Inverse Nonnegative Local Coordinate Factorization for Visual Tracking

Fanghui Liu, Tao Zhou, Chen Gong, Keren Fu, Li Bai, and Jie Yang

Abstract— **Recently, nonnegative matrix factorization (NMF) with part-based representation has been widely used for appearance modeling in visual tracking. Unfortunately, not all the targets can be successfully decomposed as "parts" unless some rigorous conditions are satisfied. To avoid this problem, this paper introduces NMF's variants into the visual tracking framework in the view of data clustering for appearance modeling. First, an initial target appearance model based on NMF is proposed to describe the target's appearance with the incorporated local coordinate factorization constraint, orthogonality of the bases, and $L_{1,1}$ norm regularized sparse residual error constraint. Second, an inverse NMF model is proposed in which each learned base vector is regarded as a clustering center in a low-dimensional subspace. Potential target samples (from the foreground) will be clustered around base vectors, while the candidate samples (from the background) are very likely to spread irregularly over the entire clustering space. Such differences can be fully exploited by the inverse NMF model to produce more discriminative encoding vectors than the conventional NMF method. Furthermore, incremental updating model is introduced into the tracking framework for online updating the initial appearance model. Experiments on object tracking benchmark suggest that our tracker is able to achieve promising performance when compared with some state-of-the-art methods in deformation, occlusion, and other challenging situations.**

Index Terms— **Local coordinate constraint, inverse nonnegative matrix factorization, incremental update, visual tracking.**

I. Introduction

VISUAL tracking is one of the most enduring topics in computer vision with a wide range of applications, such as video surveillance, autonomous driving, and robotic navigation [1], [2]. Although much progress has been made in the past decades [3]–[5], visual tracking still cannot meet the requirements of practical applications due to some challenging factors such as occlusions, shape deformation, etc.

One essential aspect of visual tracking is appearance modelling. According to the adopted appearance model, current modelling methods are either generative [6], [7] or discriminative [8], [9]. Generative methods aim to find the most similar candidate to the target by minimizing the reconstruction error, whilst discriminative methods cast the tracking problem as a supervised/semi-supervised classification problem [11]–[13] to separate the foreground target from the background.

As a representative modelling method, nonnegative matrix factorization (NMF) and its variants have been successfully applied to visual tracking [14]–[16]. NMF decomposes the nonnegative data matrix $\mathbf{X}$ into the multiplication of two nonnegative matrices $\mathbf{U}$ and $\mathbf{V}$ ($\mathbf{X} \approx \mathbf{UV}$), where $\mathbf{U}$ is called base matrix and the columns of $\mathbf{V}$ are coefficient vectors. Here each column of the data matrix $\mathbf{X}$ can be represented by a linear combination of base vectors (i.e. the columns of the base matrix $\mathbf{U}$). Due to the nonnegative constraints on $\mathbf{U}$ and $\mathbf{V}$, NMF learns a part-based representation for visual tracking, in which the target can be spatially represented by "parts" (base vectors) to enhance the representation ability in appearance modelling. However, Donoho and Stodden [17] point out that not all the data can be successfully identified as "parts". Such decomposition requires additional conditions such as separated support and factorial sampling, which are not satisfied in many practical situations. Some example images from [18] are shown in Fig. 1. We can see that NMF successfully learns part-based representation on the CBCL face dataset but fails on the ORL face dataset.

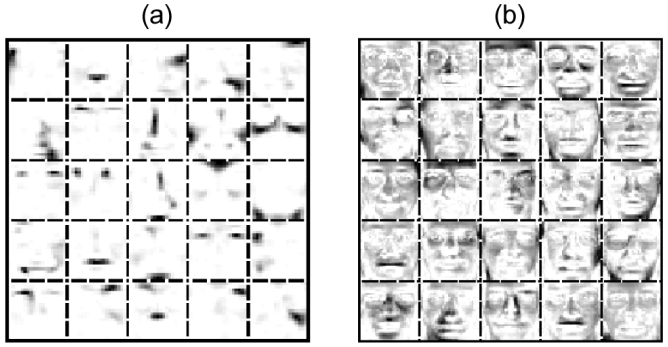


Fig. 1. The learned bases by NMF on (a) CBCL and (b) ORL dataset. A face in (a) CBCL dataset can be successfully decomposed by the learned bases with part-based representation (e.g. nose, eye, mouth, etc.). However, NMF cannot learn these "parts" to represent a face in (b) ORL dataset.

Manuscript received March 7, 2016; revised July 11, 2016 and December 5, 2016; accepted April 22, 2017. Date of publication April 28, 2017; date of current version August 3, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61572315, Grant 6151101179, and Grant 61602246, in part by the 863 Plan of China under Grant 2015AA042308, in part by the Royal Society/National Natural Science Foundation of China International Exchanges under Grant IE131664, in part by the China Postdoctoral Science Foundation under Grant 2016M601597, and in part by the Open Project Program of the Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University, under Grant MJUKF201723. This paper was recommended by Associate Editor Y. Wu. *(Corresponding author: Jie Yang.)*

F. Liu, T. Zhou, and J. Yang are with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: lfhsgre@outlook.com; zhou.tao@sjtu.edu.cn; jieyang@sjtu.edu.cn).

C. Gong is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China, and also with the Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou 350108, China (e-mail: chen.gong@njust.edu.cn).

K. Fu is with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: fkrsuper@gmail.com).

L. Bai is with the School of Computer Science, University of Nottingham, Nottingham NG8 1BB, U.K. (e-mail: bai.li@nottingham.ac.uk).

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.2017.2699676

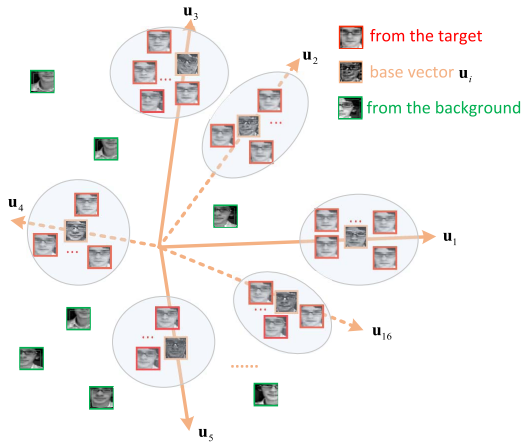


Fig. 2. Samples spread among 16 base vectors by NMF's variant from a perspective of data clustering where these base vectors are regarded as clustering centers in a subspace. The target samples are clustered around base vectors, while background samples spread among the space.

Besides, considering that the representation ability of the traditional NMF methods is limited to linear factorization, some regularization terms have been incorporated to enhance the representation ability. Graph regularization [19] is incorporated into the conventional NMF, termed GNMF [20]. Guan *et al*. [18] utilize two different classes of adjacent graphs for the data matrix to enhance GNMF's discriminative ability. The sparsity term (i.e. group sparsity [21], or $\ell_{\frac{1}{2}}$ term [22]) is added to the objective function to exploit the data structure. Specifically, to simultaneously take similarity and sparsity into account, Chen *et al*. [23] introduce a local coordinate constraint into the standard NMF that is termed Nonnegative Local Coordinate Factorization (NLCF).

Based on the above discussion, we argue that part-based representation in NMF is not suitable for visual tracking, and attempt to illustrate it from the clustering viewpoint [24], [25]. Therein, base vectors understood as the centroids of clusters in $\mathbf{U}$ represent latent semantic information of the original data in a subspace. In visual tracking, these base vectors can be regarded as data (the target) cluster centroids from different cues in a low-dimensional space. Fig. 2 demonstrates that positive candidates sampled around the potential target are similar to the base vectors, so they are located around the base vectors. In contrast, the negative candidates corresponding to background regions may spread among the clustering space. Therefore, this difference between positive and negative candidates for clustering can be effectively exploited to separate the target from the background. The main contributions of this paper are as follows:

1) NLCF is introduced to appearance modelling for visual tracking. Some additional constraints (the base orthogonality and the $L_{1,1}$ norm regularization) are imposed on $\mathbf{U}$ and error matrix $\mathbf{E}$, respectively, which helps to obtain a robust appearance model.
2) An inverse NMF representation that is called inverse NLCF (inv-NLCF) is proposed to produce discriminative feature vectors, leading to strong discriminative ability between the target and the background.
3) An incremental learning scheme is proposed for online updating the target appearance.

This paper is the extended version of our previous work [26]. The tracker described here differs from [26] in several aspects. Firstly, we introduce local coordinate constraint, orthogonality constraint and sparse error term in the appearance model, and then the corresponding algorithm is designed solve such problem. Second, the accelerated proximal gradient (APG) [27] in the inverse NLCF model is replaced by the multiplicative updating rule for the optimization problem. Thirdly, we design an incremental update rules for the appearance model. Lastly, we provide more experimental results on Object Tracking Benchmark (OTB) dataset, and also present parameter analysis, and computational complexity analysis.

II. Related Works

Since most NMF based trackers belong to generative methods, here we will mainly review some representative generative trackers including sparse representation based trackers and subspace learning based methods.

1) Sparse Representation Based Trackers: Sparse representation has been introduced into visual tracking with demonstrated success [28], [29]. The fundamental assumption is that a candidate can be represented by a sparse linear representation of target templates, where the coefficients can be solved via a constrained ℓ_1 minimization problem. Wang *et al*. [6] propose an online robust nonnegative dictionary learning algorithm based on ℓ_1 tracker [30] for updating the object appearance. Subsequently, local sparse representation [31] and structured sparse representation [28] are introduced into visual tracking framework. In [32], the dual group structures of both candidate samples and dictionary templates are formulated as the sparse representation problem at group level. Besides, reverse sparse representation formulation [33] is proposed to seek for discriminative weight for each candidate sample.

2) Subspace Learning Based Trackers: Generative trackers commonly use subspace learning (eg. PCA, NMF, tensor) for appearance modelling. The assumption is that the target lies in a low-dimensional space. The incremental PCA subspace representation [34] is adopted to learn and update the target appearance in visual tracking. The assumptions of sparse error and trivial templates are used in [35] to reduce the sensitivity to partial occlusion and also enhance the robustness of the appearance model. In [36], multiple linear and nonlinear subspaces are learned to better model the nonlinear relationship of different appearances conveyed by single object.

Some representative NMF based works include the Orthogonal Projective Nonnegative Matrix Factorization (OPNMF) [15], Constrained Incremental Nonnegative Matrix Factorization (CINMF) [37], and Constraint Online Nonnegative Matrix Factorization (CONMF) with sparsity constraint and smooth constraint [14]. These generative methods, employ NMF with different constraints (e.g. sparsity constraint, graph-based regularization) for appearance modelling. Different from above three generative methods, in [16], NMF serves as a method for feature extraction. After solving the nonnegative encoding vectors $\mathbf{v}_i$ based on $\mathbf{U}$,

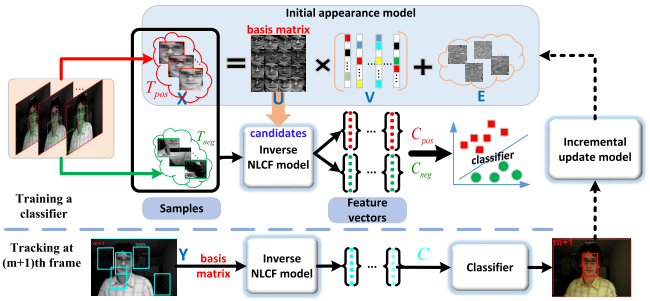


Fig. 3. Flowchart of the proposed tracker.

a Naive Bayes classifier is trained to distinguish the target from the background.

