Consistency-constrained Non-negative Coding for Tracking

Xiaolin Tian, Licheng Jiao, Zhipeng Gan, Chaohui Wang, Xiaoli Zheng

Consistency-constrained Non-negative Coding for Tracking

Xiaolin Tian, *Member, IEEE,* Licheng Jiao, *Senior Member, IEEE,* Zhipeng Gan, Chaohui Wang, *Member, IEEE,* and Xiaoli Zheng

***Abstract*—A novel visual object tracking method based on consistency-constrained non-negative coding (CNC) is proposed in this paper. For the purpose of computational efficiency, superpixels are firstly extracted from each observed video frame. And then CNC is performed based on those obtained superpixels, where the locality on manifold is preserved by enforcing the temporal and spatial smoothness. The coding result is achieved via an iterative update scheme, which is proved to converge. The proposed method enhances the coding stability and makes the tracker more robust for object tracking. The tracking performance has been evaluated based on ten challenging benchmark sequences involving drastic motion, partial or severe occlusions, large variation in pose, and illumination variation. The experimental results demonstrate the superior performance of our method in comparison with ten state-of-art trackers.**

***Index Terms*—Object tracking, non-negative matrix factorization, non-negative coding, classifier.**

I. Introduction

VISUAL object tracking is one of the most active research topics in computer vision and has attracted great attention for many years. For developing a really accurate and efficient object tracker, there are still some challenging problems to be solved. Among them, appearance modeling is a key problem [1], [2]. When the difference between an object of interest and the background is not obvious, identifying the object from the background is a basic requirement for a tracker. Moreover, the ability of dealing with object appearance variations during tracking is also very crucial. Such variations can be caused by both extrinsic variations (occlusion, illumination, etc.) and intrinsic variations (shape deformation, pose change, etc.). Therefore, a better appearance model should meet two essential requirements: robustness and adaptability. Robustness mainly refers to the stability with respect to extrinsic variations, and adaptability means that a method possesses certain adjustability to intrinsic appearance variations.

The manifold assumption [3] assumes that if two data points are close in the geometrical structure of the original data distribution, the representations of the two points in a new basis space are also close to each other. The manifold plays an important role in exploiting various types of methods including action recognition [4], dimensionality reduction [3], manifold learning theory [5], [6], etc. Non-negative matrix factorization (NMF) [7] aims to obtain two non-negative matrices whose product is a good approximation to the original matrix, where the involved non-negative constraints compose a parts-based representation, by allowing only additive combinations. NMF is optimal for learning object parts [8]. However, NMF does not take into consideration the geometrical structure of the data space. Considering spatial and temporal consistency of the object representation between two consecutive frames, we introduce the manifold structure into NMF to construct a robust tracking model. Based on the manifold assumption, we propose a novel method named consistency-constrained non-negative coding (CNC). Similar to sparse representation, we represent an object using a set of feature vectors and use one of two factors of NMF [7] as the dictionary and the other as the corresponding coefficients (or codes) for those features vectors. To preserve the locality on manifold, we choose manifold constraint to incorporate the aforementioned temporal and spatial consistency into the non-negative coding. The proposed tracking method has the ability of estimating the feature correspondence between two adjacent frames. Meanwhile, similar codes are achieved for the neighboring locations in a same frame, which allows a description of uncertainty using the proposed CNC. This coding method is able to interpret small misalignments or partial occlusions as unlikely events other than impossible events, which weakens the oversensitivity to spatial structure. Thanks to the consideration of the spatial and temporal consistency (i.e., manifold) of the object, the CNC is able to robustly track an object under complex environments. Experimental results show that the proposed CNC achieves an impressive object tracking performance with a linear support vector machine (SVM) classifier [9]–[11]. The contributions of this paper are listed below:

1) The proposed CNC method which encodes spatial and temporal information of video sequence is proposed, in which we design a new non-negative matrix factorization objective function and incorporate the context constraint based on two adjacent frames and the neighboring superpixels.

2) An optimization scheme which solves the CNC objective function via iterative updates is developed. The convergence proof of the scheme is also provided.

The proposed method is illustrated by the flowchart in Fig. 1, and the algorithms in Algorithm 1 and 2.

This work was supported by the National Natural Science Foundation of China under Grant 61571342, 61203303, and 61202176; by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2014JM8301; by the National Basic Research Program of China under Grant 2013CB329402; by the Fundamental Research Funds for the Central Universities.

