This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818132, IEEE Transactions on Pattern Analysis and Machine Intelligence

Multi-Task Structure-aware Context Modeling for Robust Keypoint-based Object Tracking

Xi Li, Liming Zhao, Wei Ji, Yiming Wu, Fei Wu, Ming-Hsuan Yang, Dacheng Tao, Ian Reid

Abstract—In the fields of computer vision and graphics, keypoint-based object tracking is a fundamental and challenging problem, which is typically formulated in a spatio-temporal context modeling framework. However, many existing keypoint trackers are incapable of effectively modeling and balancing the following three aspects in a simultaneous manner: temporal model coherence across frames, spatial model consistency within frames, and discriminative feature construction. To address this problem, we propose a robust keypoint tracker based on spatio-temporal multi-task structured output optimization driven by discriminative metric learning. Consequently, temporal model coherence is characterized by multi-task structured keypoint model learning over several adjacent frames; spatial model consistency is modeled by solving a geometric verification based structured learning problem; discriminative feature construction is enabled by metric learning to ensure the intra-class compactness and inter-class separability. To achieve the goal of effective object tracking, we jointly optimize the above three modules in a spatio-temporal multi-task learning scheme. Furthermore, we incorporate this joint learning scheme into both single-object and multi-object tracking scenarios, resulting in robust tracking results. Experiments over several challenging datasets have justified the effectiveness of our single-object and multi-object trackers against the state-of-the-art.

Index Terms—keypoint tracking, context modeling, structure learning, multi-task learning, metric learning.

1 Introduction

In computer vision and graphics, keypoint-based object tracking has been one of the most fundamental and challenging problems. Because of its efficiency and effectiveness, keypoint-based object tracking [1] [2] [3] [4] is widely used in video processing such as realistic scene reconstruction [5], automated surveillance [6] and video compression [7]. Typically, keypoint-based object tracking involves three key aspects: feature representation [8], object model learning [9], and structured object localization across frames [10]. For effective keypoint feature representation, a wide variety of low-level descriptors have been adopted to characterize the visual properties of object appearance during tracking, such as SIFT [11], SURF [12], BRIEF [13], etc. Therefore, a key issue for robust tracking is how to effectively make feature representation adapt to object appearance variations across successive frames. Moreover, object model learning [14] mainly concentrates on building discriminative keypoint-specific object models for effective keypoint matching. Structured object localization aims at capturing the global geometric structural properties of tracked objects while ensuring spatio-temporal keypoint matching consistency across frames. Therefore, the focus of this work is on designing a joint learning scheme that simultaneously models the above three aspects for robust object tracking.

More specifically, we propose a joint learning approach that is capable of well balancing the following three important parts: temporal model coherence across frames, spatial model consistency within frames, and discriminative feature construction. As illustrated in Figure 1, the joint learning approach ensures the temporal model coherence by building a multi-task structured model learning scheme, which encodes the cross-frame interaction information by simultaneously optimizing a set of mutually correlated learning subtasks (i.e. a common model plus different biases) over several successive frames. As a result, the interaction information induced by multi-task learning can guide the tracker to produce stable tracking results [15]. Moreover, the proposed approach explores the keypoint-specific structural information on spatial model consistency by performing geometric verification based on structured output learning, which aims to estimate a geometric transformation while associating cross-frame keypoints, such as a 3D pose or 2D perspective transformation. In this work, structured output learning is carried out over 2D perspective transformations. Thus, we concentrate on the problem of structured planar object tracking based on 2D perspective transformation estimation. In order to make the keypoint descriptors well adapt to time-varying tracking situations, the proposed approach naturally embeds metric learning to the structured SVM learning process [16], which enhances the discriminative power of inter-class separability [17].

The main contributions of this work are summarized as follows:

1) We propose a multi-task joint learning scheme to learn structured keypoint models by simultane-

- *Xi Li Liming Zhao, Wei Ji, Yiming Wu, and Fei Wu are with College of Computer Science, Zhejiang University, Hangzhou, China. E-mail: {xilizju, zhaoliming, jiwei, ymw, wufei}@zju.edu.cn*
- *Ming-Hsuan Yang is with Department of Electrical Engineering and Computer Science at University of California, Merced, USA. E-mail: mhyang@ucmerced.edu*
- *Dacheng Tao is with School of Information Technologies, The University of Sydney, Australia. E-mail: dacheng.tao@sydney.edu.au*
- *Ian Reid is with School of Computer Science, University of Adelaide, Australia. E-mail: ian.reid@adelaide.edu.au*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818132, IEEE Transactions on Pattern Analysis and Machine Intelligence

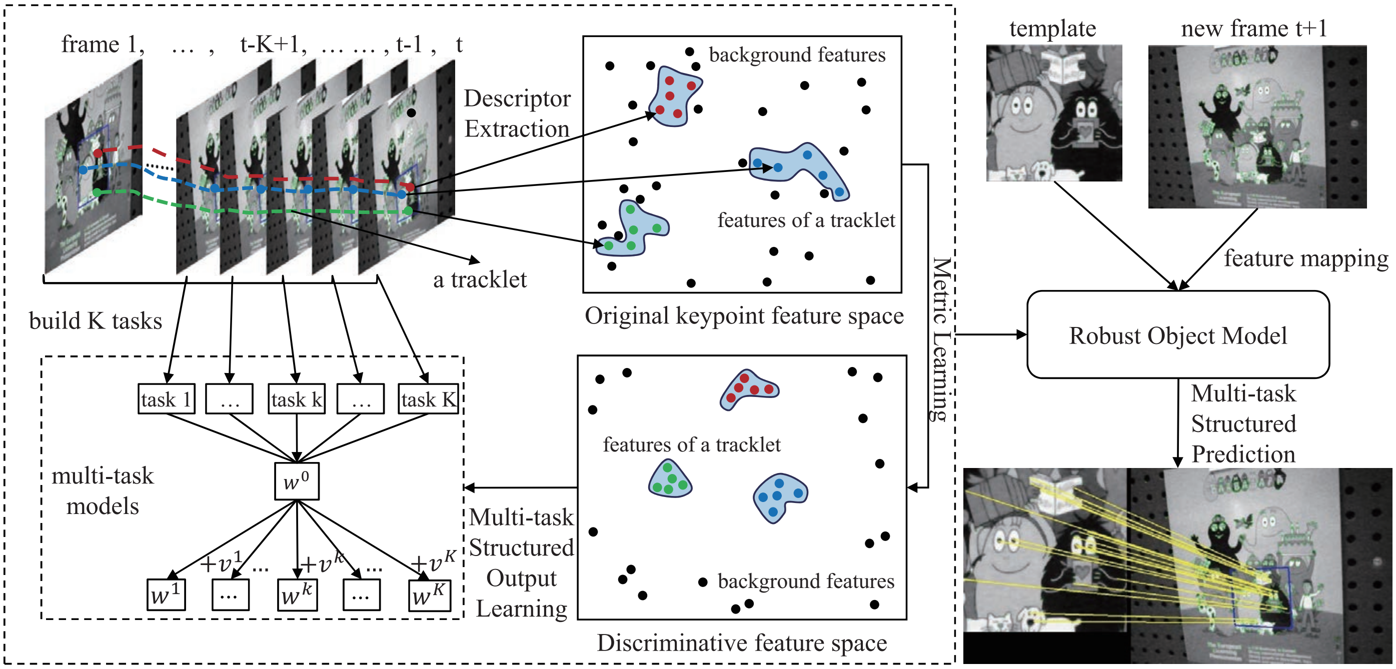


Fig. 1. Illustration of our tracking approach. Given a template picture of target, we find the corresponding location across frames through feature matching. Combined with multi-task and metric learning, a robust object model can be established to make structured prediction.

ously considering spatial model consistency, temporal model coherence, and discriminative feature learning based on our early work [18]. An online optimization algorithm is further presented to efficiently and effectively solve the proposed scheme.

2) We propose a multi-object multi-task structured learning approach, which jointly models the inter-object interaction [19] [20] while optimizing keypoint-based object-specific structural SVM problems. In addition, we create and release a new benchmark video dataset containing several challenging video sequences (covering several complicated scenarios) for experimental evaluations, which is available at http://zhaoliming.net/research/keypoint.

2 Related Work

Object tracking is of broad interest and has been broadly investigated [21]. Our work mainly builds on the following two aspects: i) feature representation; ii) object model learning. We give a brief overview of the most relevant work in each of these areas.

Feature representation. In the task of object tracking, it is very important to choose an appropriate feature [22] [23] to represent the object, which is effective enough to discriminate the object from the background during the tracking process [24] [25]. Typically, keypoint is a common way of object representation used in tracking-by-detection. For keypoint representation, a variety of keypoint descriptors are proposed to encode the local invariance information on object appearance such as SIFT [11], SURF [12] and GO [26]. To further speed up the feature extraction process, a number of binary local descriptors emerge, including BRIEF [13], ORB [27], BRISK [28], FREAK [29], etc. Since the way of feature extraction is handcrafted and fixed all the time, these keypoint descriptors are usually incapable of effectively and flexibly adapting to complex time-varying appearance variations as tracking proceeds. Therefore, some work seeks for the combination strategy for feature fusion. For instance, in [30], Bouachir et al. perform feature fusion between color distribution features and SIFT features. Petit et al. [31] propose the hybrid methods which integrate Harris corner keypoints, complementary edge and color features in model-based tracking. Recently, the research focus of designing local patch descriptors has gradually shifted from handcrafted ones (e.g., SIFT) to feature learning based ones. Deep learning based methods [32], [33], [34], [35] have been proposed to learn highly discriminative features via convolutional neural networks.

Object model learning. Object model learning aims to learn a set of keypoint-specific models for object tracking. Typically, such models are formulated in the form of classifiers [36] [37] [38]. For example, Grabner et al. [39] use boosting to learn classifiers online for feature representation, which are used to establish correct matches of keypoints across frames. Lepetit & Fua [38] build randomized trees to classify all the individual keypoints extracted from the object image. Usually, the appearance modeling methods are trained without considering geometric information. In practice, geometric information plays a very important role in object tracking. Therefore, some work incorporates various geometric constraints into the object tracking process. For example, Hare et al. [40] take advantage of RANSAC [41] [42] [43] to compute the cross-frame homography transformations, which further work as discriminative learning constraints within the tracking-by-detection framework. Lebeda et al. [44] track a 3D object and keep the 3D features correspondences by validating a global epipolar geometry model in the projected 2D observations. Cehovin et al [45] use a soft constraint of affine transformation on the local deformations of spatial neighbor patches extracted from the whole target image. Vojir & Matas [46] propose a neighborhood consistency predictor and a Markov predictor on the results from multiple local trackers to figure out the inliers and outliers. Wang & Ling [47] propose a geometric graph matching framework to incorporate transformation cues into the graph matching process.

