Robust Visual Tracking With Multitask Joint Dictionary Learning

Heng Fan and Jinhai Xiang

***Abstract*— Dictionary learning for sparse representation has been increasingly applied to object tracking, however, the existing methods only utilize one modality of the object to learn a single dictionary. In this paper, we propose a robust tracking method based on multitask joint dictionary learning. Through extracting different features of the target, multiple linear sparse representations are obtained. Each sparse representation can be learned by a corresponding dictionary. Instead of separately learning the multiple dictionaries, we adopt a multitask learning approach to learn the multiple linear sparse representations, which provide additional useful information to the classification problem. Because different tasks may favor different sparse representation coefficients, yet the joint sparsity may enforce the robustness in coefficient estimation. During tracking, a classifier is constructed based on a joint linear representation, and the candidate with the smallest joint decision error is selected to be the tracked object. In addition, reliable tracking results and augmented training samples are accumulated into two sets to update the dictionaries for classification, which helps our tracker adapt to the fast time-varying object appearance. Both qualitative and quantitative evaluations on CVPR2013 visual tracking benchmark demonstrate that our method performs favorably against state-of-the-art trackers.**

***Index Terms*— Joint dictionary learning, multitask learning (MTL), sparse coding, visual tracking.**

I. Introduction

VISUAL tracking is one of the most fundamental components of computer vision with many applications, such as surveillance, human–computer interaction, and robotics [1]. In order to obtain a robust adaptive tracker, numerous methods have been proposed. Despite reasonable good results of these approaches, object tracking remains a challenge due to appearance changes caused by occlusion, illumination, pose, and motion. To address these problems, a wide range of appearance models have been presented [2]. Roughly speaking, these appearance models can be categorized into two types: 1) based on a discriminative model [3], [6], [8], [9], [12], [13] and 2) based on a generative model [4], [7], [10], [11].

Recently, sparse representation has drawn increasing attention in computer vision, such as object recognition [20], [21], detection [22], and classification [23], [24], because of its capability to capture most essential information of the object and resist noise, which are desirable for modeling a robust appearance. Inspired by this, many sparse representation-based trackers [5], [15]–[19], [28], [46]–[48] are proposed and achieved good performance to some extent. They represent the object appearance model with a dictionary learned from just one certain feature (e.g., shape or texture or color) of the target. In this case, the dictionary may be only discriminative to the particular situation. For example, the dictionary learned from texture feature may be robust to illumination variation; however, it fails to distinguish the target in the presence of deformation. Likewise, color feature is effective to deal with deformation but sensitive to occlusion. Therefore, one problem for designing a robust discriminative appearance model, which is able to handle multiple complex scenes, is how to combine these multiple features in an effective way.

Multitask learning (MTL) [37] has received a lot of research interests in machine learning and computer vision. The idea behind this paradigm is that, when the tasks to be learned share some latent factors and are similar enough or related in some scene, it may be advantageous to consider these relations between tasks in the model. A large body of works has provided evidence on the benefit of such a framework in the problems of computer vision, such as objection recognition [38], classification [39], and visual tracking [17], [40].

In this paper, we exploit multiple features of the object for modeling its appearance. For each modality of feature, a corresponding discriminative dictionary can be obtained. Instead of separately learning multiple dictionaries, a joint way is proposed to learn these dictionaries, which is called multitask joint dictionary learning (MJDL). By using the term multitask, we mean that there are more than one linear representation model, which is simultaneously estimated with proper regularization on parameters across all the models. For instance, given a set of training samples, we extract K different features (e.g., color, shape, and texture) for each sample and can obtain K different linear sparse representations. Each linear representation can learn a dictionary. We adopt a joint learning way to combine the K linear sparse representations, which provide additional useful information to the classification problem, since different tasks may favor different sparse representation coefficients, yet the joint sparsity may enforce the robustness in coefficient estimation. After obtaining the dictionaries, the quality of each tracking candidate is measured based on a joint linear representation. Fig. 1 illustrates the proposed tracking algorithm. Considering object appearance changes, the MJDL algorithm adaptively updates the dictionaries and classifier by using the tracked target in a new frame.

Manuscript received January 6, 2015; revised June 25, 2015 and August 11, 2015; accepted October 7, 2015. Date of publication January 8, 2016; date of current version May 3, 2017. This work was supported by the Fundamental Research Funds for the Central Universities under Program 2014BQ083. This paper was recommended by Associate Editor J.-M. Odobez. *(Corresponding author: Jinhai Xiang.)*

H. Fan is with the College of Engineering, Huazhong Agricultural University, Wuhan 430074, China (e-mail: hfan@webmail.hzau.edu.cn).

J. Xiang is with the College of Informatics, Huazhong Agricultural University, Wuhan 430074, China (e-mail: jimmy_xiang@mail.hzau.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2016.2515738

1051-8215 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

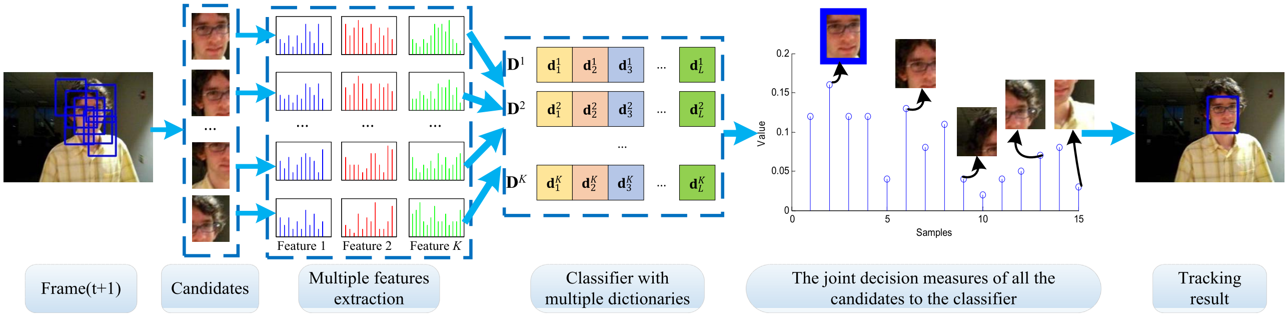

Feature 1
Feature 2
Feature K
$\mathbf{D}^1$
$\mathbf{D}^2$
$\mathbf{D}^K$
$\mathbf{d}_1^1$ $\mathbf{d}_2^1$ $\mathbf{d}_3^1$ ... $\mathbf{d}_L^1$
$\mathbf{d}_1^2$ $\mathbf{d}_2^2$ $\mathbf{d}_3^2$ ... $\mathbf{d}_L^2$
$\mathbf{d}_1^K$ $\mathbf{d}_2^K$ $\mathbf{d}_3^K$ ... $\mathbf{d}_L^K$
Value
0.2
0.15
0.1
0.05
0
Samples
0
5
10
15
Frame(t+1)
Candidates
Multiple features extraction
Classifier with multiple dictionaries
The joint decision measures of all the candidates to the classifier
Tracking result

Fig. 1. Proposed tracking method. First, we sample the candidates in frame $(t+1)$ within Bayesian framework and then extract multiple features for each candidate. After obtaining the features, a classifier based on MJDL is utilized to compute the joint decision measure of each candidate. The candidate with the maximum joint decision measure is selected to be the tracking result.

Through the above analysis, the main contributions of this paper can be summed up in the following three aspects.

1) First, we exploit multiple features of the object for developing a discriminative appearance model and propose a robust tracking method based on MJDL. In this paper, through extracting multiple different features of the target, multiple linear sparse representations are obtained. Each sparse representation can be learned by a corresponding dictionary. Instead of separately learning these multiple dictionaries, we propose an MTL strategy to learn these multiple linear sparse representations, which provide additional useful information to differentiate the object from the background.
2) Second, a joint classifier is utilized to improve tracking accuracy. In contrast to previous works, which only rely on reconstruction error to evaluate the reliability of candidates, we use both reconstruction errors and classification errors of multiple modalities to evaluate each candidate and propose a joint decision measure to compute the probability of each candidate being the target. Through this way, the tracking results are more reliable.
3) Third, the proposed tracker achieves favorable results with 51% in success plots and 67.8% in precision plots based on the CVPR2013 visual tracking benchmark [1], showing the power of MJDL.

The remainder of this paper is organized as follows. Section II briefly reviews the related works. The details of MJDL are given in Section III. Section IV describes the proposed tracking method. Experiments are shown in Section V and Section VI concludes this paper.

II. Related Work

Numerous studies have been done on visual tracking. Several classic algorithms have been presented and achieved impressive results. In [4], the incremental visual tracking introduces an online approach for efficiently learning and updating a low-dimensional principal component analysis (PCA) subspace representation for the object. However, this PCA subspace-based representation scheme is sensitive to partial occlusion. Kalal *et al.* [9] suggested a P-N learning algorithm to learn effective features from both positive and negative samples for object tracking. Kwon and Lee [10] decomposed the appearance model into multiple basic observation models to cover a wide range of illumination and deformation.

Owing to the strong representative power of sparse coding, many sparse representation-based tracking methods have been proposed. Mei and Ling [16] are the first to employ a sparse representation to track an object. However, it only simply utilizes holistic target templates to construct the dictionary ignoring background information, and computes sparse coefficients by solving ℓ_1 minimization. No dictionary learning and systematic update strategy are adopted, which makes the tracker sensitive to object appearance variations. Zhang *et al.* [19] proposed a tracking algorithm by learning a discriminative dictionary using both target information and background information. Liu *et al.* [5] constructed a dictionary by a k-selection approach before tracking. Although this method considers background information in dictionary learning, the dictionary is fixed during the whole process, therefore may not be adaptive, and the object appearance changes. To better improve the discriminative power, Zhong *et al.* [15] combined a sparsity based on both global and local representations. Nevertheless, the two parts are mutually independent and combined in a heuristic way. Jia *et al.* [42] suggested an alignment pooling approach to acquire global sparse representations from local object patches. The templates are dynamically updated to capture object appearance changes via substituting old templates with the new ones; however, no dictionary learning is utilized in this method. Zhuang *et al.* [47] proposed a tracking algorithm based on a discriminative sparse similarity map, which is obtained via a multitask reverse sparse coding approach with Laplacian constraint. Wang *et al.* [48] presented an online nonnegative dictionary learning method for updating the target templates, so that each learned template can capture a distinctive aspect of the object. Wu *et al.* [46] presented a structural appearance model via multiscale max pooling on weighted local sparse codes and the online multiple instance metric learning and applied it to visual tracking.