III. The New Appearance Model and Its Incremental Update

The flowchart of the proposed tracking method is shown in Fig. 3. It contains three main models: the initial appearance model, the inverse NLCF model, and the incremental updating model. The details are explained as follows.

A simple tracker (e.g. IVT [34]) is used to initialize the tracking process at the first m frames to collect a certain amount of target patches represented in graysacle values. Each image patch is normalized to 32×32 pixels and then stacked to a vector in our tracker. The patch near the tracking result is sampled as a positive template $\mathbf{T}_p^i \in \mathbb{R}^M$ $(M = 1024)$; while the patch far away from the tracked target is taken as a negative template $\mathbf{T}_n^i \in \mathbb{R}^M$. This forms the positive template set $\mathbf{T}_{pos} = [\mathbf{T}_p^1, \mathbf{T}_p^2, \cdots, \mathbf{T}_p^N] \in \mathbb{R}^{M\times N}$ (or called the initial data matrix $\mathbf{X}$), and the negative template set $\mathbf{T}_{neg} = [\mathbf{T}_n^1, \mathbf{T}_n^2, \cdots, \mathbf{T}_n^r] \in \mathbb{R}^{M\times r}$ constituting the background, where N and r are the number of positive templates and negative templates, respectively. The positive template set $\mathbf{T}_{pos}$ is decomposed into the base matrix $\mathbf{U} \in \mathbb{R}^{M\times K}$ and coefficient matrix $\mathbf{V} \in \mathbb{R}^{K\times N}$ by using NMF variants in the initial appearance model, where K is the number of base vectors. After the mth frame, S new candidate patches are sampled via the particle filter framework [38], forming $\mathbf{Y}_{1:S} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_S\} \in \mathbb{R}^{M\times S}$, where each particle $\mathbf{y}_i$ denotes a candidate sample.

The proposed inv-NLCF model is regarded as a feature coder, which encodes base matrix $\mathbf{U}$ by positive templates $\mathbf{T}_{pos}$ and negative templates $\mathbf{T}_{neg}$ respectively. The corresponding encoding vectors in $\mathbf{C}_{pos}$ and $\mathbf{C}_{neg}$ are then fed into SVM classifier for training. Each row of $\mathbf{C}$ corresponding to a candidate forms an encoding vector, which is classified as target (positive) or background (negative) by SVM classifier. The tracking result is delivered to our incremental updating model every ten frames to dynamically update the base matrix $\mathbf{U}$, the newly coefficient vector in $\mathbf{V}$, and the latest error vector in $\mathbf{E}$ for appearance model.

The initial appearance model is based on the traditional NMF with two additional constraints. In the following subsections, we mainly introduce the initial appearance model and its incremental updating model.

A. The Conventional NMF and Its Variants: A Review

Some conventional NMF methods are briefly summarized here for the ease of explanations for the proposed new method. NLCF method incorporates a coordinate coding constraint [39] into the conventional NMF, namely:

$$\mathcal{Q} = \sum_{i=1}^{N} \mu \sum_{k=1}^{K} v_{ki} \|\mathbf{u}_k - \mathbf{x}_i\|_2^2 = \mu \sum_{i=1}^{N} \|(\mathbf{x}_i \mathbf{1}^\top - \mathbf{U})\Lambda_i^{1/2}\|_{\mathrm{F}}^2, \quad (1)$$

where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K] \in \mathbb{R}^{M\times K}$, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K] \in \mathbb{R}^{K\times N}$, $\Lambda_i \in \mathbb{R}^{K\times K}$ is a diagonal matrix with the jth diagonal element defined by v_{jj}, and μ is the regularization parameter. The notation $\mathbf{1} \in \mathbb{R}^K$ denotes the all-one vector. The columns of the base matrix $\mathbf{U}$ can be considered as a set of anchor points, and thus each data point in the original space can be linearly represented by only a few anchor points [23]. Therefore, minimizing Eq. (1) requires that the new coordinate of $\mathbf{x}_i$ regarding $\mathbf{u}_k$ to be one if $\mathbf{x}_i$ is sufficiently close to the anchor point $\mathbf{u}_k$.

Besides, to discover the intrinsic geometrical structure in a manifold space, graph based regularizer is incorporated into NLCF's objective function, that is:

$$\mathcal{O} = \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_{\mathrm{F}}^2 + \lambda \mathrm{tr}(\mathbf{V}\mathbf{L}\mathbf{V}^\top) + \mu \sum_{i=1}^{N} \|(\mathbf{x}_i \mathbf{1}^\top - \mathbf{U})\Lambda_i^{1/2}\|_{\mathrm{F}}^2, \quad (2)$$

where λ is graph-based regularization parameter. The graph Laplacian matrix is $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$ and $\mathbf{W}$ is the weight matrix:

$$W_{ij} = \begin{cases} \mathrm{e}^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}} & \text{if } \mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i); \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

In Eq. (3), $\mathcal{N}_k(\mathbf{x}_i)$ denotes the k nearest neighbors of $\mathbf{x}_i$, and σ is the kernel width to be tuned. By such two regularization terms, NLCF not only considers the similarity between a data point and the learned base vector, but also maximally guarantees the sparsity.

B. The Initial Appearance Model

Based on NLCF, two additional measures including base orthogonality constraint and $L_{1,1}$ norm regularization are used to improve data representation ability in the appearance model.

1) Constraints: Rather than using the ℓ_2 orthogonal constraint $\mathbf{U}\mathbf{U}^\top = \mathbf{I}$, we use another form described in [42]:

$$\sum_{i\neq j} \mathbf{u}_i^\top \mathbf{u}_j = \mathrm{tr}(\mathbf{U}\mathbf{O}\mathbf{U}^\top), \quad \mathbf{O} = \bar{\mathbf{I}} - \mathbf{I}, \quad (4)$$

where $\bar{\mathbf{I}}$ signifies the matrix whose elements are all one. The derivation details are presented in Appendix. A. Minimizing Eq. (4) aims to enforce the inner product of two base vectors $< \mathbf{u}_i, \mathbf{u}_j > = \mathbf{u}_i^\top \mathbf{u}_j (i \neq j)$ to be as small as possible in different positions. Compared to the conventional orthogonal constraint $\mathbf{U}\mathbf{U}^\top = \mathbf{I}$, we do not need to guarantee that the base matrix is orthogonal.

In addition, the residual error $\|\mathbf{X} - \mathbf{U}\mathbf{V}\|_{\mathrm{F}}^2$ plays an important role in appearance modelling. In IVT [34], the error is assumed to obey Gaussian distribution with zero mean and

small variance. In [35], the error is regarded as sparse noise. We form this residual error as an error matrix $\mathbf{E}$ incorporated into our objective function. To measure the sparsity of the error matrix $\mathbf{E}$ in appearance model, we introduce the mixed norm $L_{p,q}$ [43] defined by:

$$\|\mathbf{E}\|_{p,q} = \left\{ \sum_j \left(\sum_i |E_{ij}|^p \right)^{\frac{q}{p}} \right\}^{\frac{1}{q}}. \tag{5}$$

In our model, we use $\|\mathbf{E}\|_{1,1} = \sum_j \sum_i |E_{ij}|$ to obtain a sparse error matrix. Compared to the ℓ_1 norm, the $L_{1,1}$ norm imposes column-wise (or row-wise) ℓ_1 norm. By incorporating above two regularization terms in Eqs. (4), (5), the new objective function is:

$$\begin{aligned}\mathcal{O}(\mathbf{U},\mathbf{V},\mathbf{E}) = {} & \|\mathbf{X}-\mathbf{U}\mathbf{V}-\mathbf{E}\|_{\mathrm{F}}^2 + \lambda \mathrm{tr}(\mathbf{V}\mathbf{L}\mathbf{V}^\top) + \beta\|\mathbf{E}\|_{1,1} \\ & + \mu \sum_{i=1}^{N} \left\| (\mathbf{x}_i \mathbf{1}^\top - \mathbf{U}) \Lambda_i^{1/2} \right\|_{\mathrm{F}}^2 + \gamma\, \mathrm{tr}(\mathbf{U}\mathbf{O}\mathbf{U}^\top), \end{aligned} \tag{6}$$

where γ and β are corresponding regularization parameters. Note that Eq. (6) is not convex in both $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{E}$, but is convex with respect to each of these three variables. Therefore, the optimal solution can be obtained by iteratively updating one variable with the other two fixed.

2) Iteration Rules: The updating rules here are similar to those in the conventional NMF updating rules in [44]. Given the optimal solution of $\mathbf{E}$, denoted as $\mathbf{E}_{opt}$, the updating rules for $\mathbf{U}$ and $\mathbf{V}$ are:

$$\begin{aligned} u_{jk}^{t+1} &\leftarrow u_{jk}^{t} \frac{(\mu+1)(\bar{\mathbf{X}}\mathbf{V}^\top)_{jk}}{(\mathbf{U}\mathbf{V}\mathbf{V}^\top + \mu\mathbf{U}\mathbf{H} + \gamma\,\mathbf{U}\mathbf{O})_{jk}} \\ v_{ki}^{t+1} &\leftarrow v_{ki}^{t} \frac{2((\mu+1)\mathbf{U}^\top\mathbf{X} + \lambda\mathbf{V}\mathbf{W})_{ki}}{(2\mathbf{U}^\top\mathbf{U}\mathbf{V} + \mu\mathbf{G} + \mu\mathbf{F} + 2\lambda\mathbf{V}\mathbf{D})_{ki}}, \end{aligned} \tag{7}$$

where $\bar{\mathbf{X}} = \mathbf{X} - \mathbf{E}_{opt}$, and $\mathbf{H}$ is the diagonal matrix, entries of which are row sums of $\mathbf{V}$. The matrix $\mathbf{G}$ is defined as $\mathbf{G} = (\mathbf{g}, \mathbf{g}, \ldots, \mathbf{g})^\top \in \mathbb{R}^{K\times N}$, where $\mathbf{g} = \mathrm{diag}(\mathbf{X}^\top\mathbf{X}) \in \mathbb{R}^N$. Likewise, $\mathbf{F} = (\mathbf{f}, \mathbf{f}, \ldots, \mathbf{f}) \in \mathbb{R}^{K\times N}$ with the definition of $\mathbf{f} = \mathrm{diag}(\mathbf{U}^T\mathbf{U}) \in \mathbb{R}^K$. The detailed derivations for iteration rules are given in Appendix. B. After several iterations, $\mathbf{U}$ and $\mathbf{V}$ will jointly converge to a stationary point $(\mathbf{U}_{opt}, \mathbf{V}_{opt})$. When $\mathbf{U}_{opt}$ and $\mathbf{V}_{opt}$ are obtained, Eq. (6) degenerates to:

$$\mathcal{O}(\mathbf{E}) = \|\hat{\mathbf{X}} - \mathbf{E}\|_{\mathrm{F}}^2 + \beta\|\mathbf{E}\|_{1,1}, \tag{8}$$

where $\hat{\mathbf{X}} = \mathbf{X} - \mathbf{U}_{opt}\mathbf{V}_{opt}$. The optimal problem is equivalent to $\mathcal{O}(\mathbf{E}) \triangleq \frac{1}{2}\|\hat{\mathbf{X}} - \mathbf{E}\|_{\mathrm{F}}^2 + \frac{\beta}{2}\|\mathbf{E}\|_{1,1}$. Therefore $\mathbf{E}_{opt}$ is obtained by the soft-threshold operator $\mathcal{S}_\lambda$ [45]:

$$\mathbf{E}_{opt} = \mathcal{S}_{\frac{\beta}{2}}(\hat{\mathbf{X}}) = \mathrm{sign}(\hat{x}_{ij}) \cdot \max(0, |\hat{x}_{ij}| - \frac{\beta}{2}). \tag{9}$$

The entire optimization is summarized in **Algorithm 1**, in which the base matrix $\mathbf{U}$ determines the data representation ability.

