i; (c) the spatial distance R_i, i.e., the more important the part is, the closer it is to the tracked object center. Since the part is close to the center of the tracked object, it should be more reliable. With the three factors, we define the part weighting function as follows,

$$\begin{aligned} w_i &= O_i \times M_i \times R_i \\ &= sign(S_i - S_T)\frac{P_i}{\sum_i P_i} exp(-\frac{\mathcal{D}(B_{0,c}, B_{i,c})}{\sigma^2}), \end{aligned} \tag{21}$$

$$sign(x) = \begin{cases} 1, if x >= 0 \\ 0, if x < 0 \end{cases} \tag{22}$$

where S_i is the score of each part's classifier, S_T is the occluding threshold, P_i is the cumulative motion consistency between the i^{th} part and the tracked object which is computed by the cosine similarity based on the motion vector of their center locations between the adjacent frames. $\mathcal{D}(.,.)$ is the Euclidean distance between the center of object $B_{0,c}$ and the center of the i^{th} part $B_{i,c}$, σ is the hyper-parameter for restraining the strength of parts' importance on the spatial distribution. Empirically, we set the $\sigma = \sqrt{W \times H}$, W, H are the width and height of the root object.

D. Part Initialization

How to select proper parts is very important for part models, especially in the online learning applications. Inspired by the deformable part-based models [59], we will search for these parts that cover high-energy regions of the root filter. Here, the "energy" of a region is defined by the norm of the positive weights in a subwindow. We will introduce how to select in the following section.

1) Learning Pixel Weights: To select discriminative parts, we need to evaluate the confidence for each pixel in an object bounding box. Considering the characteristic of exemplar-SVM [60], we use it to choose discriminative parts. The idea of Exemplar-SVM is that, for each positive sample (also regarded as an exemplar), each Exemplar-SVM is trained by the corresponding set of samples, where there is only one positive sample and the rest are all negative.

For each sample, we extract the HOG [51], [61] features using Piotr Dollár's toolbox [62] where the size of cell is set as 4. Then we concatenate them to one vector to present one positive sample (exemplar) $\mathbf{x}_E$ in the first frame or negative samples. We randomly sample negative windows as the negative samples, N_E, with the regions $r \geq 50$ (r is the distance between the centers of them and the positive sample). Thus, the object function of Exemplar-SVM $f_E(\mathbf{x})$ can be described as:

$$f_E(\mathbf{x}) = \mathbf{w}_E^T\mathbf{x} + b_E, \tag{23}$$

The weight $\mathbf{w}_E$ and the offset b_E can be solved by optimizing the following convex objective function,

$$\Omega(\mathbf{w}, b) = ||\mathbf{w}||^2 + C_1 h(\mathbf{w}^T\mathbf{x}_E + b) + C_2 \sum_{\mathbf{x}\in N_E} h(-\mathbf{w}^T\mathbf{x} - b), \tag{24}$$

where the hinge loss function is used in $h(x) = max(0, 1-x)$, C_1 and C_2 are both regularization parameters. For simplicity, we adopted LibSVM [63], where $C_1 = 50$, $C_2 = 1$ and $N_E = 50$. The training processes of exemplar-SVMs for the intra-object and the context parts are same except the training samples. With Eq. (24), we can obtain the confidence value for each pixel.

2) Discriminative part selection: Based on Exemplar-SVM, we get the optimized value $\mathbf{w}_E$, and utilize the correspondence between the weight $\mathbf{w}_E$ and the pixel location in the annotated bounding box to mine the discriminative parts, which include intra-object parts and context parts.

Intra-object Parts: We initialize the intra-object part B_i at the location in which the weights of the initial object Exemplar-SVM $\mathbf{w}_E$ are large and positive, because these correspond to features that are highly indicative of object presence. Mathematically, we denote the object part B_i as:

$$B_i = \underset{B_i \subset B}{\operatorname{argmax}} \sum_{(x_i, y_i)\in B_i} max(0, \mathbf{w}_{E(x_i, y_i)}) \tag{25}$$

where B denotes the bounding box of the object and (x_i, y_i) represents a pixel location. We fix the number of parts in advance, setting it to 2; we fix the width and height of the part bounding box to 60% of the object bounding box's size empirically. Once a part is placed, the weights of the covered pixels are set to zero, and we look for the next part based on Eq. (25), until 2 parts are chosen.