For sparse representation, the dictionary is very crucial. To improve the representative and discriminative power of the dictionary for sparse coding, varieties of dictionary learning methods are proposed. Unsupervised dictionary learning algorithms aim to minimize the residual for reconstruction. In particular, group features with the k-means clustering algorithm are employed in [31]. The K-singular value decomposition (K-SVD) algorithm [32], [33] generalizes the k-means clustering algorithm to learn an overcomplete dictionary. A dictionary learned by these approaches can well reconstruct the object but may not be suitable for classification. Recently, supervised dictionary learning methods have been presented for better classification [25], [26]. A simple yet effective way is to learn a dictionary for each class label, which assigns a specific label to each dictionary item. In this way, the dictionary learned is both reconstructive and discriminative. Jiang *et al.* [27] presented a discriminative dictionary learning method for sparse coding in image classification. This method improves both discriminative and reconstructive power of the learned dictionary by using class labels of training data and associating label information with each dictionary item.

The most related works to ours are [17], [28], and [40]. In [28], a discriminative dictionary learning method is proposed for visual tracking, in which the discriminative power of the dictionary is obtained by minimizing both the reconstruction error and the classification error. Although we adopt this strategy to learn the dictionary, the proposed tracker is different from [28]. In this paper, we take advantages of multiple different features of the target, which is different from [28] where only one certain feature is utilized. With the help of MTL, we can learn multiple dictionaries from these features in a joint way and construct a discriminative appearance model for robust visual tracking. Zhang *et al.* [17] presented an MTL tracking framework, in which they exploits the interdependences between particles to improve tracking performance and overall computational complexity. In this paper, nevertheless, we propose an MJDL method to learn the relationship between multiple features in appearance modeling. Though using MTL, our approach is different from [17] in two aspects. First, Zhang *et al.* [17] applied MTL to learning the interdependences between particles, and only one modality of the target is utilized in this process, however, we use MTL to learn the relationship of multiple different features in appearance modeling. Second, we adopt a discriminative dictionary learning algorithm for the sparse representation, while no learning method is adopted in [17]. Mei *et al.* [40] proposed a similar tracking method, where MTL is utilized to learn the underlying relationship between multiple sparse representations of different features, while the proposed method varies from [40] in the dictionary learning. In [40], the templates for sparse representation are obtained from target without learning, while, in this paper, the target templates are learned from both foreground information and background information. Through this way, we can develop a more discriminative and robust appearance model.

III. Multitask Joint Dictionary Learning

A. Problem Formulation

Assume that $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\} \in \mathbb{R}^{d\times K\times N}$ is a training set with N target and background templates, in which each sample has K different modalities of features (e.g., color, shape, and texture), and $\mathbf{Y} = \{-1, 1\}$ are the class labels. Our goal is to learn multiple dictionaries, which are discriminative to distinguish the target from the background. For the ith sample, $\mathbf{X}_i = \{\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^K\} \in \mathbb{R}^{d\times K} (i = 1, 2, \ldots, N)$ denotes its feature matrix, in which $\mathbf{x}_i^k \in \mathbb{R}^d$ represents the kth$(k = 1, 2, \ldots, K)$ feature associated with the ith sample. Note that the dimensions of the K different features are stretched to the same in this paper in the experiments.[1] For each modality of feature, a corresponding dictionary can be learned. We use dictionary set $\mathbf{D} = \{\mathbf{D}^1, \mathbf{D}^2, \ldots, \mathbf{D}^K\} \in \mathbb{R}^{d\times L\times K}$ to represent the K dictionaries, in which each $\mathbf{D}^k = \{\mathbf{d}_1^k, \mathbf{d}_2^k, \ldots, \mathbf{d}_L^k\} \in \mathbb{R}^{d\times L} (k = 1, 2, \ldots, K)$ is a single dictionary with L items for the kth modality. Given the value of $\mathbf{D}$, each x_i^k can be reconstructed by a linear combination of few items from the kth dictionary $\mathbf{D}^k$ as

$$x_i^k = \mathbf{D}^k \mathbf{c}_i^k + \boldsymbol{\varepsilon}_i^k \tag{1}$$

where $\boldsymbol{\varepsilon}_i^k \in \mathbb{R}^d$ is the error vector and $\mathbf{c}_i^k \in \mathbb{R}^L$ is the sparse code of x_i^k and can be obtained by ℓ_1 minimization

$$\mathbf{c}_i^k = \arg\min_{\mathbf{c}} \left\|\mathbf{x}_i^k - \mathbf{D}^k\mathbf{c}\right\|_2^2 + \lambda_1\|\mathbf{c}\|_1 \tag{2}$$

where λ_1 is a tradeoff parameter between reliable reconstruction and sparse regularization. After obtaining the $\mathbf{c}_i^k$, it can be directly utilized as a feature for classification as follows:

$$\mathbf{y}_i^k = f\left(\mathbf{W}^k, \mathbf{c}_i^k\right) + \epsilon_i^k \tag{3}$$

where $f(\mathbf{W}^k, \mathbf{c}_i^k) = \mathbf{W}^k\mathbf{c}_i^k$ is a linear classifier, $\mathbf{y}_i^k = [0, \ldots, 1, \ldots, 0]^T \in \mathbb{R}^m$ is a label vector ($m = 2$ in tracking problem), in which the nonzero index indicates the class label of $\mathbf{x}_i^k$, $\mathbf{W}^k \in \mathbb{R}^{m\times L}$ is a classification parameter, and $\epsilon_i^k \in \mathbb{R}^m$ is the residual term. $\{\mathbf{W}^k\}_{k=1}^K$ can be estimated by fitting the following least squared regression (LSR) model:

$$\arg\min_{\mathbf{W}} \left\{\mathcal{F}(\mathbf{W}) := \sum_{k=1}^{K}\sum_{i=1}^{N} \left\|\mathbf{y}_i^k - \mathbf{W}^k\mathbf{c}_i^k\right\|_2^2\right\} \tag{4}$$

where $\mathbf{W} = \{\mathbf{W}^1, \mathbf{W}^2, \ldots, \mathbf{W}^K\} \in \mathbb{R}^{m\times L\times K}$ is a set of classification parameters. To avoid singularity of linear systems, an additional regularization term $\lambda_2\|\mathbf{W}\|_F^2$ is typically imposed in (4) and it can be rewritten as

$$\arg\min_{\mathbf{W}} \left\{\mathcal{F}(\mathbf{W}) := \sum_{k=1}^{K}\left\{\sum_{i=1}^{N} \left\|\mathbf{y}_i^k - \mathbf{W}^k\mathbf{c}_i^k\right\|_2^2 + \lambda_2\|\mathbf{W}^k\|_F^2\right\}\right\}. \tag{5}$$

From the viewpoint of MTL, problem (5) is a multitask regression model with $K \times N$ independent LSR models.

Formulation (5), however, does not consider the reconstruction error of sparse coding, which makes $\mathbf{D}$ suboptimal for classification. To combine the power of reconstruction for classification, we improve the multitask regression model by

[1] For each feature, we reshape it into a feature vector with dimension $1 \times \mathfrak{n}$. For example, for the features, such as image pixels denoted by a matrix $\mathbf{A}$ with size $\mathfrak{p} \times \mathfrak{q}$, we can reshape it into a feature vector $\mathfrak{f}$ with dimension $1 \times (\mathfrak{p} \times \mathfrak{q})$ by columns. For all the features with different dimensions, we stretch them to the same by filling low-dimension feature vectors with zero vectors. For instance, $\mathfrak{f}_1$, $\mathfrak{f}_2$, and $\mathfrak{f}_3$ are different features with dimension $1 \times \mathfrak{n}_1$, $1 \times \mathfrak{n}_2$, and $1 \times \mathfrak{n}_3$ respectively, where $\mathfrak{n}_1 < \mathfrak{n}_2 < \mathfrak{n}_3$. We then keep $\mathfrak{f}_3$ unchanged and stretch $\mathfrak{f}_1$ and $\mathfrak{f}_2$ by filling zero vectors, and we can get two new feature vectors $\mathfrak{f}_1^{'} = [\mathfrak{f}_1\ \boldsymbol{I}_0^1]$ and $\mathfrak{f}_2^{'} = [\mathfrak{f}_2\ \boldsymbol{I}_0^2]$, where $\mathbf{I}_0^1$ and $\mathbf{I}_0^2$ are zero vectors, and their dimensions are $1 \times (\mathfrak{n}_3 - \mathfrak{n}_1)$ and $1 \times (\mathfrak{n}_3 - \mathfrak{n}_2)$, respectively. Through this way, all the features are with the same dimension.

imposing a sparse coding error item as in [27]. Then, (5) can be rewritten as follows:

$$\arg\min_{\mathbf{D},\mathbf{W}} \left\{ \mathcal{F}(\mathbf{D},\mathbf{W}) := \sum_{k=1}^{K} \left\{ \sum_{i=1}^{N} \left\{ (1-\mu)\left\|\mathbf{y}_i^k - \mathbf{W}^k\mathbf{c}_i^k\right\|_2^2 + \mu\left\|\mathbf{l}_i^k - \mathbf{c}_i^k\right\|_2^2 \right\} + \lambda_2\|\mathbf{W}^k\|_F^2 \right\} \right\}$$

$$\text{s.t. } \mathbf{c}_i^k = \arg\min_{\mathbf{c}} \left\|\mathbf{x}_i^k - \mathbf{D}^k\mathbf{c}\right\|_2^2 + \lambda_1\|\mathbf{c}\|_1 \tag{6}$$

where $\|\mathbf{l}_i^k - \mathbf{c}_i^k\|_2^2$ is the sparse coding error and $\mathbf{l}_i^k = [l_{i,1}^k, l_{i,2}^k, \ldots, l_{i,L}^k]^T = [0, \ldots, 1, 1, \ldots, 0]^T \in \mathbb{R}^L$ is the ideal discriminative sparse code for $\mathbf{x}_i^k$. If $l_{i,j} = 1$ $(j = 1, 2, \ldots, L)$, $\mathbf{x}_i^k$ and the dictionary item $\mathbf{d}_j^k$ share the same label, while $l_{i,j} = 0$ means that they belong to different classes. For instance, let $\mathbf{D}^k = \{\mathbf{d}_1^k, \mathbf{d}_2^k, \mathbf{d}_3^k, \mathbf{d}_4^k, \mathbf{d}_5^k\} \in \mathbb{R}^{d\times 5}$ be the dictionary for the kth, and $\mathbf{x}_i^k$ represents the kth modality of the ith training sample. If $\mathbf{x}_i^k$ is in the same class with dictionary items $\mathbf{d}_1^k$, $\mathbf{d}_2^k$, and $\mathbf{d}_3^k$, and different classes with $\mathbf{d}_4^k$ and $\mathbf{d}_5^k$, then the ideal discriminative sparse code $\mathbf{l}_i^k$ can be defined as $\mathbf{l}_i^k = [1, 1, 1, 0, 0]^T$.

