Visual Tracking via Nonnegative Multiple Coding

Fanghui Liu, Chen Gong, Member, IEEE, Tao Zhou, Keren Fu, Xiangjian He, Senior Member, IEEE, and Jie Yang

Abstract—It has been extensively observed that an accurate appearance model is critical to achieving satisfactory performance for robust object tracking. Most existing top-ranked methods rely on linear representation over a single dictionary, which brings about improper understanding on the target appearance. To address this problem, in this paper, we propose a novel appearance model named as “Nonnegative Multiple Coding” (NMC) to accurately represent a target. First, a series of local dictionaries are created with different pre-defined numbers of nearest neighbors, and then the contributions of these dictionaries are automatically learned. As a result, this ensemble of dictionaries can comprehensively exploit the appearance information carried by all the constituted dictionaries. Second, the existing methods explicitly impose the nonnegative constraint to coefficient vectors, but in the proposed model, we directly deploy an efficient $\ell_2$ norm regularization to achieve the similar nonnegative purpose with theoretical guarantees. Moreover, an efficient occlusion detection scheme is designed to alleviate tracking drifts, and it investigates whether negative templates are selected to represent the severely occluded target. Experimental results on two benchmarks demonstrate that our NMC tracker is able to achieve superior performance to state-of-the-art methods.

Index Terms—approximated Locality-constrained Linear Coding, nonnegative constraint, occlusion detection, visual tracking

I. INTRODUCTION

Visual tracking aims to track the interested object in a video sequence given its precise location at the first frame. It plays a significant role in computer vision with broadly applications, such as video surveillance, human interaction, and vehicle navigation [1], [2]. Although a large number of approaches such as those in [3], [4], [5] have been developed, it is still difficult to find a lasting solution to achieve a robust tracking performance due to its intrinsic factors (e.g. fast motion, rotation in-plane or out-of-plane, and shape deformation) and extrinsic factors (e.g. partial occlusions, illumination variation, and background clutter).

In visual tracking, satisfactory performance depends on the accurate appearance modeling for the target. According to the adopted appearance model, the tracking methods can be categorized into two classes: generative trackers [6], [7] and discriminative trackers [8], [9], [10]. The generative methods attempt to find the most similar candidate region to the target by the minimal reconstruction error; whereas discriminative methods cast the tracking problem as a binary classification problem [11], [12] to separate the foreground target from the background. Our method belongs to the former one, and in the next we will briefly review the typical generative trackers.

Extensive research in generative methods demonstrate that sparse representation has been successfully used in visual tracking. The rationale of sparse representation based methods is that the target can be sparsely represented by the atoms in a dictionary $T$ (e.g. a target template set in visual tracking area) with a sparse coefficient vector. According to the adopted representation scheme, these trackers can be grouped into global template representation [13], [14], [15], local sparse model [16], [17], [18], joint sparse appearance model [19], [20], sparse collaborative model [21], and structured sparse model [22]. These methods learn the target representation from different cues and thus show promising performance against various challenging factors. To be specific, in a local sparse model [16], [17], the local patches within a possible target candidate are sparsely represented by the patches in a dictionary. The joint sparse appearance model [19], [20] exploits the intrinsic relationship among particles to represent the target jointly. The sparse collaborative model [21] combines the advantages of generative and discriminative methods, and thus is able to exploit both holistic templates and local representations to describe the target. Furthermore, the structural sparse appearance model [22] not only exploits the intrinsic relationship among target candidates through sparse representations, but also retains the spatial structure among the local patches within every target candidate.

The above sparse representation based approaches are often used to model the appearance of object through a linear representation way, in which the mapping relationship between the original sample space and coding space has been largely ignored. When the dictionary $T$ is used to represent the target, the similarity between two samples should be reflected in both the corresponding dictionaries and coefficient vectors. To be specific, Local Coordinate Coding (LCC) [23] is proposed to investigate the relationship among data points. It uses a set of anchor points to create a dictionary, and each data point is linearly represented by only a few anchor points. Locality-constrained Linear Coding (LLC) [24] investigates...
the similarity between data points and their corresponding dictionaries. These coding methods have been successfully incorporated into a visual tracking framework [25], [26], [27]. In [26], the authors apply the histograms of sparse coefficients encoded by the LLC algorithm and a local optimal searching scheme for object tracking. Unlike other works, LLC in [26] is employed for state search instead of appearance modeling. The sampling procedure is formulated as an optimization problem, and thus it can be efficiently solved on a convex hull.

However, there are two main drawbacks of these sparse coding based trackers. First, these methods simply use a fixed number of \(k\)-nearest neighbors to construct the local dictionary. This single dictionary cannot accurately and adequately capture the appearance of the target due to the insufficient emphasis on the target appearance information. As a result, they cannot tackle appearance variations of the target, which leads to tracking drifts easily. Furthermore, it is usually difficult to determine the number of neighbors \(k\) in advance, and an improper choice of \(k\) may degrade the tracking performance. Second, there is no nonnegative constraint on the coefficient vectors, where such constraint, as we shown in this paper, is essential for better tracking performance. Fig. 1 shows two tracking results reconstructed by two sets of selected positive templates in the LLC method with/without the nonnegative constraint. Although both the good tracking result (in a green bounding box) and the bad result (in a red bounding box) are represented by the positive template set with the indices ranging from \(1\) to \(60\), the bad result suffers severe drifts and contains much background information. This is because the bad tracking result is reconstructed by some positive template images with negative coefficients (e.g. \#6, \#32, and \#47), whereas all the coefficients for the good result are nonnegative. Therefore, we argue that positive templates with negative coefficients would lead to an unreliable tracking result.

Based on the above observations, this paper develops a robust Nonnegative Multiple Coding (NMC) tracker by exploiting an ensemble of multiple dictionaries and the nonnegative constraint to accurately characterize the target appearance. The contributions of this paper are summarized as follows.

1) A series of local dictionaries are built with regards to different pre-defined numbers of nearest neighbors, and their related weights are automatically learned in our NMC model. Thereby, the obtained coefficient vectors and the ensemble of multiple dictionaries can be learned in a uniform framework.

2) We demonstrate through both theory and experiments that the incorporation of \(\ell_2\) norm regularization term to our NMC model is able to make the obtained coefficient vector nonnegative. It achieves the similar effect with the exact nonnegative constraint with the provable guarantee of the lower bounded regularization parameter.

3) The occlusion detection criterion is proposed to mitigate drifting problem which investigates whether the negative templates (i.e. background) are used to represent the target.

Fig. 1. Illustration of two tracking results in the sequence Singer1. The good tracking result (in green) is represented by eight positive template images (#53, #54, #55, #56, #57, #58, #59 and #60) with nonnegative coefficients. The bad tracking result (in red) is represented by eight selected positive template images (#6, #7, #23, #32, #45, #47, #54 and #55) with both positive and negative coefficients as illustrated by the figure in the top corner.