Algorithm 1 Algorithm for the Initial Appearance Model

Input: data matrix $\mathbf{X} \in \mathbb{R}^{M\times N}$, $1 \le K \le \min\{M, N\}$, related regularization parameters: λ, μ, γ and β
Output: base matrix $\mathbf{U} \in \mathbb{R}^{M\times K}$, coefficient matrix $\mathbf{V} \in \mathbb{R}^{K\times N}$, and error matrix $\mathbf{E} \in \mathbb{R}^{M\times N}$

Set: stopping error ε.
Construct the weight matrix $\mathbf{W}$ by using Eq. (3) and the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$.
Initialize $i = 0$, $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{E}$ with random positive values.
Repeat
  Update $\mathbf{U}^{i+1}$ and $\mathbf{V}^{i+1}$ by Eq. (7);
  Update $\mathbf{E}^{i+1}$ by Eq. (9);
  $i := i + 1$;
Until $\frac{\|\mathbf{U}^{i+1} - \mathbf{U}^i\|_{\mathrm{F}}}{\|\mathbf{U}^i\|_{\mathrm{F}}} \le \varepsilon$;

C. Incremental Learning Model

The traditional updating rules are used in many applications such as document analysis [46] and face recognition [20]. However, it is not applicable to visual tracking because of the unaffordable computational and storage costs. An incremental updating scheme is therefore proposed based on [47] and [48]. The assumption behind the incremental NMF (INMF) is that the previous coefficient matrix $\mathbf{V}$ has no effect on the incremental process when a new sample $\mathbf{x}$ is added, namely: $[\mathbf{X}, \mathbf{x}] \approx \mathbf{U} \times [\mathbf{V}, \mathbf{v}]$. In our method, we propose another assumption that the previous error matrix $\mathbf{E}$ does not change during the incremental process except when a new sample arrives, namely: $\mathbf{x} = \mathbf{U}\mathbf{v} + \mathbf{e}$. In other words, incremental updating scheme can be used for $\mathbf{v}$ and $\mathbf{e}$, whilst $\mathbf{U}$ needs to be recalculated entirely.

In the incremental updating model, $\mathbf{X}_{t+1} = [\mathbf{X}_t, \mathbf{x}]$, $\mathbf{V}_{t+1} = [\mathbf{V}_t, \mathbf{v}]$, $\mathbf{E}_{t+1} = [\mathbf{E}_t, \mathbf{e}]$, where $\mathbf{U}_{t+1}$, $\mathbf{W}_{t+1}$, and $\mathbf{D}_{t+1}$ are the corresponding matrices when the $(t+1)$-th sample arrives. Therefore, the corresponding objective function $\mathcal{O}_{t+1}(\mathbf{U}_{t+1}, \mathbf{v}, \mathbf{e})$ is rewritten as:

$$\begin{aligned}\mathcal{O}_{t+1} = {} & \|\mathbf{X}_{t+1} - \mathbf{U}_{t+1}\mathbf{V}_{t+1} - \mathbf{E}_{t+1}\|_{\mathrm{F}}^2 + \beta\|\mathbf{E}_{t+1}\|_{1,1} \\ & + \lambda \mathrm{tr}\left(\mathbf{V}_{t+1}\mathbf{L}_{t+1}\mathbf{V}_{t+1}^\top\right) + \gamma\, \mathrm{tr}\left(\mathbf{U}_{t+1}\mathbf{O}\mathbf{U}_{t+1}^\top\right) \\ & + \mu \sum_{i=1}^{t+1} \left\| \left(\mathbf{x}_i \mathbf{1}^\top - \mathbf{U}_{t+1}\right) \Lambda_i^{1/2} \right\|_{\mathrm{F}}^2. \end{aligned} \tag{10}$$

1) Incremental Updating Rules for $\mathbf{U}_{t+1}$: Given $\mathbf{E}_{t+1}$ and $\mathbf{V}_{t+1}$, $\bar{\mathbf{X}}_{t+1} = \mathbf{X}_{t+1} - \mathbf{E}_{t+1}$, Eq. (10) is equivalent to the following formulation:

$$\begin{aligned}\mathcal{O}_{t+1}(\mathbf{U}_{t+1}) = {} & \left\| \bar{\mathbf{X}}_{t+1} - \mathbf{U}_{t+1}\mathbf{V}_{t+1} \right\|_{\mathrm{F}}^2 + \gamma\, \mathrm{tr}\left(\mathbf{U}_{t+1}\mathbf{O}\mathbf{U}_{t+1}^\top\right) \\ & + \mu \sum_{i=1}^{t+1} \left\| \left(\bar{\mathbf{x}}_i \mathbf{1}^\top - \mathbf{U}_{t+1}\right) \Lambda_i^{1/2} \right\|_{\mathrm{F}}^2. \end{aligned} \tag{11}$$

Let $\Psi = [\psi_{pq}]$ be a Lagrange multiplier for the nonnegative constraint on $\mathbf{U}_{t+1}$, then the related Lagrange function is $\mathcal{L}_{\mathbf{U}_{t+1}} = \mathcal{O}_{t+1}(\mathbf{U}_{t+1}) + \mathrm{tr}(\Psi\mathbf{U}_{t+1})$. The partial derivative of $\mathcal{L}_{\mathbf{U}}$ with respect to $\mathbf{U}_{t+1}$ is therefore computed as:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathbf{U}}}{\partial \mathbf{U}_{t+1}} = {} & -2\left(\bar{\mathbf{X}}_{t+1} - \mathbf{U}_{t+1}\mathbf{V}_{t+1}\right)\mathbf{V}_{t+1}^\top + \Psi \\ & + \gamma\, \mathbf{U}_{t+1}\left(\mathbf{O} + \mathbf{O}^\top\right) + \mu \sum_{i=1}^{t+1} \left(-2\bar{\mathbf{x}}_i \mathbf{1}^\top \Lambda_i + 2\mathbf{U}_{t+1}\Lambda_i \right). \end{aligned} \tag{12}$$

$$(\mathbf{U}_{t+1})_{pq} \leftarrow (\mathbf{U}_{t+1})_{pq} \cdot \frac{[(\mu+1)(\bar{\mathbf{X}}_t\mathbf{V}_t^\top + \bar{\mathbf{x}}\mathbf{v}^\top)]_{pq}}{[\mathbf{U}_{t+1}\mathbf{V}_t\mathbf{V}_t^\top + \mathbf{U}_{t+1}\mathbf{v}\mathbf{v}^\top + \mu\mathbf{U}_{t+1}\mathbf{H}_t + \mathbf{U}_{t+1}\,\mathrm{diag}(\mathbf{v}) + \gamma_1\mathbf{U}_{t+1}(\mathbf{O}+\mathbf{O}^\top)]_{pq}} \tag{13}$$

$$v_j \leftarrow v_j \cdot \frac{[(\mu+1)\mathbf{U}_{t+1}^\top\bar{\mathbf{x}} - \mu_1\mathbf{1}\bar{\mathbf{x}}^\top\bar{\mathbf{x}} + \lambda\mathbf{V}_t(\mathbf{W}_{t+1})_{:,t+1} + \lambda\mathbf{v}w_{end}]_j}{[\mathbf{U}_{t+1}^\top\mathbf{U}_{t+1}\mathbf{v} + \lambda\mathbf{V}_t(\mathbf{D}_{t+1})_{:,t+1} + \lambda\mathbf{v}d_{end} + \mu_1\,\mathrm{diag}(\mathbf{U}_{t+1}^\top\mathbf{U}_{t+1})]_j} \tag{14}$$

By using the Karush-Kuhn-Tucker (KKT) condition, $\mathbf{X}_{t+1} = [\mathbf{X}_t, \mathbf{x}]$ and $\mathbf{V}_{t+1} = [\mathbf{V}_t, \mathbf{v}]$, the updating rule for $\mathbf{U}_{t+1}$ is formulated as Eq. (13), as shown at the top of this page, where $\gamma_1 = \frac{1}{2}\gamma$.

2) Incremental Updating Rules for $\mathbf{v}$*:* It is nontrivial to update $\mathbf{v}$ involved with Λ_i. After the factorization operation on $\|\cdot\|_\mathrm{F}$ and $\mathbf{V}$, and also omitting some irrelevant terms (more details are provided in Appendix. C), Eq. (10) can be transformed to:

$$\begin{aligned}\mathcal{O}_{t+1}(\mathbf{v}) = \|\bar{\mathbf{x}} - \mathbf{U}_{t+1}\mathbf{v}\|_2^2 + 2\lambda\mathbf{v}^\top\mathbf{V}_t(\mathbf{L}_{t+1})_{:,t+1} + \mu\bar{\mathbf{x}}^\top\bar{\mathbf{x}}\mathbf{1}^\top\mathbf{v} \\ + \lambda\mathbf{v}^\top\mathbf{v}_t l_{end} - 2\mu\mathbf{v}^\top\mathbf{U}_{t+1}^\top\bar{\mathbf{x}} + \mu\mathbf{v}^\top\mathrm{diag}(\mathbf{U}_{t+1}^\top\mathbf{U}_{t+1}).\end{aligned} \tag{15}$$

where $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{e}$, $(\mathbf{L}_{t+1})_{:,t+1}$ represents the $(t+1)$th column of the Laplacian matrix $\mathbf{L}$, and $l_{end} = (\mathbf{L}_{t+1})_{t+1,t+1}$ denotes the element of $\mathbf{L}$ in the last row and last column. Let ϕ_j be a Lagrange multiplier for the nonnegative constraint on $\mathbf{v}$, then the relevant Lagrange function is $\mathcal{L}_\mathbf{v} = \mathcal{O}_{t+1}(\mathbf{v}) + Tr(\phi\mathbf{v})$. Hence the partial derivative of $\mathcal{L}_\mathbf{v}$ with respect to $\mathbf{v}$ is:

$$\begin{aligned}\frac{\partial\mathcal{L}_\mathbf{v}}{\partial\mathbf{v}} = -2(\mu+1)\mathbf{U}_{t+1}^\top\mathbf{x} + \mu\mathbf{1}\mathbf{x}^\top\mathbf{x} + 2\mathbf{U}_{t+1}^\top\mathbf{U}_{t+1}\mathbf{v} + \phi \\ + 2\lambda\mathbf{V}_t(\mathbf{L}_{t+1})_{:,t+1} + 2\lambda\mathbf{v}^\top\mathbf{v}_t l_{end} + \mu\mathrm{diag}(\mathbf{U}_{t+1}^\top\mathbf{U}_{t+1}).\end{aligned} \tag{16}$$

By using the KKT condition, the updating rule for $\mathbf{v}$ is formulated in Eq. (14), as shown at the top of this page, where $\mu_1 = \frac{1}{2}\mu$.