X. Tian, L. Jiao, Z. Gan and X. Zheng are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an 710071, China (e-mail: xltian@mail.xidian.edu.cn).

C. Wang is with Laboratoire d'Informatique Gaspard Monge - CNRS UMR 8049, Université Paris-Est, 77454 Marne-la-Vallée Cedex 2, France.

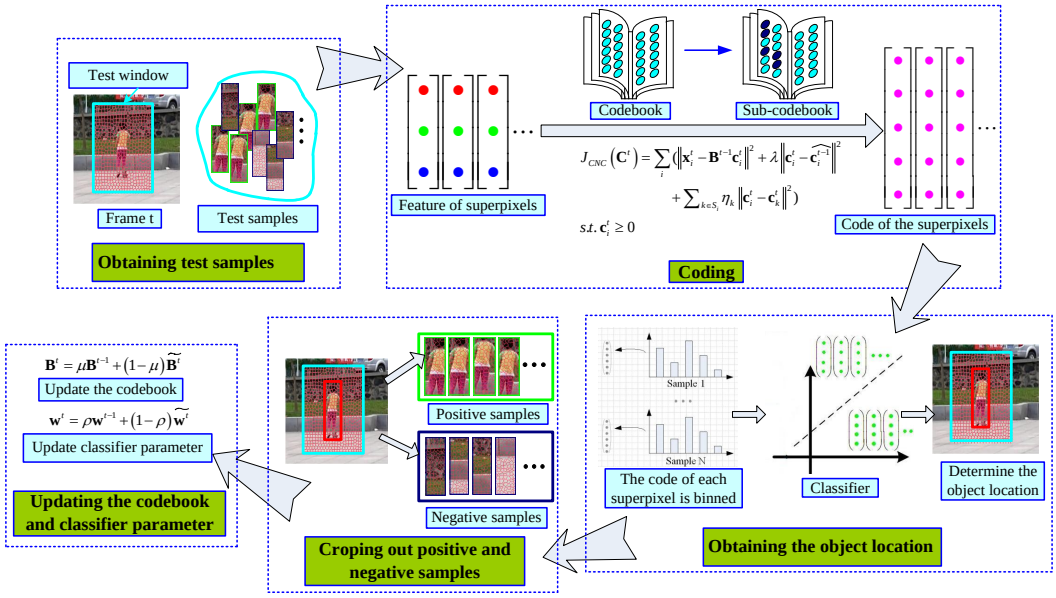

Fig. 1. Flowchart of CNC. When a new frame arrives, the test window in the frame is segmented into superpixels and all test samples are encoded. The test sample with the maximum score generated by SVM is the predicted object configuration. Based on the estimated object location, we crop out positive and negative samples, and then update the codebook and the corresponding classifier parameter.

II. Related Work

To correctly model object/background appearance and deal with tracking drift, an increasing number of algorithms have been proposed. For example, the incremental visual tracking (IVT) method [12] and the NormalHedge method [13] achieved efficient tracking through online model update. Adam et al. introduced a robust fragments-based tracking using the integral histogram (Frag) [14]. In the same year, online boosting (OAB) was used in [15] to update appearance model online. Essentially, the OAB method is an online AdaBoost feature selection algorithm based on which the classifier is adapted online during the tracking process. Two years later, Grabner et al. further proposed a semi-supervised online boosting algorithm for robust tracking (SemiB) [16], in order to alleviate drift in tracking applications. To deal with more challenging problems, visual tracking decomposition (VTD) [17] was introduced, leading to a more efficient representation of observations and motions. Babenko et al. proposed a multiple instance learning-based (MIL) [18], [19] method which can avoid degrading classifier accuracy and further drift. Also based on MIL, a real-time compressive tracking (CT) [20] compresses samples of the object of interest and the background using a same sparse measurement matrix. Distribution fields (DF) were used for object tracking in [21], which significantly alleviates the image information destruction caused by image blurring and reduces the effect of outliers during tracking.

Sparse representation and compressed sensing [22], [23] have also been widely applied to object tracking [24]–[27], where an observation is approximated by a sparse linear combination of a given basis. For example, Mei et al. proposed a robust visual tracking using L1 minimization (L1-track) [28], [29], which performs robust visual tracking by projecting the object of interest onto a set of trivial templates. Least soft-threshold squares tracking (LSST) was proposed in [30], which derives a new distance to measure the difference between an observation sample and the dictionary by maximizing the joint likelihood of parameters. The method deals with object/background appearance change via a proper updating scheme and also is more effective in dealing with outliers. Liu et al. proposed spatial neighborhood-constrained linear coding for visual object tracking [31]. This method exploits a new spatial neighborhood-constrained linear coding strategy by embedding spatial layout information into the involved coding process, together with a co-learning approach to update the dictionaries.