The variable μ controls the contributions of the sparse coding error and the reconstruction error. Through formulation (6), the dictionaries, which are learned in a joint way, are both reconstructive and discriminative.

B. Optimization

To start with, we give a definition of function set $\mathcal{L} = \{\mathcal{L}^1, \mathcal{L}^2, \ldots, \mathcal{L}^K\}$, where each $\mathcal{L}^k$ is defined as follows:

$$\mathcal{L}^k(\mathbf{D}^k,\mathbf{W}^k) = \sum_{i=1}^{N} \left\{ (1-\mu)\left\|\mathbf{y}_i^k - \mathbf{W}^k\mathbf{c}_i^k\right\|_2^2 + \mu\left\|\mathbf{l}_i^k - \mathbf{c}_i^k\right\|_2^2 \right\} + \lambda_2\|\mathbf{W}^k\|_F^2$$

$$\text{s.t. } \mathbf{c}_i^k = \arg\min_{\mathbf{c}} \left\|\mathbf{x}_i^k - \mathbf{D}^k\mathbf{c}\right\|_2^2 + \lambda_1\|\mathbf{c}\|_1. \tag{7}$$

Then, (6) can be rewritten as

$$\arg\min_{\mathbf{D},\mathbf{W}} \left\{ \mathcal{F}(\mathbf{D},\mathbf{W}) := \sum_{k=1}^{K} \mathcal{L}^k(\mathbf{D}^k,\mathbf{W}^k) \right\}. \tag{8}$$

The objective function in (8) is a nonlinear and nonconvex problem, therefore, we resort to stochastic gradient descent [26]. The gradient with respect to $\mathbf{W}^k$ is

$$\nabla_{\mathbf{W}^k}\mathcal{L}^k(\mathbf{D}^k,\mathbf{W}^k) = (1-\mu)\left(\mathbf{W}^k\mathbf{c}_i^k - \mathbf{y}_i^k\right)\left(\mathbf{c}_i^k\right)^T + \lambda_2\mathbf{W}^k. \tag{9}$$

However, the dictionary $\mathbf{D}^k$ is not explicitly defined in $\mathcal{L}^k$ but implicitly defined on the sparse code $\mathbf{c}_i^k$. To obtain the gradient with respect to $\mathbf{D}^k$, we use the implicit differentiation algorithm on the fixed point equations as in [26]. The gradient with respect to $\mathbf{D}^k$ can be acquired via[2]

$$\nabla_{\mathbf{D}^k}\mathcal{L}^k(\mathbf{D}^k,\mathbf{W}^k) = -\mathbf{D}^k\boldsymbol{\delta}^k\left(\mathbf{c}_i^k\right)^T + \left(\mathbf{x}_i^k - \mathbf{D}^k\mathbf{c}_i^k\right)(\boldsymbol{\delta}^k)^T \tag{10}$$

where $\boldsymbol{\delta}^k \in \mathbb{R}^L$ is a vector that relies on $\mathbf{c}_i^k$, $\mathbf{y}_i^k$, $\mathbf{l}_i^k$, $\mathbf{D}^k$, and $\mathbf{W}^k$ with

$$\boldsymbol{\delta}^k_{\bar{\wedge}} = 0 \text{ and } \boldsymbol{\delta}^k_{\wedge} = \left(\left(\mathbf{D}^k_{\wedge}\right)^T\mathbf{D}^k_{\wedge}\right)^{-1}\nabla_{\mathbf{c}_i^k}\mathcal{L}^k(\mathbf{D}^k,\mathbf{W}^k) \tag{11}$$

where $\wedge$ denotes the indices of the nonzero coefficients of all nonzero values in $\mathbf{D}^k$, and $\nabla_{\mathbf{c}_i^k}\mathcal{L}^k(\mathbf{D}^k,\mathbf{W}^k)$ can be obtained with

$$\nabla_{\mathbf{c}_i^k}\mathcal{L}^k(\mathbf{D}^k,\mathbf{W}^k) = (1-\mu)(\mathbf{W}^k)^T\left(\mathbf{W}^k\mathbf{c}_i^k - \mathbf{y}_i^k\right) + \mu\left(\mathbf{c}_i^k - \mathbf{l}_i^k\right). \tag{12}$$

Through this way, the gradients with respect to $\mathbf{W}^k$ and $\mathbf{D}^k$ are available. We adopt the learning rate in [26], which is $\min(\eta, \eta i_0/i)$, where η is a constant and $i_0 = Q/10$, where Q is the number of iteration. The MJDL is shown in Algorithm 1.

Algorithm 1 MJDL

```
Require: Training sample features $\mathbf{X}$ with labels $\mathbf{Y}$, $\lambda_1$,
         $\lambda_2$, $\mu$, $\eta$, $M$, $\mathbf{W}_1 = \{\mathbf{W}_1^k\}_{k=1}^K$, $\mathbf{D}_1 = \{\mathbf{D}_1^k\}_{k=1}^K$;
for $q = 1$ to $Q$
    for $k = 1$ to $K$
        Generate training sample $\mathbf{X}$;
        for $i = 1$ to $N$
            Obtain $\mathbf{y}_i^k$ for $\mathbf{x}_i^k$;
            Sparse coding: compute $\mathbf{c}_i^k$ according to
            Equation (2);
            Compute the active set $\wedge$ and variable $\boldsymbol{\delta}^k$
            via Equation (11);
            Choose learning rate $\eta_q = \min(\eta, \eta i_0/i)$;
            Compute the gradients of $\mathbf{W}_q^k$ and $\mathbf{D}_q^k$ via
            Equation (9) and (10)
            Update $\mathbf{W}_q^k$ and $\mathbf{D}_q^k$ with
            $\mathbf{W}_q^k = \mathbf{W}_q^k - \eta_q\nabla_{\mathbf{W}^k}\mathcal{L}_i^k(\mathbf{D}_q^k, \mathbf{W}_q^k)$ and
            $\mathbf{D}_q^k = \mathbf{D}_q^k - \eta_q\nabla_{\mathbf{D}^k}\mathcal{L}_i^k(\mathbf{D}_q^k, \mathbf{W}_q^k)$;
        end for
        Let $\mathbf{W}_{q+1}^k = \mathbf{W}_q^k$ and $\mathbf{D}_{q+1}^k = \mathbf{D}_q^k$;
    end for
    $\mathbf{W}_{q+1} = \{\mathbf{W}_{q+1}^k\}_{k=1}^K$ and $\mathbf{D}_{q+1} = \{\mathbf{D}_{q+1}^k\}_{k=1}^K$;
end for
Return: New $\mathbf{W}$ and $\mathbf{D}$;
```

C. Initialization

First, we randomly select N target and background templates, and extract K features for each sample to form $\mathbf{X}$. Then, the K-SVD algorithm [32], [33] is performed on training samples K times to initialize the dictionary $\mathbf{D}_0 = \{\mathbf{D}_0^k\}_{k=1}^K$. Given the dictionary, we calculate the sparse code $\mathbf{c}_i^k$ for $\mathbf{x}_i^k$ to form matric $\mathbf{C}^k$, which contains the sparse code of all samples for the kth modality, and then employ the ridge regression model [34] to obtain $\mathbf{W}_0^k$ by

$$\mathbf{W}_0^k = \arg\min_{\mathbf{W}_0^k} \left\|\mathbf{F}^k - \mathbf{W}_0^k\mathbf{C}^k\right\|^2 + \lambda_3\left\|\mathbf{W}_0^k\right\|_2^2 \tag{13}$$

where $\mathbf{F}^k$ is the label matrix of all samples for the kth modality. After obtaining $\mathbf{W}_0^k$, $\mathbf{W}_0$ can be initialized as $\mathbf{W}_0 = \{\mathbf{W}_0^k\}_{k=1}^K$.

[2] The details of the implicit differentiation algorithm on the fixed point equations can be shown in [26].

The solution for formulation (13) is

$$\mathbf{W}_0^k = \mathbf{F}^k(\mathbf{C}^k)^T(\mathbf{C}^k(\mathbf{C}^k)^T + \lambda_3\mathbf{I})^{-1} \tag{14}$$

where $\mathbf{I}$ is the identity matrix.