Since object tracking is a time-varying dynamic process in the spatial and temporal dimensions, it is necessary for trackers to capture the spatio-temporal correlations during the model learning process [48]. For temporal consistency, much work resorts to multi-task learning, where the tasks are mutually correlated and share dependencies in features or learning parameters to enhance the performance of each individual task. It has a wide range of vision applications such as image classification [49] and image annotation [50]. In [51], Zhang et al. formulate object tracking as a multi-task sparse learning problem in a particle filtering framework. For spatial consistency, structured learning is typically adopted in an inter-frame geometric verification fashion [52] [53] [54] [55].

3 Approach

3.1 Problem Formulation

Given a planar object template image O denoted as a set of keypoints $\{(u_i, \mathbf{q}_i)\}_{i=1}^{N^O}$, the tracking problem aims to dynamically estimate the localization states (characterized by homography transformations) of an object within each input video frame $I = \{(v_j, \mathbf{d}_j)\}_{j=1}^{N^I}$. Here, u_i and v_j stand for the keypoint locations while $\mathbf{q}_i$ and $\mathbf{d}_j$ represent their corresponding keypoint descriptors. As a result, the tracking problem is actually converted to that of keypoint-based object matching between the template image and the input frame. In general, a crucial issue in keypoint-based object matching is to build an effective keypoint compatibility scoring function $F(C, \mathbf{y})$, which is used to measure the compatibility between keypoint correspondences C and any possible homography transformation $\mathbf{y}$. The scoring function is supposed to have the capability of well capturing the intrinsic spatio-temporal structure information during tracking.

More specifically, the compatibility score for each keypoint pair $\{u_i, v_j\}$ is calculated as $s_{ij} = \langle \mathbf{w}_i, \mathbf{d}_j \rangle$, where $\mathbf{w}_i$ is a linear model weight parameter vector for a given template keypoint u_i. Consequently, we have a set of keypoint correspondences $C = \{(u_i, v_j, s_{ij}) | (u_i, \mathbf{q}_i) \in O, (v_j, \mathbf{d}_j) \in I, s_{ij} = \langle \mathbf{w}_i, \mathbf{d}_j \rangle\}$ with $\langle \cdot, \cdot \rangle$ being the inner product. Hence, the tracking task is accomplished by solving the following structured prediction problem:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}} F(C, \mathbf{y}), \quad (1)$$

where $\mathcal{Y}$ is the homography transformation space (usually generated from RANSAC). In order to make object tracking well adapt to complicated time-varying scenarios, the compatibility scoring function $F(C, \mathbf{y})$ ought to be dynamically updated to ensure the spatial model consistency within frames as well as the temporal model coherence across frames in a discriminative feature space. To achieve this goal, we propose a metric learning driven spatio-temporal multi-task structured learning scheme.

As for the multi-object tracking case, we need to take the inter-object interactions into account. Therefore, we have the multi-object objective function formulated as:

$$(\hat{\mathbf{y}^1}, \cdots, \hat{\mathbf{y}^M}) = \arg\max_{\mathbf{y}^1, \cdots, \mathbf{y}^M \in \mathcal{Y}} \sum_{m=1}^{M} (F_m(C^m, \mathbf{y}^m) + \sum_{n \neq m} Itr_{mn}), \quad (2)$$

where $F_m(C^m, \mathbf{y}^m)$ represents the internal constraint part of each single object, and Itr_{mn} represents the interaction part between any two objects. More details of the term definitions can be found in Section 4.2

To sum up, our tracking algorithm is mainly divided into two parts: structured learning and structured prediction. Namely, an object model is first learned by a multi-task structured learning scheme in a discriminative feature space (induced by metric learning), which is shown in the left part of Figure 1. Based on the learned object model, our approach subsequently produces the tracking results through structured prediction. Using the tracking results, a set of training samples are further collected for structured learning. The tracking process recursively run the above procedures.

In the following subsections, we give a detailed description of our structured learning parts, including structured keypoint model learning, discriminative feature learning, and multi-task learning.

3.2 Structured Keypoint Model Learning

In this subsection, we need to build the keypoint tracking model by learning the compatibility scoring function during the tracking process. Before describing the keypoint tracking model, we first give the definition of the inlier set with a specific transformation $\mathbf{y}$:

$$H(C, \mathbf{y}) = \{(u_i, v_j) | (u_i, v_j) \in C, \|\mathbf{y}(u_i) - v_j\| < \tau\}, \quad (3)$$

where $\mathbf{y}(u_i)$ is the transformed location in the input frame of the template keypoint location u_i, $\tau \in \mathbb{R}$ is a spatial distance threshold, and $\|\cdot\|$ denotes the Euclidean norm.

The compatibility function is defined as the sum of inlier scores, which is practically equivalent to the MLESAC scoring function [56], [57]:

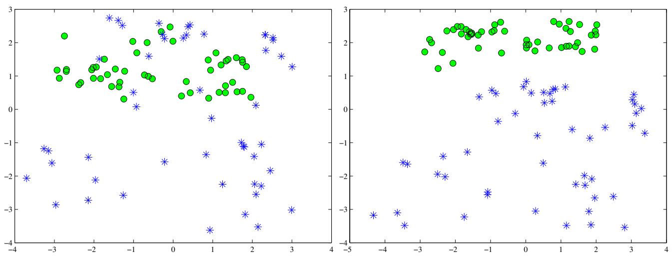
$$F(C, \mathbf{y}) = \sum_{(u_i, v_j) \in H(C, \mathbf{y})} \langle \mathbf{w}_i, \mathbf{d}_j \rangle = \langle \mathbf{w}, \mathbf{\Phi}(C, \mathbf{y}) \rangle, \quad (4)$$

where $\mathbf{w}_i$ is the model parameter vector for the i-th template keypoint. $\mathbf{\Phi}(C, \mathbf{y})$ is a joint feature mapping vector concatenated by $\boldsymbol{\phi}_i(C, \mathbf{y})$ which is defined as:

$$\boldsymbol{\phi}_i(C, \mathbf{y}) = \begin{cases} \mathbf{d}_j & \exists (u_i, v_j) \in C : \|\mathbf{y}(u_i) - v_j\| < \tau \\ \mathbf{0} & \text{otherwise.} \end{cases} \quad (5)$$

Given T training samples $\{(C_t, \mathbf{y}_t)\}_{t=1}^{T}$ (each C_t is the hypothetical correspondences of the frame I_t, and $\mathbf{y}_t$ is the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818132, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) before mapping (b) after mapping

Fig. 2. Visualization of keypoint features using PCA. The 50 green circle points represent the keypoints from 50 successive frames corresponding to the same keypoint in the template (semantically similar keypoints). The blue asterisk points represent any other keypoints, which are dissimilar to the above 50 keypoints. In figure (a), all the keypoints mix together in the original feature space. In figure (b), there is a large margin between dissimilar keypoints in current learned feature space.

homography transformation), a basic keypoint model can be learned by solving the following problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + \nu_1 \sum_{t=1}^{T} \boldsymbol{\alpha}_t + \nu_2 \sum_{t=1}^{T} \boldsymbol{\beta}_t, \tag{6}$$

where $\boldsymbol{\alpha}_t$ is the term of structural loss measurement for t-th training sample, and $\boldsymbol{\beta}_t$ is the term of feature distance for separating positive and negative keypoints. The weighting parameter ν_1 determines the trade-off between accuracy and regularization and ν_2 is also a weight parameter when matching points.

First, the $\boldsymbol{\alpha}_t$ term is defined as a penalty function on any other transformation $\mathbf{y}$ against the transformation $\mathbf{y}_t$ of the t-th training sample:

$$\boldsymbol{\alpha}_t = [\max_{\mathbf{y} \neq \mathbf{y}_t} \{\Delta(\mathbf{y}_t, \mathbf{y}) - \delta F_t(\mathbf{y})\}]_+, \tag{7}$$

where $[z]_+ = \max(z, 0)$, $\delta F_t(\mathbf{y}) = F(C_t, \mathbf{y}_t) - F(C_t, \mathbf{y})$ is the difference of compatibility scores, and $\Delta(\mathbf{y}_t, \mathbf{y})$ is a structural loss function which measures the difference of two transformations. Following [40], we use the same loss function $\Delta(\mathbf{y}_t, \mathbf{y}) = |H(C, \mathbf{y}_t)| - |H(C, \mathbf{y})|$ based on the inlier numbers. In principle, given a geometric transformation, the commonality between the template image and the video frame is characterized by the transformation-specific keypoint correspondence inliers (i.e., common matched keypoints between images). The number of these inliers reflects the template-frame overlapping correlation degree regarding transformation-specific keypoint correspondences. Hence, $\Delta(\mathbf{y}_t, \mathbf{y})$ measures the differences of the template-frame overlapping correlation degrees associated with the two geometric transformations $(\mathbf{y}_t, \mathbf{y})$.

Second, the $\boldsymbol{\beta}_t$ term is defined as a penalty function on any negative feature $\mathbf{d}_{j'}$ against the positive feature $\mathbf{d}_j$ of the matched j-th keypoint for current i-th object keypoint:

$$\boldsymbol{\beta}_t = \sum_{(u_i, v_j) \in H(C_t, \mathbf{y}_t)} [\max_{j' \neq j} \{1 - D_i(\mathbf{d}_j, \mathbf{d}_{j'})\}]_+, \tag{8}$$

where $D_i(\mathbf{d}_j, \mathbf{d}_{j'}) = \langle \mathbf{w}_i, \mathbf{d}_j - \mathbf{d}_{j'} \rangle$ is the difference of keypoint scores weighted by $\mathbf{w}_i$. Here the $\boldsymbol{\beta}_t$ term aims to maximize the score difference D_i between $\mathbf{d}_j$ and $\mathbf{d}_{j'}$ by a margin of 1.

3.3 Discriminative Feature Learning

In order to further enhance the discriminative power of the tracker and make the keypoint descriptors well adapt to time-varying tracking situations, we wish to learn a mapping function $f(\mathbf{d})$ that maps the original feature space to another discriminative feature space, in which the semantically similar keypoints are close to each other while the dissimilar keypoints are far away from each other. This procedure can be formulated as a metric learning process [58] [59]. We then use the mapped feature $f(\mathbf{d})$ to replace the original feature $\mathbf{d}$ in the structured learning process to enhance its inter-class discriminability.