Context Parts: The procedure to choose discriminative parts in context region is similar with intra-object. We define the context part C_j as:

$$C_j = \underset{C_j \subset C}{\operatorname{argmax}} \sum_{(x_j, y_j)\in C_j} max(0, \mathbf{w}_{E(x_j, y_j)}) \tag{26}$$

where C represents the bounding box of the context. Different from intra-object parts, the size of context parts is set to 75% of the whole context region, and the context region size is 0.618 times larger than the object.

IV. Experiments

To evaluate the performance of the proposed approach, extensive experiments are performed with public dataset. Firstly, we perform a comprehensive evaluation of part initialization for visual tracking. Secondly, we evaluate the proposed update scheme for parts. Thirdly, we evaluate our prior label distribution for visual tracking. Fourthly, we evaluate the importance of the temporal information. Finally, we provide both quantitative and attribute-based comparisons with state-of-the-art trackers.

A. Experimental Setup

Our method is implemented with Matlab 2013b. The experiments are performed on an Intel core i5 3.1 GHz CPU with 20 GB RAM. The proposed WPCT tracker ran at 10 FPS in average for all sequences, which was sped up compared with our previous tracker PCT tracker. We initialize the parts in first frame where the part scale is 60% times of the object, the numbers of in-parts and context parts are both set to 2. We test our tracker with fixed parameters in a public benchmark including 50 video sequences [64]. The dataset is collected from many previous works, so we can prevent the training process from the danger of overfitting to a small subset. The sequences used in our experiments pose challenging situations such as heavy occlusions, deformation, out-of-view, motion blur, illumination changes, scale variation, in-plane and out-of-plane rotations, background clutter and low resolution.

Evaluation Methodology: To validate the performance of our proposed approach, we follow the protocol used in [64]. The results are presented using three evaluation metrics based on [38], [64]: distance precision (DP), and overlap precision (OP). DP is the relative number of frames in the sequence where the center location error is smaller than a certain threshold. We use the same parameter as [38], [64] and report DP values at a threshold of 20 pixels. OP is defined as the percentage of frames where the bounding box overlap exceeds a threshold $t \in [0, 1]$ between the identified bounding box and the ground-truth bounding box. If the overlap ratio of bounding boxes exceeds 0.5, it is considered to be successful in tracking for each frame and we can call OP as correct detection rate (CDR). The results are summarized over all 50 sequences with 51 objects. We present them with precision and success plots [64].

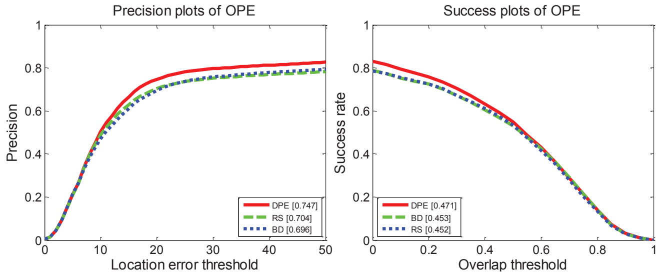


Fig. 3: Plots of overall performance comparison for the benchmark [64] with different part selection strategies.

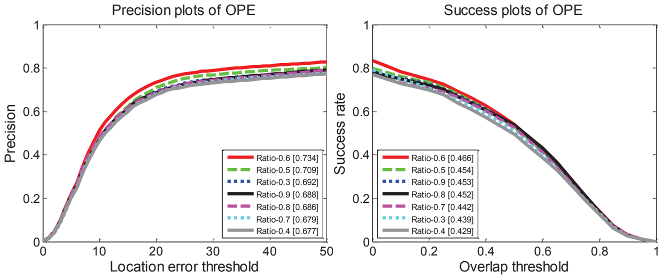


Fig. 4: Plots of overall performance comparison with different ratios between the object size and the context region.

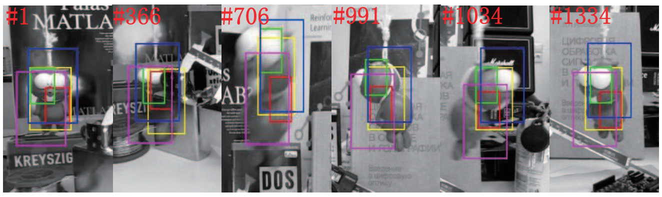


Fig. 5: Part visualization in the tracking process. The yellow rectangle represents the bounding box of object and two smaller rectangles denote the bounding box of intra-object parts, and the larger rectangles denote the bounding box of context parts.