D. Classification

After learning the dictionaries, we can classify a test sample through multiple features. The key point is to combine the similarity between the new sample and the training set with the classification score from the classifier. In doing so, we adopt the joint decision measure strategy used in [28]. For a new sample $\mathbf{s}$, we extract its multiple feature matrix $\mathbf{X}_s = \{\mathbf{x}_s^1, \mathbf{x}_s^2, \ldots, \mathbf{x}_s^K\}$ and compute the corresponding sparse codes $\mathbf{C}_s = \{\mathbf{c}_s^1, \mathbf{c}_s^2, \ldots, \mathbf{c}_s^K\}$. Note that there are K modalities of features used in this paper, and thus, the joint decision measure can be defined as

$$\varphi(\mathbf{s}) = 1 - \sum_{k=1}^{K} \varphi^k(\mathbf{s}) \tag{15}$$

where $\varphi(\mathbf{s})$ is the joint decision measure, which represents the probability of $\mathbf{s}$ belonging to object class. The more the $\varphi(\mathbf{s})$ is, the more probably the s is the object. $\varphi^k(\mathbf{s})$ is the joint decision error under the kth modality and it can be obtained via

$$\varphi^k(\mathbf{s}) = (1-\alpha)\left\|\mathbf{s}_{\text{tr}}^k - \mathbf{D}^k\mathbf{c}_s^k\right\|^2 + \alpha\left\|\mathbf{y}_s^k - \mathbf{W}^k\mathbf{c}_s^k\right\|^2 \tag{16}$$

where $\mathbf{s}_{\text{tr}}^k$ represents the average of the real kth modality of the tracking results. Taking object appearance variations into account, we utilize the tracking results of several latest frames instead of just one frame to calculate the kth modality of the tracking results. We collect the kth modalities of these latest into a set, which is dynamic (see T^k in Section IV-A), and compute the average of the set to obtain $\mathbf{s}_{\text{tr}}^k$; $\|\mathbf{s}_{\text{tr}}^k - \mathbf{D}^k\mathbf{c}_s^k\|^2$ and $\|\mathbf{y}_s^k - \mathbf{W}^k\mathbf{c}_s^k\|^2$ are the reconstruction error and the classification error of the kth modality, respectively, α is a tradeoff parameter, and $\mathbf{y}_s^k = [0, \ldots, 1, \ldots, 0]^T \in \mathbb{R}^m$ is a label vector, in which the nonzero index indicates the corresponding class. In this paper, m is set to 2 because there are only two classes: target and background. Therefore, the label vector $\mathbf{y}_s^k$ is fixed to $[1, 0]^T$ (for target class) or $[0, 1]^T$ (for background class).

IV. Proposed Tracking Method

A. Tracking Formulation

Our tracker is implemented via the Bayesian framework. Given the observation set of targets $Y^t = \{y_1, y_2, \ldots, y_t\}$ up to the frame t, where y_τ ($\tau = 1, 2, \ldots, t$) represents the observation of target in frame τ, we can obtain estimation $\widehat{X}_t$ by computing the maximum *a posterior* via

$$\widehat{X}_t = \arg\max_{X_t^i} p\left(X_t^i | Y^t\right) \tag{17}$$

where $\widehat{X}_t$ denotes the ith sample at the state of X_t. The posterior probability $p(X_t^i|Y^t)$ can be obtained by the Bayesian theorem recursively via

$$p(X_t|Y^t) \propto p(y_t|X_t)\int p(X_t|X_{t-1})p(X_{t-1}|Y^{t-1})dX_{t-1} \tag{18}$$

where $p(X_t|X_{t-1})$ and $p(y_t|X_t)$ represent the dynamic model and the observation model, respectively.

The dynamic model indicates the temporal correlation of the target state between consecutive frames. We apply affine transformation to model the target motion between two consecutive frames within the particle filter framework. The state transition can be formulated as

$$p(X_t|X_{t-1}) = \mathcal{N}(X_t; X_{t-1}, \Psi) \tag{19}$$

where Ψ is a diagonal covariance matrix whose elements are the variance of affine parameters. The observation model $p(y_t|X_t)$ represents the probability of the observation y_t at state X_t. In this paper, the observation is designed by

$$p(y_t|X_t) \propto \varphi(X_t) \tag{20}$$

where $\varphi(X_t)$ is the joint decision measure of the tth candidate. Through Bayesian inference, we can determine the candidate sample with the maximum joint decision measure as the tracking result.

To obtain the reconstruction error $\|\mathbf{s}_{\text{tr}}^k - \mathbf{D}^k\mathbf{c}_s^k\|^2$ in (16), we accumulate multiple features extracted from the tracking results into K sets $\{T^k\}_{k=1}^K$, in which each T^k is used to store the kth features of the tracking results. For the kth modality, the feature of the tracking result in current frame is added to T^k, while those from older frames are deleted from T^k, so that each T^k has a fixed number of elements, denoted by U. We assign each element in T^k the weight $w^k = e^{-\varphi^k}$, where φ^k is the joint decision error under the kth modality. Note that for each newly added element, its weight is computed for only one time based on its joint decision error and will keep unchangeable in the set T^k. $\mathbf{s}_{\text{tr}}^k$ in (16) is then computed as the weighted average of the elements in T^k, since the elements with different reliabilities should have a different importance on the combined sample $\mathbf{s}_{\text{tr}}^k$. It is worth noticing that we just use the normalization of the weights to compute their weighted average and do not change their weights. Thus, the weights of elements are actually invariable and determined when they are added into T^k. Initially, each T^k comprises only one element, i.e., the selected target box, whose weight is 1.

B. Online Update

Due to the appearance variations of target, update is essential. In this paper, an effective mechanism is proposed to update $\mathbf{D}$ and $\mathbf{W}$ periodically.

To start with, we design a set S. In each frame, after locating the target, we randomly extract some positive and negative samples for updating a dictionary. In this paper, we learn the dictionary using both target information and background information. Through this way, the dictionary learned is much more discriminative than that learned by only using target information. In doing so, we need to extract both positive samples containing more target information and negative samples containing more background information. Therefore, the positive samples should be around the target, and the negative samples are far away from the object. It is worth noticing that, however, far away from the target does not mean that the negative samples could be extracted at any

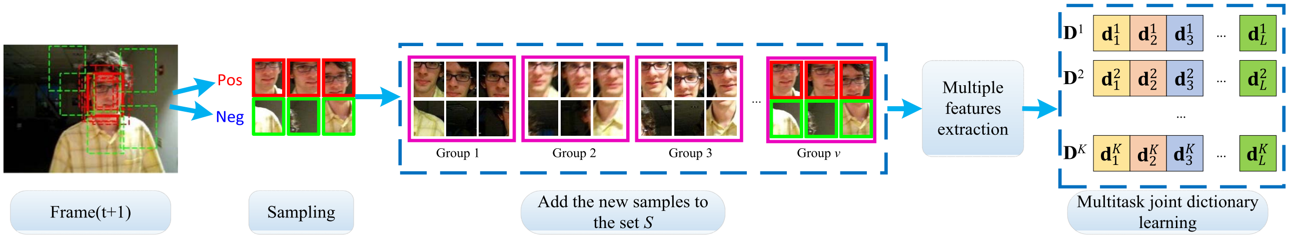

Pos
Neg
Frame(t+1)
Sampling
Group 1
Group 2
Group 3
...
Group v
Add the new samples to the set S
Multiple features extraction
$\mathbf{D}^1$ $\mathbf{d}_1^1$ $\mathbf{d}_2^1$ $\mathbf{d}_3^1$ ... $\mathbf{d}_L^1$
$\mathbf{D}^2$ $\mathbf{d}_1^2$ $\mathbf{d}_2^2$ $\mathbf{d}_3^2$ ... $\mathbf{d}_L^2$
...
$\mathbf{D}^K$ $\mathbf{d}_1^K$ $\mathbf{d}_2^K$ $\mathbf{d}_3^K$ ... $\mathbf{d}_L^K$
Multitask joint dictionary learning

Fig. 2. Illustration of the update process.

positions in the frame, it is relative distance. By controlling the distance from the tracked object, we ensure that most negative samples contain pure background, so that they are capable of differentiating from the target to the most extent. These samples are collected as a group, which consists of two subgroups: 1) the positive subgroup to store positive samples and 2) the negative subgroup to store negative samples and is added into the set S. When the set size υ reaches a threshold V, we extract multiple features from all the groups in S and apply MJDL to update dictionaries, and empty S in the end.

However, when accumulating elements into S, the tracking result may contain significant noise, and thus, is not reliable, if the optimal location of the bounding box determined by our tracker has a high reconstruction error $\varepsilon(\mathbf{s})_{\text{re}} = (1/K)\sum_{k=1}^{K} \|\mathbf{s}_{\text{tr}}^k - \mathbf{D}^k\mathbf{c}_s^k\|^2$ or a high classification error $\varepsilon(\mathbf{s})_{\text{cl}} = (1/K)\sum_{k=1}^{K} \|\mathbf{y}_s^k - \mathbf{W}^k\mathbf{c}_s^k\|^2$. In this case, we skip this frame to avoid introducing noise into S. Two thresholds t_{re} and t_{cl} are adopted to determine whether this frame is skipped. If $\varepsilon(\mathbf{s})_{\text{re}} \geq t_{\text{re}}$ or $\varepsilon(\mathbf{s})_{\text{cl}} \geq t_{\text{cl}}$, the tracking result is not reliable and this frame will be skipped; otherwise it will be added into S. The update process can be shown in Fig. 2. Note that when a frame is skipped, the kth feature of the tracking result is not added into the set T^k as well.

So far, we have introduced the overall procedure of the proposed tracking algorithm, as shown in Algorithm 2.

Algorithm 2 Tracking by MJDL

```
Require: Frames $1, 2, \cdots, t, \cdots$;
Initialization:
  Select initial target $X_1$ and sample $N^+$ positive
    samples and $N^-$ negative samples;
  Extract multiple features from samples to form
    $\mathbf{X}$ with label $\mathbf{Y}$;
  Initialize $\mathbf{W}_1 = \{\mathbf{W}_1^k\}_{k=1}^K$, $\mathbf{D}_1 = \{\mathbf{D}_1^k\}_{k=1}^K$;
  Add $K$ features of $X_1$ and $\mathbf{X}$ to $\{T_k\}_{k=1}^K$
    and $S$ respectively;
Tracking:
  for $t = 2$ to the end of the sequence
    Sample $J$ candidates around $X_{t-1}$;
    Extract $K$ features from each candidate
      and compute the corresponding sparse codes;
    Compute joint decision measure error for each
      candidate based on Equation (15) and (16);
    Select the candidate with smallest joint
      decision measure error to be the tracking
      result $X_t$;
    Sample $N^+$ positive samples and $N^-$ negative
      samples to form new $\mathbf{X}_{new}$;
    Add $K$ features of $X_t$ and $\mathbf{X}_{new}$ to $\{T_k\}_{k=1}^K$
      and $S$ respectively;
    For each $T^k$, if its size is greater than $U$, delete
      the oldest element from $T^k$;
    If the size of set $S$ is equal to $V$, update
      $\mathbf{D}$ and $\mathbf{W}$ according to Algorithm 1, and then
      empty set $S$;
  end for
Return: Tracking results $X_1, X_2, \cdots, X_t$;
```

V. Experimental Results

A. Setup

1) Parameter Setting: The proposed algorithm is implemented in MATLAB on a 3.2-GHz Intel E3-1225 v3 Core PC with 8-GB memory. Since we do not update the dictionaries every frame, our implementation is very efficient. The average frames per second is 4. In our experiment, we select two modalities of features, i.e., histogram of oriented gradient (HOG) feature [35] and local binary pattern (LBP) feature [36]. Both the two features are implemented in the VLFeat tool.[3] The cell size of HOG is 8 × 8, the block size is 16 × 16, and its number of bins is set to 9. The parameters P and R in LBP are 8 and 1, respectively, and the patter of LBP is uniform. The parameters of the proposed tracker are as follows. The size of each dictionary is fixed to 200, which comprises 100 items for positive samples and 100 items for negative samples. The number of particles in the Bayesian framework is set to 300–800. The iteration numbers for initialization and learning are 5 and 30, respectively. The learning rate is set to 0.2. In the initial frame, both the numbers of positive and negative samples are 200 for initialization and 100 for update, respectively. The sizes for the set T and S are fixed to 20 and 5, respectively. The thresholds t_{re} and t_{cl} are both empirically set to 0.3 according to our experiment results.