Figure 2 shows an example of such feature space transformation. Before the mapping procedure, the object keypoints and the background keypoints can not be discriminated in the original feature space. After the transformation, the keypoints in different frames corresponding to the same keypoint in the template, which are semantically similar, get closer to each other in the mapped feature space, while the features of the other keypoints have a distribution in another side with a large margin.

The following describes how to learn the mapping function. Given the learned model $\mathbf{w}_i$, the distance between a doublet $(\mathbf{d}_j, \mathbf{d}_{j'})$ is defined as follows:

$$D_i(\mathbf{d}_j, \mathbf{d}_{j'}) = \langle \mathbf{w}_i, f(\mathbf{d}_j) - f(\mathbf{d}_{j'}) \rangle \,. \tag{9}$$

For convenience, we assume that the binary matrix $p_{jj'} \in \{0, 1\}$ indicates whether or not the features $\mathbf{d}_j$ and $\mathbf{d}_{j'}$ are semantically similar (if they are similar, $p_{jj'} = 1$). Therefore, the hinge loss function on a doublet is defined as:

$$\ell_i(\mathbf{d}_j, \mathbf{d}_{j'}) = [(-1)^{p_{jj'}}(1 - D_i(\mathbf{d}_j, \mathbf{d}_{j'}))]_+. \tag{10}$$

To learn the effective feature in our mapping process and maintain its consistency, we wish to find the group-sparsity of the features. So we utilize $\ell_{2,1}$-norm [60] [61] to learn the discriminative information and feature correlation consistently. Since we use a linear transformation $f(\mathbf{d}) = \mathbf{M}^\mathsf{T}\mathbf{d}$ as our mapping function, the $\ell_{2,1}$-norm for the mapping matrix $\mathbf{M}$ is defined as: $\|\mathbf{M}\|_{2,1} = \sum_i \sqrt{\sum_j \mathbf{M}_{ij}^2}$.

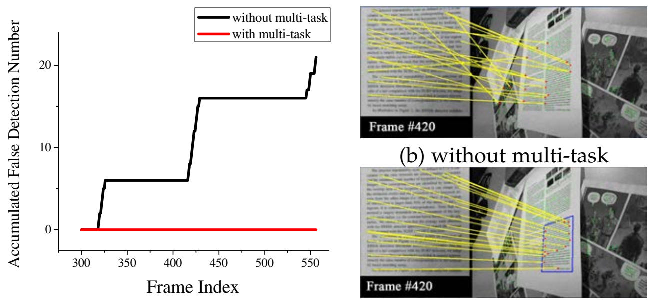
Given all the keypoint features from the video frames $\{I_t\}_{t=1}^T$, we collect all possible combinations of the features as the training set, which is denoted as $\mathcal{A} = \{(\mathbf{d}_j, \mathbf{d}_{j'}) | \mathbf{d}_j \in \{I_t\}_{t=1}^T, j' \neq j, \mathbf{d}_{j'} \in \{I_t\}_{t=1}^T\}$. We obtain the binary matrix $p_{jj'}$ by using the tracking results (if $\mathbf{d}_j$ and $\mathbf{d}_{j'}$ from different frames correspond to the same keypoint in the template, $p_{jj'}$ is set to 1; otherwise, $p_{jj'}$ is set to 0). We wish to minimize the following cost function consisting of the empirical loss term and the $\ell_{2,1}$-norm regularization term:

$$\min_{\mathbf{w}^0, \mathbf{M}} \frac{1}{2}\|\mathbf{w}\|^2 + \lambda\|\mathbf{M}\|_{2,1} + \nu_1 \sum_{t=1}^{T} \boldsymbol{\alpha}_t + \nu_2 \sum_{t=1}^{T} \boldsymbol{\beta}_t. \tag{11}$$

3.4 Multi-task Joint Learning

Due to the consistent relationships of objects in the spatio-temporal dimension, the tracking task is context sensitive within frames. Jointly learning multiple related tasks has been empirically as well as theoretically shown to significantly improve performance relative to learning each task independently [62]. Hence, it is reasonable to bring

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818132, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) tracking result (b) without multi-task (c) with multi-task

Fig. 3. Example tracking results. Figure (a) shows the quantitative results of the trackers with and without multi-task learning in the accumulated number of falsely detected frames (lower is better), the tracker with multi-task learning produces a stable tracking result. Figure (b) and (c) show the qualitative tracking results. The blue bounding box represents the location of the detected object, and the yellow line represents a keypoint correspondence. In figure (b), the tracker without multi-task learning fails to match keypoints correctly.

in the multi-task approach for our tracking task. During the tracking process, the keypoints in the successive frames $\{I_1, I_2, \dots\}$ corresponding to the i-th keypoint u_i in the template image form a tracklet $\{v^1, v^2, \dots\}$. Based on the observation that the adjacent keypoints in a tracklet are similar to each other, the models learned for the frames $\{\mathbf{w}_i^1, \mathbf{w}_i^2, \dots\}$ should be mutually correlated. So we construct K learning tasks over several adjacent frames. For example, task k learns a model $\mathbf{w}^k$ over the training samples collected from the frames I_1 to I_{T-K+k}. We model each $\mathbf{w}^k$ as a linear combination of a common model $\mathbf{w}^0$ and a unique part $\mathbf{v}^k$ [63]:

$$\mathbf{w}^k = \mathbf{w}^0 + \mathbf{v}^k, \quad k = 1, \dots, K, \tag{12}$$

where all the vectors $\{\mathbf{v}^k\}_{k=1}^K$ are "small" when the tasks are similar to each other. We will estimate all the vectors $\{\mathbf{v}^k\}_{k=1}^K$ as well as the common model $\mathbf{w}^0$ simultaneously in a multi-task model learning scheme. After considering the multi-task case,the formulation (4) can be changed to:

$$F^k(C, \mathbf{y}) = \sum_{\substack{(u_i, v_j)\in \\ H(C,\mathbf{y})}} \left\langle \mathbf{w}_i^k, \mathbf{d}_j \right\rangle = \left\langle \mathbf{w}^k, \mathbf{\Phi}(C, \mathbf{y}) \right\rangle, \tag{13}$$

where $\mathbf{w}_i^k$ is the model parameter vector for the i-th template keypoint, and $\mathbf{w}^k = [\mathbf{w}_1^k, \dots, \mathbf{w}_{N^O}^k]^\intercal$ is the column concatenation of the model parameter vectors.

Given training samples $\{(C_t, \mathbf{y}_t)\}_{t=1}^T$, we introduce a nonnegative λ_1 as the weight parameter for multiple tasks, so a structured output maximum margin framework [64] is used to learn models, which can be expressed by the following optimization problem:

$$\begin{aligned}\min_{\mathbf{w}^0, \mathbf{v}^k, \mathbf{M}} \; & \frac{1}{2}\|\mathbf{w}^0\|^2 + \frac{\lambda_1}{2K}\sum_{k=1}^K \|\mathbf{v}^k\|^2 + \lambda_2 \|\mathbf{M}\|_{2,1} \\ & + \sum_{k=1}^K \Big(\nu_1 \sum_{t=1}^{T-K+k} \boldsymbol{\alpha}_{kt} + \nu_2 \sum_{t=1}^{T-K+k} \boldsymbol{\beta}_{kt} \Big).\end{aligned} \tag{14}$$

where $\boldsymbol{\alpha}_{kt} = [\max_{\mathbf{y}\neq\mathbf{y}_t}\{\Delta(\mathbf{y}_t, \mathbf{y}) - \delta F_t^k(\mathbf{y})\}]_+$ for the k-th model. And $\boldsymbol{\beta}_{kt} = \sum_{(u_i, v_j)\in H(C_t, \mathbf{y}_t)} [\max_{j'\neq j}\{1 - D_i^k(\mathbf{d}_j, \mathbf{d}_{j'})\}]_+$ where $D_i^k(\mathbf{d}_j, \mathbf{d}_{j'}) = \langle \mathbf{w}_i^k, f(\mathbf{d}_j) - f(\mathbf{d}_{j'}) \rangle$.

To better describe the contribution of the multi-task learning, the tracking results over sample frames with and without multi-task learning are shown in Figure 3. From Figure 3(b), we observe that the independent model fails to match the keypoints in the case of drastic rotations, while the multi-task model enables the temporal model coherence to capture the information of rotational changes, thus produces a stable tracking result.

After all the models $\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^K$ are learned, we use the last model $\mathbf{w} = \mathbf{w}^K$ to predict the result of new frame I_t. We use the RANSAC method to generate hypothetical transformations. Based on the model $\mathbf{w}$, we predict the expected transformation $\mathbf{y}_t$ from all hypothetical transformations by maximizing Eq. (13). The hypothetical correspondence set C_t of the frame I_t and the predicted transformation $\mathbf{y}_t$ are then added to our training set. We use all the training samples collected from the results of previous T frames to update our model. Then the above process is repeated as tracking proceeds.

4 Model Solving for Object Tracking

In this section, we elaborate the model solving process of our processed multi-task structured learning approach with applications to single-object and multi-object tracking.

4.1 Single-Object Model Solving

In this subsection, we describe the detailed optimization procedure for single-object keypoint model solving for Eq. (14). For descriptive convenience, let J denote the term of $\nu_1 \sum_{t=1}^{T-K+k} \boldsymbol{\alpha}_{kt} + \nu_2 \sum_{t=1}^{T-K+k} \boldsymbol{\beta}_{kt}$ in Eq. (14). We solve the optimization in an alternating manner.

Fix $\{\mathbf{v}^k\}_{k=1}^K$ and $\mathbf{w}^0$, solve $\mathbf{M}$. Firstly, we fix all $\{\mathbf{v}^k\}_{k=1}^K$ and $\mathbf{w}^0$, and learn the transformation matrix $\mathbf{M}$ by solving the following problem:

$$\min_{\mathbf{M}} \|\mathbf{M}\|_{2,1} + \frac{1}{\lambda_2}\sum_{k=1}^K J. \tag{15}$$

Let $\mathbf{M}^i$ denote the i-th row of $\mathbf{M}$, and $Tr(\cdot)$ denote the trace operator. In mathematics, the Eq. (15) can be converted to the following form:

$$\min_{\mathbf{M}} Tr(\mathbf{M}^\intercal \mathbf{D} \mathbf{M}) + \frac{1}{\lambda_2}\sum_{k=1}^K J, \tag{16}$$

where $\mathbf{D}$ is the diagonal matrix of $\mathbf{M}$, and each diagonal element is $\mathbf{D}_{ii} = \frac{1}{2\|\mathbf{M}^i\|_2}$. We use an alternative algorithm to calculate $\mathbf{D}$ and $\mathbf{M}$ respectively. We calculate $\mathbf{M}$ with the current $\mathbf{D}$ by using gradient descent method, and then update $\mathbf{D}$ according to the current $\mathbf{M}$. The details of solving Eq. (16) are shown in the supplementary materials.