3) Incremental Updating Rules for $\mathbf{e}$*:* Given $\mathbf{U}_{t+1}$ and $\mathbf{v}$, by defining $\hat{\mathbf{x}} = \mathbf{x} - \mathbf{U}_{t+1}\mathbf{v}$, the incremental objective function Eq. (10) with respect to $\mathbf{e}$ is converted to:

$$\mathcal{O}_{t+1}(\mathbf{e}) = \|\hat{\mathbf{x}} - \mathbf{e}\|_2^2 + \beta\|\mathbf{e}\|_1. \tag{17}$$

We omit the derivation of the incremental updating scheme on $\mathbf{e}$, as it is the same as the updating of $\mathbf{E}$ in Eq. (9). The completed incremental updating scheme for our incremental learning model is shown in **Algorithm 2**.

The convergence of the three iterative models can be easily proved in the similar way as in [14], [18], [20], [23], and [48].

Algorithm 2 Algorithm for Incremental Updating Model

Input: the new data matrix $\mathbf{X}_{t+1} = [\mathbf{X}_t, \mathbf{x}] \in \mathbb{R}^{M\times(t+1)}$, base matrix $\mathbf{U}_t \in \mathbb{R}^{M\times K}$, coefficient matrix $\mathbf{V}_t \in \mathbb{R}^{K\times t}$, error matrix $\mathbf{E}_t \in \mathbb{R}^{M\times t}$, and the corresponding parameters

Output: base matrix $\mathbf{U}_{t+1} \in \mathbb{R}^{M\times K}$, coefficient matrix $\mathbf{V}_{t+1} = [\mathbf{V}_t, \mathbf{v}] \in \mathbb{R}^{K\times(t+1)}$, and error matrix $\mathbf{E}_{t+1} = [\mathbf{E}_t, \mathbf{e}] \in \mathbb{R}^{M\times(t+1)}$

Set: stopping error ε.
Construct the weight matrix $\mathbf{W}_{t+1}$ by using Eq. (3) and the Laplacian matrix $\mathbf{L}_{t+1} = \mathbf{D}_{t+1} - \mathbf{W}_{t+1}$.
Initialize $i = 0$ and calculate γ_1, $\mathbf{H}_t$ and w_{end}.
Repeat
  Update $\mathbf{U}_{t+1}^{i+1}$ by Eq. (13);
  Update $\mathbf{v}^{i+1}$ by Eq. (14);
  Update $\mathbf{e}^{i+1}$ by Eq. (9);
  $i := i + 1$;
Until $\frac{\|\mathbf{U}_{t+1}^{i+1} - \mathbf{U}_{t+1}^{i}\|_F}{\|\mathbf{U}_{t+1}^{i}\|_F} \le \varepsilon$;

IV. The Inverse NLCF Model

In this section, we will analyse the inv-NMF/inv-NLCF model[1] from data clustering viewpoint. Compared to the conventional NMF, the base matrix $\mathbf{U}$ in inv-NMF model is spanned by the candidates $\mathbf{Y}$, namely, $\mathbf{U} \approx \mathbf{YC}$. Each column $\mathbf{c}_i$ in $\mathbf{C}$ denotes the coefficients of a certain base vector $\mathbf{u}_i$ projected by all candidates. Each row $\mathbf{c}^{(i)}$ in $\mathbf{C}$ corresponds to the responses of one candidate on the base matrix $\mathbf{U}$, which can be regarded as discriminative feature for classification in visual tracking. Specifically, if the data points are similar to each other (or they are from the same class) as shown in Fig. 2, these base vectors will manifest clustering property in different cues, rather than an oversimplified projection (i.e. PCA) in a low-dimensional space.

A. Inverse NMF Versus NMF

In the conventional NMF for feature coding [16], a candidate $\mathbf{y}$ is represented by a linear combination of base vectors $\mathbf{U}$ with the nonnegative coefficient vector $\mathbf{v}$. These coefficient vectors can be regarded as the discriminative features for different candidates. However, this operation renders that when the base vectors are used to represent a "bad" candidate (i.e. the background region), the reconstruction error ($\|\mathbf{y} - \mathbf{U}\mathbf{v}\|_2^2$) would be large. In this case, the coefficient vector $\mathbf{v}$ cannot accurately represent the candidate. As shown in Fig. 4, a bad candidate is represented by these base vectors with relatively similar encoding coefficients, which means that the encoding coefficients are not discriminative enough to distinguish good candidates from the bad one.

Comparably, the goal of the inverse NMF model is not to reconstruct a base vector by all candidates with the corresponding column vector $\mathbf{c}_i$. Instead, it aims to generate each row $\mathbf{c}^{(i)}$ of $\mathbf{C}$ as a feature vector, which can be regarded as a probability that one candidate projects on the base matrix $\mathbf{U}$. As shown in Fig. 4, the encoding coefficients for a "good" candidate denote that this candidate resembles some base vectors with high response values, while a "bad" candidate has

[1]The main difference between these two models is that the inv-NLCF model incorporates the local coordinate constraint while the inverse NMF does not.

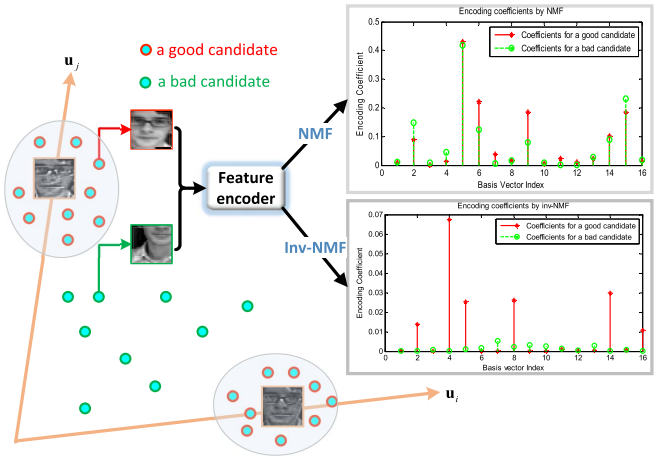


Fig. 4. Encoding coefficients obtained by NMF and inv-NMF feature coder. In NMF coding scheme, the encoding coefficients for a "good" candidate or a "bad" candidate are similar and thus lacking discriminative ability. In our inv-NMF coding scheme, a "good" candidate distributes near a base vector, which indicates that this candidate resembles certain base vectors with high response values. Whilst a "bad" candidate spreads over the whole space, and has extremely low response values to all these base vectors.

extremely low response values to all these base vectors. Such difference of responses for various candidates can be fully exploited to produce discriminative feature vectors, which help to separate the target from the background.

B. Inverse NMF Versus Reverse Sparse Representation

Compared with the reverse sparsity theory in [7] and [33], the encoding vector $\mathbf{c}^{(i)}$ in our method has more distinct advantages with the following two merits.

First, rather than directly using the templates in inverse sparse representation, our NMF based method can explicitly capture information in different cues.

Second, from the perspective of data clustering, if $\mathbf{Y}$ contains a set of good candidates (i.e. similar to the target), these good candidates will spread among the base vectors as shown in Fig. 4. In this case, a few nonzero coefficients in $\mathbf{c}^{(i)}$ are obtained, as base vectors can be easily represented by their good neighbouring candidates. As a result, the sparsity of our method is naturally guaranteed without any additional sparsity constraint. For bad candidates, they are incoherent in the subspace spanned by base vectors. There is no definite relationship between the background and the target representation $\mathbf{U}$. If $\mathbf{Y}$ contains a set of bad candidates sampled from the background, it is difficult for these bad candidates to represent base vectors accurately and sparsely. As a result, these corresponding coefficient vectors do not hold the sparsity property as that in the positive sample case. By exploiting this difference between the candidates in the target and the background, the base matrix $\mathbf{U}$ can be mapped onto the associated coefficient vectors. The resulting coefficient vectors are used as discriminative features to separate the target from the background.

C. Identifying Candidates in Inv-NLCF Model

The local coordinate constraint is incorporated into the inverse NMF method, which helps to preserve the similarity of coefficient vectors. We estimate the positive coefficient vector $\mathbf{C}_{pos}$ using the target patches $\mathbf{T}_{pos}$ as base vectors, namely:

$$\min_{\mathbf{C}_{pos}} \|\mathbf{U}-\mathbf{T}_{pos}\mathbf{C}_{pos}\|_{\mathrm{F}}^2+\zeta\sum_{k=1}^{K}\left\|(\mathbf{u}_k\mathbf{1}^\top-\mathbf{T}_{pos})\Gamma_k^{1/2}\right\|^2 \quad s.t.\ \ \mathbf{C}_{pos}\geq 0, \tag{18}$$

where $\mathbf{1}\in\mathbb{R}^N$ and $\Gamma_k\in\mathbb{R}^{N\times N}$ is a diagonal matrix spanned by the kth column of $\mathbf{C}_{pos}$. Similarly, we can easily derive the formula for estimating $\mathbf{C}_{neg}$ from the negative templates $\mathbf{T}_{neg}$, which is:

$$\min_{\mathbf{C}_{neg}} \|\mathbf{U}-\mathbf{T}_{neg}\mathbf{C}_{neg}\|_{\mathrm{F}}^2+\tau\sum_{k=1}^{K}\left\|(\mathbf{u}_k\mathbf{1}^\top-\mathbf{T}_{neg})\Upsilon_k^{1/2}\right\|_{\mathrm{F}}^2 \quad s.t.\ \ \mathbf{C}_{neg}\geq 0, \tag{19}$$

where $\Upsilon_k\in\mathbb{R}^{r\times r}$ is a diagonal matrix spanned by the kth column of $\mathbf{C}_{neg}$. For training process, each row of $\mathbf{C}_{pos}$ and $\mathbf{C}_{neg}$ corresponding to a positive and a negative feature vector, is sent to SVM classifier for training process. For testing process, the candidates $\mathbf{Y}$ are sampled at each frame, and then are used to estimate the coefficient matrix $\mathbf{C}$ by solving the following constrained optimization problem:

$$\min_{\mathbf{C}} \|\mathbf{U}-\mathbf{Y}\mathbf{C}\|_{\mathrm{F}}^2+\zeta\sum_{k=1}^{K}\left\|(\mathbf{u}_k\mathbf{1}^\top-\mathbf{Y})\Omega_k^{1/2}\right\|_{\mathrm{F}}^2 \quad s.t.\ \ \mathbf{C}\geq 0, \tag{20}$$

where ζ is regularization parameter and $\Omega_k\in\mathbb{R}^{S\times S}$ is a diagonal matrix spanned by the kth column of $\mathbf{C}$.