2) Data Set: We evaluate the proposed algorithm on the CVPR2013 tracking benchmark [1], which contains the results of 29 tracking algorithms on 50 fully annotated videos (∼26 000 frames). For better evaluation and analysis of the strength and weakness of the tracking algorithms, the sequences are categorized according to 11 attributes, including illumination variation, scale variation, occlusion, deformation,

[3] VLFeat is an open source library and available at http://www.vlfeat.org/.

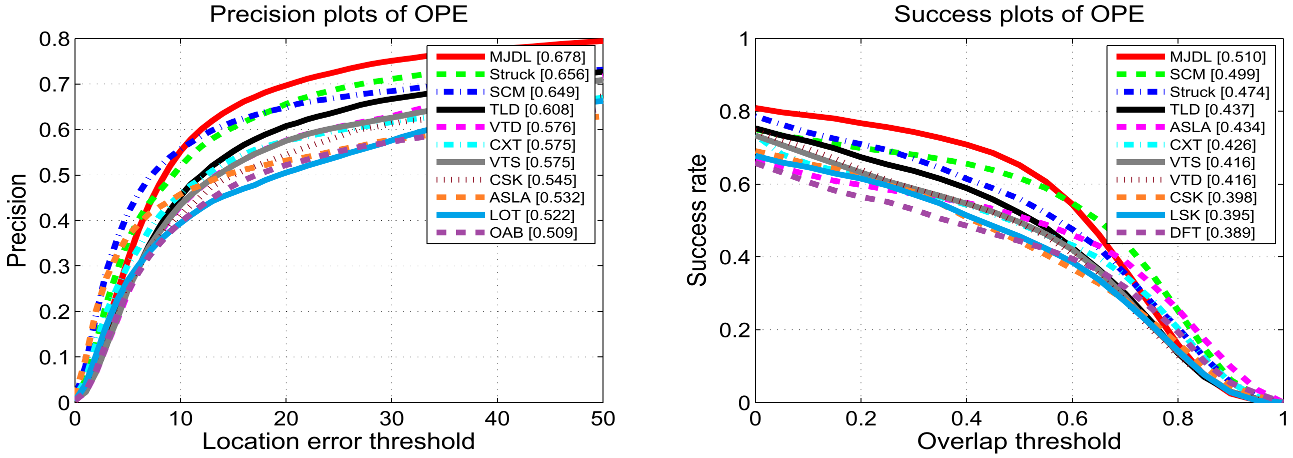


Fig. 3. Precision plots and success plots of OPE for the proposed tracker and the top ten trackers in the benchmark. The performance score for each tracker is shown in the legend. The performance score of precession plot is at error threshold of 20 pixels, while the performance score of success plot is the AUC value. Best viewed on color display.

motion blur, fast motion, in-plane rotation, out-of-plane, out of view, background clutter, and low resolution.

3) Evaluation Metric: We employ the precision plot and the success plot defined in [1] to evaluate the robustness of the tracking algorithms. The precision plot demonstrates the percentage of frames whose estimated average center location errors are within the given threshold distance to the ground truth, in which the average center location is defined as the average Euclidean distance between the center locations of the tracker target and the manually labeled ground truth. The score at the threshold 20 pixels is defined as the precision score. Success plot shows the percentage of successful frames at the threshold ranging from 0 to 1. The successful frame is defined as the overlap score more than a fixed value, where the overlap ratio is defined as score $= (\text{area}(R_{\text{GT}} \cap R_T)/\text{area}(R_{\text{GT}} \cup R_T))$ with the groundtruth R_{GT} and the tracking result R_T. For fair evaluation, the area under curve (AUC) is preferred to measure the success ratio. The one-pass evaluation (OPE) based on the average precision and the success rate given the groundtruth of the first frame is used to evaluate the robustness of our algorithm.

B. Quantitative Comparison

1) Overall Performance: Fig. 3 demonstrates the overall comparison of the proposed tracker and top ten evaluated tracking methods (e.g., SCM [15], Struck [41], TLD [9], ASLA [42], CXT [43], VTS [11], VTD [10], CSK [44], LSK [5], DFT [45], OAB [8], and LOT [30]) in terms of precision plot and success plot. Note that SCM, Struck, TLD, ALSA, CXT, VTS, VTD, CSK, LOT, and OAB are top ten trackers in terms of precision plot; while SCM, Struck, TLD, ALSA, CXT, VTS, VTD, CSK, LSK, and DFT are top ten trackers in terms of success plot. Therefore, we compare our tracker with 12 tracking methods. The proposed MJDL obtains favorable results in terms of both precision plot and success plot: the precision score of MJDL is 0.678; meanwhile, in the success plot, our MJDL achieves the score of AUC 0.510.

2) Attribute-Based Performance: To facilitate analyzing strength and weakness of the proposed algorithm, we further evaluate MJDL on videos with 11 attributes. Since the AUC score of the success plot is more accurate than that at the representative threshold (e.g., 20 pixels) of the precision plot, in the following, we mainly analyze MJDL based on the success plot.

Fig. 4 shows the success plot of videos with attributes that our method achieves favorable results, in which MJDL ranks within top 3 on 9 out of 11 attributes. For the videos with attributes, such as background clutter, in-plane rotation, out-of-plane rotation, deformation, and low resolution, MJDL ranks first among all evaluated algorithms. For the sequences with occlusion, scale variation and illumination variation, MJDL ranks second among the evaluated algorithms, while SCM ranks first. Both MJDL and SCM represent the object with sparse representation. MJDL exploits multiple features of the object and learn multiple sparse dictionaries in a joint way, while the SCM learns the local features from target with sparse representation. Furthermore, both MJDL and SCM utilize the background information to improve the discriminative power of the dictionary. On videos with motion blur, our MJDL is still able to locate the target, which can be attributed to the discriminative appearance model by combining multiple discriminative features in a joint way.

Fig. 5 shows that MJDL cannot perform well with two attributes, such as out of view and fast motion. For out-of-view attribute, the MJDL is not able to handle this case. For fast motion attribute, MJDL ranks fourth, while the top three (i.e., Struck, TLD, and CXT) track the object based on dense sampling and search for the target in the whole frame by sliding window, while our MJDL only relies on a simple dynamic model based on stochastic search.

To further qualitatively analyze the performance of MJDL, Fig. 6 shows some sampled results on the benchmark. We can observe that MJDL favorably performs on 38 out of 50 videos without suffering from severe drift. Note that there exist many challenging factors in these videos that MJDL achieves favorable results. For example, the videos *Car4*, *CarDark*, *David*, *Mhyang*, *Singer1*, *Sylvester*, and *Trellis* also have the deformation and in- or out-of-plane rotation attributes,

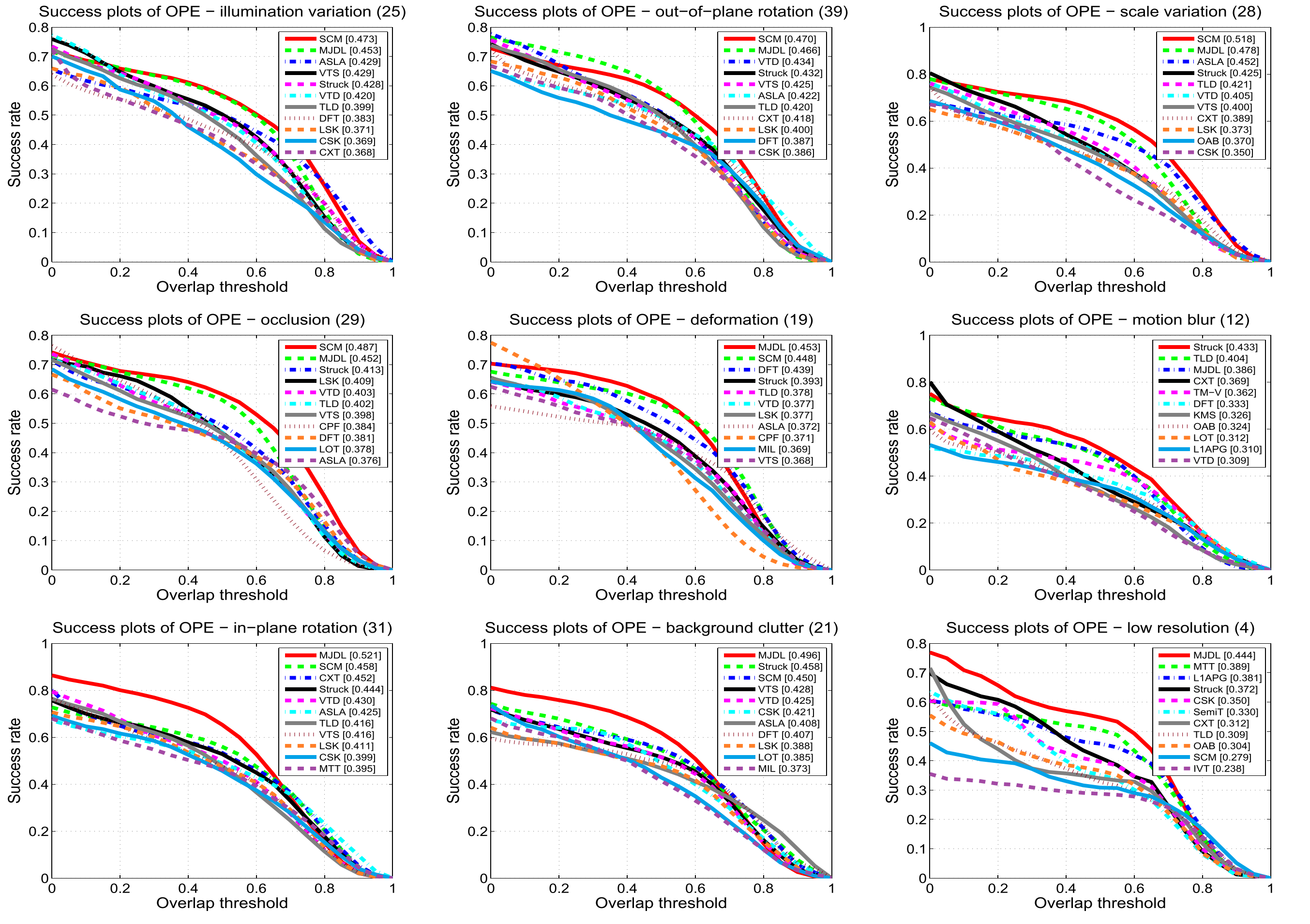


Fig. 4. Success plots of different image sequences attributes that the proposed MJDL can achieve favorable results (within the top three). Best viewed on color display.