This paper is the extended version of our previous work [29]. The tracker described here differs from [29] in several aspects. Firstly, to update the positive template set, PCA-based vectors and additional trivial templates are used to represent the target by the approximated LLC method. The updating scheme is accomplished by gradually absorbing the latest positive samples and eliminating out-of-date samples. Secondly, we theoretically prove that the introduced \(\ell_2\) norm regularization is able to achieve nonnegative effects if its regularization parameter is larger than a lower bound. Thirdly, Histogram of Gradient (HOG) feature is introduced into our tracking framework to improve the performance. Lastly, we present more experimental results on two benchmark datasets, and investigate the parametric sensitivity and computational complexity analysis of the proposed tracker.

The remainder of the paper is organized as follows. Section II explains the details of the proposed Nonnegative Multiple Coding model. Section III introduces the proposed NMC tracker. Section IV shows the evaluation results of the proposed tracker with other state-of-the-art methods on two popular benchmarks. Finally, conclusion is given in Section V.

II. NONNEGATIVE MULTIPLE CODING MODEL

In this section, we first provide an overview of the LLC method that will be used in our NMC model, and then detail the employed nonnegative constraint and dictionary ensemble. Particularly, the nonnegative constraint is replaced by the \(\ell_2\) norm regularization term to achieve the similar nonnegative purpose with theoretical guarantees.

A. Review of LLC and Approximated LLC

The idea of the LLC method [24] implies that if two data points are close to each other in the original space, they should be represented by two similar dictionaries, and thus have similar encoding coefficients. To improve its efficiency, LLC applies a locality constraint to select similar bases, and then it learns a linear combination of these bases in the local...
dictionary $B$ with a coefficient vector $c_i$ to represent a data point $y_i \in \mathbb{R}^M$ as follows:

$$\min_{c_i} \sum_{i=1}^{N} \|y_i - Bc_i\|^2_2 + \lambda \|d_i \odot c_i\|^2_2,$$  

subject to $1^T c_i = 1, \forall i = 1, 2, \cdots, N,$

where $N$ is the number of data points, and the constraint $1^T c_i = 1$ ensures shift-invariant where 1 denotes the all-one vector. The distance $d_i = \exp(-\frac{\text{dist}(y_i, B)}{\sigma})$ with $\sigma$ being a scale factor parameter is defined by $\text{dist}(y_i, B) = \text{dist}(y_i, b_1), \text{dist}(y_i, b_2), \cdots, \text{dist}(y_i, B_N)^T,$ where the distance metric $\text{dist}(y_i, b_j)$ represents the Euclidean distance between $y_i$ and $b_j$.

The approximated LLC method [24], as a simplified version of LLC, takes local sparsity into consideration. To be specific, it simply selects $k$ nearest neighbors of $y_i$ as the local dictionary $B_i$ instead of using the metric $\text{dist}(y_i, B)$. In this way, the approximated LLC method not only obtains local similarity but also achieves sparsity due to the judicious selection of $k$ nearest neighbors. Hence, a test sample $y_i$ can be sparsely represented by the linear combination of several local base vectors in the local dictionary $B_i$ with the coefficient vector $c_i$ as follows:

$$\min_{c_i} \|y_i - Bc_i\|^2_2 \text{ subject to } 1^T c_i = 1, \forall i = 1, 2, \cdots, N,$$

from which the closed-form solution is $c_i = [(B_i^T - 1y_i^T)(B_i - y_i1^T)]^{-1}1.$

### B. Nonnegative Constraint on Approximated LLC

Note that, in nonnegative matrix factorization (NMF) [30], a nonnegative data point is modeled as the nonnegative linear combination of a set of nonnegative bases, which could implicitly grab the structure information contained within data distribution. Therefore it is straightforward to require the obtained coefficients in our NMC model to be nonnegative. Specifically, the illustration in the introduction also demonstrates the superiority of adopting the nonnegative coefficients (see Fig. 1). In this paper, higher priority is given to adopt the nonnegative coefficients based on Eq. (2), which is written as:

$$\min_{c_i} \|y_i - Bc_i\|^2_2 \text{ subject to } 1^T c_i = 1, c_i \geq 0, \forall i = 1, 2, \cdots, N.$$  

Several optimization algorithms for such quadratic programming problem with linear constraint have been proposed, such as interior point method, accelerated proximal gradient method (APG) [15], and alternating direction method of multipliers (ADMM) [32]. However, Eq. (3) contains a nonnegative inequality constraint, so the above methods cannot yield a closed-form solution here. Consequently, above iterative algorithms are not very efficient for visual tracking. To tackle this problem, we introduce a conventional $\ell_2$ norm regularization term to replace the nonnegative constraint to achieve the similar nonnegative effect. The rationality of such replacement will be theoretically demonstrated in Section II-D with provable guarantees. Herein, Eq. (3) is reformulated as:

$$\min_{c_i} \|y_i - Bc_i\|^2_2 + \lambda \|c_i\|^2_2 \text{ subject to } 1^T c_i = 1,$$

where $\lambda$ is the tuning parameter. Let $\gamma$ be the Lagrange multiplier for the shift-invariant constraint $1^T c_i = 1$, then the related Lagrange function can be formulated as:

$$\mathcal{L} = c_i^T (y_i1^T - B_i) (y_i1^T - B_i)c_i + \lambda c_i + \gamma (1 - 1^T c_i).$$

The partial derivative of $\mathcal{L}$ with respect to $c_i$ is:

$$\frac{\partial \mathcal{L}}{\partial c_i} = 2(y_i1^T - B_i)^T (y_i1^T - B_i)c_i + 2\lambda c_i - \gamma 1.$$  

By using the Karush-Kuhn-Tucker condition, the analytic solution can be obtained as:

$$c_i = \frac{\gamma}{2} (B_i^T - 1y_i^T)(B_i - y_i1^T) + \lambda 1.$$  

Specifically, $\gamma$ can be further obtained by the shift-invariant constraint. In this case, we can directly obtain the closed-form solution $c_i$ without employing or designing any inefficient iterative method for this particular purpose.