Fix $\mathbf{M}$ and $\{\mathbf{v}^k\}_{k=1}^K$, solve $\mathbf{w}^0$. Secondly, after $\mathbf{M}$ is learned, let $\{\mathbf{v}^k\}_{k=1}^K$ have been the optimal solution of Eq. (14). Then $\mathbf{w}^0$ can be obtained by the combination of $\mathbf{v}^k$ according to [65]:

$$\mathbf{w}^0 = \frac{\lambda_1}{K}\sum_{k=1}^K \mathbf{v}^k, \tag{17}$$

where the proof can be found in our supplementary materials.

Algorithm 1: Online Optimization for Single Object Tracking

Input: Input frame I_t and previous models $\{\mathbf{w}^1, \ldots, \mathbf{w}^K\}$
Output: The predicted transformation $\mathbf{y}_t$, updated models and mapping matrix for metric learning

/* The structured prediction part */
Calculate the correspondences C_t based on the model $\mathbf{w}^K$;
Estimate hypothetical transformations $\mathbf{y}$ using RANSAC on C_t;
Calculate the inlier set of each $\mathbf{y}$ using Eq. (3);
Predict the expected $\mathbf{y}_t$ by maximizing Eq. (13);
/* The structured learning part */
Collect the training samples $\{(C_{t-k}, \mathbf{y}_{t-k})\}_{k=0}^{K-1}$;
repeat
  Calculate $\bar{\mathbf{w}}$ according to Eq. (14);
  for $k = 1, \ldots, K$ **do**
    Update each model $\mathbf{w}^k$ using Eq. (19)
  end
  Update the mapping matrix $\mathbf{M}$ by solving Eq. (16);
until *Alternating optimization convergence*;
return $\mathbf{y}_t$, $\{\mathbf{w}^1, \ldots, \mathbf{w}^K\}$ and $\mathbf{M}$;

Fix $\mathbf{M}$ and $\mathbf{w}^0$, solve $\{\mathbf{v}^k\}_{k=1}^K$. Finally, $\{\mathbf{v}^k\}_{k=1}^K$ can be learned one by one using gradient descent method. In fact, we learn $\mathbf{w}^k = \mathbf{w}^0 + \mathbf{v}^k$ instead of $\mathbf{v}^k$ for convenience. Let $\bar{\mathbf{w}} = \frac{1}{K}\sum_{k=1}^K \mathbf{w}^k$ be the average vector of all $\mathbf{w}^k$. Then the optimization problem for each $\mathbf{w}^k$ can be rewritten as:

$$\min_{\mathbf{w}^k} \rho_1 \|\mathbf{w}^k\|^2 + \rho_2 \|\mathbf{w}^k - \bar{\mathbf{w}}\|^2 + J \tag{18}$$

where $\rho_1 = \lambda_1/(\lambda_1+1)$ and $\rho_2 = \lambda_1^2/(\lambda_1+1)$.

Given training samples $\{(C_{t-k}, \mathbf{y}_{t-k})\}_{k=0}^{K-1}$ at time t, the sub-gradient of Eq. (18) with respect to $\mathbf{w}^k$ is calculated, and we perform a gradient descent step according to:

$$\mathbf{w}^k \leftarrow (1 - \frac{1}{t})\mathbf{w}^k + \eta\rho_2\bar{\mathbf{w}} - \eta\frac{\partial J}{\partial \mathbf{w}^k} \tag{19}$$

where $\eta = 1/(\rho_1 t + \rho_2 t)$ is the step size (the details of the term J is described in the supplementary materials). We repeat the procedure to obtain an optimal solution until the algorithm converges (on average after 5 iterations).

All the above is summarized in Algorithm 1.

4.2 Multi-Object Extension

As for the multi-object case, we should give each object template a homography transformation in the successive frames. If we have M templates totally, the templates and transformations are represented by $O^m = \{(u_i, \mathbf{q}_i)\}_{i=1}^{N^{O^m}}$ and $\mathbf{y}^m$ respectively, where $m \in \{1, \ldots, M\}$. We define the compatibility function which simultaneously considers the unary compatibility for each template as well as the interactions between templates [66] [67] [68] [69].

For a particular template $\mathbf{y}^m$ in the tracking process, the compatibility function is formulated as:

$$F^k_{multi}(C^m, \mathbf{y}^m) = F^k(C^m, \mathbf{y}^m) + \sum_{n \neq m} Itr_{mn} \tag{20}$$

where $F^k(C^m, \mathbf{y}^m)$ and $\sum_{n \neq m} Itr_{mn}$ represent the unary and interaction function respectively.

Specifically, the unary object modeling is the same as the single object tracking case.

$$F^k(C^m, \mathbf{y}^m) = \sum_{\substack{(u_i, v_j) \in \\ H(C^m, \mathbf{y}^m)}} \left\langle \mathbf{w}_i^{km}, \mathbf{d}_j \right\rangle \tag{21}$$

For any two templates m and n, their transformations are $\mathbf{y}^m$ and $\mathbf{y}^n$, through the homography transformation, we define their overlapping ratio as the IoU (intersection over union) between two quadrangles denoted by $U(\mathbf{y}^m, \mathbf{y}^n)$. Along with the transformation in the t-th frame, we think the keypoints located in the overlapping regions possess a lower matching confidence than the other keypoints. So the interaction part is formulated as:

$$Itr_{mn} = \mu_1 \sum_{(u_i, v_j) \in R(m,n)} (g(\boldsymbol{\lambda}_i) \left\langle \mathbf{w}_i^{km}, \mathbf{d}_j \right\rangle) + \mu_2 \boldsymbol{\psi}_o(\mathbf{y}_{t-1}^m, \mathbf{y}^m) \tag{22}$$

where $R(m,n)$ represents the keypoints pairs which are the inliers of m but located in the overlapping region, $g(\boldsymbol{\lambda}_i) = 1/(1 + e^{-\boldsymbol{\lambda}_i})$ is the sigmoid function, which adaptively activates the weight vector of these keypoints. Moreover, μ_1 and μ_2 are negative penalty parameters for enforcing the interaction terms to be small. Specifically, μ_1 penalizes the keypoints located in the overlapping regions, and μ_2 penalizes the large variation with respect to IoUs for two adjacent frames. Therefore, their corresponding terms respectively aim to discourage the overlapping keypoints from different objects, and encouraging the temporal smoothness between two adjacent frames. Therefore, these two parameters are negatively correlated with the compatibility function such that smaller interaction terms lead to a larger score. Here

$$\boldsymbol{\psi}_o(\mathbf{y}_{t-1}^m, \mathbf{y}^m) = \sum_{l \neq m} |U(\mathbf{y}_{t-1}^m, \mathbf{y}_{t-1}^l) - U(\mathbf{y}^m, \mathbf{y}^l)| \tag{23}$$

penalizes the large variation with respect to IoUs for two adjacent frames.

Given the collected training samples $\{\mathbf{C}_t, \mathbf{Y}_t\}_{t=1}^T$ ($\mathbf{C}_t = \{C_t^1, \ldots, C_t^M\}$ are the hypothesis correspondences and $\mathbf{Y}_t = \{Y_t^1, \ldots, Y_t^M\}$ are predicted transformations), the optimization problem is expressed as:

$$\min_{\substack{\mathbf{w}^{0m} \\ \mathbf{v}^{km}, \mathbf{M}^m}} \sum_{m=1}^M \Big(\frac{\|\mathbf{w}^{0m}\|^2}{2} + \frac{\lambda_1}{2K} \sum_{k=1}^K \|\mathbf{v}^{km}\|^2 + \lambda_2 \|\mathbf{M}^m\|_{2,1} + \sum_{k=1}^K \Big(\nu_1 \sum_{t=1}^{T-K+k} \boldsymbol{\alpha}_{ktm} + \nu_2 \sum_{t=1}^{T-K+k} \boldsymbol{\beta}_{ktm} \Big) \Big). \tag{24}$$

where $\boldsymbol{\alpha}_{ktm} = [\max_{\mathbf{y} \neq \mathbf{y}_t} \{\Delta(\mathbf{y}_t^m, \mathbf{y}^m) - \delta F_t^k(\mathbf{y}^m)\}]_+$ is extended as multi-object version for m-th object, and here $\delta F_t^k(\mathbf{y}^m) = F^k_{multi}(C_t^m, \mathbf{y}_t^m) - F^k_{multi}(C_t^m, \mathbf{y}^m)$. The $\boldsymbol{\beta}$ term is $\boldsymbol{\beta}_{ktm} = \sum_{(u_i, v_j) \in H(C_t, \mathbf{y}_t)} [\max_{j' \neq j} \{1 - D^k_{im}(\mathbf{d}_j, \mathbf{d}_{j'})\}]_+$ where $D^k_{im}(\mathbf{d}_j, \mathbf{d}_{j'}) = \langle \mathbf{w}_i^k, f(\mathbf{d}_j) - f(\mathbf{d}_{j'}) \rangle$. Again, the unary optimization is similar to single object optimization, it could be solved briefly by following the steps in Section 4.1. Algorithm 2 summaries the optimization for multi-object tracking.