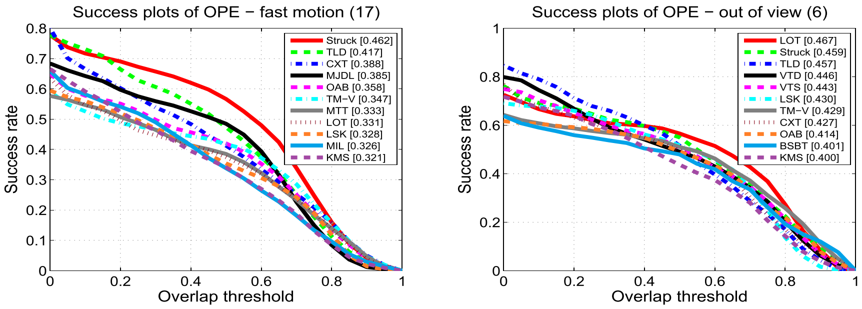


Fig. 5. Success plots of different image sequences attributes that the proposed MJDL cannot perform well (outside the top three). Best viewed on color display.

therefore, making them much more challenging; furthermore, the long videos *Dog1*, *Doll*, and *Sylvester* have the attributes of scale variation, in- and out-of-plane rotations; the videos *CarDark*, *Crossing*, *Dudek*, and *Mhyang* have the attribute of background clutter. Nevertheless, the proposed MJDL performs persistently well from beginning to the end. However, in some videos with complex compound attributes, such as illumination variation mixed with deformation, occlusion, fast motion, and motion blur, our tracker easily drifts to the background, such as *Matrix* (#46/100), *Soccer* (#98/392), *Skiing* (#5/81), and *Skating1* (#270/400).

C. Qualitative Comparison

1) Background Clutter: Fig. 7 shows the sampled experimental results of sequences *Deer*, *Football*, and *Football1*, which are challenging for background clutter caused by multiple similar targets. In *Deer*, both the water and the furriery background make trackers confusing, since they are similar to the object. In *Football* and *Football1*, the target player is running across a field full of other players, whose outfits and helmets highly resemble the target. We observe that the SCM, LSK, ASLA, VTS, and VTD trackers drift to background in *Deer* (e.g., #38, #50, and #70), the Struck

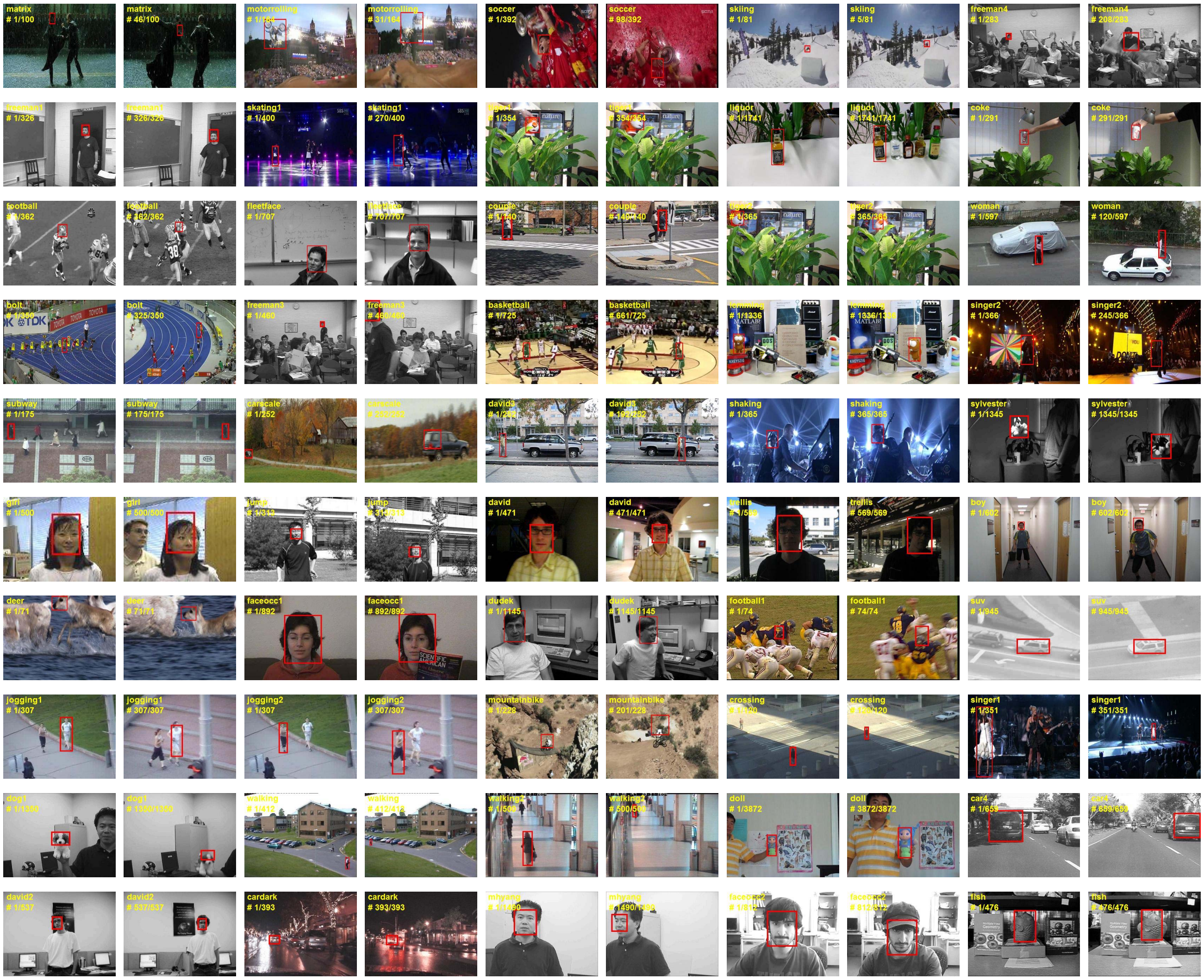


Fig. 6. Sampled results of MJDL on the benchmark. For the frame pair of each image sequence, the left-hand side image shows the first frame with the bounding box of the target, while the right-hand side image one shows the beginning frame that suffers from severe drift. If the severe drift does not happen in that image sequence, the right one shows the last frame with the tracking result. Best viewed on color and high-resolution display.

tracker drifts to background in *Football* (e.g., #278), and the LSK, SCM, ASLA, CSK, and VTS trackers drift to background in *Football1* (e.g., #71). The TLD, CXT, and our MJDL are able to distinguish the target from background in the presence of multiple similar objects. The TLD tracker can track the object by detection the object in whole frame, and the CXT is able to use context information to differentiate the object from background. The proposed MJDL locates the target with two strategies. First, our MJDL focuses on both discriminative and reconstructive power of the dictionary in appearance modeling. In this paper, the tracking task is also considered as a binary classification problem. Apart from reconstruction error, classification error is considered. Through this way, the proposed algorithm learns a sparse dictionary and a linear classifier simultaneously, which is able to represent the object well and differentiate the target from multiple similar background objects. Second, we exploit multiple features of the target in its appearance modeling. For multiple similar objects, certain features of them might be close to each other separately, such as color feature, texture feature, and shape feature, and it is difficult to use single one of them to distinguish the tracked target from multiple similar objects. Thus, we combine these multiple features of the object and develop a discriminative appearance model. In this method, we construct a multiple feature collaborative appearance model for tracking, which is able to well address the problem of background clutter. With the help of these two strategies above, the MJDL is able to handle the problem of background.

2) Occlusion: Fig. 8 demonstrates the sampled experimental results of sequences *Tiger2*, *Jogging1*, and *Faceocc2*, which undergo occlusion. In *Tiger2*, the object not only suffers from occlusion but also illumination variation (e.g., #148, #221, and #349). In *Jogging1*, the target person is fully occluded by the background and undergoes deformation (e.g., #55). In *Faceocc2*, the face is occluded by the book and rotates meanwhile (e.g., #491). We observe that, overall, the proposed MJDL can well deal with the occlusion in these

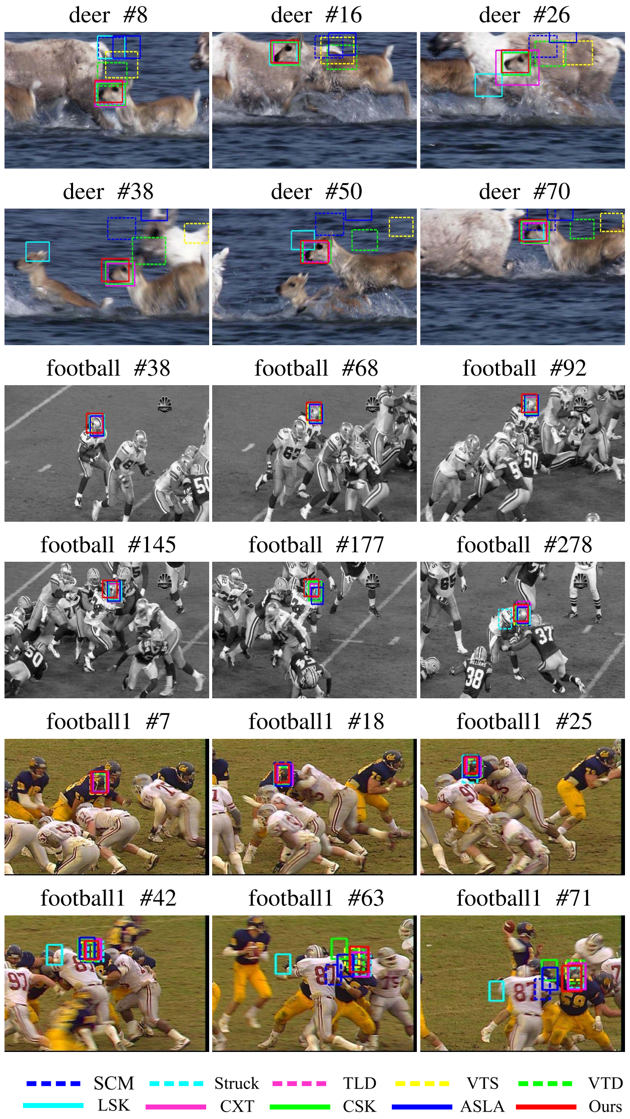


Fig. 7. Qualitative results of ten trackers over sequences *Deer*, *Football*, and *Football1* with background clutter. Best viewed on high-resolution display.