### C. Ensemble of Multiple Local Dictionaries

With regards to the existing problems, the single dictionary cannot faithfully represent the appearance of the target. In this paper, we mainly focus on designing an ensemble of dictionaries to sufficiently capture the appearance information revealed by the target. The local dictionaries $(B_1, B_2, \cdots, B_m)$ for every sample $y_i$ with different numbers of nearest neighbors are constructed by:

$$\begin{align*}
\min_{c_i} \|y_i - B_i^1c_i\|^2_2 \text{ subject to } 1^T c_i = 1, c_i \geq 0, \\
\min_{c_i} \|y_i - B_i^2c_i\|^2_2 \text{ subject to } 1^T c_i = 1, c_i \geq 0, \\
\vdots \\
\min_{c_i} \|y_i - B_i^mc_i\|^2_2 \text{ subject to } 1^T c_i = 1, c_i \geq 0,
\end{align*}$$

where $B_i^j \in \mathbb{R}^{M \times k_j}$ is established from the sample $y_i$’s $k_j$ nearest neighbors, and the associated coefficient vector is $c_i^j \in \mathbb{R}^{k_j}$. After obtaining these coefficient vectors $(c_i^1, c_i^2, \cdots, c_i^m)$, the next step is to accurately integrate these vectors into a final coefficient vector. By introducing a weight vector $w = [w_1, w_2, \cdots, w_m]^T$, which satisfies $\sum_{s=1}^{m} w_s = 1$ and $w_s \geq 0$ for $s = 1, 2, \cdots, m$, the weight vector $w$ can be obtained by optimizing the following objective function:

$$\min_{w} \left\| y_i - \sum_{j=1}^{m} w_j(B_i^j c_i^j) \right\|^2_2$$  

subject to $1^T w = 1, \; w_j \geq 0 \; \forall j = 1, 2, \cdots, m$.

Furthermore, by defining $D = \left[ B_1^1 c_1^1, B_1^2 c_1^2, \cdots, B_m^m c_m^m \right] \in \mathbb{R}^{M \times m}$, we formulate Eq. (9) as:

$$\min_{w} \left\| y_i - Dw \right\|^2_2 \text{ subject to } 1^T w = 1, w_j \geq 0 \; \forall j.$$  

This objective function shares the same expression as the approximate LLC problem with the nonnegative constraint in
Eq. (3). Hence, an $\ell_2$ norm regularization term can be added to Eq. (10) to achieve a similar nonnegative effect, that is:

$$\min_w |y_i - Dw|_2^2 + \beta |w|_2^2 \quad s.t. \quad 1^T w = 1, \quad \forall j,$$

where $\beta$ is the regularization parameter. Similar to the solution of Eq. (4), the optimal $w$ of Eq. (11) is $w = [(D^T - 1y^T_j)(D - y_j 1^T)] + \beta I^{-1} 1$.

D. Theoretical Analysis on $\ell_2$ Norm Regularization

As mentioned in Section II-B, the adopted $\ell_2$ norm regularization shows the similar effect with the hard nonnegative constraint. In this section, we theoretically demonstrate the requirements for the nonnegative coefficient vector with theoretical guarantees. As both Eqs. (4) and (11) incorporate the $\ell_2$ norm to generate nonnegative coefficient vectors $c_i$ and $w$ respectively, here we study their general formation, namely:

$$\min_x \|y - Bx\|_2^2 + \alpha \|x\|_2^2 \quad s.t. \quad 1^T x = 1,$$

where $y$ is a data point, $B \in \mathbb{R}^{M \times k}$ is the base matrix, $x$ is the coefficient vector, and $\alpha$ is the tuning parameter that represents $\lambda$ and $\beta$ in Eqs. (4) and (11) respectively. Likewise, the optimizer of Eq. (12) is:

$$x = [(B^T - 1y^T)(B - y1^T)] + \alpha I^{-1} 1.$$

By defining $A = B - y1^T$ and $F = (A^T A + \alpha I) \in \mathbb{R}^{k \times k}$, the solution in Eq. (13) can be reformulated as:

$$x = (A^T A + \alpha I)^{-1} 1 = F^{-1} 1.$$

Specifically, when $\alpha$ is extremely large, the optimization problem in Eq. (12) approaches the following problem:

$$\min_x \alpha \|x\|_2^2 \quad s.t. \quad 1^T x = 1.$$

It is obvious that the minimizer of Eq. (15) is $x_1 = x_2 = \cdots = x_k = \frac{1}{\sqrt{\alpha}}$. This formula implies that when the tuning parameter $\alpha$ approaches to infinity, the elements in $x$ tend to be nonnegative and equivalent. It means that, there must exist a lower bound of $\alpha$ that ensures the coefficient vector $x$ to be nonnegative. Next the following theorem will demonstrate how to compute this bound.

**Theorem 1.** Let $A^T A$ be a $k \times k$ matrix, then the coefficient vector $x = (A^T A + \alpha I)^{-1} 1$ in Eq. (13) is nonnegative if $\alpha$ satisfies:

$$\alpha \geq (k + 1)\|A^T A - 2\Gamma\|_\infty + k\|A^T A\|_\infty,$$

where $\Gamma$ is a diagonal matrix with its $i$th diagonal element $(A^T A)_{ii}$.

**Proof.** The proof is presented in Appendix A.

Theorem 1 theoretically demonstrates that replacing the nonnegative constraint by $\ell_2$ norm regularization term is reasonable and effective if $\alpha$ is lower bounded. Therefore, in all experiments we simply set $\alpha$ (i.e. $\lambda$ and $\beta$) to be its lower bound rather than empirically choosing this parameter.

III. NONNEGATIVE MULTIPLE CODING TRACKER

Visual tracking problem is usually formulated as a particle filter framework [35, 31]. The implicit rationale behind particle filter is to estimate the posterior distribution $p(x_t|y_{1:t})$, which can be approximately computed by a finite set of randomly sampled particles. Given some observed image patches at the $t$th frame $y_{1:t} = \{y_1, y_2, \ldots, y_t\}$, the state of the target $x_t$ can be estimated recursively by the following two stages:

Prediction: $p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1}) dx_{t-1},$

Updating: $p(x_t|y_{1:t-1}) \propto p(y_t|x_t)p(x_{t-1}|y_{1:t-1}).$

By defining the target state $x_t = [l_x, l_y, \theta, s, \alpha, \phi]^T$, where $l_x$, $l_y$, $\theta$, $s$, $\alpha$, and $\phi$ denote the translations in the direction of $x$ and $y$, rotation angle, scale, aspect ratio, and skew respectively, we apply the affine transformation with the above six parameters to model the target motion. The motion model $p(x_t|x_{t-1})$ represents the state transition of the target between two consecutive frames. The observation model $p(y_t|x_t)$ denotes the likelihood of the observation $y_t$ at state $x_t$. In our method, the observation model is formulated by our NMC model, which reflects the similarity between a candidate sample and the target templates. The optimal state at the $t$th frame is obtained by maximizing the posterior probability:

$$x_t^* = \arg \max_{x_t} p(y_t|x_t)p(x_t|x_{t-1}), \quad i = 1, 2, \ldots, N,$$

where $x_t^*$ is the $i$th sample of the state $x_t$, and $y_t^*$ is a candidate sample predicted by $x_t^*$.

