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 models, 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.
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 $\ell_1$ 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 exploit 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 $X = \{X_1, X_2, \ldots, X_N\} \in \mathbb{R}^{d \times K \times N}$ 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 $Y = \{-1, 1\}$ are the class labels. Our goal is to learn multiple dictionaries, which are discriminative to distinguish the target from the background. For the $i$th sample, $X_i = \{x_1^i, x_2^i, \ldots, x_d^i\} \in \mathbb{R}^{d \times K} (i = 1, 2, \ldots, N)$ denotes its feature matrix, in which $x_k^i \in \mathbb{R}^d$ represents the $k$th ($k = 1, 2, \ldots, K$) feature associated with the $i$th sample. Note that the dimensions of the $K$ different features are stretched to the same in this paper in the experiments. For each modality of feature, a corresponding dictionary can be learned. We use dictionary set $D = \{D^1, D^2, \ldots, D^K\} \in \mathbb{R}^{d \times L \times K}$ to represent the $K$ dictionaries, in which each $D^k = \{d_1^k, d_2^k, \ldots, d_d^k\} \in \mathbb{R}^{d \times L \times K}$ is a single dictionary with $L$ items for the $k$th modality. Given the value of $D$, each $x_k^i$ can be reconstructed by a linear combination of few items from the $k$th dictionary $D^k$ as

$$x_k^i = D_k^k c_k^i + e_k^i$$

where $e_k^i \in \mathbb{R}^d$ is the error vector and $c_k^i \in \mathbb{R}^L$ is the sparse code of $x_k^i$ and can be obtained by $\ell_1$ minimization

$$c_k^i = \arg \min_c \|x_k^i - D_k^k c_k^i\|_2^2 + \lambda_1 \|c\|_1$$

where $\lambda_1$ is a tradeoff parameter between reliable reconstruction and sparse regularization. After obtaining the $c_k^i$, it can be directly utilized as a feature for classification as follows:

$$y_k^i = f(W_k^i, c_k^i) + e_k^i$$

where $f(W_k^i, c_k^i)$ is a linear classifier, $y_k^i = [0, \ldots, 1, \ldots, 0]^T \in \mathbb{R}^m$ is a label vector ($m = 2$ in tracking problem), in which the nonzero index indicates the class label of $x_k^i$. $W_k^i \in \mathbb{R}^{m \times L}$ is a classification parameter, and $e_k^i \in \mathbb{R}^m$ is the residual term. $(W_k^i)^\dagger_{k=1}$ can be estimated by fitting the following least squared regression (LSR) model:

$$\arg \min_W \left\{ \mathcal{F}(W) := \sum_{k=1}^{K} \sum_{i=1}^{N} \|y_k^i - W_k^i c_k^i\|_2^2 \right\}$$

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

$$\arg \min_W \left\{ \mathcal{F}(W) := \sum_{k=1}^{K} \sum_{i=1}^{N} \|y_k^i - W_k^i c_k^i\|_2^2 + \lambda_2 \|W_k^i\|_F^2 \right\}.$$  

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 $D$ 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 $1 \times n$. For example, for the features, such as image pixels denoted by a matrix $A$ with size $p \times q$, we can reshape it into a feature vector $f$ with dimension $1 \times (p \times q)$ 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, $f_1^1$, $f_2$, and $f_3$ are different features with dimension $1 \times n_1$, $1 \times n_2$, and $1 \times n_3$ respectively, where $n_1 < n_2 < n_3$. We then keep $f_1^1$ unchanged and stretch $f_1$ and $f_2$ by filling zero vectors, and we can get two new feature vectors $f_1^2 = [f_1^1, f_2^1]$ and $f_1^3 = [f_1^1, f_2^2]^T$, where $f_1^1$ and $f_2^1$ are zero vectors, and their dimensions are $1 \times (n_1 - n_2)$ and $1 \times (n_1 - n_2)$ 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:

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

s.t. \( c_i^k = \arg \min_c \|x_i^k - D^k c_i^k \|^2_2 + \lambda_1 \|c_i^k \|_1 \) \quad (6)

where \( \|t_i^k - c_i^k \|^2_2 \) is the sparse coding error and \( t_i^k = [l_{i1}^k, l_{i2}^k, \ldots, l_{iL}^k]^T \) is the ideal discriminative sparse code for \( x_i^k \). If \( l_{ij} = 1 \) (\( j = 1, 2, \ldots, L \)), \( x_i^k \) and the dictionary item \( d_i^k \) share the same label, while \( l_{ij} = 0 \) means that they belong to different classes. For instance, let \( D^k = \{d_1^k, d_2^k, d_3^k, d_4^k, d_5^k\} \in \mathbb{R}^{d \times 5} \) be the dictionary for the \( k \)-th element, and \( x_i^k \) represents the \( k \)-th modality of the \( i \)-th training sample. If \( x_i^k \) is in the same dictionary class with \( d_1^k, d_2^k, \) and \( d_3^k \), and different classes with \( d_4^k \) and \( d_5^k \), then the ideal discriminative sparse code \( t_i^k \) can be defined as \( t_i^k = [1, 1, 1, 0, 0]^T \).

The variable \( \mu \) 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:

$$\mathcal{L}^k(D^k, W^k) = \sum_{i=1}^{N} \left\{ (1-\mu)\|y_i^k - W^k c_i^k \|^2_2 + \mu \|t_i^k - c_i^k \|^2_2 \right\} + \lambda_2 \|W^k\|^2_F$$

s.t. \( c_i^k = \arg \min_c \|x_i^k - D^k c_i^k \|^2_2 + \lambda_1 \|c_i^k \|_1 \). \quad (7)

Then, (6) can be rewritten as

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

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

$$\nabla_w \mathcal{L}^k(D^k, W^k) = (1-\mu)(W^k c_i^k - y_i^k)(c_i^k)^T + \lambda_2 W^k. \quad (9)$$

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

$$\nabla_d \mathcal{L}^k(D^k, W^k) = -D^k \delta^k (c_i^k)^T + (x_i^k - D^k c_i^k)(\delta^k)^T \quad (10)$$

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

### Algorithm 1 MJDL

**Require:** Training sample features \( X \) with labels \( \mathbf{y}, \mathbf{\lambda}_1, \mathbf{\lambda}_2, \mu, \mathbf{M}, \mathbf{W}_0 \), \( D_0 = \{D_0^k\}_{k=1}^{K} \); \( D_1 = \{D_1^k\}_{k=1}^{K} \); \( \eta = \min(\eta, \eta_i/\mu) \);

1. **for** \( q = 1 \) to \( Q \)
2. **for** \( k = 1 \) to \( K \)
3. Generate training sample \( X \);
4. **for** \( i = 1 \) to \( N \)
5. Obtain \( y_i^k \) for \( x_i^k \);
6. Sparse coding: compute \( c_i^k \) according to Equation (2); \( \bar{\lambda} = \{\lambda_2^k\} \)
7. Compute the active set \( \bar{\lambda} \) and variable \( \delta^k \) via Equation (11);
8. Choose learning rate \( \eta_q = \min(\eta, \eta_i q / \mu) \);
9. Compute the gradients of \( W^k_0 \) and \( D_0^k \) via Equation (9) and (10);
10. Update \( W_{q+1}^k \) and \( D_{q+1}^k \) with \( W_{q+1}^k = W_q^k - \eta_q \nabla_{w_q} \mathcal{L}_q^k(D_q^k, W_q^k) \) and \( D_{q+1}^k = D_q^k - \eta_q \nabla_{D_q} \mathcal{L}_q^k(D_q^k, W_q^k) \);
11. **end for**
12. Let \( W_{q+1}^k = W_{q+1}^k \) and \( D_{q+1}^k = D_{q+1}^k \);
13. **end for**
14. \( W_{q+1}^k = \{W_{q+1}^k\}_{k=1}^{K} \) and \( D_{q+1}^k = \{D_{q+1}^k\}_{k=1}^{K} \);
15. **end for**