Different from sparse coding representations, NMF with non-negativity constraints can effectively implement an image-patch-based representation by allowing only additive combination [7]. In addition, NMF yields non-negative factors and has the advantage of interpretability, which has been applied to various data sets. In [7], two types of multiplicative algorithms for NMF were analyzed, whose monotonic convergence can be proven using an auxiliary function. Moreover, convex and semi-nonnegative matrix factorizations [32] further extend the application of NMF, by allowing a data matrix to contain mixed signs. Wu et al. applied NMF to visual tracking successfully [8], where the appearance of an object is represented with a non-negative linear combination of non-negative components and is learned from examples observed in previous frames. Furthermore, an efficient appearance-model updating scheme based on an online iterative learning algorithm and a particle filter framework is considered in this method. Based on online robust non-negative dictionary learning, Wang et al. [33] proposed a sparse tracker under the particle filter framework, and each learned template can capture a distinctive aspect of an object of interest. Qian et

al. [34] studied an extended incremental non-negative matrix factorization and developed an effective appearance model for visual tracking. Zhang et al. [35] developed an incremental non-negative matrix factorization (INMF) scheme, together with dual-norm constraints, to reduce the effect of noise during tracking. The method takes partial occlusions into the likelihood function and updates its object appearance model to alleviate tracking drift. We extend the non-negative matrix factorization algorithm by defining a new regularization term to preserve the locality on manifold. The proposed method improves the coding stability and makes the tracker more robust for object tracking. Experimental evaluation has been performed on ten challenging benchmark sequences, which demonstrates that our approach achieves or exceeds the state-of-the-art performance in visual object tracking.

III. Superpixel-based CNC

A. Non-negative Matrix Factorization (NMF)

Non-negative matrix factorization (NMF) is a matrix factorization algorithm that aims to factorize a data matrix into two non-negative matrices. Let $\mathbf{X}$ be a set of N D-dimensional feature vectors, i.e., $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in R^{D\times N}$. $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_M] \in R^{D\times M}$ and $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N] \in R^{M\times N}$ are two non-negative matrices, denoting a codebook of size M and the coefficients (*codes*) for those N feature vectors, respectively. The NMF minimizes the following objective function:

$$J_{NMF} = \|\mathbf{X} - \mathbf{BC}\|^2 = \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{B}\mathbf{c}_i\|^2 \quad s.t.\ \mathbf{B} \geq 0, \mathbf{C} \geq 0 \tag{1}$$

The following iterative update algorithm [7] can be used to efficiently minimize the above objective function:

$$b_{m,n} \leftarrow b_{m,n} \frac{(\mathbf{X}\mathbf{C}^T)_{m,n}}{(\mathbf{B}\mathbf{C}\mathbf{C}^T)_{m,n}} \quad c_{m,n} \leftarrow c_{m,n} \frac{(\mathbf{B}^T\mathbf{X})_{m,n}}{(\mathbf{B}^T\mathbf{B}\mathbf{C})_{m,n}} \tag{2}$$

where $\mathbf{b}_m = [b_{m,1}, b_{m,2}, \ldots, b_{m,D}]^T$, $\mathbf{c}_m = [c_{m,1}, c_{m,2}, \ldots, c_{m,M}]^T$, and $(\cdot)_{m,n}$ represents the $(m,n)^{\text{th}}$ entry of a matrix.

B. Consistency-constrained Non-negative Coding (CNC)

Given an input image, we extract superpixels from it using an existing image superpixelization method (e.g., *TurboPixels* [36]), and describe each superpixel i by a HSV feature vector $\mathbf{x}_i = (x_1, x_2, x_3)$ that consists of the average value of HSV features of all pixels in superpixel i. To maintain the coding stability of each superpixel, we integrate the consistency constraint into the non-negative coding, leading to the proposed consistency-constrained non-negative coding (CNC).