Here we briefly introduce how to obtain the templates module in the beginning frames of a video sequence. To better capitalize on the distinction between the foreground and background, plenty of negative templates are collected apart from positive templates. In other words, the template set $T$ contains positive and negative template sets, defined as $T^{pos} = \{T_1, T_2, \ldots, T_p\}$ and $T^{neg} = \{T_{p+1}, T_{p+2}, \ldots, T_{p+n}\}$, where $p$ and $n$ denote the numbers of positive and negative templates respectively. The base dictionary $B_i = [B_{i1}^2, B_{i2}^{T2}, \ldots, B_{ik}^{Tk}] \in \mathbb{R}^{M \times k}$ is built by selecting $k$ nearest neighbors of a candidate sample $y_i$ from the template set $T$.

A. Observation Model

The proposed NMC model outputs a coefficient vector $c$ to calculate the reconstruction error $\|y - Tc\|_2^2$ that represents the similarity between a candidate and the target. A small reconstruction error indicates that the test sample $y$ is similar to the target with a high probability, and vice versa. The detailed implementation issues are explained as follows.

Under the aforementioned results, the local dictionary $B_{ij}^2 \in \mathbb{R}^{M \times k_j}$ is established by $k_j$ ($j = 1, 2, \ldots, m$) nearest neighbors of a candidate sample $y_i$, which are selected from the template set $T$. To record the selected templates in $B_{ij}^2$, we define a selection vector $d_{ij}^2 \in \mathbb{R}^{p+n}$ of which the dimensionality equals to the number of templates. Therefore $c_{ij}^2 \in \mathbb{R}^{k_j}$ in Eq. (8) is plugged into $d_{ij}^2$ where the locations...
of these \(k_j\) elements correspond to the indices of the selected templates in \(T\).

After that, once the weight vector \(w\) and the coefficient vector \(d_i\) are successfully obtained, the final coefficient vector \(d_i \in \mathbb{R}^{p+n}\) is a linear combination of \(d_i^j (j = 1, 2, \ldots, m)\) weighted by the elements in the vector \(w\). The final coefficient vector \(d_i\) is solved by Algorithm 1.

Algorithm 1: Algorithm for the final coefficient vector \(d_i\)

| Input: a candidate \(y_i\), local dictionaries: \(B_i^1, B_i^2, \ldots, B_i^n\). |
| Output: the final coefficient vector \(d_i\) related to \(y_i\). |

1. Set: iteration number \(\eta\).
2. for \(j = 1 : m\) do
   3. Compute \(c_i^j\) by Eq. (4).
   4. Obtain the selection vector \(d_i^j\);
5. end
6. Solve the weight vector \(w\) by Eq. (11).
7. Obtain the final coefficient vector \(d_i = \sum_{j=1}^{m} w_j d_i^j\).

In our method, \(d_i \in \mathbb{R}^{p+n}\) is divided into two parts: \(d_i = [d_i^{pos}; d_i^{neg}]\) with respect to \(T^{pos}\) and \(T^{neg}\) respectively, where \(d_i^{pos} \in \mathbb{R}^{p}\) and \(d_i^{neg} \in \mathbb{R}^{n}\). Then the observation model for the candidate \(y_i\) is formulated as:

\[
p(y_i|x_i) \propto \exp(-\psi(e_i^{pos} - e_i^{neg})),
\]

where \(e_i^{pos} = \|y_i - T^{pos}d_i^{pos}\|_2^2\) is the reconstruction error between the candidate \(y_i\) and the positive template set \(T^{pos}\). Likewise, \(e_i^{neg} = \|y_i - T^{neg}d_i^{neg}\|_2^2\) is the reconstruction error between the candidate \(y_i\) and the negative template set \(T^{neg}\). The parameter \(\psi\) is a normalization factor fixed to 2.5 in our experiments.

B. Occlusion Detection Scheme

In tracking process, severe occlusion is one of the most significant issues which lead to tracking drifts or failures. Therefore, designing a suitable occlusion detection module is important to ensure accurate appearance modeling and avoid unexpected tracking drifts.

Generally, the target without occlusions will be entirely represented by positive templates. If the target suffers from severe occlusions, negative templates will be selected to represent the target with a high probability. Such scenario motivates us to propose an efficient approach to detect severe occlusions. The occlusion detection criterion is mainly based on whether negative templates are selected to represent the tracking target. If several negative templates are used to reconstruct the target, the target probably suffers from serious occlusions, and vice versa. In the proposed NMC tracker, if more than one negative template (the number of selected negative templates is denoted as \(\text{Num}(\neg e)\)) are used to represent the target, the target in the current frame is regarded as “occluded”. In this case, the model updating should be strictly prohibited, otherwise the appearance model would be contaminated by inaccurate positive templates.

1Note that the time index \(t\) is omitted for simplicity.

C. Template Updating Scheme

Usually, tracking with fixed templates is prone to fail in dynamic scenes due to the inevitable appearance changes. However, constantly updating the template set easily leads to error accumulation and tracking drifts. According to this, we propose an incremental updating scheme for the positive template set based on the approximated LLC method. The flowchart of this updating scheme is shown in Fig. 3. Similar to [37], the estimated target can be modeled by a linear combination of PCA base vectors and additional trivial templates. In the proposed NMC tracker, the local dictionary \(B\) is composed of \(k\) nearest neighbors of the optimal candidate \(y^*\) from the codebook as shown in Fig. 3 which arrives at:

\[
y^* = Bc = UX + e = \begin{bmatrix} U & I \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}_{\triangleq B}
\]

where \(y^*\) is the current optimal candidate sample, \(U\) is the matrix composed of PCA base vectors in the local dictionary \(B\), \(x\) is the corresponding PCA coefficient vector, and \(e\) is assumed to be sparse noise. The coefficient vector \(c\) can be obtained by Eq. (4) in the proposed NMC model. In
this formula, we use the reconstructed sample $y_{rs} = Ux$ for positive template set updating rather than the current optimal candidate $y^*$. Compared to directly using the current tracking result $y^*$ with the reconstruction error smaller than a pre-defined threshold as in [29], in the proposed template updating scheme, the appearance model can effectively avoid being contaminated when the optimal candidate $y^*$ is slightly occluded. For example, in Fig. 3 the reconstructed sample $y_{rs}$ is much more accurate than the optimal candidate $y^*$.

Furthermore, to save the storage space and reduce the computational cost, the “First-in and First-out” procedure is used to maintain the number of positive templates. In this way, the newly reconstructed sample $y_{rs}$ is added to the template set, while the oldest positive template is thrown away.