Note that the objective function in Eqs. (18), (19) and (20) with respect to $\mathbf{C}_{pos}$, $\mathbf{C}_{neg}$ and $\mathbf{C}$ are convex functions w.r.t. the variables to be optimized. Therefore, there are many off-the-shelf methods for solving this constraint linear quadratic programming problem, such as interior point method, and APG [27]. To seek for the unified solving algorithm framework, we still use the similar updating rule for $\mathbf{V}$ in NLCF shown in Eq. (2). For example, each element in the feature vectors $\mathbf{C}$ with respect to candidates $\mathbf{Y}$ is obtained by:

$$c_{sk}^{t+1}\leftarrow c_{sk}^{t}\frac{2(\zeta+1)(\mathbf{Y}^\top\mathbf{U})_{sk}}{(2\mathbf{Y}^\top\mathbf{Y}\mathbf{C}+\zeta\mathbf{G}_1+\zeta\mathbf{F}_1)_{sk}}, \tag{21}$$

where $\mathbf{G}_1=(\mathbf{f},\mathbf{f},\ldots,\mathbf{f})^\top\in\mathbb{R}^{S\times K}$. Similarly, the column vector is defined as $\mathbf{f}_1=\mathrm{diag}(\mathbf{Y}^\top\mathbf{Y})\in\mathbb{R}^S$, and $\mathbf{F}_1=(\mathbf{f}_1,\mathbf{f}_1,\ldots,\mathbf{f}_1)\in\mathbb{R}^{S\times K}$. After several iterations, the optimal $\mathbf{C}$ (also includes $\mathbf{C}_{pos}$ and $\mathbf{C}_{neg}$) is obtained. Subsequently, the SVM classifier is employed to assign the encoding feature vector $\mathbf{c}^{(i)}$ to the target or the background.

V. NLCF Variants in Tracking Framework

In this section, we incorporate the above models into our tracking framework.

A. Particle Filter in Visual Tracking Framework

Generally, particle filter is based on the theory of Bayesian inference. The rationale behind particle filter is to estimate the posterior distribution $p(\mathbf{z}_t|\mathbf{Y}_{1:t})$ by a finite set of randomly

sampled particles. Given some observed image patches at the tth frame $\mathbf{Y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{t-1}\}$, the state of the target $\mathbf{z}_t$[2] can be recursively estimated as follows:

$$p(\mathbf{z}_t|\mathbf{Y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{z}_t)\int p(\mathbf{z}_t|\mathbf{z}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{Y}_{1:t-1})\mathrm{d}\mathbf{x}, \quad (22)$$

where $p(\mathbf{z}_t|\mathbf{z}_{t-1})$ is a motion model depicting state transition between two consecutive frames, subject to Gaussian distribution with mean $\mathbf{z}_{t-1}$ and variance σ^2. The optimal state at tth frame is then obtained by maximizing the approximate posterior probability:

$$\mathbf{z}_t^* = \underset{\mathbf{z}_t}{\operatorname{argmax}}\, p(\mathbf{y}_t|\mathbf{z}_t)p(\mathbf{z}_t|\mathbf{z}_{t-1}). \quad (23)$$

In our proposed observation model, the observation likelihood can be measured by the reconstruction error of each observed image patch, namely:

$$p(\mathbf{y}_t^i|\mathbf{z}_t^i) \propto \exp(-\|\mathbf{y}_t^i - \mathbf{U}\mathbf{v}_t^i\|_2^2), \quad \forall i. \quad (24)$$

To make our algorithm more robust, a coarse-to-fine search scheme for the optimal candidate is proposed. After obtaining $\mathbf{Y}^+$ and $\mathbf{Y}^-$, we do not simply choose the candidate with the highest classification confidence value as our tracking result.[3] The observation likelihood can be measured by the reconstruction error of positive candidates $\mathbf{Y}^+$ (noting that the time index t is omitted for simplicity):

$$p(\mathbf{y}_i^+|\mathbf{z}_i) = \underset{j}{\operatorname{argmax}} \exp(-\|\mathbf{y}_i^+ - \mathbf{U}\mathbf{v}_j\|_2^2), \quad \forall j, \quad (25)$$

where $\mathbf{y}_i^+$ represents the ith positive candidate from $\mathbf{Y}^+$, and $\mathbf{v}_j$ denotes the jth column of coefficient matrix $\mathbf{V}$. The optimal state $\mathbf{z}^*$ from the positive samples $\mathbf{Y}^+$ with the minimal reconstruct error is chosen as the tracking result. Such searching scheme incorporates the merit of generative methods into the discriminative classification problem.

For incremental updating in visual tracking, every ten frame, the tracking result closest to the mean value of these ten results is chosen as the newly added sample $\mathbf{x}^*$ into $\mathbf{X}$. Note that, to ensure the dimension of $\mathbf{X}$ unchanged to save memory space, we discard the element in $\mathbf{X}$ closest to $\mathbf{x}^*$. By our incremental updating model, the base matrix $\mathbf{U}$ is totally updated. The corresponding coefficient vector $\mathbf{v}^*$ and error vector $\mathbf{e}^*$ are updated while the remaining vectors are kept unchanged with details in **Algorithm 2**. Finally, we summarize the proposed tracker in **Algorithm 3** below.

Algorithm 3 Algorithm for the Proposed Tracker

for $t = 1$ *to* m **do**
  Use a simple tracker;
  Extract samples $\mathbf{T}_{pos}$ and $\mathbf{T}_{neg}$ in tth frame;
end
Obtain the data matrix $\mathbf{X}$ and the initial appearance model by **Algorithm 1**.
for $t = m + 1$ *to the end of the sequence* **do**
  S particles $\mathbf{Y}_{1:S}$ are sampled;
  Inverse NLCF: obtain encoding vectors in each row of $\mathbf{C}_{pos}$, $\mathbf{C}_{neg}$ and $\mathbf{C}$ by Eq. (21);
  Train a SVM classifier by $\mathbf{C}_{pos}$, $\mathbf{C}_{neg}$, and then conduct classification on the encoding vector $\mathbf{C}$;
  for *each positive particle* $\mathbf{y}_i^+$ **do**
    Compute their likelihood by Eq. (25);
  end
  Choose $\mathbf{x}_t^*$ with the minimal reconstruction error;
  Update: **for** *each* 10 *frames* **do**
    Choose the best tracking result $\mathbf{x}_{m_*}^*$ and then replace $\mathbf{x}^*$;
    Recalculate $\mathbf{W}$ and $\mathbf{D}$ by Eq. (3);
    Employ the incremental scheme in **Algorithm 2** after the updated $\mathbf{X}$ is obtained;
  end
end

VI. Experiments

In this section, we test the proposed tracker on the Object Tracking Benchmark (OTB) [1] with 29 trackers and 51 video sequences. Besides, SST [28] based on sparsity theory is also compared. Experiments including parameter analysis and computational complexity analysis are also further provided.

Setup: Our tracker was implemented in MATLAB on a PC with Intel Xeon E5506 CPU (2.13 GHz) and 24 GB memory. The following parameters were used for our tests: the graph-regularized parameter was set to $\lambda = 1$; the number of initial positive templates and the negative templates were $N = 140$ and $r = 280$, respectively; the number of base vectors was $K = 16$; the corresponding regularization parameters in Eq. (6) were $\mu = 0.001$, $\gamma = 0.001$ and $\beta = 1$; the local coordinate regularization parameter in Eq. (18) and Eq. (20) were $\zeta = 0.1$;

A. Qualitative Evaluation

1) Evaluation Metrics: Similar to [1], two evaluation methods are used in one pass evaluation (OPE): precision plot and success plot. They show the percentage of successfully tracked frames measured by two widely used metrics: mean center location error (CLE) and Pascal VOC Overlap Ratio (VOR) [49]. Small CLE value indicates accurate and good tracking result. The overlap ratio measures the overlapping rate between the tracked bounding box and the ground truth box, which is defined as $e = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)}$, where R_T and R_G are the areas of the tracked and ground truth boxes, respectively.

To rank these trackers, two types of ranking metrics are provided in [1]. One is the Area Under the Curve (AUC) metric for the success plot, and the other is the representative precision score at threshold of 20 pixels for the precision plot.

2) Overall Performance: We show the overall performance of OPE for our tracker and compare it with some other state-of-the-arts (ranked within top 10) as shown in Fig. 5. The top 5 trackers on success plot include SCM [38], SST [28],

[2]The state $\mathbf{z}_t = [p_x, p_y, \theta, s, \alpha, \phi]$ represents translation on X, Y direction, rotation angle, scale, aspect ratio, and skew respectively.

[3]In our experiments, statistical results show that the number of $\mathbf{Y}^+$ accounts for about 10% of the whole $\mathbf{Y}$.

TABLE I

AVERAGE CENTER LOCATION ERRORS (CLE) AND AVERAGE VOC OVERLAP RATIO (VOR) OF VARIOUS ALGORITHMS FROM OTB WITH 51 TASKS. THE FIRST, SECOND AND THIRD BEST SCORES ARE HIGHLIGHTED BY **BOLD**, UNDERLINE AND *Italic*, RESPECTIVELY

Method	Ours	Ours	MUSTER [3]	MEEM [4]	KCF [8]	IMT [52]	CFLB [53]	CN [54]	IST [7]	NMF	Sparse	Classification
Feature	HOG	Gray	HOG	4-channel[1]	HOG	Mutiple[2]	Gray	Color	Gray	Gray	Gray	Gray
CLE	*33.7*	42.3	**17.3**	22.3	35.5	48.7	90.6	64.8	85.3	78.1	70.6	48.3
VOR(%)	*54.2*	50.9	**65.0**	57.9	51.9	51.7	37.8	44.8	38.1	40.2	41.1	49.7

[1] 3-channel in CIE Lab color space and a non-parametric local rank transformation on L channel.
[2] IMT method incorporates three features including gray intensity, HOG and Haar.

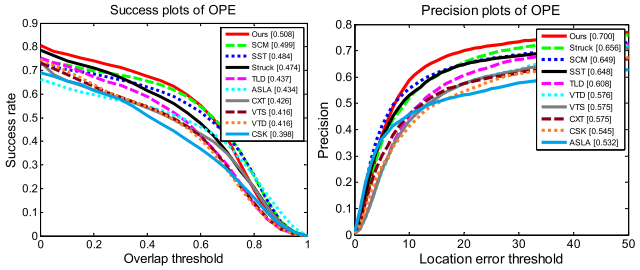


Fig. 5. Plots of OPE. The performance score for each tracker is shown in the legend. For each figure, the top 10 trackers are presented for clarity.