Let $\mathbf{X}^t = [\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_N^t] \in R^{D\times N}$ be a set of N feature vectors extracted from frame t. It is factorized into $\mathbf{B}^{t-1}$ and $\mathbf{C}^t$ via NMF. $\mathbf{B}^{t-1} = [\mathbf{b}_1^{t-1}, \mathbf{b}_2^{t-1}, \ldots, \mathbf{b}_M^{t-1}] \in R^{D\times M}$ is the codebook which is formed in frame $t-1$. Since $\mathbf{B}^{t-1}$ is formed in frame $t-1$, it is considered to be constant for the superpixel coding in frame t. $\mathbf{C}^t = [\mathbf{c}_1^t, \mathbf{c}_2^t, \ldots, \mathbf{c}_N^t] \in R^{M\times N}$ is the code for those N feature vectors, where $\mathbf{c}_i^t = [c_{i,1}^t, c_{i,2}^t, \ldots, c_{i,M}^t]^T$. The constructed objective function J_{CNC} of CNC can be written as follow:

$$J_{CNC}(\mathbf{C}^t) = \sum_i (\|\mathbf{x}_i^t - \mathbf{B}^{t-1}\mathbf{c}_i^t\|^2 + \lambda\|\mathbf{c}_i^t - \widehat{\mathbf{c}_i^{t-1}}\|^2 + \sum_{k\in\mathcal{S}_i} \eta_k \|\mathbf{c}_i^t - \mathbf{c}_k^t\|^2) \quad s.t.\ \mathbf{c}_i^t \geq 0 \tag{3}$$

where $\widehat{\mathbf{c}_i^{t-1}}$ is the superpixel code in frame $t-1$ corresponding to $\mathbf{c}_i^t$ (the correspondence is described in Section IV-D), $\lambda \geq 0$ and $\eta \geq 0$ are two regularization parameters, and $\mathbf{c}_k^t$ is the superpixel code neighboring with superpixel i in frame t. For each superpixel i, $\mathcal{S}_i$ denotes the set of its neighboring superpixels. By assuming that the influence of neighboring superpixels is identical, i.e., $\eta_k = \eta$, then Eq. (3) becomes:

$$J_{CNC}(\mathbf{C}^t) = \sum_i (\|\mathbf{x}_i^t - \mathbf{B}^{t-1}\mathbf{c}_i^t\|^2 + \lambda\|\mathbf{c}_i^t - \widehat{\mathbf{c}_i^{t-1}}\|^2 + \sum_{k\in\mathcal{S}_i} \eta \|\mathbf{c}_i^t - \mathbf{c}_k^t\|^2) \tag{4}$$

In the above equation, the first term encodes the residual after projecting $\mathbf{x}_i^t$ on the non-negative factorized subspace $\mathbf{B}^{t-1}$. The second term is used to enforce the consistency between the codes of two adjacent frames, ensuring that similar superpixel patches will have similar codes based on the codebook $\mathbf{B}^{t-1}$. The third term is employed to enforce the consistency between the codes of neighboring superpixels. Both of the consistency constraints preserve the locality on manifold and make two neighboring superpixels of original space should also be close in the new space spanned by $\mathbf{B}^{t-1}$.

For each superpixel i, $\mathcal{S}_i$ consists of two closest neighboring superpixels, determined by minimizing the Euclidean distance between a superpixel and the superpixel i. The choice is made by considering the correlation between the spatial distance and the appearance similarity, as well as the computational complexity. The values of the regularization parameters λ and η are described in Section IV-F. We rearrange Eq. (4) and obtain the following form:

$$\begin{aligned}\underset{\mathbf{c}^t\geq 0}{\operatorname{argmin}} J_{CNC}(\mathbf{C}^t) = \underset{\mathbf{c}^t\geq 0}{\operatorname{argmin}} \sum_i \Bigg(&\left\| \begin{bmatrix} \mathbf{x}_i^t \\ 0_{M\times 1} \\ 0_{M\times 1} \end{bmatrix} - \begin{bmatrix} \mathbf{B}^{t-1} \\ \sqrt{\lambda}\mathbf{I}_M \\ \sqrt{S\eta}\mathbf{I}_M \end{bmatrix} \mathbf{c}_i^t \right\|^2 \\ &- 2\lambda \mathbf{c}_i^{t^T} \widehat{\mathbf{c}_i^{t-1}} - 2\eta \sum_{k\in\mathcal{S}_i} \mathbf{c}_i^{t^T} \mathbf{c}_k^t \Bigg)\end{aligned} \tag{5}$$