If the occlusion is not detected, the positive and negative template sets are normally updated every 5 frames by our template updating scheme. Once an occlusion is detected, the positive template set should stop updating, while the negative template set is still regularly updated for every 5 frames.

Finally, the flowchart of the NMC tracker is summarized in Algorithm 2.

**Algorithm 2:** Algorithm for the NMC tracker

1. **Initialization**: Extract templates $T$ in the first 5 frames.
2. **for** $t = 6$ to the end of the sequence **do**
3.   **for** $i = 1 : N$ **do**
4.     **Construct** local dictionaries $B_1^i, B_2^i, ..., B_m^i$ for each candidate $y_i$;
5.     **Obtain** $d_i$ by Algorithm 1;
6.     **Calculate** the likelihood of $y_i$ by Eq. (19) in the observation model;
7. **end**
8. **Obtain** the optimal candidate $y^*$ and the reconstructed sample $y_{rs}$ by Eq. (20);
9. **Update**: **for** each 5 frames **do**
10.   **Update** the negative template set $T_{neg}^n$;
11.   **if** $Num(neg_{rs}) \leq 1$ **then**
12.     **Update** the positive template set;
13. **end**
14. **end**

**IV. EXPERIMENTS**

This section evaluates the proposed NMC tracker on two benchmarks including Princeton Tracking Benchmark (PTB) [38] and Object Tracking Benchmark (OTB) [39]. We perform an extensive comparison of our NMC tracker with the existing state-of-the-art trackers on two benchmarks. The compared methods include SST [22], KCF [10], LNLT [25], and IMT [40]. In addition, the results on parameter sensitivity and computational complexity analyses are also provided.

**Setup:** The proposed NMC 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: each sample (i.e. the patch of image) was normalized to 32 × 32 pixels ($M = 1024$); the number of positive templates and negative templates $p$ and $n$ were set to 100 and 150 respectively; three local dictionaries (i.e. $m = 3$) were selected with three different numbers of nearest neighbors $k_1 = 5$, $k_2 = 8$, and $k_3 = 10$; the tuning parameters $\alpha$ and $\beta$ for $\ell_2$ norm regularization were chosen by Eq. (16). In the template updating scheme, the number of PCA base vectors was set to 16, and eight nearest neighbors from the codebook were selected to represent the optimal candidate $y^*$.

**Evaluation metrics:** In PTB and OTB, two fundamental metrics namely mean center location error (CLE) and the Pascal VOC Overlap Ratio (VOR) [41] are used to evaluate the tracking performances of all the compared methods. CLE is defined as the pixel distance between the centroid of the tracking result and the ground truth, in which small CLE value indicates a precise tracking result. The overlap ratio measures the overlapping rate between the bounding box yielded by a tracker and the ground truth box. It is defined as $o = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)}$, where $R_T$ and $R_G$ denote the area of bounding boxes of the tracker and ground truth, respectively.

**A. Results on PTB**

The Princeton Tracking Benchmark contains 95 video clips collected by both RGB and depth sensors from a standard Microsoft Kinect 1.0. These sequences are grouped into the following aspects: target type (human, animal and rigid), target size (large and small), movement (slow and fast), presence of occlusion, and motion type (passive and active). For fair comparison, the proposed tracker is only compared with RGB competitors without considering the depth information. The nine colored trackers includes KCF [10], Struck [42], VTD [43], CT [44], TLD [45], MIL [46], SemiB [47], RGBdet [38] and OF [38].

The average overlap ratio and ranking results of these methods are shown in Tab. 1. One can see that our method ranks first, and it is followed by the KCF tracker. From the target type aspect, the proposed NMC tracker performs better on human and animal target than KCF method due to an accurate appearance model. Besides, the proposed tracker achieves promising performance in terms of target size, fast movement, occlusion, and active motion when compared with other methods.

**B. Results on OTB**

Object Tracking Benchmark includes 29 trackers evaluated on 51 sequences with accurate bounding box annotations. Two evaluation criteria are used in OTB: precision plot and success plot. Both two criteria show the percentage of successfully tracked frames based on the two metrics CLE and VOR mentioned above under different thresholds. In precision plot, the tracking result in a frame is considered successful if the CLE falls below a pre-defined threshold. In success plot, the target in a frame is declared to be successfully tracked if the current VOR exceeds a certain threshold. Two ranking metrics are used to evaluate all compared trackers: one is the Area Under the Curve (AUC) metric at the success plot, and the other is the precision score at threshold of 20 pixels for the precision plot.
TABLE I
RESULTS ON THE PRINCETON TRACKING BENCHMARK: SUCCESS RATES AND RANKINGS (IN PARENTHESES) FOR DIFFERENT SEQUENCE CATEGORIZATIONS. THE BEST TWO RESULTS ARE HIGHLIGHTED BY GREEN AND RED, RESPECTIVELY.

<table>
<thead>
<tr>
<th>Method</th>
<th>Avg. Rank</th>
<th>target type</th>
<th>target size</th>
<th>movement</th>
<th>occlusion</th>
<th>motion type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours(HOG)</td>
<td>1.0(1)</td>
<td>0.4(1)</td>
<td>0.52(3)</td>
<td>0.35(2)</td>
<td>0.3(1)</td>
<td>0.4(1)</td>
</tr>
<tr>
<td>KCF</td>
<td>1.0(1)</td>
<td>0.4(1)</td>
<td>0.52(3)</td>
<td>0.35(2)</td>
<td>0.3(1)</td>
<td>0.4(1)</td>
</tr>
<tr>
<td>Struck</td>
<td>3.7(3)</td>
<td>0.3(1)</td>
<td>0.47(3)</td>
<td>0.53(1)</td>
<td>0.3(1)</td>
<td>0.4(1)</td>
</tr>
<tr>
<td>VTD</td>
<td>4.9(1)</td>
<td>0.1(1)</td>
<td>0.64(3)</td>
<td>0.7(1)</td>
<td>0.3(1)</td>
<td>0.4(1)</td>
</tr>
<tr>
<td>RGBNet</td>
<td>4.7(1)</td>
<td>0.2(1)</td>
<td>0.47(3)</td>
<td>0.35(2)</td>
<td>0.3(1)</td>
<td>0.4(1)</td>
</tr>
<tr>
<td>MIL</td>
<td>6.0(1)</td>
<td>0.3(1)</td>
<td>0.5(1)</td>
<td>0.35(2)</td>
<td>0.3(1)</td>
<td>0.4(1)</td>
</tr>
<tr>
<td>TLD</td>
<td>6.9(1)</td>
<td>0.2(1)</td>
<td>0.5(1)</td>
<td>0.35(2)</td>
<td>0.3(1)</td>
<td>0.4(1)</td>
</tr>
<tr>
<td>CT</td>
<td>7.0(1)</td>
<td>0.3(1)</td>
<td>0.5(1)</td>
<td>0.35(2)</td>
<td>0.3(1)</td>
<td>0.4(1)</td>
</tr>
<tr>
<td>SiamRbb</td>
<td>8.6(1)</td>
<td>0.2(1)</td>
<td>0.5(1)</td>
<td>0.35(2)</td>
<td>0.3(1)</td>
<td>0.4(1)</td>
</tr>
<tr>
<td>OF</td>
<td>10(1)</td>
<td>0.1(1)</td>
<td>0.5(1)</td>
<td>0.35(2)</td>
<td>0.3(1)</td>
<td>0.4(1)</td>
</tr>
</tbody>
</table>

Fig. 4. Success plot and precision plot. The performance score for each tracker is shown in the legend.