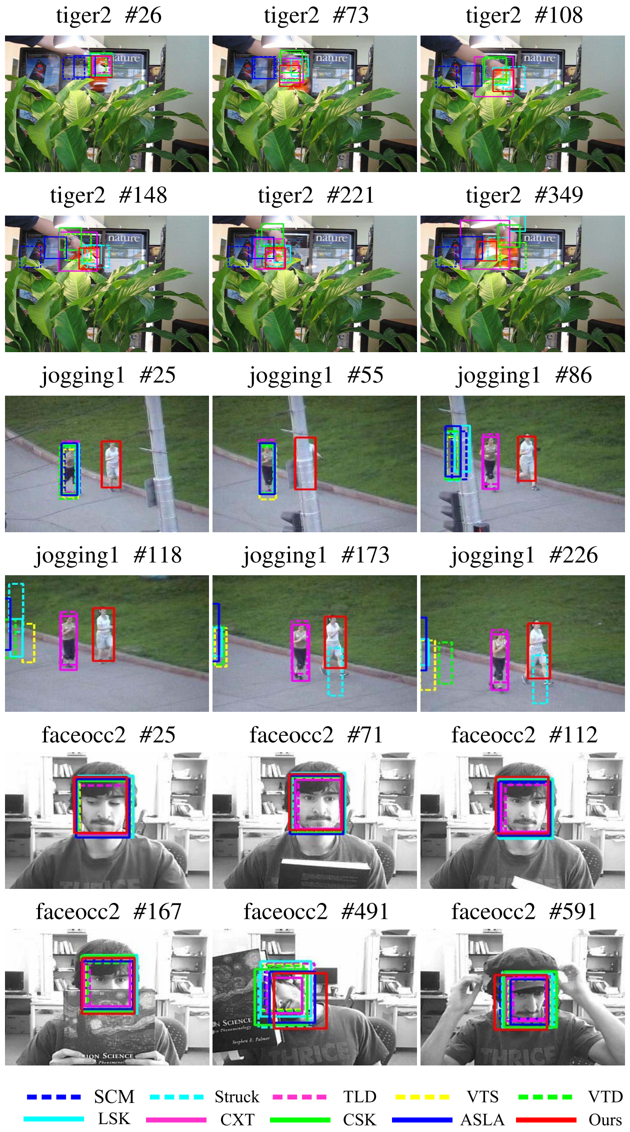


Fig. 8. Qualitative results of ten trackers over sequences *Tiger2*, *Jogging1*, and *Faceocc2* with occlusion. Best viewed on high-resolution display.

three videos, which is attributed to three aspects. First, the proposed method is based on sparse representation and can take advantage of the robustness to occlusion from it. Besides, we use both target and background information to learn the dictionary. Through this way, the object appearance model based on the learned dictionary is more discriminative to distinguish the target and the background. When occlusion occurs, our discriminative appearance model is able to utilize the unoccluded information to determine the tracked target from the candidate set. Second, we use both reconstruction error and classification to improve the discriminative power of the dictionary. In this paper, visual tracking is not only viewed as a reconstruction problem but also a binary classification task. The dictionary is learned by minimizing both the reconstruction error and classification error and more discriminative. Therefore, the object appearance model based on this dictionary is more discriminative to locate the object even in the presence of occlusion. Third, multiple different features of the target are exploited and combined in a joint way. Certain one single feature may be sensitive to occlusion, however, multiple feature combination can make the appearance model much more robust and discriminative, and help our tracker better resist occlusion. Besides, the update mechanism also helps our tracker improve the robustness to occlusion. Each tracking result will be evaluated and determined whether or not being added into the update set according to its reconstruction error and classification error. By this means, we can avoid updating background into an object appearance model.

3) Illumination Variation: Fig. 9 shows the sampled experimental results of sequences *David*, *Singer1*, and *CarDark* with drastic illumination. In these sequences, the object suffers from not only illumination but also deformation, scale variation, and background clutter. The VTS, CXT, LSK, and Struck trackers drift gradually. Although the TLD, VTD, and CSK trackers performance well in

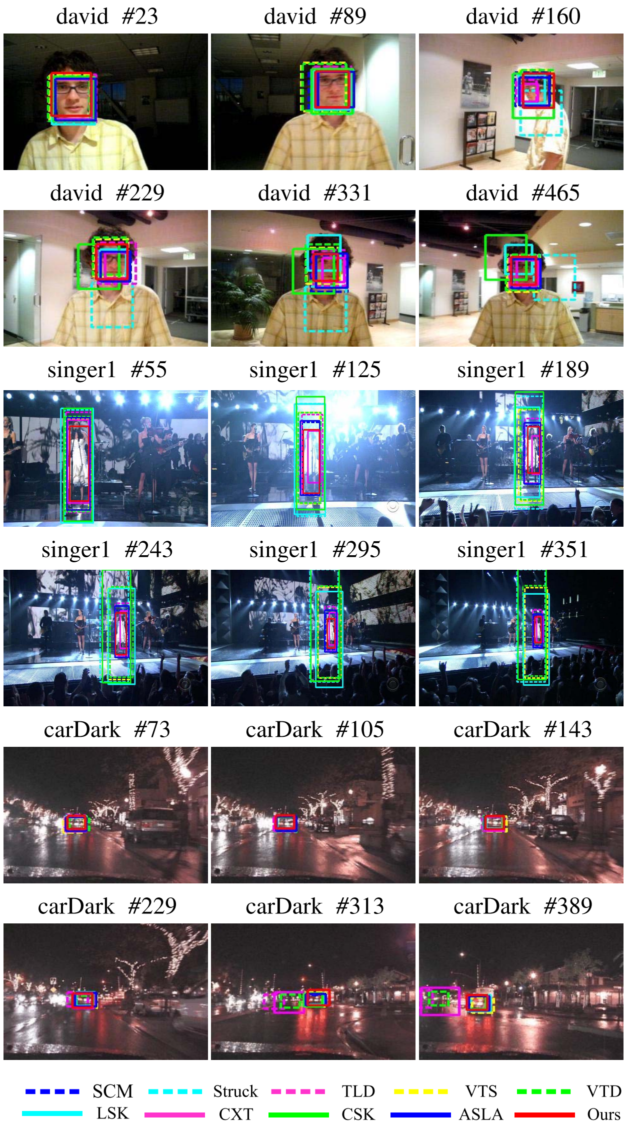


Fig. 9. Qualitative results of ten trackers over sequences *David*, *Singer1*, and *CarDark* with illumination variation. Best viewed on high-resolution display.

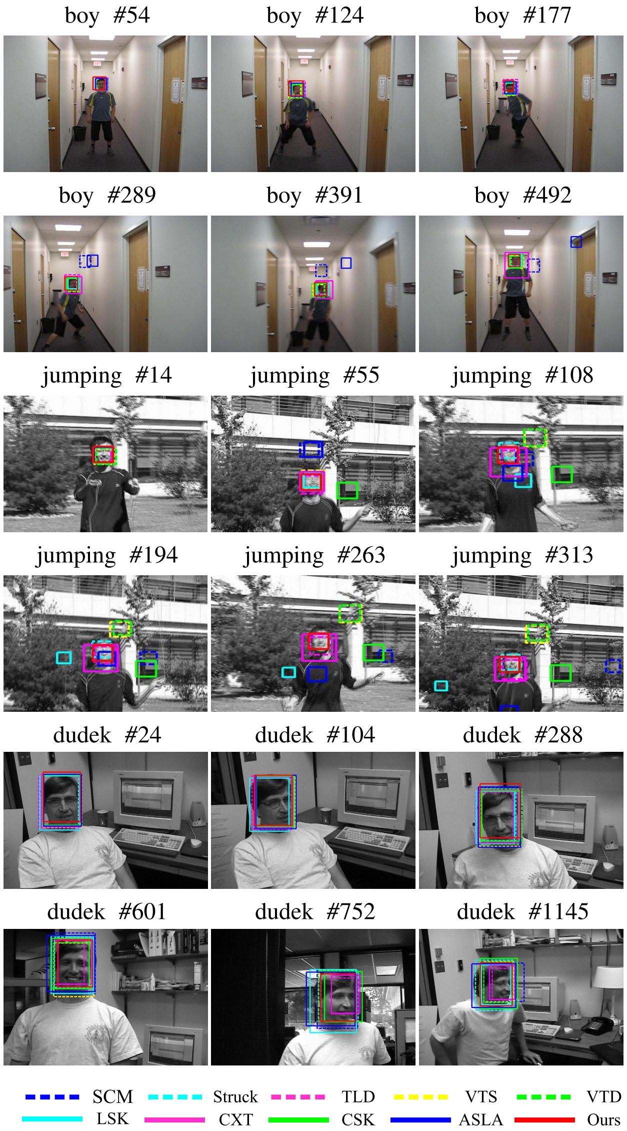


Fig. 10. Qualitative results of ten trackers over sequences *Boy*, *Jumping*, and *Dudek* with other challenges. Best viewed on high-resolution display.