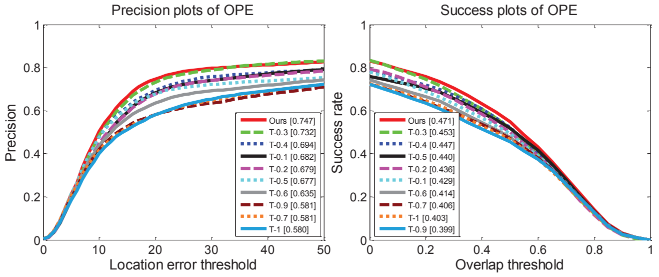


Fig. 6: lots of overall performance comparison with different update thresholds.

B. Part Initialization

We can adopt the examplar-SVM to choose the representative and discriminative parts based on the first frame. To evaluate the effectiveness of the part selection strategy in the proposed method, we compared with different part selection strategies in Fig. 3.

Random Selection (RS) We randomly choose the intra-object and context parts with the same size in the tracked object and the context region respectively.

Block Division (RD) We divide the object into four overlapping parts in the range of the target object equally.

Discriminative Parts with Examplar-SVM (DPE) We choose the discriminative parts as described in the section III-D.

We choose four parts besides the tracked object based on SPOT [13] and our algorithm. Here the part number is predefined empirically. Choosing small number of parts can hardly utilize enough context information while more parts may increase the computational complexity. It is necessary to pay attention on that all the strategies will only adopted two parts (the tracked object and context region part) while the tracked object size is too small so as not to choose appropriate parts. To get one relative confident part size of the context region, we test it in the benchmark dataset. The context region is set based on the object size, i.e., ζ is the ratio between the object size and the size of the context region. Different ζ results to different performance are shown in Fig. 4. Using the sequence Lemming for example, the part visualization in the tracking process is shown in Fig. 5.

C. Online Update Scheme

1) Update Function: In order to evaluate the performance of our update strategy, we conduct the experiments to compare different update thresholds and our strategy. As shown in Fig. 6, we can see that the update threshold affects the performance of our trackers heavily because it determines the learning rate and the leverage between adaptivity and stability of the tracker. As shown in Fig. 6, our update function can get better performance because it not only considers the prior knowledge, but also takes the random perturbation into consideration.

2) Part Weights: To investigate the importance of part reweighting on our results, we next perform a set of experiments against different part weighting methods. To achieve this we modify our tracking framework such that the parts are reweighted by different strategies. They are the strategies of Occluded, Motion, Spatial, and Integration. In more details, Occluded is only using the occlusion function for validating, Motion is only using the motion information for decision, Spatial only endows the weights using spatial distance between the object and the parts, and Integration uses the three strategies simultaneously. As shown in Fig. 7, the method of Integration is worse than the Spatial method mainly because of the decision whether the object or parts are occluded or move consistently may be inaccurate. We can see from these results that overall the median precision and correct detection ratio for Spatial strategy are better than other methods, which demonstrate that the Spatial reweighting framework we use is able to produce gains in accuracy over other approaches.

D. Evaluation of Prior Label Distribution

We evaluate the proposed tracker with and without prior label distribution in Fig. 8. From Fig. 8, we can see that prior label distribution has led to a significant improvement in the performance of the tracker.

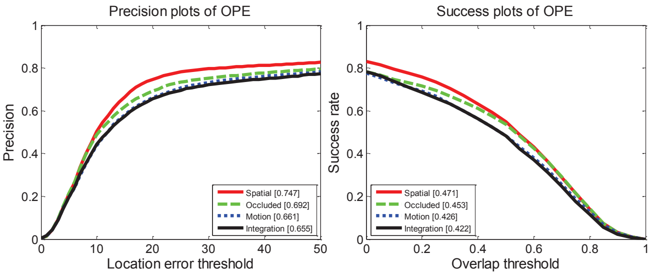


Fig. 7: Plots of overall performance comparison with different part weight strategies.

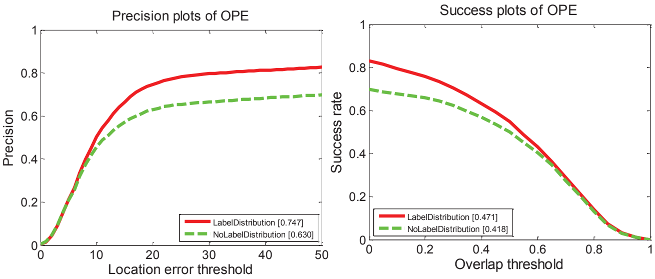


Fig. 8: Plots of overall performance whether adding prior label distribution.