1) Overall performance: To show the overall performance of the proposed NMC tracker, we compare it with some other state-of-the-art methods on success plots and precision plots. The results in Fig. 4 show that the top 10 trackers on success plot are IMT [40], LNLT [25], SCM [21], SST [22], Struck [42], TLD [45], ASLA [16], CXT [48] and our two methods with HOG feature and raw grayscale feature, respectively. The proposed tracker with HOG feature ranks the first on success plot and precision plot. Besides, the NMC tracker with raw grayscale feature also achieves comparable performance with other state-of-the-art methods.

Apart from the above methods, a representative deep learning based tracker (DLT) [49] is also incorporated for comparisons and its performance is recorded in Tab. I. It can be observed that the NMC tracker performs better than DLT with 3.8% improvement on average VOR.

2) Attribute based performance analysis: The sequences in the OTB dataset are annotated by several challenging factors in visual tracking. To prove the superiority of the proposed algorithm, we plot the success rates of various methods on the subsets with eight main attributes (Occlusion, Motion Blur, Deformation, Illumination Variation, In-plane Rotation, Out-of-plane Rotation, Background Clutter, and Out of View). The results shown in Fig. 5 illustrate that the proposed NMC tracker with HOG feature ranks first or second on these eight attributes. Specifically, on Deformation attribute, our proposed method ranks the first and obtains 11% improvement than the second best tracker on success plot due to an accurate appearance model. Besides, the occlusion detection scheme prevents the proposed tracker from updating the positive templates when the target is heavily occluded. The satisfactory performance on Occlusion attribute verifies that the template updating scheme introduced in this paper is effective.

3) Qualitative analysis: Fig. 6 shows the qualitative results comparing three best baseline trackers on several representative frames. It can be observed that the proposed tracker is robust to the occlusions (Suv and Jogging), effectively grabs the appearance changes (Basketball, Freeman1, and David3), and resists on undesirable illumination variations (Tiger2) when compared with other baseline methods in these sequences. Overall, the proposed NMC tracker performs well on these challenging sequences, and it can be attributed to an accurate appearance model learned from an ensemble of multiple local dictionaries, and an effective model updating scheme to avoid degrading the appearance model.

C. Selection of Regularization Parameters

In this subsection, we firstly analyze the superiority of the adopted regularization term over the exact nonnegative constraint on the final tracking performance, and then investigate the influence of different regularization parameters to the tracking results.

Tab. I compares our NMC tracker with the “Nonnegative” method using the exact nonnegative constraint. Apart from high computational complexity, this setting also yields an inferior performance with 1.1% reduction on success plot than the proposed NMC tracker. In contrast, the introduced \( \ell_2 \) regularization term is able to effectively avoid over-fitting and make the coefficient vector more stable.

Besides, to quantitatively analyze the influence of the two regularization parameters \( \lambda \) and \( \beta \) to the final tracking performance, we provide the corresponding tracking results on OTB under different selections of \( \lambda \) and \( \beta \). Fig. 7 shows the influence of the tuning parameter \( \lambda \) with different values \( 0, 0.01, 0.1, 1, 10 \) on success plot and precision plot. We see that without any regularization term (i.e. \( \lambda = \beta = 0 \)), the success rate and precision are extremely low, and these indicate that the tracker is very fragile. When \( \lambda \) ranges from 0.01 to 10 (\( \beta \) is fixed to 0.1), the success rate and precision steadily increase to the summit at \( \lambda = 1 \) and then fall down to a low record. If \( \lambda \) is too small (e.g. \( \lambda \in [0.01, 0.1] \)), we see that the variation of this parameter plays a little role in determining the final tracking performance. If \( \lambda \) is too large, it will lead to over-fitting with a large reconstruction error.

The above experimental results suggest that the fixed values of \( \lambda \) and \( \beta \) will degrade the tracking performance. Therefore,
TABLE II
AVERAGE CENTER LOCATION ERRORS (CLE) AND AVERAGE VOC OVERLAP RATIO (VOR) ON OTB WITH 51 TASKS.

<table>
<thead>
<tr>
<th>Method</th>
<th>Ours(Gray)</th>
<th>DLT [49]</th>
<th>Nonnegative</th>
<th>M=24×24</th>
<th>M=36×36</th>
<th>k_1 = 5</th>
<th>k_2 = 8</th>
<th>k_3 = 10</th>
<th>K_1 = [3,5,8]</th>
<th>K_2 = [8,10,15]</th>
</tr>
</thead>
<tbody>
<tr>
<td>average CLE(pixels)</td>
<td>48.5</td>
<td>61.7</td>
<td>55.3</td>
<td>67.2</td>
<td>52.4</td>
<td>66.3</td>
<td>65.0</td>
<td>78.7</td>
<td>65.1</td>
<td>64.5</td>
</tr>
<tr>
<td>average VOR(%)</td>
<td>47.6%</td>
<td>43.8%</td>
<td>46.5%</td>
<td>44.2%</td>
<td>47.1%</td>
<td>43.4%</td>
<td>44.8%</td>
<td>42.7%</td>
<td>44.8%</td>
<td>43.4%</td>
</tr>
</tbody>
</table>

Fig. 5. The curves of success rates of various compared methods on eight main attributes.

Fig. 6. The comparison of our method with three best baseline trackers on the challenging frames. Subfigures from left to right, top to bottom: Tiger2, Bolt, and Shaking; Freeman4, Basketball, and Carscale; Trellis, David3, and Jogging; Freeman1, Suv, and Skiing.

in our method the parameter $\lambda$ is adaptively selected by Eq. (16), which brings about 2.2% improvement when compared with the manually tuned $\lambda = 1$ revealed by the success plot. Therefore the automatic way for setting $\lambda$ presented in Section II-D is reasonable and effective.

D. Key Component Validation

This subsection validates the effectiveness of several key components in our tracker.