5 Experiments

We have performed two sets of experiments to evaluate our trackers. In the first set of experiments, we evaluate the performance of the single object tracker SMM (SSVM + ML + MT), and compare it with the state-of-the-art trackers. In the second set of experiments, we evaluate the multi-object tracker MSMM (Multi-object + SSVM + ML + MT), compared with multiple single object trackers.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818132, IEEE Transactions on Pattern Analysis and Machine Intelligence

Algorithm 2: Online Optimization for Multi-object Tracking

Input: Input frame I_t and M previous models $\{\{\mathbf{w}^{11}, \dots, \mathbf{w}^{1M}\}, \dots, \{\mathbf{w}^{K1}, \dots, \mathbf{w}^{KM}\}\}$
Output: The predicted transformation $\mathbf{Y}_t$, updated models and mapping matrix for metric learning

/* The structured prediction part */
Calculate the correspondences $\mathbf{C}_t$ based on the model $\{\mathbf{w}^{K1}, \dots, \mathbf{w}^{KM}\}$;
Estimate hypothetical transformations $\mathbf{Y} = \{\mathbf{y}^1, \dots, \mathbf{y}^m, \dots, \mathbf{y}^M\}$ using RANSAC on $\mathbf{C}_t$;
Calculate the inlier set of each $\mathbf{y}^m$;
Predict the expected $\mathbf{Y}_t$ by maximizing Eq. (20);
/* The structured learning part */
Collect the training samples $\{(\mathbf{C}_{t-k}, \mathbf{Y}_{t-k})\}_{k=0}^{K-1}$;
repeat
  for $m = 1, \dots, M$ **do**
    Calculate $\bar{\mathbf{w}}^m$ according to Eq. (24);
    for $k = 1, \dots, K$ **do**
      Update model $\mathbf{w}^{km}$ using Eq. (19)
    end
    Update the mapping matrix $\mathbf{M}^m$ by solving Eq. (16);
  end
until *Alternating optimization convergence*;
return $\mathbf{Y}_t$, $\{\{\mathbf{w}^{11}, \dots, \mathbf{w}^{1M}\}, \dots, \{\mathbf{w}^{K1}, \dots, \mathbf{w}^{KM}\}\}$ and $\{\mathbf{M}^1, \dots, \mathbf{M}^M\}$;

Dataset. We use several sequences in the experiments for single object tracking. There are five sequences are from [40], and the four sequences (i.e., "chart", "keyboard", "food", "book") are recorded by ourselves. All these sequences recorded in the natural scene cover several complicated scenarios such as background clutter, object zooming, object rotation, illumination variation, motion blurring and partial occlusion, some example frames are shown in Figure 4. Furthermore, to evaluate the performance under different complicated conditions, we also report the results on a public planar object tracking dataset: UCSB [70]. The dataset is relatively large and contains 96 video sequences with complicated scenarios, including geometric distortions (panning, zoom, tilting, rotation), nine levels of motion blur, as well as different lighting conditions.

In the experiments for multi-object tracking, we use the extra three video sequences (i.e. "interaction", "twobooks", "twocards", some example frames are shown in Figure 4). These video sequences are recorded with several objects of interest. On one hand, we use several single object trackers to track the different objects as a baseline approach, on the other hand, we use the approach proposed in Section 4.2 as the multi-object tracker. The detailed description of our dataset is shown in the supplementary material.

5.1 Setup

Evaluation Metrics. We use the same criteria as [40] with a scoring function between the predicted homography $\mathbf{y}$ and the ground-truth homography $\mathbf{y}^*$:

$$S(\mathbf{y}, \mathbf{y}^*) = \frac{1}{4}\sum_{i=1}^{4} \|\mathbf{y}(c_i) - \mathbf{y}^*(c_i)\|_2 \tag{25}$$

where $\{c_i\}_{i=1}^4 = \{(-1,-1)^\mathsf{T}, (1,-1)^\mathsf{T}, (-1,1)^\mathsf{T}, (1,1)^\mathsf{T}\}$ is a normalized square. For each frame, it is regarded as a successfully detected frame if $S(\mathbf{y}, \mathbf{y}^*) < 10$, and a falsely detected frame otherwise. The average success rate is defined as the number of successfully detected frames divided by the length of the sequence, which is used to evaluate the performance of the tracker. To provide the tracking results frame by frame, we present a criterion on the accumulated false detection number, which is defined as the accumulated number of falsely detected frames as tracking proceeds.

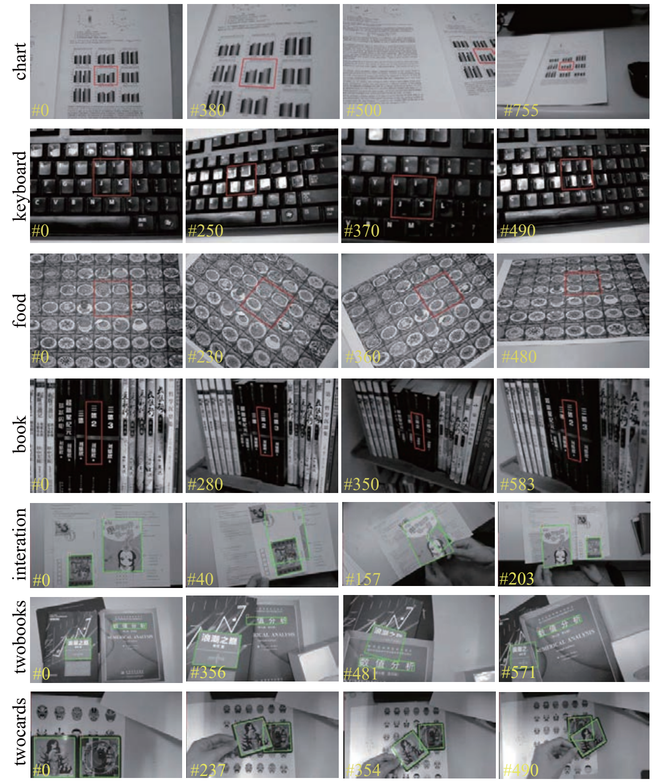


Fig. 4. The example frames in the single object tracking dataset. The #0 frame is the first frame for initializing the template. In the single object tracking dataset, the red boxes in the subsequent frames show the groundtruth in these frames. In the multi-object tracking dataset, the green boxes show the groundtruth. *Figure is best viewed in color.*

Implementation Details. For keypoint feature extraction, we use the FAST keypoint detector [71] with 256-bit BRIEF descriptor [13]. For metric learning, the linear transformation matrix $\mathbf{M}$ is initialized to be an identity matrix. For multi-task learning, the number of tasks K is chosen as 5 and we update all the multi-task models frame by frame. Weighting parameters $\lambda_1, \lambda_2, \nu_1, \nu_2$ are set to 1. In the experiments of multi-object tracking, we empirically set the values of μ_1 and μ_2 to balance the three parts in the compatibility function. The μ_2 part is related to the number of keypoints, so we simply set $|\mu_2|$ as the number of keypoints. The μ_1 part controls the contribution degree of the interaction part. We empirically set μ_1 to -0.6. Similar to [40], we consider the tracking process of estimating homography transformation on the object as a tracking-by-detection task. Particularly, for multi-object tracking, the RANSAC samples are too many to traverse, so we use only the best ten RANSAC samples to calculate the homography.

We implement our approach in C++ and OPENCV. On average, our algorithm takes 0.0746 second to process one frame with a quad-core 2.4GHz Intel Xeon E5-2609 CPU and 16GB memory.

TABLE 1
Comparisons with nine state-of-the-art methods in the average tracking accuracy ($\pm$ standard deviation) on the UCSB dataset. The best result on each categories of sequences is shown in bold font.

Motion Task	SSVM	IC	ESM	GPF	GOESM	KCF	SRDCF	TLD	Gracker	Ours
panning(6)	0.84±0.25	0.29±0.25	0.68±0.31	0.90±0.02	0.35±0.41	0.90±0.03	0.92±0.03	0.79±0.13	0.96±0.02	**0.99±0.01**
tilting(6)	0.73±0.41	0.82±0.30	0.90±0.19	0.73±0.28	**0.90±0.18**	0.67±0.33	0.68±0.33	0.62±0.38	0.85±0.24	0.83±0.04
rotation(6)	0.65±0.33	0.74±0.21	0.80±0.14	0.79±0.12	0.79±0.14	0.66±0.18	0.59±0.24	0.65±0.18	**0.80±0.13**	0.73±0.21
zoom(6)	0.73±0.34	0.73±0.29	0.92±0.07	0.87±0.07	0.88±0.11	0.55±0.28	0.88±0.06	0.77±0.21	0.89±0.07	**0.99±0.01**
lighting(12)	0.80±0.33	0.68±0.39	0.83±0.21	0.90±0.02	**0.99±0.01**	0.92±0.01	0.91±0.02	0.58±0.42	**0.99±0.01**	**0.99±0.01**
blur(54)	0.40±0.41	0.29±0.37	0.44±0.40	0.81±0.07	0.36±0.43	0.82±0.07	0.87±0.03	0.65±0.33	0.88±0.11	**0.93±0.15**
unconstrained(6)	0.36±0.34	0.07±0.22	0.16±0.24	0.42±0.39	0.12±0.22	0.28±0.18	0.66±0.15	0.33±0.34	0.58±0.37	**0.83±0.08**
Total(96)	0.53±0.41	0.41±0.34	0.56±0.32	0.80±0.10	0.52±0.31	0.77±0.10	0.83±0.07	0.64±0.32	0.88±0.11	**0.92±0.14**

5.2 Single Object Tracking

We first evaluate the performance of the SMM tracker on video sequences in which single object need to be tracked.

Comparison with State-of-the-art Methods. On the public dataset UCSB, we compare our method with nine state-of-the-art baselines, including six planar object trackers (Gracker [47], GOESM [26], GPF [72], SSVM [40], IC [73], ESM [74]), and three generic object trackers (TLD [75], KCF [76], and SRDCF [77]). The Gracker [47] method and the proposed method are both keypoint based planar object trackers. The results in Table 1 show that the proposed method achieves favorable performances on such a large and complicated benchmark dataset. We obtain the best tracking results on most sequences except the categories of "tilting" and "rotation" sequences. We also find out that the tracker's performance drops greatly when there are very few keypoints detected in some dramatically tilted or rotated frames. It probably results from the BRIEF descriptor, which is not rotationally invariant. For the other 84 sequences with challenging scenarios, the proposed tracker achieves highly accurate and stable results with the help of discriminative feature learning and structured multi-task modeling.

We compare our approach with some state-of-the-art approaches, including boosting based approach [39], structured SVM (SSVM) approach [40] and a baseline static tracking approach (without model updating). All these approaches are implemented by making use of their publicly available code. Table 2 shows the experimental results of all four approaches in the average success rate. As shown in this table, our approach performs best on all sequences. Multi-task learning module guarantees the temporal smoothness over the changes of object motion and metric learning module learns a more discriminative feature. And thus we get a higher average accuracy.

To provide an intuitive illustration, we report the detection result on each frame in Figure 5. We observe that both the "Boosting" and "SSVM" approaches obtain a number of incorrect detection results on some frames of the test sequences, while our approach achieves stable tracking results in most situations (the curve corresponding to our approach grows slowly and is almost horizontal).

Figure 6 shows the tracking results on some sample frames. These sequences containing background clutter are challenging for keypoint based tracking. Due to metric learning and multi-task learning, our approach still performs well in some complicated scenarios with drastic object appearance changes.

TABLE 2
Comparison with state-of-the-art methods in the average success rate (higher is better). The best result on each sequence is shown in bold. We observe that our method performs best on all the sequences.