where $\mathbf{I}_M$ is the identity matrix with size $M \times M$. Let $\widehat{\mathbf{x}_i^t} = \begin{bmatrix} \mathbf{x}_i^t \\ 0_{M\times 1} \\ 0_{M\times 1} \end{bmatrix}$, $\widehat{\mathbf{B}^{t-1}} = \begin{bmatrix} \mathbf{B}^{t-1} \\ \sqrt{\lambda}\mathbf{I}_M \\ \sqrt{S\eta}\mathbf{I}_M \end{bmatrix}$. Eq. (5) can then be further reformulated as:

$$\begin{aligned}\underset{\mathbf{c}^t\geq 0}{\operatorname{argmin}} J_{CNC}(\mathbf{C}^t) = \underset{\mathbf{c}^t\geq 0}{\operatorname{argmin}} \sum_i \Bigg(&\left\| \widehat{\mathbf{x}_i^t} - \widehat{\mathbf{B}^{t-1}} \mathbf{c}_i^t \right\|^2 \\ &- 2\lambda \mathbf{c}_i^{t^T} \widehat{\mathbf{c}_i^{t-1}} - 2\eta \sum_{k\in\mathcal{S}_i} \mathbf{c}_i^{t^T} \mathbf{c}_k^t \Bigg)\end{aligned} \tag{6}$$

Let $\Phi = [\varphi_1, \varphi_2, \ldots, \varphi_N] \in R^{M\times N}$, $\varphi_i = [\phi_{i,1}, \phi_{i,2}, \ldots, \phi_{i,M}]^T$, and $\phi_{i,j}$ denote the Lagrange multiplier for the constraint $c_{i,j}^t > 0$. We can get the following Lagrangian function:

$$\begin{aligned}L(\mathbf{C}^t, \Phi) = \sum_i \Bigg(&\left\| \widehat{\mathbf{x}_i^t} - \widehat{\mathbf{B}^{t-1}} \mathbf{c}_i^t \right\|^2 - 2\lambda \mathbf{c}_i^{t^T} \widehat{\mathbf{c}_i^{t-1}} \\ &- 2\eta \sum_{k\in\mathcal{S}_i} \mathbf{c}_i^{t^T} \mathbf{c}_k^t + \varphi_i^T \mathbf{c}_i^t \Bigg)\end{aligned} \tag{7}$$

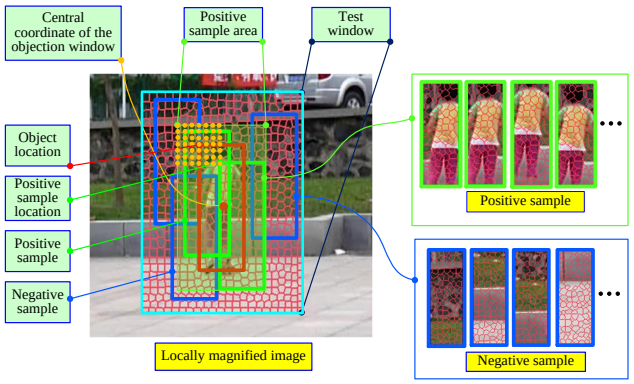

Fig. 2. Illustration of the acquisition of position and negative samples.

The partial derivatives of $L(\mathbf{C}^t, \Phi)$ with respect to $\mathbf{c}_i^t$ are

$$\frac{\partial L}{\partial \mathbf{c}_i^t} = -2\widehat{\mathbf{B}^{t-1}}^T \widehat{\mathbf{x}_i^t} + 2\widehat{\mathbf{B}^{t-1}}^T \widehat{\mathbf{B}^{t-1}} \mathbf{c}_i^t - 2\lambda \widehat{\mathbf{c}_i^{t-1}} - 2\eta \sum_{k \in \mathcal{S}_i} \mathbf{c}_k^t + \varphi_i \tag{8}$$

Using the KKT conditions [37] $\phi_{m,n} c_{m,n}^t = 0$, we can get $[-(\widehat{\mathbf{B}^{t-1}}^T \widehat{\mathbf{x}_m^t})_n + (\widehat{\mathbf{B}^{t-1}}^T \widehat{\mathbf{B}^{t-1}} \mathbf{c}_m^t)_n - (\lambda \widehat{\mathbf{c}_m^{t-1}} + \eta \sum_{k \in \mathcal{S}_i} \mathbf{c}_k^t)_n](c_{m,n}^t) = 0$. This equation requires either of the two factors is zero or both are zero. If $c_{m,n}^t = 0$, then $(c_{m,n}^t)^2 = 0$. We get the following equation:

$$\begin{aligned} &-(\widehat{\mathbf{B}^{t-1}}^T \widehat{\mathbf{x}_m^t})_n (c_{m,n}^t)^2 + (\widehat{\mathbf{B}^{t-1}}^T \widehat{\mathbf{B}^{t-1}} \mathbf{c}_m^t)_n (c_{m,n}^t)^2 \\ &-(\lambda \widehat{\mathbf{c}_m^{t-1}} + \eta \sum_{k \in \mathcal{S}_i} \mathbf{c}_k^t)_n (c_{m,n}^t)^2 = 0 \end{aligned} \tag{9}$$