**Return:** New \( W \) and \( D \);

where \( \delta^k \in \mathbb{R}^L \) is a vector that relies on \( c_i^k, y_i^k, t_i^k, D^k, \) and \( W^k \) with

$$\delta^k_\bar{\lambda} = 0$$

$$\delta^k_\bar{\lambda} = (D_0^k)^T (W_0^k)^{-1} \nabla_{c_i^k} \mathcal{L}_0^k(D_0^k, W_0^k)$$

where \( \bar{\lambda} \) denotes the indices of the nonzero coefficients of all nonzero values in \( D^k \), and \( \nabla_{c_i^k} \mathcal{L}_0^k(D_0^k, W_0^k) \) can be obtained with

$$\nabla_{c_i^k} \mathcal{L}_0^k(D_0^k, W_0^k) = (1-\mu)(W_0^k c_i^k - y_i^k)(W_0^k c_i^k - y_i^k)^T + \mu (c_i^k - t_i^k). \quad (12)$$

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

### C. Initialization

First, we randomly select \( N \) target and background templates, and extract \( K \) features for each sample to form \( X \). Then, the K-SVD algorithm [32, 33] is performed on training samples \( K \) times to initialize the dictionary \( D_0 = \{D_0^k\}_{k=1}^{K} \). Given the dictionary, we calculate the sparse code \( c_i^k \) for \( x_i^k \) to form matrix \( C^k \), which contains the sparse code of all samples for the \( k \)-th modality, and then employ the ridge regression model [34] to obtain \( W_0^k \) by

$$W_0^k = \arg \min_{W_0^k} \|F^k - W_0^k C^k\|^2_2 + \lambda_3 \|W_0^k\|^2_2$$

where \( F^k \) is the label matrix of all samples for the \( k \)-th modality. After obtaining \( W_0^k \), \( W_0 \) can be initialized as \( W_0 = \{W_0^k\}_{k=1}^{K} \).
The solution for formulation (13) is
\[ W_0^k = F^k (C^k)^T \left( C^k (C^k)^T + \lambda^k I \right)^{-1} \]
where \( I \) is the identity matrix.

\[ \text{D. Classification} \]

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 \( s \), we extract its multiple feature matrix \( X_s = \{ x_1^1, x_2^1, \ldots, x_K^1 \} \) and compute the corresponding sparse codes \( C_s = \{ c_1^1, c_2^1, \ldots, c_K^1 \} \). Note that there are \( K \) modalities of features used in this paper, and thus, the joint decision measure can be defined as
\[ \varphi(s) = 1 - \sum_{k=1}^{K} \varphi^k(s) \]
where \( \varphi(s) \) is the joint decision measure, which represents the probability of \( s \) belonging to object class. The more the \( \varphi(s) \) is, the more probably the \( s \) is the object. \( \varphi^k(s) \) is the joint decision error under the \( k \)th modality and it can be obtained via
\[ \varphi^k(s) = (1 - \alpha) \| s^k_t - D^k_c^k \|^2 + \alpha \| y^k_t - W^k_c^k \|^2 \]
where \( s^k_t \) represents the average of the real \( k \)th 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 \( k \)th modality of the tracking results. We collect the \( k \)th 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 \( s^k_t \). \( \| s^k_t - D^k_c^k \|^2 \) and \( \| y^k_t - W^k_c^k \|^2 \) are the reconstruction error and the classification error of the \( k \)th modality, respectively, \( \alpha \) is a tradeoff parameter, and \( y^k_t = [0, 1, \ldots, 0]^T \in \mathbb{R}^m \) 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 \( y^k_t \) is fixed to \([1, 0]^T \) (for target class) or \([0, 1]^T \) (for background class).

\[ \text{IV. PROPOSED TRACKING METHOD} \]

\[ \text{A. Tracking Formulation} \]