Sequence	Average Success Rate(%)			
	Static	Boosting	SSVM	Ours
barbapapa	19.7138	89.0302	94.1176	**94.4356**
comic	42.5000	57.6042	98.1250	**98.8542**
map	81.1295	82.0937	**98.7603**	**98.7603**
paper	05.0267	03.8502	82.7807	**88.2353**
phone	88.1491	84.9534	96.6711	**98.4021**
chart	13.1461	01.9101	53.0337	**77.5281**
keyboard	27.8607	57.7114	62.3549	**94.5274**
food	32.8173	67.4923	85.7585	**99.6904**
book	08.5616	08.9041	55.8219	**81.6781**

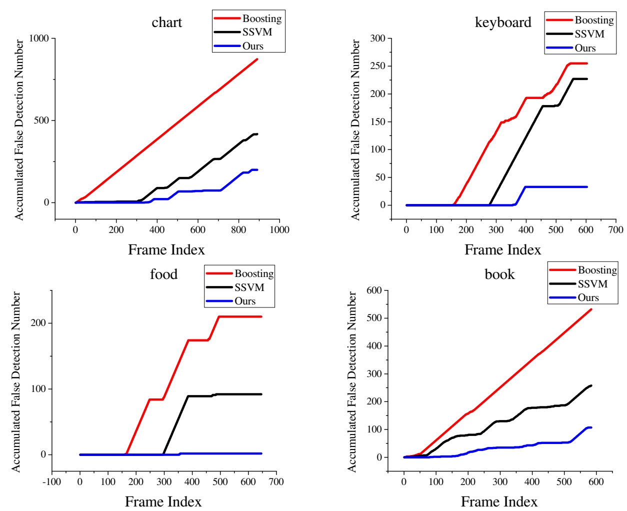
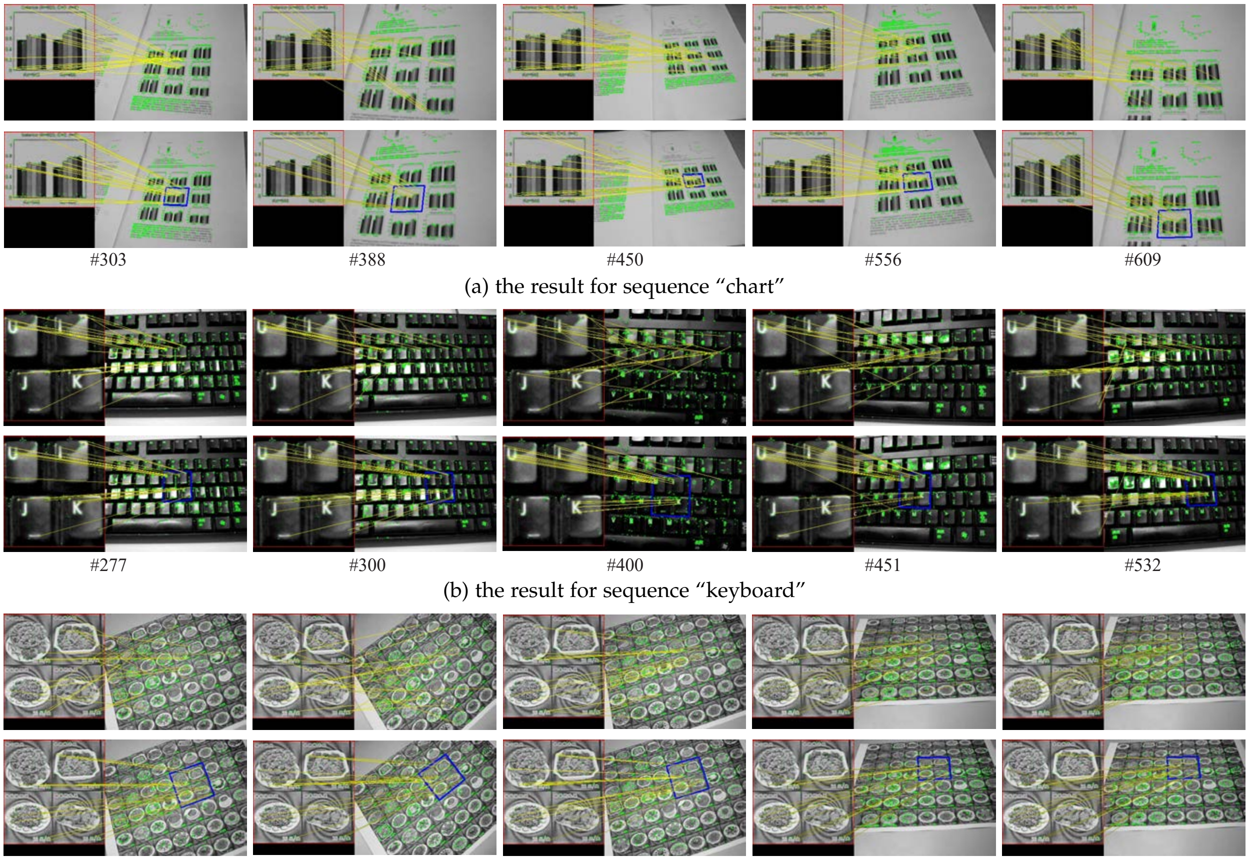


Fig. 5. Comparison of three approaches in the accumulated number of falsely detected frames (lower is better). The curve corresponding to our approach grows slowly and is almost horizontal, which means that our tracking result is stable.

Evaluation of Our Individual Components. To explore the contribution of each component in our approach, we compare the performances of the approaches with individual parts, including SSVM (structured SVM), SML (SSVM + metric learning), SMT (SSVM + multi-task learning), and SMM (SSVM + ML + MT, which is exactly our approach). The experimental results of all these approaches in the average success rate are reported in Table 3.

From Table 3, we find that the geometric verification based structured learning approach achieves good tracking results in most situations. Furthermore, we observe from Figure 8 that multi-task structured learning guides the tracker to produce a stable tracking result in the complicated scenarios, and metric learning enhances the capability of the tracker to separate keypoints from background clutter. Our

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818132, IEEE Transactions on Pattern Analysis and Machine Intelligence



#303 #388 #450 #556 #609

(a) the result for sequence "chart"

#277 #300 #400 #451 #532

(b) the result for sequence "keyboard"

#301 #350 #384 #472 #484

(c) the result for sequence "food"

Fig. 6. Example tracking results on our test video sequences. In each picture, the left part highlighted in red bounding box is the template image. The blue box shows the location of the detected object in the frame. And in each set of pictures, the first row is the result of approach "SSVM", the second row is the result of approach "SMM"(exactly our approach). Our model has adapted to obtain correct detection results in the complicated scenarios with drastic object appearance changes. *Figure is best viewed in color.*

TABLE 3
Evaluation of our individual components in the average success rate. The best result on each sequence is shown in bold font. We find that both metric learning and multi-task learning based approach obtain a higher success rate than the structured SVM approach, and our joint learning approach achieves the best performance.

Sequence	Average Success Rate(%)			
	SSVM	SML	SMT	SMM
barbapapa	94.1176	**94.4356**	94.2766	**94.4356**
comic	98.1250	98.5417	98.6458	**98.8542**
map	**98.7603**	98.6226	**98.7603**	**98.7603**
paper	82.7807	86.2032	87.3797	**88.2353**
phone	96.6711	97.2037	97.6032	**98.4021**
chart	53.0337	62.0225	61.1236	**77.5281**
keyboard	62.3549	73.6318	76.6169	**94.5274**
food	85.7585	88.0805	99.3808	**99.6904**
book	55.8219	71.5753	74.8288	**81.6781**

TABLE 4
Comparison of the proposed method with and without metric learning by using BRIEF or SIFT descriptor

Sequence	BRIEF		SIFT	
	w/o ML	w/ ML	w/o ML	w/ ML
paper	87.38	**88.24**	93.37	**97.22**
chart	61.12	**77.53**	73.37	**83.60**

approach consisting of all these components then generates a robust tracker.

From Table 3, we also observe that the tracker is very robust to illumination with the help of metric learning, but not so much to rotation/viewpoint changes in some sequences

TABLE 5
Results of multi-object tracking between SMM and MSMM. We find that multi-object tracker is better than several single object trackers especially in the sequences "interaction" and "twocards", which contain plenty of interaction frames among all the frames.

Sequence	Average Success Rate(%)	
	SMM	MSMM
interaction	67.1171	**72.0721**
	67.1171	**68.9189**
twobooks	55.4472	**57.561**
	88.2927	**89.7561**
twocards	61.1607	**65.4018**
	76.5625	**79.5759**

(e.g., "paper" and "chart"). The reason is that the BRIEF descriptor is not rotationally invariant. Thus we use a more powerful descriptor SIFT and evaluate the performance differences for the proposed method with and without metric learning. The results on sequences "paper" and "chart" (difficult to BRIEF due to large rotation/viewpoint changes) are reported in Table 4. We observe that using more powerful SIFT descriptor achieves better results than the BRIEF one. With the help of metric learning, the performance can still be improved for both BRIEF and SIFT descriptors. Due to the simplicity and efficiency, we use the BRIEF binary descriptor for balancing accuracy and speed in the experiments.