This is a fixed point equation that the solution must satisfy at convergence. We get the following formula for iterative updating and estimating $c_{m,n}^t$.

$$c_{m,n}^t \leftarrow c_{m,n}^t \sqrt{\frac{(\widehat{\mathbf{B}^{t-1}}^T \widehat{\mathbf{x}_m^t} + \lambda \widehat{\mathbf{c}_m^{t-1}} + \eta \sum_{k \in \mathcal{S}_i} \mathbf{c}_k^t)_n}{(\widehat{\mathbf{B}^{t-1}}^T \widehat{\mathbf{B}^{t-1}} \mathbf{c}_m^t)_n}} \tag{10}$$

The convergence property of Eq. (10) is proved in Appendix.

In essence, $\mathbf{B}^{t-1}$ represents the local structure of image frame. Although updating $\mathbf{B}^{t-1}$ is beneficial for reducing the reconstruction error of $\mathbf{X}^t$, the optimized entries in $\mathbf{B}^{t-1}$ will not correctly represent the basic structure of video sequences, which will lead to instability of $\mathbf{C}^t$ and thus tracking drift. Hence, we update $\mathbf{B}^{t-1}$ by combining the codebook of the previous frame and the new observation in the current frame via a component-wise convex combination of them (see Section IV-B).

IV. CNC for Tracking

A. Positive and Negative Samples

Since object tracking is to locate a specific object in a test area of a new frame, and the object locations in two adjacent frames are close, the object of interest is assumed to be inside a certain area referred to as test window (shown in Fig. 2), which corresponds to a surrounding area of the object rather than the entire image and is obtained by expanding the object window 60 pixels in vertical and horizontal direction in our experiments. Superpixel segmentation is done in the area. When a new frame arrives, the object is searched in the range of the defined test window and the test sample with the maximum score is the estimated location of the object.

After obtaining the estimated object location L_t^* in frame t, we crop out the positive samples and negative samples, which are to be used for updating the tracker for the next frame. If the number of acquired positive samples is too large, a part of positive samples will not be able to correctly model the object appearance, and as a result, the tracking model will become confused and its discriminative power will become weak. We illustrate the acquisition of positive and negative samples in Fig. 2. Let L_t^* denote the coordinates of the pixel at the top left corner of the object window (the red box in Fig. 2). Positive samples are those windows that are centered within the 7×7 neighborhood of the object location and have the same size as the objection window (49 positive samples), which are illustrated by green boxes in Fig. 2. These positive samples form a positive sample area (green area). We randomly select 49 samples as negative samples (blue boxes) from the area between the positive sample area and the test window. The size of the negative samples is also equal to the object window. It is worth noting that local overlapping between the negative sample and the positive sample areas is possible. Accordingly, the positive sample set consists of the parts containing almost the whole object (with the estimated location), and the negative sample set is mainly composed with the background.

B. Initialization and Update of the Codebook

It is known that in general, the more codewords a codebook consists of, the more discriminative the coding method is. However, a larger number of codewords in a codebook will increase the computational complexity. In experiments, we use 300 codewords to build the codebook. For the first frame, we execute k-means clustering on each superpixel feature in the search window, and the cluster centroids are used as initial entries of the codebook $\mathbf{B}^1$. The initial value of the code $\mathbf{c}_i^1$ of superpixel i is obtained by the membership of $\mathbf{x}_i^1$ belonging to each entry of $\mathbf{B}^1$ [38].