Struck [50], TLD [51] and our method. It can be observed that our method ranks first on the success plot and precision plot. The satisfactory performance is largely dependent on the accurate appearance model and discriminative feature vectors generated by the inv-NLCF model in our method.

Besides, in Tab. I, we also give the corresponding results (CLE and VOR) of recent state-of-the-art methods including correlation filter based trackers MUSTER [3], CFLB [53], KCF [8] and CN [54]; ensemble based trackers MEEM [4] and IMT [52]. Moreover, we compare the extended version of the proposed method with HOG feature. Among these trackers, the top three trackers are MUSTER, MEEM, and our method (with HOG feature) respectively. By comparing with the two best existing trackers MUSTer and MEEM, we see that our tracker still has room for improvement. For example, the proposed method can combine short-term and long-term templates to obtain better performance.

3) Attribute Based Performance Analysis: Each sequence in OTB [1] is annotated with eleven attributes that indicate which challenging factors are included. In Fig. 6, we present the top 10 trackers on success plots and precision plots in terms of four main attributes. On *Deformation*, our method outperforms the other methods and obtains 12% improvements on the precision plot than the second best tracker. This is mainly due to the accurate appearance model in the NMF variants. On *Occlusion*, the proposed tracker achieves the top level performance in the precision plot and the success plot. On the remaining attributes, our method also yields promising performance.

4) Qualitative Evaluation: Fig. 7 shows a qualitative comparison of our tracker with four baseline methods on 16 extremely challenging videos. We see that IVT, ASLA and Struck often lose the target completely when it suffers from severe occlusions (e.g. *SUV* and *Jogging.2*) expect our method and SCM. However, SCM often fails to track the object with pose variations (e.g. *Basketball* and *Singer2*), background clutters (e.g. *Freeman4*). Apart from SCM, most tracker are also not able to accurately predict the location of the target when abrupt motion (*Deer*) and motion blur (*Jumming*) occurs. In contrast, our tracker effectively tackles the above challenging factors such as occlusions, shape deformation, undesirable illuminations, and motion blur, etc.

B. Key Component Validation

Above comparisons have shown that our tracker is superior to other existing methods, and this section studies the effect of every key component in our algorithm, and see how these components contributes to improving the performance.

1) Influence of Different Feature Coding Methods: We quantitatively analyse the influence of four feature coding model (our inv-NLCF model, NMF, reverse sparse representation based tracker IST [7], and sparse representation) in Tab. I. They are named as "Ours (Gray)", "NMF", IST [7] and "sparse" respectively. In "NMF" method, the inverse NMF representation is substituted by the conventional NMF method that is used in our inv-NLCF model. To verify the importance of locality coding methods, "sparse" method is proposed, which does not consider local coordinate coding constraint. In the appearance model and inv-NLCF model, the locality constraint is replaced by a $L_{1,1}$ norm for sparse representation in Eq. (6), Eq. (18), Eq. (19), and Eq. (20). In terms of average CLE and average VOR, our method based on inverse NMF representation outperforms standard NMF for coding owing to the more discriminative feature. Compared to inverse sparse representation, the proposed inverse NMF model provides a justification for natural sparsity property.

Besides, we analyze the coarse-to-fine search to verify the effectiveness of the inverse NMF coding scheme. As mentioned before, a coarse-to-fine search scheme is proposed to obtain the optimal candidate, which chooses a positive candidate with the minimal reconstruction error. Here, we only use SVM classifier to choose the optimal candidate with the highest classification confidence, which serves as a coarse search scheme (termed as "Classification"). Compared to the proposed method, the "Classification" method shows a slight decline in tracking performance in terms of CLE and VOR as shown in the last column of Tab. I. Although this scheme may not obtain an optimal candidate, it still guarantees a sub-optimal candidate as demonstrated.

2) Influence of Different Constraints: We quantitatively show the influence of different regularization terms on OTB tracking results in Fig. 8. When β is set to zero, the initial

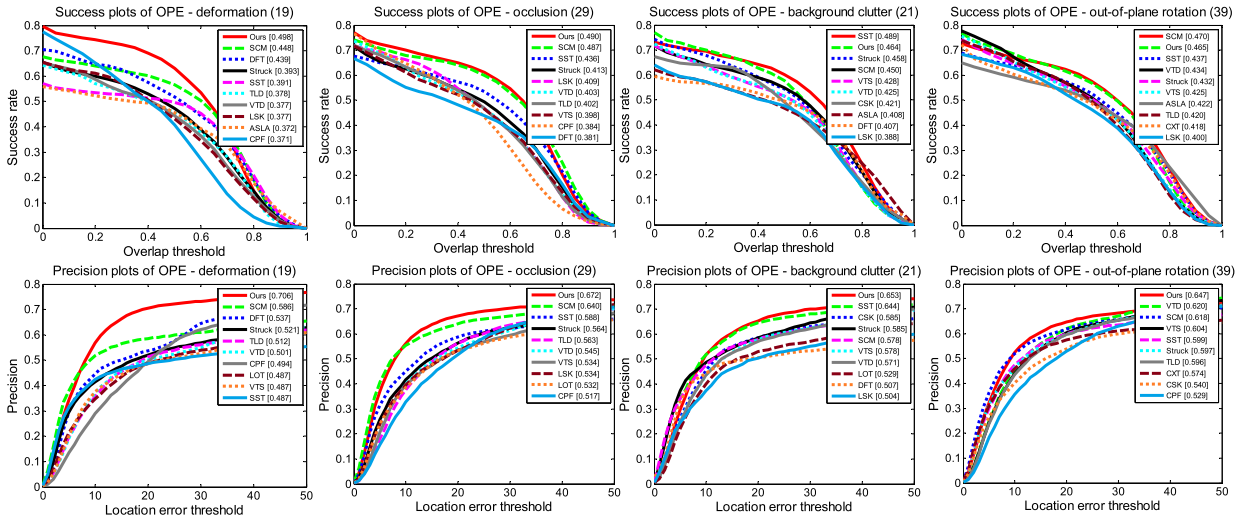


Fig. 6. Success plots and precision plots of OPE on four main attributes (*Deformation*, *Occlusion*, *Background Clutter*, and *Out-of-plane Rotation*).

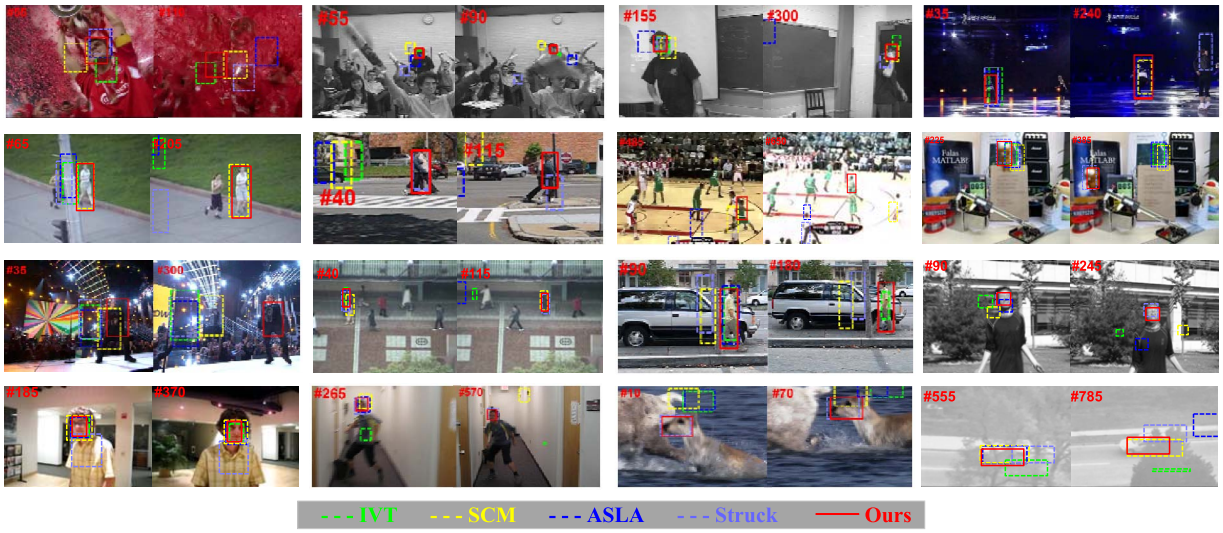


Fig. 7. Representative frames in the tracking results. The subfigures from top to bottom, left to right are sequences: *Soccer*, *Freeman4*, *Freeman1*, and *Skating1*; *Jogging.2*, *Couple*, *Basketball*, and *Lemming*; *Singer2*, *Subway*, *David3*, and *Jumping*; *David*, *Boy*, *Deer*, and *Suv*.

appearance model overlooks the sparse error constraint, named as "No error" method. Its success rate degenerates about 8.1% when compared to the proposed method. Without NLCF constraint ("No NLCF") in the initial appearance model, the success rate drops to 44.5%. When ζ ("No inv-NLCF") is not taken into consideration by inv-NLCF model, the success rate is as low as 44.3%. This result is similar to that in the initial appearance model without μ. The success rate decreases to 43.1% ($\lambda = 0$) in "No graph" method and 43.4% ($\gamma = 0$) in "No orth" method respectively.

Among these five regularization terms, sparse error constraint mostly affects the tracking results. The second-ranked constraint is orthogonality constraint with a reduction of 7.7%. The remaining three constraints also decreases the final tracking results, with specific values from 50.8% to around 44%.

C. Convergence and Computational Complexity Analysis

We show the convergence curves of different NMF variants for the initial appearance model. These curves have been averaged on 51 sequences on OTB as shown in Fig. 9(a). Compared with the conventional NMF and GNMF, the proposed method achieves higher accuracy precision with less iterations to reach steady state. In Fig. 9(b), we give a comparison between NMF and NLCF in inverse NLCF model.

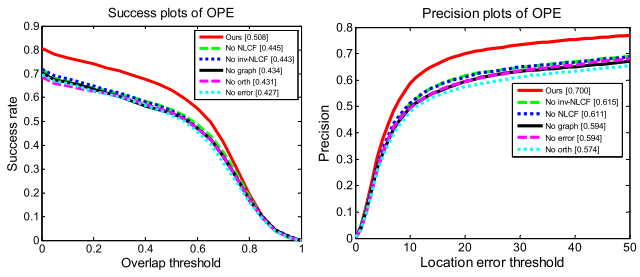


Fig. 8. Plots of OPE. The success plot and precision plot of our proposed tracker versus different regularization terms, where "No graph" is associated with λ, "No error" is with respect to β, "No NLCF" is with μ, and "No orth" relates to γ in Eq. (6). ζ is the regularization parameter in inv-NLCF model associated to "No inv-NLCF".