In the experiments, we use all the keypoint features extracted from the object for modeling and tracking. For

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818132, IEEE Transactions on Pattern Analysis and Machine Intelligence

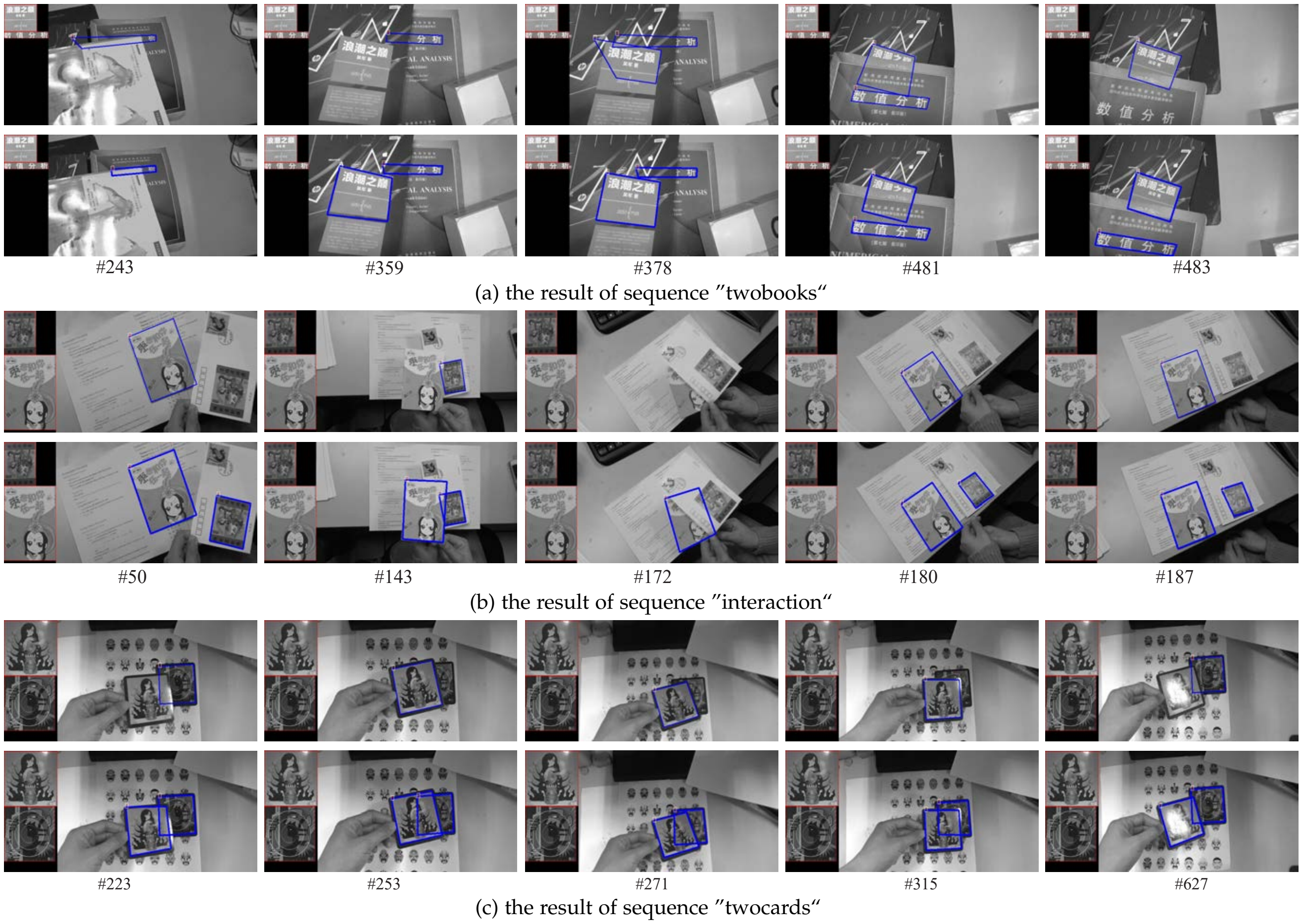


Fig. 7. Example multi-object tracking results, in each set of pictures, first row is the result of SMM, second row is the result of MSMM. In each picture, the tracked objects are in the bounding box (for two-object video sequence, the object is bounded by green and red box; for three-object video sequence, the object is bounded by blue, red and green box respectively), the number under the picture is the frame number in the sequence. *Figure is best viewed in color.*

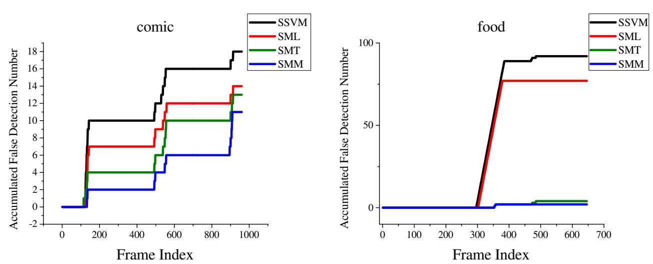


Fig. 8. Evaluation of our individual components in the accumulated number of falsely detected frames (lower is better). We observe that both metric learning and multi-task learning can improve the robustness of the tracker.

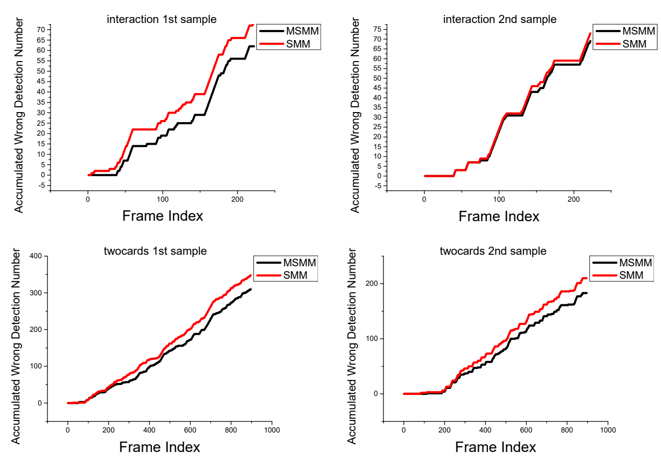


Fig. 9. Evaluation of multi-object tracker and several single object trackers in the accumulated number of falsely detected frames(lower is better). We observe that the SMM tracker obtains a higher incorrect detection number with the increase of frame number.

cases where the object is with a large scale and consists of a significant number of feature points and descriptors, we utilize a feature pool for efficiency and explore the influence of different sizes of feature pools. Specifically, we keep the top N strongest keypoints with high response scores from keypoint detector. The results shown in Table 6 indicate that using a smaller feature pool size leads to a better performance. That is probably because keypoint selection (guided by detection response) can reduce the influence of noisy keypoints and refine the keypoint model learning process. In the case of large objects, we can decrease the feature pool size for tracking. In the current experiments, we just use all the keypoints in the experiments for simplicity.

5.3 Multi-Object Tracking

In the multi-object experiment, we compare the SMM tracker and MSMM tracker in the three multi-object sequences (i.e. "interaction", "twobooks", "twocards"). We evaluate the trackers by measuring the accuracy and accumulated number of falsely detected frames, Table 5 gives

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818132, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 6
Results of using different sizes of descriptor pools.

Sequence	$N = 100$	$N = 75$	$N = 50$	$N = 30$
Comic	98.85	99.47	**99.79**	99.58
Chart	77.52	79.89	**85.73**	83.93
Food	99.69	99.69	**100**	98.45

the experiment results.

From Table 5, we find out that the MSMM tracker achieves a better performance than the SMM tracker throughout the three datasets. Especially in the sequence "interaction" and "twocards", there are plenty of object interactions among all the frames. As a result, the MSMM tracker obtain more robust tracking results, and the tracking accuracy for MSMM is about 5 percent higher than SMM. Moreover, in the sequence "twobooks", it is challenging to track the objects because of object vanishment in several frames. Numbers of accumulated wrong detections for each video sequence are shown in Figure 9. From Figure 9, we observe that the MSMM tracker obtains lower accumulative detection errors as tracking proceeds.

To provide an intuitive illustration, we show tracking results on some example frames in Figure 7. From Figure 7, we observe that the proposed MSMM tracker achieves a more accurate and stable tracking performance when occlusions take place.

6 Conclusion

In this paper, we have presented novel and robust keypoint trackers by solving a multi-task structured output optimization problem driven by metric learning. Our joint learning approach has simultaneously considered spatial model consistency, temporal model coherence, and discriminative feature construction during the tracking process. We have shown in extensive experiments that structured learning based on geometric verification has modeled the spatial model consistency to generate a robust tracker in most scenarios; multi-task structured learning has characterized the temporal model coherence to produce stable tracking results even in complicated scenarios with drastic changes; metric learning enhances the discriminability of the tracker by discriminative feature construction. Experimental results on the single-object and multi-object tracking datasets have demonstrated the effectiveness of our tracker.

References

[1] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision." in *IJCAI*, vol. 81, pp. 674–679, 1981.
[2] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "Prost: Parallel robust online simple tracking," in *CVPR*, pp. 723–730. IEEE, 2010.
[3] M. E. Maresca and A. Petrosino, "Matrioska: A multi-level approach to fast tracking by learning," in *ICIAP*, pp. 419–428. Springer, 2013.
[4] G. Nebehay and R. Pflugfelder, "Consensus-based matching and tracking of keypoints for object tracking," in *WACV*, pp. 862–869. IEEE, 2014.
[5] K. Ng, V. Sequeira, S. Butterfield, D. Hogg, and J. G. Gonçalves, "An integrated multi-sensory system for photo-realistic 3d scene reconstruction," *International archives of Photogrammetry and Remote Sensing*, vol. 32, pp. 356–363, 1998.
[6] O. Javed and M. Shah, "Tracking and object classification for automated surveillance," in *ECCV*, pp. 343–357. Springer, 2002.
[7] R. Frederick, "Experiences with real-time software video compression," in *International Workshop on Packet Video*, pp. 26–27. Citeseer, 1994.
[8] T. Nowak, P. Najgebauer, J. Romanowski, M. Gabryel, M. Korytkowski, R. Scherer, and D. Kostadinov, "Spatial keypoint representation for visual object retrieval," in *IJAISC*, pp. 639–650. Springer, 2014.
[9] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and k-selection," in *CVPR*, pp. 1313–1320. IEEE, 2011.
[10] L. Wen, Z. Cai, Z. Lei, D. Yi, and S. Z. Li, "Online spatio-temporal structural context learning for visual tracking," in *ECCV*, pp. 716–729. Springer, 2012.
[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[12] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
[13] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *ECCV*, pp. 778–792, 2010.
[14] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.
[15] L. Zhang and L. Maaten, "Structure preserving object tracking," in *CVPR*, pp. 1838–1845, 2013.
[16] S. Branson, P. Perona, and S. Belongie, "Strong supervision from weak annotation: Interactive training of deformable part models," in *ICCV*, pp. 1832–1839. IEEE, 2011.
[17] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007.
[18] L. Zhao, X. Li, J. Xiao, F. Wu, and Y. Zhuang, "Metric learning driven multi-task structured output optimization for robust keypoint tracking," in *AAAI*, 2015.
[19] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1531–1536, 2004.
[20] A. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu, "Multi-object tracking through simultaneous long occlusions and split-merge conditions," in *CVPR*, vol. 1, pp. 666–673. IEEE, 2006.
[21] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *CVPR*, pp. 2411–2418, 2013.
[22] D.-N. Ta, W.-C. Chen, N. Gelfand, and K. Pulli, "Surftrac: Efficient tracking and continuous object recognition using local feature descriptors," in *CVPR*, pp. 2937–2944. IEEE, 2009.
[23] B. Yang and R. Nevatia, "Multi-target tracking by online learning a crf model of appearance and motion patterns," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 203–217, 2014.
[24] Z. Han, Q. Ye, and J. Jiao, "Combined feature evaluation for adaptive visual object tracking," *Computer Vision and Image Understanding*, vol. 115, no. 1, pp. 69–80, 2011.
[25] M. Özuysal, V. Lepetit, F. Fleuret, and P. Fua, "Feature harvesting for tracking-by-detection," in *ECCV*, pp. 592–605. Springer, 2006.
[26] L. Chen, F. Zhou, Y. Shen, X. Tian, H. Ling, and Y. Chen, "Illumination insensitive efficient second-order minimization for planar object tracking," in *ICRA*, pp. 4429–4436, 2017.
[27] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *ICCV*, pp. 2564–2571. IEEE, 2011.
[28] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *ICCV*, pp. 2548–2555. IEEE, 2011.
[29] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *CVPR*, pp. 510–517. IEEE, 2012.
[30] W. Bouachir and G.-A. Bilodeau, "Structure-aware keypoint tracking for partial occlusion handling," in *WACV*, pp. 877–884. IEEE, 2014.
[31] A. Petit, E. Marchand, and K. Kanani, "Combining complementary edge, keypoint and color features in model-based tracking for highly dynamic scenes," in *ICRA*, pp. 4115–4120. IEEE, 2014.
[32] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolu-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818132, IEEE Transactions on Pattern Analysis and Machine Intelligence

tional feature point descriptors," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 118–126, 2015.