It is important for the codebook to be updated, so as to capture the appearance variation caused by illumination or pose change and accordingly to alleviate accumulated error and tracker drift. We achieve this by combining the codebook of the previous frame and the new observation in the current frame using a component-wise convex combination of them, which is described in Eq. (11). When dealing with frame t, the codebook of the previous frame is represented as $\mathbf{B}^{t-1}$. The superpixel features in the current frame can be clustered (e.g., k-means clustering) using the codewords of the codebook $\mathbf{B}^{t-1}$ as the cluster centroids. By averaging all feature vectors in one cluster, the new centroid of the cluster can be computed. The new centroids of all clusters construct a new codebook $\widetilde{\mathbf{B}^t}$. Finally, the codebook of current frame t is obtained as follow:

$$\mathbf{B}^t = \alpha \mathbf{B}^{t-1} + (1-\alpha)\widetilde{\mathbf{B}^t} \tag{11}$$

The parameter α controls the rate at which the codebook is updated, which is described in Section IV-F.

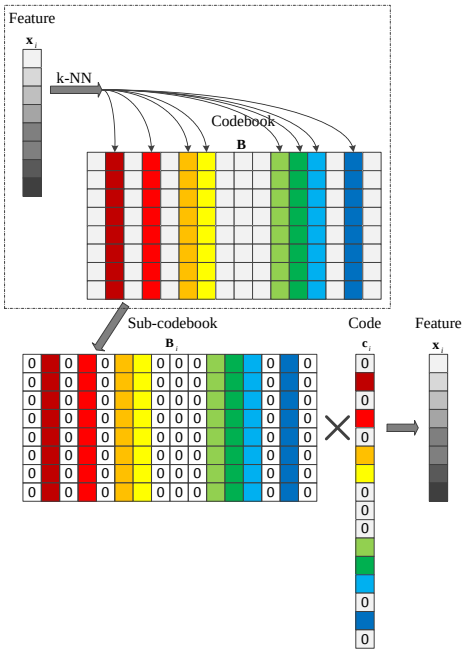

Fig. 3. Sub-codebook construction and feature encoding.

C. Image Representation

The encoding process aims to use a few codebook entries (codewords) to effectively represent typical feature vectors, which implies that most of the coefficients/codes for some feature vector are close to zero and only a few take significantly non-zero values. In our method, similar to locality-constrained linear coding (LLC) [39], each feature is projected into a local coordinate system to form a sub-codebook. We use k-nearest neighbors (k-NN) to select several principal entries of the codebook $\mathbf{B}$ as the sub-codebook $\mathbf{B}_i$ for superpixel i, so as to represent a feature using typical entries in the codebook $\mathbf{B}$. The sub-codebook method is sparse in the sense that the code $\mathbf{c}_i$ only has a few significant values. The idea of sub-codebook construction is illustrated in Fig. 3. In our experiments, we select the nearest 10 codewords from the codebook $\mathbf{B}^{t-1}$ for the feature vector $\mathbf{x}_i^t$ of superpixel i in frame t, and construct a sub-codebook $\mathbf{B}_i^{t-1}$ for superpixel i.

During CNC processing, we adopt the max pooling and the spatial pyramid matching (SPM) [39], [40], where the codes of features for each spatial sub-region and the original image are pooled to obtain the corresponding pooled features. These pooled features are normalized and concatenated as the final image representation. The procedure of the max pooling and SPM is shown in Fig. 4.

D. Pre-coding

For a new frame t, the pre-coding is achieved by the basic NMF, i.e., the second equation of Eq. (2). After we obtain the code of each superpixel by the pre-coding, we search for the corresponding $\widehat{\mathbf{c}_i^{t-1}}$ for superpixel i according to the following formulas:

Algorithm 1 Preparation

Input: Video frame number 1

1: Segment the test window into superpixels and compute their HSV feature vectors.
2: Use k-means to obtain a codebook $\mathbf{B}^1$.
3: Crop out positive and negative samples based on the given object location.
4: Encode each sample based on $\mathbf{B}^1$ via the second equation of Eq. (2).
5: Train the classifier parameter $\mathbf{w}^1$ based on the code and the label of each sample.

Output: An initial codebook $\mathbf{B}^1$ and an initial classifier parameter $\mathbf{w}^1$.

$$\begin{aligned}\widetilde{\mathbf{c}_i^t} &= \underset{\mathbf{C}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\mathbf{x}_i^t - \mathbf{B}_i^{t-1}\mathbf{c}_i^t\|^2 \\ \widehat{\mathbf{c}_i^{t-1}} &= \underset{\mathbf{c}_{\partial i}^{t-1}}{\operatorname{argmin}} \|\widetilde{\mathbf{c}_i^t} - \mathbf{c}_{\partial i}^{t-1}\|^2\end{aligned} \tag{12}$$

According to the first line of Eq. (12), we first estimate the code $\widetilde{\mathbf{c}_i^t}$ of $\mathbf{x}_i^t$. Then we search for the corresponding code $\widehat{\mathbf{c}_i^{t-1}}$ in frame $t-1$ for $\mathbf{c}_i^t$ through the second line of Eq. (12), and the search area is represented as ∂i consisting of the neighboring superpixels of superpixel i in frame $t-1$. Finally, we obtain the corresponding code $\widehat{\mathbf{c}_i^{t-1}}$ of $\mathbf{c}_i^t$.