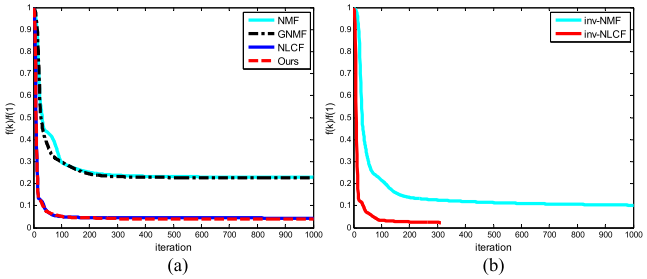


Fig. 9. Objective values versus iteration numbers for NMF variants in (a) the initial appearance model and (b) the inverse NLCF model, where the stopping error is set to 10^{-5}, and the maximal iteration time is fixed to 1000.

Fig. 10. Two failed tracking cases: *Bolt* (the left); *MotorRolling* (the right).

These two iteration curves have been respectively averaged on the corresponding curves at each frame in *Car4* sequences. Compared to NMF ($\zeta = 0$ in Eq. (20)), NLCF obtains a smaller residual error and also needs fewer iterations (about 300 iterations). Next we will analyze the computational complexity of the proposed method involved with the initial appearance model and the inv-NLCF model.

For the initial appearance model, suppose that the multiplicative updates stop after t iterations, the total cost for NLCF is $\mathcal{O}(tMNK)$ demonstrated in [20] and [23]. Besides, the graph regularizer needs $\mathcal{O}(pN^2M)$ to construct the p-nearest neighbour graph. Therefore the overall time cost for graph-based NLCF is $\mathcal{O}(tMNK + pN^2M)$. Based on this, the orthogonality constraint is introduced into our appearance model. Although this term increases MK^2 fladd (a floating-point addition), MK^2 flmlt (a floating-point multiplication) and MN fladd on $\hat{\mathbf{X}}$, the computational complexity still remains unchanged. And also, the introduced sparse error constraint in Eq. (9) incurs $\mathcal{O}(MK)$ due to the computational complexity of shrinkage operation. Finally, the overall computational complexity of our proposed initial appearance model is $\mathcal{O}(t(MNK + MK) + pN^2M)$. Likewise, the computational complexity for our inv-NLCF model is the same as that of NLCF algorithm, namely $\mathcal{O}(tMNK)$.

D. Failure Cases

As shown in our experiments, the proposed method can address these factors including deformation, occlusion, out-of-plane rotation and other attributes. However, our method may fail if the object is affected by drastic appearance variations, and abrupt motion, especially when the target undergoes strong illumination and background clutters as shown in Fig. 10.

In *Bolt* sequence, due to the appearance variations and fast motion of Bolt (i.e. the runner), it is difficult to accurately predict the location of Bolt. In *MotorRolling* sequence, when the motorcyclist undergoes illumination variation and background clutters, the corresponding feature vectors lack discriminative ability to separate the target from the background. From this point of view, the proposed method can still be improved to handle some extreme cases.

VII. Conclusion

This paper introduces the new inv-NLCF model for visual tracking from the viewpoint of data clustering. It combines merits of generative tracking methods and discriminative methods to learn an accurate appearance model and discriminative encoding vectors. Such two scheme help our tracker yield enhanced discriminant ability, and thus effectively separates the target from the background during the tracking process. Quantitative and qualitative comparisons on OTB have demonstrated the effectiveness and robustness of the proposed tracker.

Appendix A
The Base Orthogonality Constraint

We define $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_1, \ldots, \mathbf{u}_n] \in \mathbb{R}^{m\times n}$, and expand the jth dominant element of $\mathbf{UOU}^\top$ in Eq. (4), which leads to:

$$(\mathbf{UOU}^\top)_{jj} = u_{j1}\sum_{i\neq 1}^{n} u_{ji} + u_{j2}\sum_{i\neq 2}^{n} u_{ji} + \cdots + u_{jn}\sum_{i\neq n}^{n} u_{ji}. \tag{26}$$

Then we seek for the relationship between $\mathrm{tr}(\mathbf{UOU}^\top)$ and $\sum_{i\neq j}\mathbf{u}_i^\top\mathbf{u}_j$ by expanding $\mathrm{tr}(\mathbf{UOU}^\top)$, and arrive at:

$$\begin{aligned}
\mathrm{tr}(\mathbf{UOU}^\top) &= \sum_{k=1}^{m}(\mathbf{UOU}^\top)_{kk} = \sum_{j=1}^{n} u_{1j}\sum_{i\neq j}^{n} u_{1i} \\
&\quad + \sum_{j=1}^{n} u_{2j}\sum_{i\neq j}^{n} u_{2i} + \cdots + \sum_{j=1}^{n} u_{mj}\sum_{i\neq j}^{n} u_{mi} \\
&= \sum_{k=1}^{m}\sum_{j=1}^{n} u_{kj}\sum_{i\neq j}^{n} u_{ki} = \sum_{k=1}^{m}\sum_{i=1}^{n} u_{ki}\sum_{j\neq i}^{n} u_{kj} \\
&= \mathbf{u}_1^\top\sum_{j\neq 1}^{n}\mathbf{u}_j + \mathbf{u}_2^\top\sum_{j\neq 2}^{n}\mathbf{u}_j + \cdots + \mathbf{u}_n^\top\sum_{j\neq n}^{n}\mathbf{u}_j \\
&= \sum_{i\neq j}^{n}\mathbf{u}_i^\top\mathbf{u}_j.
\end{aligned} \tag{27}$$

Appendix B
Iteration Rules for the Initial Appearance Model

We first omit the irrelevant term with respect to $\mathbf{E}$, and Eq. (6) is rewritten as:

$$\begin{aligned}
\mathcal{O}(\mathbf{U},\mathbf{V}) &= \|\bar{\mathbf{X}} - \mathbf{UV}\|_{\mathrm{F}}^2 + \lambda\,\mathrm{tr}(\mathbf{VLV}^\top) \\
&\quad + \mu\sum_{i=1}^{N}\|(\mathbf{x}_i\mathbf{1}^\top - \mathbf{U})\Lambda_i^{1/2}\|_{\mathrm{F}}^2 + \gamma\,\mathrm{tr}(\mathbf{UOU}^\top).
\end{aligned} \tag{28}$$

Let ψ_{jk} and ϕ_{ki} be Lagrange multipliers for nonnegative constraints $u_{ik} \geq 0$ and $v_{ki} \geq 0$, respectively, and define the

matrix $(\Psi)_{jk} = \psi_{jk}$ and $(\Phi)_{ki} = \phi_{ki}$, then the Lagrange function $\mathcal{L}$ is:

$$\mathcal{L} = \mathrm{tr}(\Psi\mathbf{U}^\top + \bar{\mathbf{X}}\bar{\mathbf{X}}^\top + \mathbf{U}\mathbf{V}\mathbf{V}^\top\mathbf{U}^\top - 2\bar{\mathbf{X}}\mathbf{V}^\top\mathbf{U}^\top + \gamma\,\mathbf{U}\mathbf{O}\mathbf{U}^\top) \\ \mathrm{tr}(\Phi\mathbf{V}) + \mathrm{tr}\left(\mu\sum_{i=1}^{N}(\bar{\mathbf{x}}_i\mathbf{1}^\top\Lambda_i\bar{\mathbf{x}}_i^\top - 2\bar{\mathbf{x}}_i\mathbf{1}^\top\Lambda_i\mathbf{U}^\top + \mathbf{U}^\top\Lambda_i\mathbf{U})\right). \tag{29}$$

The partial derivatives of $\mathcal{L}$ with respect to $\mathbf{U}$ and $\mathbf{V}$ are:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{U}} = 2\mathbf{U}\mathbf{V}\mathbf{V}^\top - 2\bar{\mathbf{X}}\mathbf{V}^\top + \mu\sum_{i=1}^{N}(-2\bar{\mathbf{x}}_i\mathbf{1}^\top\Lambda_i \\ + 2\mathbf{U}\Lambda_i) + 2\gamma\,\mathbf{U}\mathbf{O} + \Psi \\ \frac{\partial\mathcal{L}}{\partial\mathbf{V}} = 2\mathbf{U}^\top\mathbf{V}\mathbf{V} - 2\mathbf{U}\bar{\mathbf{X}} + \mu(\mathbf{G} - 2\mathbf{U}\bar{\mathbf{X}} + \mathbf{F}) + 2\lambda\mathbf{V}\mathbf{D} + \Phi, \tag{30}$$

where $\mathbf{G}$ and $\mathbf{F}$ have been defined as mentioned in Section.III-B. By using the KKT conditions and some straightforward algebraic manipulations, we can obtain the following updating rules:

$$u_{jk}^{t+1} \leftarrow u_{jk}^{t}\frac{\mu(\bar{\mathbf{X}}\mathbf{V}^\top + \mu\sum_{i=1}^{N}\bar{\mathbf{x}}_i\mathbf{1}^\top\Lambda_i)_{jk}}{(\mathbf{U}\mathbf{V}\mathbf{V}^\top + \mu\sum_{i=1}^{N}\mathbf{U}\Lambda_i + \gamma\,\mathbf{U}\mathbf{O})_{jk}} \\ v_{ki}^{t+1} \leftarrow v_{ki}^{t}\frac{2((\mu+1)\mathbf{U}^\top\bar{\mathbf{X}} + \lambda\mathbf{V}\mathbf{W})_{ki}}{(2\mathbf{U}^\top\mathbf{U}\mathbf{V} + \mu\mathbf{G} + \mu\mathbf{F} + 2\lambda\mathbf{V}\mathbf{D})_{ki}}. \tag{31}$$

Note that $\sum_{i=1}^{N}\bar{\mathbf{x}}_i\mathbf{1}^\top\Lambda_i = \bar{\mathbf{X}}\mathbf{V}^\top$ and $\sum_{i=1}^{N}\mathbf{U}\Lambda_i = \mathbf{U}\mathbf{H}$, we see that Eq. (31) can be exactly rewritten as Eq. (7).