1) Validation of occlusion detection scheme: The only difference between the “NoOcc” method and the proposed NMC tracker is that the “NoOcc” method does not leverage the aforementioned occlusion detection scheme. Fig. 8 shows that the average overlap rate of the “NoOcc” method decreases dramatically from 47.6% to 43.1%, and the average center location error significantly rises from 48.5 to 70.4. Consequently, we see that the developed occlusion detection scheme plays an important role in boosting the tracking performance.

Furthermore, we also change the range of the number of negative templates $\text{Num}(\text{neg})$ (from 0 to 6) to see its effect on
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2708424, IEEE Transactions on Multimedia

We can see that the proposed tracker with an ensemble of \( k \) nearest neighbors for dictionary construction are shown in Tab. II. A large \( k \) might incorporate dissimilar candidates to construct the local dictionary, which would cause error accumulation and degrade the tracking accuracy. The insufficient number of positive and negative templates (e.g., \( \mathcal{K}_2 = \{8, 10, 15\} \)) models the target’s appearance inaccurately, leading to an unreliable tracking performance. Accordingly, a suitable ensemble should take diverse \( k \)-nearest neighbors into consideration to fully exploit their individual advantages, such as \( \mathcal{K} = \{5, 8, 10\} \). Tab. III shows that the NMC tracker with the selection of \( \mathcal{K} = \{5, 8, 10\} \) achieves the highest record on average VOR among all the compared parametric settings, which suggests that the selected \( \mathcal{K} = \{5, 8, 10\} \) in this paper is effective.

3) Numbers of positive and negative templates: We investigate how the numbers of positive and negative templates sampled at the beginning of a video sequence affect the final tracking performance. The corresponding success rate and precision are shown in Fig. 9, in which “P200N300” denotes that the number of positive and negative templates are set to be 200 and 300, respectively. It can be observed that the insufficient number of positive templates (e.g., “P10N30”) would incur unsatisfactory performance with a low record, namely 41.1% on success rate and 57.4% on precision. This is because that eight templates are selected from only ten positive templates in this case, and thus the output coefficient vector by such setting does not achieve sparse effect.

By contrast, numerous positive templates (e.g., “P200N300”, “P50N100”) sampled at the beginning frames do not have significant effects on the final tracking performance. Specifically, we can see that the “P200N300” setting achieves 48.4% on average overlap rate, which is slightly higher than our method because much more information has been taken into consideration. However, because superabundant templates are involved in the “P200N300” setting, it arrives at 0.9fps (frame per second), which is twice slower than our NMC tracker.

4) Size of the image patch: In our experiments, the image patch is normalized to \( M = 32 \times 32 \). Here we analyze the influence of different feature dimensions (i.e., the size of an image patch) on the final tracking performance. Tab. III shows the results of two settings with different sizes, where the “\( M = 36 \times 36 \)” method leads to slight fluctuation on the average CLE and VOR. In addition, the performance of “\( M = 24 \times 24 \)” method dramatically decreases from 47.6% to 44.2% on average VOR due to its insufficient feature representation.
E. Computational Complexity

We present the computational complexity analysis of the proposed algorithm as well as the existing IVT and ℓ1 tracker. Assume that the size of the base matrix is \( M \times k \), and the iteration number is \( \eta \), the computation in the IVT method involves matrix-vector multiplication and the complexity is \( O(Mk) \). The computational complexity of the \( \ell_2 \) tracker for computing the sparse coefficients using the Lasso algorithm is \( O(M^2 + Mk) \). For our method, the base matrix is obtained from \( k \)-nearest neighbors, and the computational complexity for each candidate is \( O(k^3) \). Regarding finding the closed-form solution for each candidate in Eq. (13), the complexity is mathematically cubic to \( \mu \), and also demonstrated that the proposed NMC tracker achieves the similar nonnegative effect. Such approximation has been demonstrated by the theoretical analysis and experimental results. Thereby, the obtained coefficient vector and the non-negative constraint have greatly enhanced the representation ability for appearance modeling of the target, which effectively improves the tracking performance. The experimental results on two benchmarks have verified the rationality of such \( \ell_2 \) norm regularization approximation and the ensemble strategy, and also demonstrated that the proposed NMC tracker achieves a favorable performance over the state-of-the-art trackers.

V. CONCLUSION

In this paper, we have proposed a novel Nonnegative Multiple Coding (NMC) tracker that uses an ensemble of dictionaries to accurately characterize the target appearance. In addition, the nonnegative constraint on the coefficient vector is replaced by an efficient \( \ell_2 \) regularization term to achieve the similar nonnegative effect. Such approximation has been demonstrated by the theoretical analysis and experimental results. Thereby, the obtained coefficient vector and the non-negative constraint have greatly enhanced the representation ability for appearance modeling of the target, which effectively improves the tracking performance. The experimental results on two benchmarks have verified the rationality of such \( \ell_2 \) norm regularization approximation and the ensemble strategy, and also demonstrated that the proposed NMC tracker achieves a favorable performance over the state-of-the-art trackers.

APPENDIX A

Proof of Theorem 1

This section provides a proof of Theorem 1.

Proof. Let \( \mathbf{A}^T \mathbf{A} = \mathbf{U}^T \Lambda \mathbf{U} \), where \( \mathbf{U} \) is an orthogonal matrix. The diagonal matrix \( \Lambda \) is defined as \( \Lambda = \text{diag}(\mu_1, \mu_2, \ldots, \mu_k) \), of which the elements are eigenvalues of \( \mathbf{A}^T \mathbf{A} \). Then \( \mathbf{F} \) and \( \mathbf{F}^{-1} \) can be reformulated as:

\[
\begin{align*}
\mathbf{F} &= \mathbf{A}^T \mathbf{A} + \alpha \mathbf{I} = \mathbf{U}^T (\Lambda + \alpha \mathbf{I}) \mathbf{U} = \sum_{i=1}^{k} (\mu_i + \alpha) \mathbf{u}^{(i)} \mathbf{u}^{(i)^T}, \\
\mathbf{F}^{-1} &= (\Lambda + \alpha \mathbf{I})^{-1} = \mathbf{U}^T (\Lambda + \alpha \mathbf{I})^{-1} \mathbf{U} = \sum_{i=1}^{k} \frac{\mathbf{u}^{(i)} \mathbf{u}^{(i)^T}}{\mu_i + \alpha},
\end{align*}
\]

where \( \mathbf{u}^{(i)} \in \mathbb{R}^{1 \times k} \) is the \( i \)th row of the orthogonal matrix \( \mathbf{U} \), and \( (\Lambda + \alpha \mathbf{I})^{-1} \) is a diagonal matrix of which the \( i \)th diagonal element is \( 1/\mu_i + \alpha \). As a result, the coefficient vector \( \mathbf{x} \) can be rewritten as:

\[
\mathbf{x} = \mathbf{F}^{-1} \mathbf{1} = \sum_{i=1}^{k} \frac{1}{\mu_i + \alpha} \mathbf{u}^{(i)^T} \mathbf{1}.
\]