E. Evaluation of Temporal Information

We evaluate the proposed tracker with and without the temporal information in Fig. 9. From Fig. 9, we can see that the temporal information has led to a significant improvement in the performance of the tracker.

F. Comparison of Different Context Part Models

To show the role of different context in object tracking, we compared the following four kinds of models: appearance model, internal context part model, external context part model and weighted part context tracker (WPCT). Appearance model is only modeling the object appearance. Internal context part model is only modeling the appearance and relations of the object and intra-object parts. External context part model only models the appearance and relations of the object and context parts. WPCT models all the appearance and relations of the object, intra-object parts and context parts. The performance of different parts in our model is shown in Fig. 10. From Fig. 10, we can see that adding the internal and external context part model to the appearance model enhances the robustness of the tracker and improves the performance respectively. Although they don't have significant improvement in the performance independently, they jointly contribute the performance. The reason is that both of them preserve some locality structure information and highly complementary each other.

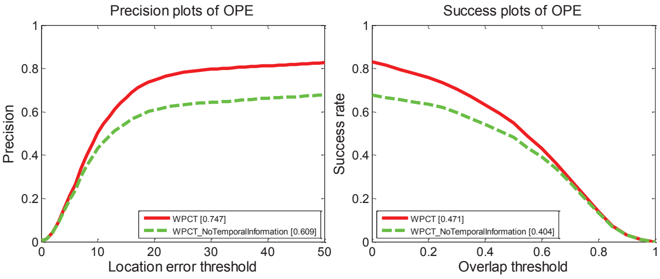


Fig. 9: Plots of overall performance whether adding temporal information.

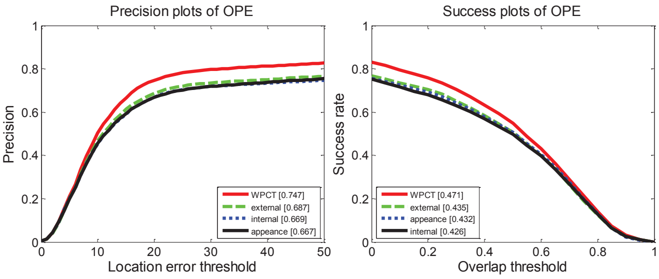


Fig. 10: Plots of overall performance comparison with different context part models

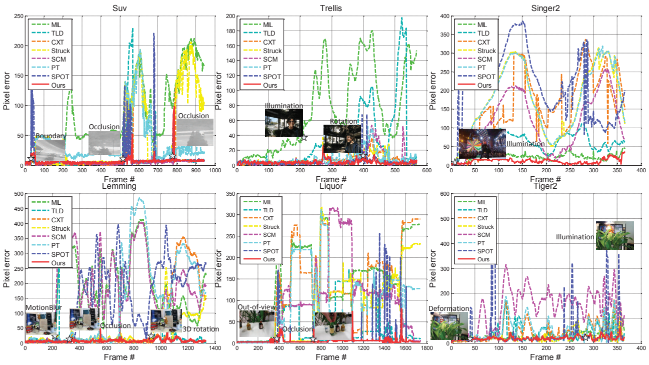


Fig. 11: Comparisons on the center distance error per frame.

G. Fine-grained Evaluation

To evaluate WPCT in more close view, we show six examples of the center distance error per frame in Fig. 11 with some part-based (e.g., PartTracker (PT) [41], structure preserving tracker (SPOT) [53]), context-based (e.g., Context-Tracker (CXT) [23]) or label-based trackers(e.g., MIL [8], Struck [9]).Their source codes or binary codes are provided by the authors and the parameters are tuned finely. All algorithms are compared in terms of the same initial positions in the first frame in [64]. Fig. 11 shows that our method can handle illumination, occlusion and rotation well.

As illustrated in Fig. 11, our tracker outperforms the structure SVM based trackers such as Struck [9], PT [41] and SPOT [53], because most of the sequences for more context information are used in our tracker. Fig. 11 shows the center location error per frame with the compared trackers and some trackers lose the target in several key frames. In general, the robustness of WPCT lies in the context parts with spatial and temporal compositional structures which are locality-preserving and discriminatively trained online to account for the variations.