Appendix C
The Objective Function of Incremental Update on $\mathbf{v}$

To tackle the incremental update on $\mathbf{v}$, Eq. (10) should be expanded on $\mathbf{V}$ and $\mathbf{v}$. By omitting the terms in Eq. (10) irrelevant to $\mathbf{V}$ and $\mathbf{v}$, we have:

$$\bar{\mathcal{O}}_{t+1} = \left\|\bar{\mathbf{X}}_t - \mathbf{U}_{t+1}\mathbf{V}_t\right\|_{\mathrm{F}}^2 + \lambda\mathrm{tr}\left(\mathbf{V}_{t+1}\mathbf{L}_{t+1}\mathbf{V}_{t+1}^\top\right) \\ + \|\mathbf{x} - \mathbf{U}_{t+1}\mathbf{v}\|_2^2 + \mu\sum_{i=1}^{t}\left\|\left(\bar{\mathbf{x}}_i\mathbf{1}^\top - \mathbf{U}_{t+1}\right)\Lambda_i^{1/2}\right\|_{\mathrm{F}}^2 \\ + \mu\|(\mathbf{x}\mathbf{1}^\top - \mathbf{U}_{t+1})\Lambda_{t+1}^{1/2}\|_{\mathrm{F}}^2, \tag{32}$$

where $\bar{\mathbf{X}}_{t+1} = \mathbf{X}_{t+1} - \mathbf{E}_{t+1}$. After expanding $\lambda\mathrm{tr}(\mathbf{V}_{t+1}\mathbf{L}_{t+1}\mathbf{V}_{t+1}^\top)$ and making some mathematical rearrangements, we have:

$$\mathrm{tr}(\mathbf{V}_{t+1}\mathbf{L}_{t+1}\mathbf{V}_{t+1}^\top) \\ = \sum_{l=1}^{r}\sum_{i=1}^{t}\sum_{j=1}^{t}(\mathbf{V}_t)_{li}(\mathbf{L}_{t+1})_{ij}(\mathbf{V}_t^\top)_{jl} \\ + 2\mathbf{v}^\top\mathbf{V}_t(\mathbf{L}_{t+1})_{:,t+1} + \mathbf{v}^\top\mathbf{v}(\mathbf{L}_{t+1})_{t+1,t+1}. \tag{33}$$

After preserving the terms related to $\mathbf{v}$ and dropping the other terms, Eq. (32) is simplified to the following formulation:

$$\mathcal{F}_{t+1} = \|\mathbf{x} - \mathbf{U}_{t+1}\mathbf{v}\|_2^2 + 2\mathbf{v}^\top\mathbf{V}_t(\mathbf{L}_{t+1})_{:,t+1} \\ + \mathbf{v}^\top\mathbf{v}(\mathbf{L}_{t+1})_{t+1,t+1} + \mu\left\|(\mathbf{x}\mathbf{1}^\top - \mathbf{U}_{t+1})\Lambda_{t+1}^{1/2}\right\|_{\mathrm{F}}^2. \tag{34}$$

By using $\|\mathbf{A}\|_{\mathrm{F}}^2 = \mathrm{tr}(\mathbf{A}^\top\mathbf{A})$ and $\mathrm{tr}(\mathbf{ABC}) = \mathrm{tr}(\mathbf{BCA}) = \mathrm{tr}(\mathbf{CAB})$, Eq. (34) is rewritten as:

$$\mathcal{F}_{t+1} = \|\mathbf{x} - \mathbf{U}_{t+1}\mathbf{v}\|_2^2 + \mu\mathbf{x}^\top\mathbf{x}\mathbf{1}^\top\mathbf{v} + \mu\mathbf{v}^\top diag(\mathbf{U}_{t+1}^\top\mathbf{U}_{t+1}) \\ + 2\lambda\mathbf{v}^\top\mathbf{V}_t(\mathbf{L}_{t+1})_{:,t+1} + \lambda\mathbf{v}^\top\mathbf{v}(\mathbf{L}_{t+1})_{t+1,t+1} \\ - 2\mu\mathbf{v}\mathbf{U}_{t+1}^\top\mathbf{x}, \tag{35}$$

which is exactly what we seek in Eq. (15).

References

[1] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[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, pp. 478–488, 2013.

[3] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (MUSTer): A cognitive psychology inspired approach to object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 749–758.

[4] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 188–203.

[5] C. Gong, K. Fu, A. Loza, Q. Wu, J. Liu, and J. Yang, "PageRank tracker: From ranking to tracking," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 882–893, Jun. 2014.

[6] N. Wang, J. Wang, and D. Yeung, "Online robust non-negative dictionary learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 657–664.

[7] D. Wang, H. Lu, Z. Xiao, and M.-H. Yang, "Inverse sparse tracker with a locally weighted distance metric," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2646–2657, Sep. 2015.

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

[9] F. Liu, T. Zhou, K. Fu, and J. Yang, "Robust visual tracking via constrained correlation filter coding," *Pattern Recognit. Lett.*, vol. 84, pp. 163–169, 2016.

[10] D. Chen, Z. Yuan, G. Hua, Y. Wu, and N. Zheng, "Description-discrimination collaborative tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 345–360.

[11] C. Gong, D. Tao, M. J. Stephen, W. Liu, G. Kang, and J. Yang, "Multi-modal curriculum learning for semi-supervised image classification," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3249–3260, Jul. 2016.

[12] C. Gong, D. Tao, K. Fu, and J. Yang, "Fick's law assisted propagation for semisupervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2148–2162, Sep. 2014.

[13] C. Gong, D. Tao, W. Liu, L. Liu, and J. Yang, "Label propagation via teaching-to-learn and learning-to-teach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1452–1465, Jun. 2017.

[14] Y. Wu, B. Shen, and H. Ling, "Visual tracking via online nonnegative matrix factorization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 3, pp. 374–383, Mar. 2014.

[15] D. Wang and H. Lu, "On-line learning parts-based representation via incremental orthogonal projective non-negative matrix factorization," *Signal Process.*, vol. 93, no. 6, pp. 1608–1623, 2013.

[16] C. Qian, Y. Zhuang, and Z. Xu, "Visual tracking with structural appearance model based on extended incremental non-negative matrix factorization," *Neurocomputing*, vol. 136, pp. 327–336, Jul. 2014.

[17] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2003, pp. 265–272.

[18] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Manifold regularized discriminative nonnegative matrix factorization with fast gradient descent," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 2030–2048, Jul. 2011.

[19] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, "Deformed graph Laplacian for semisupervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2261–2274, Oct. 2015.

[20] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[21] J. Kim, R. Monteiro, and H. Park, "Group sparsity in nonnegative matrix factorization," in *Proc. SIAM Int. Conf. Data Mining*, 2012, pp. 851–862.

[22] Y. Qian, S. Jia, J. Zhou, and A. Kelly, "Hyperspectral unmixing via $L_{1/2}$ sparsity-constrained nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4282–4297, Nov. 2011.

[23] Y. Chen, J. Zhang, D. Cai, W. Liu, and X. He, "Nonnegative local coordinate factorization for image representation," *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 969–979, Mar. 2013.

[24] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proc. SIAM Int. Conf. Data Mining*, vol. 13. 2013, pp. 252–260.

[25] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proc. SDM*, vol. 5. 2005, pp. 606–610.

[26] F. Liu, T. Zhou, K. Fu, I. Gu, and J. Yang, "Robust visual tracking via inverse nonnegative matrix factorization," in *Proc. Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 1491–1495.

[27] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust ℓ_1 tracker using accelerated proximal gradient approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1830–1837.

[28] T. Zhang *et al.*, "Structural sparse tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 150–158.

[29] B. Ma, J. Shen, Y. Liu, H. Hu, L. Shao, and X. Li, "Visual tracking using strong classifier and structural local sparse descriptors," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1818–1828, Oct. 2015.

[30] 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.

[31] X. Jia, H. Lu, and M. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1822–1829.

[32] F. Li, H. Lu, D. Wang, Y. Wu, and K. Zhang, "Dual group structured tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1697–1708, Sep. 2016.

[33] 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.

[34] 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.

[35] D. Wang, H. Lu, and M. Yang, "Robust visual tracking via least soft-threshold squares," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 9, pp. 1709–1721, Sep. 2016.

[36] L. Ma, X. Zhang, W. Hu, J. Xing, J. Lu, and J. Zhou, "Local subspace collaborative tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4301–4309.

[37] H. Zhang, S. Hu, X. Zhang, and L. Luo, "Visual tracking via constrained incremental non-negative matrix factorization," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1350–1353, Sep. 2015.

[38] 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.

[39] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2009, pp. 2223–2231.

[40] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 3360–3367.

[41] B. Ma, H. Hu, J. Shen, Y. Zhang, and F. Porikli, "Linearization to nonlinear learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4400–4407.

[42] D. Cai, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proc. 12th ACM SIGKDD, Int. Conf. Knowl. Disc. Data Min.*, 2006, pp. 126–135.

[43] 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.

[44] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2001, pp. 556–562.

[45] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[46] B. Cao, D. Shen, J. Sun, X. Wang, Q. Yang, and Z. Chen, "Detect and track latent factors with online nonnegative matrix factorization," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2007, pp. 2689–2694.

[47] S. S. Bucak and B. Gunsel, "Incremental subspace learning via non-negative matrix factorization," *Pattern Recognit.*, vol. 42, no. 5, pp. 788–797, 2009.

[48] Z. Yu, Y. Liu, B. Li, S. Pang, and C. Jia, "Incremental graph regulated nonnegative matrix factorization for face recognition," *J. Appl. Math.*, vol. 2014, no. 11, pp. 1–10, 2014.

[49] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

[50] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 263–270.

[51] 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.

[52] J. Yoon, M. Yang, and K. Yoon, "Interacting multiview tracker," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 5, pp. 903–917, May 2016.

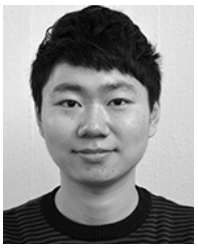
[53] H. Galoogahi, T. Sim, and S. Lucey, "Correlation filters with limited boundaries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4630–4638.

[54] M. Danelljan, F. Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1090–1097.

Fanghui Liu received the B.S. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2014. He is currently pursuing the Ph.D. degree with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, under the supervision of Prof. J. Yang. His research areas mainly include computer vision and machine learning with respect to visual tracking, kernel learning, and Bayesian learning.

Tao Zhou received the M.S. degree in computer application technology from Jiangnan University in 2012 and the Ph.D. degree in pattern recognition and intelligent system from the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, in 2016. His current research interests include object detection, visual tracking, and machine learning.

Chen Gong received dual Ph.D. degrees from Shanghai Jiao Tong University, China, and The University of Technology Sydney, Australia, in 2016, under the supervision of Prof. J. Yang and Prof. D. Tao, respectively. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. He has authored over 30 technical papers in prominent journals and conferences, such as the IEEE T-NNLS, the IEEE T-IP, the IEEE T-CYB, CVPR, AAAI, and IJCAI. His current research interests include machine learning, data mining, and learning-based vision problems.



Keren Fu received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2011, and the dual Ph.D. degrees from Shanghai Jiao Tong University, Shanghai, China, and the Chalmers University of Technology, Gothenburg, Sweden, in 2016, under the joint supervision of Prof. J. Yang and Prof. I. Yu-Hua Gu. He is currently with the College of Computer Science, Sichuan University. His current research interests include visual computing, saliency analysis, and machine learning.

Li Bai received the B.Sc. and M.Sc. degrees in mathematics and the Ph.D. degree in computer science from the University of Nottingham in 1994, where she is currently with the School of Computer Science. She is an expert in pattern recognition, computer vision, and medical imaging, with research grants and numerous research papers published in these areas.

Jie Yang received the Ph.D. degree from the Department of Computer Science, Hamburg University, Germany, in 1994. He is currently a Professor with the Institute of Image Processing and Pattern recognition, Shanghai Jiao Tong University, China. He has led many research projects, such as the National Science Foundation and the 863 National High Tech. Plan, had one book published in Germany, and authored over 300 journal papers. His major research interests are object detection and recognition, data fusion and data mining, and medical image processing.