Note that \( \mathbf{F}^{-1} = \mathbf{I} \) and \( \mathbf{u}^{(i)} \mathbf{u}^{(i)^T} = 1 \), we have:

\[
\sum_{i=1}^{k} \mathbf{u}^{(i)^T} \mathbf{1} = \mathbf{u}^{(1)^T} \mathbf{1} + \cdots + \mathbf{u}^{(k)^T} \mathbf{1} = \mathbf{1}. \tag{23}
\]

Hence, the \( i \)th element of \( \mathbf{x} \) can be analytically expressed by:

\[
x_i = \sum_{r=1}^{k} \frac{1}{\mu_r + \alpha} \mathbf{u}_{ir} \mathbf{u}_{ir}, \quad \text{where} \quad \sum_{r=1}^{k} \mathbf{u}_{ir} \mathbf{u}_{ir} = 1.
\]

Consequently, our target is to prove that \( x_i \) in Eq. (24) is not less than 0 when \( \alpha \) satisfies the condition in Eq. (16). Without loss of generality, we assume that the first \( k \) elements in \( \mathbf{F}^{-1} \) are nonnegative, namely \( \mathbf{u}_{i1} \geq 0 \) for \( q = 1, 2, \ldots, \nu \). Therefore \( x_i \geq 0 \) is reformulated to Eq. (25), where \( \mathbf{P} = \sum_{r=1}^{k} \frac{1}{\mu_r + \alpha} \mathbf{u}_{ri} \mathbf{u}_{ri} \) and \( \mathbf{Q} = \sum_{r=1}^{k} \mathbf{u}_{ri} \mathbf{u}_{ri} \).

Furthermore, by defining that \( \mu_s = \min\{\mu_1, \mu_2, \ldots, \mu_k\} \) and \( \mu_t = \max\{\mu_{\nu+1}, \mu_{\nu+2}, \ldots, \mu_k\} \), \( \mathbf{P} \) and \( \mathbf{Q} \) can be bounded by:

\[
\begin{align*}
\mathbf{P} &\geq \frac{1}{\mu_t + \alpha} \left( \sum_{j=1}^{\nu} \mathbf{u}_{j1} \mathbf{u}_{j1} + \cdots + \mathbf{u}_{k1} \mathbf{u}_{k1} \right) \geq \mathbf{P}_1 \\
\mathbf{Q} &\leq \frac{1}{\mu_s + \alpha} \left( \sum_{j=1}^{\nu} \mathbf{u}_{j1} \mathbf{u}_{j1} + \cdots + \mathbf{u}_{\nu1} \mathbf{u}_{\nu1} \right) \geq \mathbf{Q}_1,
\end{align*}
\]

where \( \mathbf{P}_1 = \sum_{r=1}^{\nu} \mathbf{u}_{ri} \mathbf{u}_{ri} \), \( \mathbf{Q}_1 = \sum_{r=1}^{\nu} \mathbf{u}_{ri} \mathbf{u}_{ri} \), which leads to \( (\mathbf{P}_1 + \mathbf{Q}_1) \leq k \). Thus the lower bound of \( \alpha \) is chosen as:

\[
\alpha \geq -\mu_s \mathbf{P}_1 + \mu_t \mathbf{Q}_1.
\]

Furthermore, we divide the proof into two cases.

Case 1. \( \mu_s \geq \mu_t \).

In this case, the lower bound of \( \alpha \) can be chosen as \( \alpha \geq -\mu_t \), which is always satisfied due to \( \alpha \geq 0 \).

Case 2. \( \mu_s < \mu_t \).

In Eq. (29), by utilizing the inequalities of arithmetic and geometric means, we have \( \mathbf{P}_1 \leq \mathbf{Q}_1 \leq \sqrt{\sum_{j=1}^{\nu} \mathbf{u}_{jj} \mathbf{u}_{jj}} \), which leads to \( (\mathbf{P}_1 + \mathbf{Q}_1) \leq k \). Thus the lower bound of \( \alpha \) is chosen as:

\[
\alpha \geq -\mu_s (\mu_t - \mu_s) k.
\]
Furthermore, we discuss the range of eigenvalues $\mu_s$ and $\mu_l$ to obtain a more elegant bound of $\alpha$. Based on the Gershgorin’s theorem, the bounds of $\mu_s$ and $\mu_l$ satisfy:

\[
\begin{align*}
|\mu_s - (A^T A)_{ss}| &\leq \sum_{j \neq s} |(A^T A)_{sj}| \leq \frac{1}{\mu_s + \alpha} \sum_{j = 1}^{k} u_{j(s+1)} \\
|\mu_l - (A^T A)_{ll}| &\leq \sum_{j \neq l} |(A^T A)_{lj}| \leq \frac{1}{\mu_l + \alpha} \sum_{j = 1}^{k} u_{j(l+1)} + \frac{1}{\mu_l + 2\alpha} \sum_{j = 1}^{k} u_{j(l+2)} + \cdots + \frac{1}{\mu_l + k\alpha} \sum_{j = 1}^{k} u_{kj}
\end{align*}
\]

(31)

By plugging Eq. (31) into Eq. (30), the right-hand side of Eq. (30) is transformed to:

\[
\begin{align*}
-\mu_s + (\mu_l - \mu_s)k &\leq \sum_{j \neq s} |(A^T A)_{sj}| - (A^T A)_{ss} + k(A^T A)_{ll} \\
- k(A^T A)_{ss} + k \sum_{j \neq s} |(A^T A)_{sj}| + k \sum_{j \neq l} |(A^T A)_{lj}| \\
= &\ (k + 1) \left\{ \sum_{j = 1}^{k} |(A^T A)_{sj}| - (A^T A)_{ss} + k \sum_{j = 1}^{k} |(A^T A)_{lj}| \right\} \\
= &\ (k + 1) \left\{ \sum_{j = 1}^{k} |(A^T A)_{sj}| - 2(A^T A)_{ss} + k \sum_{j = 1}^{k} |(A^T A)_{lj}| \right\} \\
\leq &\ (k + 1) \|A^T A - 2I\|_\infty + k\|A^T A\|_\infty,
\end{align*}
\]

(32)

which concludes the proof.

\[\square\]

ACKNOWLEDGEMENT

The authors would like to thank Prof. Xiaolin Huang and Dr. Masoumeh Zareapoor from Shanghai Jiao Tong University for their proofreading and the anonymous reviewers for their insightful comments.

REFERENCES