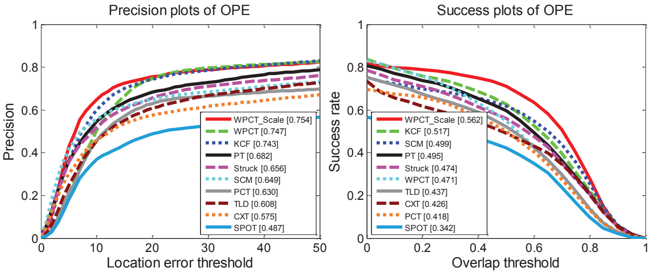


Fig. 12: Plots of overall performance comparison for the 50 videos in the benchmark [64]. The proposed methods WPCT and WPCT_Scale obtain the top-2 and top-1 performance in precision plot (left) and the top-5 and top-1 performance in success plot (right).

H. Comparison with State-of-the-art Trackers

We compare our method with 9 different state-of-the-art trackers. The trackers used for comparison are: SPOT [53], TLD [10], CXT [23], Struck [9], SCM [65], KCF [66], PT [41] and PCT [29] are shown in Fig. 12. Their source codes or binary codes are provided by the authors and the parameters are tuned finely. All algorithms are compared in terms of the same initial positions in the first frame in [64]. They are also provided with the benchmark evaluation [64] except KCF. Here, KCF uses HOG feature and the gaussian kernel which gets the best performance in [66]. For handling scale variation, we introduced scale estimation into the proposed tracker WPCT, denoted as WPCT_Scale. The scale estimation was using nearest neighbor search in multiple scale spaces based on the estimated object center of WPCT.

Fig. 12 shows precision and success plots which contains the mean distance and overlap precision over all the 50 sequences. The values in the legend are the mean precision score and AUC, respectively. Our approaches PCT, WPCT, and WPCT_Scale both improve the baseline SPOT tracker with a relative reduction in accuracy. Specifically, our PCT tracker improves the distance precision rate of the baseline method SPOT from 48.7% to $\mathbf{63.0\%}$ and WPCT boosts the PCT tracker with a gain of $\mathbf{11.7\%}$, and then WPCT_Scale approximates WPCT. In addition, our PCT, WPCT and W-PCT_Scale trackers improve the success rate of their baseline methods from 34.2% to $\mathbf{41.8\%}$ and from 41.8% to $\mathbf{47.1\%}$, and then from 47.1% to $\mathbf{56.2\%}$, respectively. Struck, which has shown to obtain the best performance in a recent evaluation until year 2013 [64]. In [66], the performance of KCF is better than Struck in precision of predicting the object state. Shown in Fig. 12, our trackers are better than the other trackers and achieves a significant gain in precision plots and success plots.

Attribute-based Evaluation: There are several factors which can affect the performance of a visual tracker. In the recent benchmark evaluation [64], the sequences are annotated with 11 different attributes, which are named as: occlusion, deformation, illumination variation, fast motion, motion blur, out-of-plane rotation, scale variation, background clutter, out-of-view, low resolution and in-plane rotation. We perform a comparison with other methods. Fig. 13 and Fig. 14 show example precision plots and success plots of different attributes.

TABLE II: Time cost distribution of the proposed approach. There are four parts for each object in the experiments.

Sequence	*Jogging-1*	*Sylvester*
Image Size	352×288	320×240
Object Size	25×101	61×51
Holistic Feature Extraction	14ms	12ms
Part Classifier Update	2ms	1ms
Part Pictorial Structure	11ms	11ms
Part Prediction	14ms	12ms
Total Time	149ms	132ms

As shown in Fig. 13, WPCT provides superior results compared to existing methods in 5 of 11 attributes, which includes illumination, out-of-plane rotation, motion blur, occlusion, low resolution and so on, mainly because of the locality-preserving structure and context information. That is because the passive-aggressive algorithm may not less effective than KCF in modeling rotation and deformation variation. Specially, although low resolution cases are lack of gradient information, our tracker can still get a reasonable results since the context region can provide rich complementary information. After introducing the scale estimation, the proposed tracker WPC-T_Scale also gets good performance in most of attributes shown in Fig. 14.