David and *CarDark*, they lose the target when the pose change and scale variation accompany the illumination variation in *Singer1*. The SCM, ALSA, and our MJDL are able to accurately track the object throughout the entire sequences.

4) Other Challenges: Fig. 10 shows the sampled experimental results where many other challenges occur in these sequences, such as in-plane rotation, out-of-plane rotation, scale variation, motion blur, and so on. In *Boy* and *Jumping* sequences, the object jumps regularly, causing motion blur and scale variation in the face (e.g., #124 and #391 in *Boy* sequence and #108, #194, and #313 in *Jumping* sequence), making it hard to track. Our MJDL performs well in this sequence because of the power of multiple feature fusion. The target in the *Dudek* sequence suffers from occlusion, rotation, and scale variation (e.g., #601 and #752), our MJDL works well due to the discriminative power of the appearance model.

D. Analysis of MJDL

In order to verify the effectiveness of MTL, we develop a tracker by directly concatenating multiple features (CMF) of the target without MTL. The quantitative results are shown in Fig. 11.

From Fig. 11, we can see that the MJDL favorably performs because we exploit multiple features of the target and learn the underlying relationship between these different features for object appearance modeling within the MTL framework. Through this way, different features can exert different powers for different cases within the MTL framework, and thus, the object appearance model based on these features is more discriminative and robust to appearance changes. On the contrary, the CMF cannot take full advantages of multiple features of the target because each feature has the same importance in the concatenating feature. As we previously put it, however, different features should have different weights in dealing with different situations. Although the CMF can still obtain

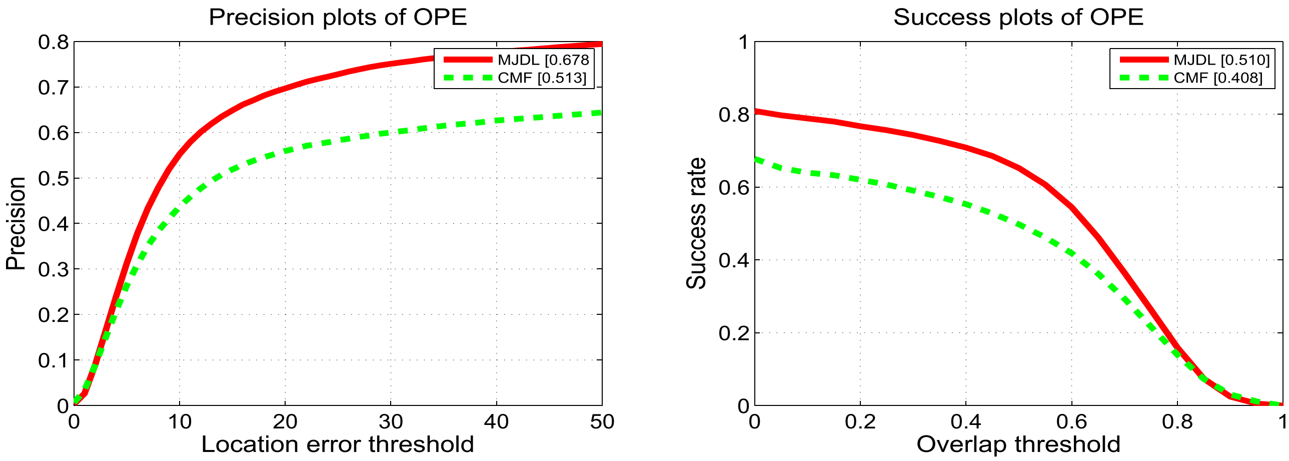


Fig. 11. Precision plots and success plots of OPE for MJDL and CMF.

a dictionary using the learning method adopted in this paper, its discriminative power has been significantly undermined.

VI. Conclusion

In this paper, we exploit multiple features of the object for modeling its appearance and propose an efficient tracking algorithm based on MJDL. This method extracts K different features for each sample and can obtain K different linear sparse representations. Each linear representation can learn a dictionary. We adopt a joint learning way to combine the K linear sparse representations, which provide additional useful information to the classification problem, since different tasks may favor different sparse representations, yet the joint sparsity may enforce the robustness in coefficient estimation, which in return improves both the reconstructive and discriminative power of the learned dictionaries. After obtaining the dictionaries, the quality of each tracking candidate is measured based on a joint linear representation. Extensive evaluation on a large benchmark data set demonstrates that the proposed tracking algorithm achieves favorable results against some state-of-the-art methods.

References

[1] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. CVPR*, 2013, pp. 2411–2418.

[2] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. Van Den Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, 2013, Art. ID 58.

[3] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 261–271, Feb. 2007.

[4] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[5] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and K-selection," in *Proc. CVPR*, 2011, pp. 1313–1320.

[6] F. Yang, H. Lu, and M.-H. Yang, "Robust superpixel tracking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1639–1651, Apr. 2014.

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

[8] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. BMVC*, 2006, pp. 47–56.

[9] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[10] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. CVPR*, 2010, pp. 1269–1276.

[11] J. Kwon and K. M. Lee, "Tracking by sampling and integrating multiple trackers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1428–1441, Jul. 2014.

[12] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[13] K. Zhang, L. Zhang, and M.-H. Yang, "Fast compressive tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2002–2015, Oct. 2014.

[14] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proc. 13th ECCV*, 2014, pp. 127–141.

[15] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2356–2368, May 2014.

[16] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2259–2272, Nov. 2011.

[17] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via structured multi-task sparse learning," *Int. J. Comput. Vis.*, vol. 101, no. 2, pp. 367–383, Jan. 2013.

[18] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem, "Robust visual tracking via consistent low-rank sparse learning," *Int. J. Comput. Vis.*, vol. 111, no. 2, pp. 171–190, 2015.

[19] S. Zhang, H. Yao, H. Zhou, X. Sun, and S. Liu, "Robust visual tracking based on online learning sparse representation," *Neurocomputing*, vol. 100, pp. 31–40, Jan. 2013.

[20] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 210–227, Feb. 2009.

[21] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. ICCV*, 2011, pp. 471–478.

[22] S. Agarwal and D. Roth, "Learning a sparse representation for object detection," in *Proc. 7th ECCV*, 2002, pp. 113–127.

[23] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Proc. NIPS*, 2006, pp. 609–616.

[24] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Kernel sparse representation for image classification and face recognition," in *Proc. 11th ECCV*, 2010, pp. 1–14.

[25] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Proc. CVPR*, 2008, pp. 1–8.

[26] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.

[27] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.

[28] F. Yang, Z. Jiang, and L. S. Davis, "Online discriminative dictionary learning for visual tracking," in *Proc. WACV*, 2014, pp. 854–861.

[29] Q. Zhang and B. Li, “Discriminative K-SVD for dictionary learning in face recognition,” in *Proc. CVPR*, 2010, pp. 2691–2698.

[30] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, “Locally orderless tracking,” in *Proc. CVPR*, 2012, pp. 1940–1947.

[31] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained linear coding for image classification,” in *Proc. CVPR*, 2010, pp. 3360–3367.

[32] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[33] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[34] G. H. Golub, P. C. Hansen, and D. P. O’Leary, “Tikhonov regularization and total least squares,” *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 1, pp. 185–194, 1999.

[35] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. CVPR*, 2005, pp. 886–893.

[36] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[37] X. Chen, W. Pan, J. T. Kwok, and J. G. Carbonell, “Accelerated gradient method for multi-task sparse learning problem,” in *Proc. 9th ICDM*, 2009, pp. 746–751.

[38] Z. Kang, K. Grauman, and F. Sha, “Learning with whom to share in multi-task feature learning,” in *Proc. 28th ICML*, 2011, pp. 521–528.

[39] X.-T. Yuan, X. Liu, and S. Yan, “Visual classification with multitask joint sparse representation,” *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 4349–4360, Oct. 2012.

[40] X. Mei, Z. Hong, D. Prokhorov, and D. Tao, “Robust multitask multiview tracking in videos,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2874–2890, Nov. 2015.

[41] S. Hare, A. Saffari, and P. H. S. Torr, “Struck: Structured output tracking with kernels,” in *Proc. ICCV*, 2011, pp. 263–270.

[42] X. Jia, H. Lu, and M.-H. Yang, “Visual tracking via adaptive structural local sparse appearance model,” in *Proc. CVPR*, 2012, pp. 1822–1829.

[43] T. B. Dinh, N. Vo, and G. Medioni, “Context tracker: Exploring supporters and distracters in unconstrained environments,” in *Proc. CVPR*, 2011, pp. 1177–1184.

[44] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “Exploiting the circulant structure of tracking-by-detection with kernels,” in *Proc. 12th ECCV*, 2012, pp. 702–715.

[45] L. Sevilla-Lara and E. Learned-Miller, “Distribution fields for tracking,” in *Proc. CVPR*, 2012, pp. 1910–1917.

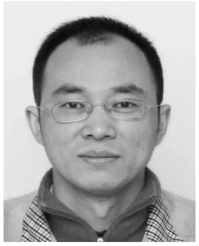
[46] Y. Wu, B. Ma, M. Yang, J. Zhang, and Y. Jia, “Metric learning based structural appearance model for robust visual tracking,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 5, pp. 865–877, May 2014.

[47] B. Zhuang, H. Lu, Z. Xiao, and D. Wang, “Visual tracking via discriminative sparse similarity map,” *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1872–1881, Apr. 2014.

[48] N. Wang, J. Wang, and D.-Y. Yeung, “Online robust non-negative dictionary learning for visual tracking,” in *Proc. ICCV*, 2013, pp. 657–664.

Heng Fan received the B.E. degree from the College of Science, Huazhong Agricultural University, Wuhan, China, in 2013, where he is currently pursuing the M.Sc. degree with the College of Engineering.

His current research interests include computer vision, pattern recognition, and machine learning.



Jinhai Xiang received the M.E. degree in computer science from the China University of Geoscience, Wuhan, China, in 2003, and the Ph.D. degree in computer architecture from the Huazhong University of Science and Technology, Wuhan, in 2014.

He is currently an Associate Professor with the College of Informatics, Huazhong Agricultural University, Wuhan. His current research interests include computer vision and machine learning.