Our tracker is implemented via the Bayesian framework. Given the observation set of targets \( Y' = \{ y_1, y_2, \ldots, y_T \} \) up to the frame \( t \), where \( y_t \) (\( t = 1, 2, \ldots, t \)) represents the observation of target in frame \( t \), we can obtain estimation \( \hat{X}_t \) by computing the maximum a posteriori via
\[ \hat{X}_t = \arg \max_{X_t} p(X_t|Y') \]
where \( \hat{X}_t \) denotes the \( t \)th sample at the state of \( X_t \). The posterior probability \( p(X_t|Y') \) can be obtained by the Bayesian theorem recursively via
\[ p(X_t|Y') \propto p(y_t|X_t) \int p(X_t|X_{t-1}) p(X_{t-1}|Y^{t-1}) dX_{t-1} \]
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 $v$ 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 $\varepsilon(\text{re}) = (1/K)\sum_{k=1}^{K} ||\mathbf{s}_k - \mathbf{D}^k\mathbf{c}_k||^2$ or a high classification error $\varepsilon(\text{cl}) = (1/K)\sum_{k=1}^{K} ||\mathbf{y}_k - \mathbf{W}^k\mathbf{c}_k||^2$. In this case, we skip this frame to avoid introducing noise into $S$. Two thresholds $t_{\text{re}}$ and $t_{\text{cl}}$ are adopted to determine whether this frame is skipped. If $\varepsilon(\text{re}) \geq t_{\text{re}}$ or $\varepsilon(\text{cl}) \geq t_{\text{cl}}$, 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 $k$th 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.

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 $8 \times 8$, the block size is $16 \times 16$, and its number of bins is set to 9. The parameters $P$ and $R$ in LBP are 8 and 1, respectively, and the pattern 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.

\(^3\)VLFeat is an open source library and available at http://www.vlfeat.org/

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 $t_{\text{re}}$ and $t_{\text{cl}}$ 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 ($\sim$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, 

Algorithm 2 Tracking by MJDL

**Require:** Frames 1, 2, \(\cdots\), $t$, \(\cdots\);

**Initialization:**

1: Select initial target $X_1$ and sample $N^+$ positive samples and $N^-$ negative samples;
2: Extract multiple features from samples to form $X$ with label $Y$;
3: Initialize $W_t = \{W^k_t\}_{k=1}^{K}$, $D_t = \{D^k_t\}_{k=1}^{K}$;
4: Add $K$ features of $X_1$ and $X$ to $\{T_k\}_{k=1}^{K}$ and $S$ respectively;

**Tracking:**

5: for $t = 2$ to the end of the sequence
6: Sample $J$ candidates around $X_{t-1}$;
7: Extract $K$ features from each candidate and compute the corresponding sparse codes;
8: Compute joint decision measure error for each candidate based on Equation (15) and (16);
9: Select the candidate with smallest joint decision measure error to be the tracking result $X_t$;
10: Sample $N^+$ positive samples and $N^-$ negative samples to form new $X_{\text{new}}$;
11: Add $K$ features of $X_t$ and $X_{\text{new}}$ to $\{T_k\}_{k=1}^{K}$ and $S$ respectively;
12: For each $T^k$, if its size is greater than $U$, delete the oldest element from $T^k$;
13: If the size of set $S$ is equal to $V$, update $D$ and $W$ according to Algorithm 1, and then empty set $S$;
14: end for

**Return:** Tracking results $X_1, X_2, \cdots, X_t$;
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 \(= \frac{\text{area}(R_{GT} \cap R_T)}{\text{area}(R_{GT} \cup R_T)}\) with the ground truth \(R_{GT}\) 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 [5], DFT [45], OAB [8], and LOT [30]) in terms of precision plot and success plot. Note that SCM, Struck, TLD, ASLA, CXT, VTS, VTD, CSK, LOT, and OAB are top ten trackers in terms of precision plot; while SCM, Struck, TLD, ASLA, CXT, VTS, VTD, CSK, LOT, 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,
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...
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
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
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
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


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.