I. Time complexity Analysis

The extraction of HOG features and the computation of the appearance score per object with pictorial structures are the main computational costs to run our tracker. And the optimization process for tracking is very fast, i.e., only takes a few milliseconds. The number of parts (i.e., in $|V|$) affects the computational complexity linearly. To accelerate the approach, we make some attempts to speed up the tracker. Firstly, we adopted fast HOG extraction method using Dollár's toolbox [62], which is almost 4 times faster than the implementation by Felzenszwalb *et al.* [51]. Secondly, the model update is fast because only a positive sample and a hard negative sample are used for online passive-aggressive algorithm. Thirdly, minimum spanning tree can reduce the complexity of pictorial structures. Using the sequences *Jogging-1* and *Sylvester* as an example, we give the time cost in Table II.

V. Conclusion

We presented a unified context framework for simultaneously tracking and learning with spatial and temporal structure context inference. The proposed tracker was robust to certain conditions of occlusion, illumination and out-of-view because of exploring different context information. The proposed update strategy alleviated the drifting problem caused by update in some extent. Additionally, the prior label distribution for the inference process of the tracker provided a significant promotion for the performance of the tracker. Experiments on challenging video sequences showed that the proposed method performed better than several state-of-the-art approaches.

References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *CSUR*, vol. 38, no. 4, p. 13, 2006.

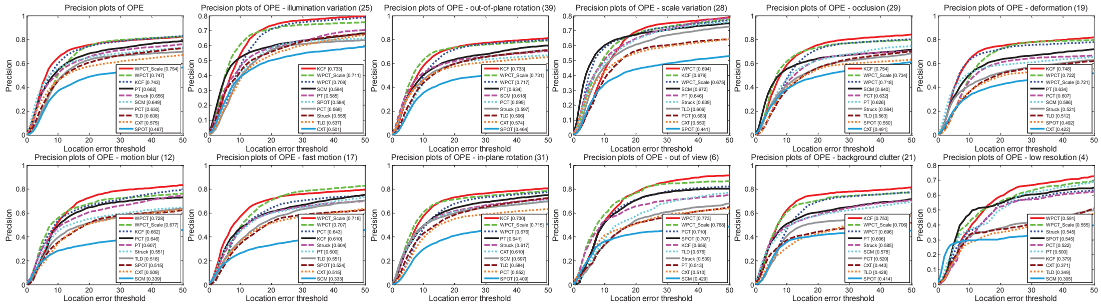


Fig. 13: Precision plots of different attributes (best-viewed on high-resolution display). The valued appearing in the title denotes the number of videos associated with the respective attribute. The proposed methods in this paper perform favorably against state-of-the-art algorithms.

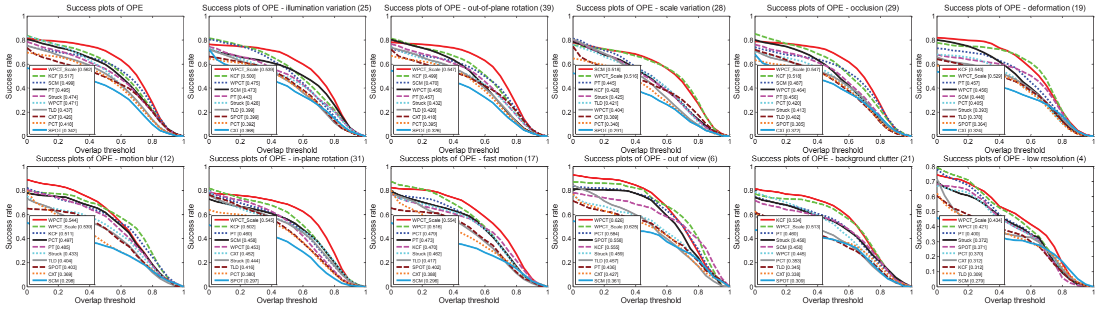


Fig. 14: Precision plots of different attributes [64]. The proposed methods (WPCT and WPCT_Scale) obtain better or comparable performance in all the subsets.

[2] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, 2011.

[3] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *TPAMI*, vol. 36, no. 7, pp. 1442 – 1468, 2013.

[4] J. Wang, W. Fu, J. Liu, and H. Lu, "Spatiotemporal group context for pedestrian counting," *TCSVT*, vol. 24, no. 9, pp. 1620–1630, 2014.

[5] J. Wang, W. Fu, H. Lu, and S. Ma, "Bilayer sparse topic model for scene analysis in imbalanced surveillance videos," *TIP*, vol. 23, no. 12, pp. 5198–5208, 2014.

[6] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *CVPR*, vol. 1. IEEE, 2006, pp. 798–805.

[7] J. Wang, L. Duan, Z. Li, J. Liu, H. Lu, and J. Jin, "A robust method for tv logo tracking in video streams," in *ICME*. IEEE, 2006, pp. 1041–1044.