[33] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[34] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[35] Y. D. Lin, J. Lu, Z. Wang, J. Feng, and J. Zhou, "Learning deep binary descriptors with multi-quantization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[36] M. Mayo and E. Zhang, "Improving face gender classification by adding deliberately misaligned faces to the training data," in *IVCNZ*, pp. 1–5. IEEE, 2008.

[37] S. Uchida, Y. Shigeyoshi, Y. Kunishige, and F. Yaokai, "A keypoint-based approach toward scenery character detection," in *ICDAR*, pp. 819–823. IEEE, 2011.

[38] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1465–1479, 2006.

[39] M. Grabner, H. Grabner, and H. Bischof, "Learning features for tracking," in *CVPR*, pp. 1–8. IEEE, 2007.

[40] S. Hare, A. Saffari, and P. H. Torr, "Efficient online structured output learning for keypoint-based object tracking," in *CVPR*, pp. 1894–1901. IEEE, 2012.

[41] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[42] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *ISMAR*, pp. 225–234. IEEE, 2007.

[43] F. Pernici and A. Del Bimbo, "Object tracking by oversampling local features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2538–2551, 2014.

[44] K. Lebeda, S. Hadfield, and R. Bowden, "2d or not 2d: Bridging the gap between tracking and structure from motion," in *Computer Vision - ACCV 2014 - 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part IV*, pp. 642–658, 2014.

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

[46] T. Vojír and J. Matas, "The enhanced flock of trackers," in *Registration and Recognition in Images and Videos*, 2014, pp. 113–136.

[47] T. Wang and H. Ling, "Gracker: A graph-based planar object tracker," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[48] Y. Bai and M. Tang, "Robust tracking via weakly supervised ranking svm," in *CVPR*, pp. 1854–1861. IEEE, 2012.

[49] X.-T. Yuan, X. Liu, and S. Yan, "Visual classification with multitask joint sparse representation," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4349–4360, 2012.

[50] A. Quattoni, X. Carreras, M. Collins, and T. Darrell, "An efficient projection for l 1, regularization," in *ICML*, pp. 857–864. ACM, 2009.

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

[52] M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in *ECCV*, pp. 2–15. Springer, 2008.

[53] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel, "Robust tracking with weighted online structured learning," in *ECCV*, pp. 158–172. Springer, 2012.

[54] S. Kim, S. Kwak, J. Feyereisl, and B. Han, "Online multi-target tracking by large margin structured learning," in *ACCV*, pp. 98–111. Springer, 2012.

[55] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[56] P. H. S. Torr and A. Zisserman, "Robust computation and parametrization of multiple view relations," in *ICCV*, pp. 727–732, 1998.

[57] ——, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[58] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, no. 2, pp. 207–244, 2009.

[59] K. Park, C. Shen, Z. Hao, J. Kim *et al.*, "Efficiently learning a distance metric for large margin nearest neighbor classification." in *AAAI*, 2011.

[60] X. Cai, F. Nie, H. Huang, and C. Ding, "Multi-class l2, 1-norm support vector machine," in *ICDM*, pp. 91–100. IEEE, 2011.

[61] Z. Li, Y. Yang, J. Liu, X. Zhou, H. Lu *et al.*, "Unsupervised feature selection using nonnegative spectral analysis." in *AAAI*, 2012.

[62] A. Evgeniou and M. Pontil, "Multi-task feature learning," *Advances in neural information processing systems*, vol. 19, p. 41, 2007.

[63] J. Zheng and L. M. Ni, "Time-dependent trajectory regression on road networks via multi-task learning." in *AAAI*, 2013.

[64] B. Taskar, C. Guestrin, and D. Koller, "Max-margin markov networks," in *Advances in Neural Information Processing Systems*, p. None, 2003.

[65] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *KDD*, pp. 109–117. ACM, 2004.

[66] Z. Khan, T. Balch, and F. Dellaert, "Mcmc-based particle filtering for tracking a variable number of interacting targets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805–1819, 2005.

[67] Y. Huang and I. Essa, "Tracking multiple objects through occlusions," in *CVPR*, vol. 2, pp. 1051–1058. IEEE, 2005.

[68] X. Song, J. Cui, H. Zha, and H. Zhao, "Vision-based multiple interacting targets tracking via on-line supervised learning," in *ECCV*, pp. 642–655. Springer, 2008.

[69] R. Gross, I. Matthews, and S. Baker, "Active appearance models with occlusion," *Image and Vision Computing*, vol. 24, no. 6, pp. 593–604, 2006.

[70] S. Gauglitz, T. Höllerer, and M. Turk, "Evaluation of interest point detectors and feature descriptors for visual tracking," *International Journal of Computer Vision*, vol. 94, no. 3, pp. 335–360, 2011.

[71] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *ECCV*, pp. 430–443. Springer, 2006.

[72] J. Kwon, H. S. Lee, F. C. Park, and K. M. Lee, "A geometric particle filter for template-based visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 625–643, 2014.

[73] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, Feb 2004.

[74] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," in *ICRA*, pp. 1843–1848, April 2004.

[75] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

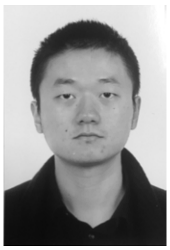
[76] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, 2015.

[77] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *ICCV*, pp. 4310–4318, 2015.

Xi Li is currently a full professor at the Zhejiang University, China. Prior to that, he was a senior researcher at the University of Adelaide, Australia. From 2009 to 2010, he worked as a postdoctoral researcher at CNRS Telecomd ParisTech, France. In 2009, he got the doctoral degree from National Laboratory of Pattern Recognition, Chinsese Academy of Sciences, Beijing, China. His research interests include visual tracking, motion analysis, face recognition, web data mining, image and video retrieval.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818132, IEEE Transactions on Pattern Analysis and Machine Intelligence

Liming Zhao is currently a fifth-year PhD student in College of Computer Science at Zhejiang University, Hangzhou, China. His advisors are Prof. Xi Li and Prof. Yueting Zhuang. Earlier, he received his bachelor's degree in Software Engineering from Shandong University in 2013. His current research interests are primarily in computer vision and machine learning, especially deep learning, visual attention, object recognition, detection and segmentation.

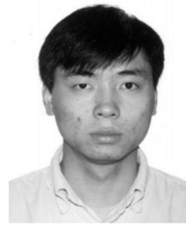


Wei Ji is currently a third-year PhD student in College of Computer Science at Zhejiang University, Hangzhou, China. His advisors are Prof. Xi Li and Prof. Yueting Zhuang. Earlier, he received his bachelor's degree in Computer Science and Technology from Nanjing University of Science and Technology in 2015. His current research interests are primarily in computer vision and machine learning, object recognition and detection.

Yiming Wu is now a third-year PhD student in College of Computer Science at ZheJiang University, Hang Zhou, China. His mentor is Professor Li Xi. Prior to that, he received a bachelor's degree in engineering from Beijing Jiaotong University. His current research direction is computer vision and machine learning, especially gesture recognition and tracking.

Fei Wu received the B.S. degree from Lanzhou University, Lanzhou, Gansu, China, the M.S. degree from Macao University, Taipa, Macau, and the Ph.D. degree from Zhejiang University, Hangzhou, China. He is currently a Full Professor with the College of Computer Science and Technology, Zhejiang University. He was a Visiting Scholar with Prof. B. Yu's Group, University of California, Berkeley, from 2009 to 2010. His current research interests include multimedia retrieval, sparse representation, and machine learning.

Ming-Hsuan Yang received the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2000. He is an associate professor in electrical engineering and computer science at the University of California, Merced. Prior to joining UC Merced in 2008, he was a senior research scientist at the Honda Research Institute working on vision problems of humanoid robots. He served as an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* from 2007 to 2011, and is an associate editor of the *International Journal of Computer Vision, Image and Vision Computing*, and *Journal of Artificial Intelligence Research*. He received the US National Science Foundation CAREER award in 2012, the Senate Award for Distinguished Early Career Research at UC Merced in 2011, and the Google Faculty Award in 2009. He is a senior member of the IEEE and the ACM.



Dacheng Tao received the BEng degree from the University of Science and Technology of China (USTC), the MPhil degree from the Chinese University of Hong Kong (CUHK), and the PhD degree from the University of London. Currently, he is a Nanyang assistant professor in the School of Computer Engineering at the Nanyang Technological University, a visiting professor at Xi Dian University, a guest professor at Wu Han University, and a visiting research fellow at the University of London. His research is mainly on applying statistics and mathematics for data analysis problems in data mining, computer vision, machine learning, multimedia, and visual surveillance. He has published more than 90 scientific papers in journals including IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Image Processing, etc., with best paper runner up awards. Previously, he gained several Meritorious Awards from the International Interdisciplinary Contest in Modeling, which is the highest level mathematical modeling contest in the world, organized by COMAP. He is an associate editor of the IEEE Transactions on Knowledge and Data Engineering, Neurocomputing (Elsevier), and Computational Statistics & Data Analysis (Elsevier). He is a member of the IEEE.

Ian Reid received the BSc degree in computer science and mathematics with first class honors from the University of Western Australia in 1987 and was awarded a Rhodes Scholarship in 1988 in order to study at the University of Oxford, where he received the DPhil degree in 1991. He is a professor of computer science at the University of Adelaide. His research interests include active vision, visual navigation, visual geometry, human motion capture, and intelligent visual surveillance, with an emphasis on real-time aspects of the computations.