[8] B. Babenko, M. H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *CVPR*. IEEE, 2009, pp. 983–990.

[9] S. Hare, A. Saffari, and P. Torr, "Struck: Structured output tracking with kernels," in *ICCV*. IEEE, 2011, pp. 263–270.

[10] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *TPAMI*, vol. 34, no. 7, pp. 1409–1422, 2012.

[11] X. Jia, H. Lu, and M. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *CVPR*. IEEE, 2012, pp. 1822–1829.

[12] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *CVPR*. IEEE, 2012, pp. 2042–2049.

[13] L. Zhang and L. van der Maaten, "Preserving structure in model-free tracking," *TPAMI*, vol. 36, no. 4, pp. 756–769, 2014.

[14] D. Wang, H. Lu, and M. Yang, "Online object tracking with sparse prototypes," *TIP*, vol. 22, no. 1, pp. 314–325, 2013.

[15] D. Wang, H. Lu, and M. Yang, "Least soft-threshold squares tracking," in *CVPR*. IEEE, 2013, pp. 2371–2378.

[16] S. Divvala, D. H. Hoiem, J. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *CVPR*. IEEE, 2009, pp. 1271–1278.

[17] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin markov networks," in *CVPR*. IEEE, 2009, pp. 975–982.

[18] C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative models for multi-class object layout," *IJCV*, vol. 95, no. 1, pp. 1–12, 2011.

[19] K. Junseok and L. K. Mu, "Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling," in *CVPR*. IEEE, 2009, pp. 1208–1215.

[20] W. Chang, C. Chen, and Y. Hung, "Tracking by parts: A bayesian approach with component collaboration," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 2, pp. 375–388, 2009.

[21] H. Grabner, J. Matas, L. V. Gool, and P. Cattin, "Tracking the invisible: Learning where the object might be," in *CVPR*. IEEE, 2010, pp. 1285–1292.

[22] Z. Kalal, J. Matas, and K. Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *CVPR*. IEEE, 2010, pp. 49–56.

[23] T. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *CVPR*. IEEE, 2011, pp. 1177–1184.

[24] X. Li, A. Dick, H. Wang, C. Shen, and A. van den Hengel, "Graph mode-based contextual kernels for robust svm tracking," in *ICCV*. IEEE, 2011, pp. 1156–1163.

[25] L. Wen, Z. Cai, Z. Lei, D. Yi, and S. Li, "Online spatio-temporal structural context learning for visual tracking," in *ECCV*. Springer, 2012, pp. 716–729.

[26] L. Cehovin, M. Kristan, and A. Leonardis, "Robust visual tracking using an adaptive coupled-layer visual model," *TPAMI*, vol. 35, no. 4, pp. 941–953, 2013.

[27] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *ECCV*. Springer, 2014, pp. 127–141.

[28] H. Song, "Robust visual tracking via online informative feature selection," *Electronics Letters*, vol. 50, no. 25, pp. 1931–1933, 2014.

[29] G. Zhu, J. Wang, C. Zhao, and H. Lu, "Part context learning for visual tracking," in *BMVC*. BMVA Press, 2014.

[30] G. Zhu, J. Wang, and H. Lu, "Object tracking with part-based discriminative context models," in *ICIP 2014*. IEEE, 2014, pp. 4932–4936.

[31] D. Ross, J. Lim, R. Lin, and M. H. Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1-3, pp. 125–141, 2008.

[32] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded

efficient $\ell1$ tracker with occlusion detection," in *CVPR*. IEEE, 2011, pp. 1257–1264.
[33] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *CVPR*. IEEE, 2010, pp. 1269–1276.
[34] S. He, Q. Yang, R. W. Lau, J. Wang, and M. Yang, "Visual tracking via locality sensitive histograms," in *CVPR*. IEEE, 2013, pp. 2427–2434.
[35] M. Isard and A. Blake, "Condensation ł conditional density propagation for visual tracking," *IJCV*, vol. 29, no. 1, pp. 5–28, 1998.
[36] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *BMVC*, 2006, pp. 47–56.
[37] R. Liu, J. Cheng, and H. Lu, "A robust boosting tracker with minimum error bound in a co-training framework," in *ICCV*. IEEE, 2009, pp. 1459–1466.
[38] B. Babenko, M. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *TPAMI*, vol. 33, no. 8, pp. 1619–1632, 2011.
[39] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "Prost: Parallel robust online simple tracking," in *CVPR*. IEEE, 2010, pp. 723–730.
[40] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *ECCV*. Springer, 2012, pp. 702–715.
[41] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel, "Part-based visual tracking with online latent structural learning," in *CVPR*, 2013.
[42] Y. Lu, T. Wu, and S. Zhu, "Online object tracking, learning, and parsing with and-or graphs," in *CVPR*. IEEE, 2014, pp. 3462–3469.
[43] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *CVPR*. IEEE, 2014.
[44] M. Fischler and R. Elschlager, "The representation and matching of pictorial structures," *TC*, vol. 22, no. 1, pp. 67–92, 1973.
[45] M. Yang, Y. Wu, and G. Hua, "Context-aware visual tracking," *TPAMI*, vol. 31, no. 7, pp. 1195–1209, 2009.
[46] S. Shahed Nejhum, J. Ho, and M. Yang, "Visual tracking with histograms and articulating blocks," in *CVPR*. IEEE, 2008, pp. 1–8.
[47] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer, "Large margin methods for structured and interdependent output variables." *JMLR*, vol. 6, no. 9, 2005.
[48] M. B. Blaschko and C. Lampert, "Learning to localize objects with structured output regression," in *ECCV*. Springer, 2008, pp. 2–15.
[49] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel, "Robust tracking with weighted online structured learning," in *ECCV*. Springer, 2012, pp. 158–172.
[50] Y. Amit and A. Trouvé, "Pop: Patchwork of parts models for object recognition," *IJCV*, vol. 75, no. 2, pp. 267–282, 2007.
[51] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *TPAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.
[52] L. Zhu, Y. Chen, A. Yuille, and W. Freeman, "Latent hierarchical structural learning for object detection," in *CVPR*. IEEE, 2010, pp. 1062–1069.
[53] L. Zhang and L. van der Maaten, "Structure preserving object tracking," in *CVPR*. IEEE, 2013, pp. 1838–1845.
[54] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *IJCV*, vol. 61, no. 1, pp. 55–79, 2005.
[55] P. Felzenszwalb and D. Huttenlocher, "Distance transforms of sampled functions," *Tech. Rep.*, 2004.
[56] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *JMLR*, vol. 7, pp. 551–585, 2006.
[57] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *ICML*. ACM, 2004, p. 104.
[58] X. Geng and R. Ji, "Label distribution learning," in *ICDMW*. IEEE, 2013, pp. 377–383.
[59] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *TPAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.
[60] T. Malisiewicz, A. Gupta, and A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *ICCV*. IEEE, 2011, pp. 89–96.
[61] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.
[62] P. Dollár, "Piotr's Computer Vision Matlab Toolbox (PMT)," http://vision.ucsd.edu/ pdollar/toolbox/doc/index.html.
[63] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM TIST*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
[64] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark," in *CVPR*. IEEE, 2013, pp. 2411–2418.
[65] W. Zhong, H. Lu, and M. Yang, "Robust object tracking via sparsity-based collaborative model," in *CVPR*. IEEE, 2012, pp. 1838–1845.
[66] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *TPAMI*, 2015.

Guibo Zhu received his B.E. degree in 2009 from Wuhan University, and M.Sc. degree from University of Chinese Academy of Sciences, Beijing, China, in 2013. He is currently a Ph.D. student of National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His current research interests include pattern recognition and machine learning, with computer vision applications in detection and tracking.

Jinqiao Wang received the B.E. degree in 2001 from Hebei University of Technology, China, and the M.S. degree in 2004 from Tianjin University, China. He received the Ph.D. degree in pattern recognition and intelligence systems from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, in 2008. He is currently an Associate Professor with Chinese Academy of Sciences. His research interests include pattern recognition and machine learning, image and video processing, mobile multimedia, and intelligent video surveillance.

Chaoyang Zhao received the B.E. and M.S. degrees from University of Electronic Science and Technology of China in 2009 and 2012 respectively. He is currently pursuing the Ph.D. degree at National Laboratory of Pattern Recognition, Chinese Academy of Sciences. His research interests include pattern recognition and machine learning, image and video processing, multimedia and intelligent video surveillance.

Hanqing Lu received his B.E. degree in 1982 and his M.E. degree in 1985 from Harbin Institute of Technology, and Ph.D. degree in 1992 from Huazhong University of Sciences and Technology. Currently, he is a Professor of National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include image and video analysis, medical image processing, object recognition, etc.