E. Classifier Parameter Update

To achieve an accurate tracker, it is important to update the involved classifier. Similar to the codebook update, we perform the classifier parameter update by combining the parameter of the previous frame and the new observation in the current frame using a component-wise convex combination of them, which is shown in Eq. (13). When dealing with frame t, the classifier parameter of the previous frame is represented as $\mathbf{w}^{t-1}$. Based on the parameter $\mathbf{w}^{t-1}$, the location of the sample with the maximum score generated by the linear SVM is used as the object location. After obtaining the new objection location, we can obtain a new positive sample set as well as a new negative one. By training these samples, we achieve the new classifier parameter $\widetilde{\mathbf{w}^t}$ of the linear SVM. Finally, the classifier parameter is obtained as follow:

$$\mathbf{w}^t = \rho\mathbf{w}^{t-1} + (1-\rho)\widetilde{\mathbf{w}^t} \tag{13}$$

The parameter ρ, discussed in Section IV-F, controls the rate at which the classifier parameter is updated.

F. Parameter Analysis

Because the parameters α, ρ, λ, and η are critical to tracking performance, we analyze the effect of each parameter during tracking, and then choose appropriate parameter values. The two parameters, α and ρ, control the rate at which the codebook and the classifier parameter are updated, respectively. The smaller the two parameters are, the faster the codebook and the classifier parameter will be updated. When α and

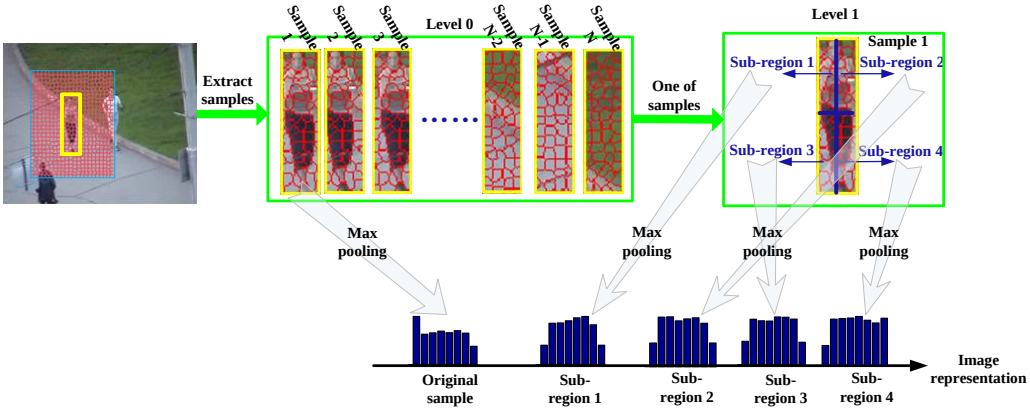

Fig. 4. Illustration of the formation of the sample representation.

Algorithm 2 Tracking

Input: A video sequences of length N

1: **for** $t = 2$ to N **do**
2: Segment the test window into superpixels and compute their feature vectors.
3: Encode each sample with $\mathbf{B}^{t-1}$ by the proposed coding method.
4: Predict the object location in the search window of frame t with the trained classifier parameter $\mathbf{w}^{t-1}$.
5: Crop out positive and negative samples based on the predicted object location.
6: Update the codebook to obtain a new codebook $\mathbf{B}^t$.
7: Encode each sample with $\mathbf{B}^t$ via the proposed coding method.
8: Update the classifier parameter $\mathbf{w}^t$.
9: **end for**

Output: The estimated object location in each frame.

ρ are too small, errors will be accumulated quickly, which leads to tracking drift. In contrast, if α and ρ are too large, the codebook and the classifier parameter will be updated very slowly, which cannot reflect appearance change in time. The parameter λ controls the consistency between the codes of two adjacent frames. η controls the consistency between the codes of two neighboring superpixels in the same frame. The larger λ is, the more similar the two corresponding superpixel codes between two adjacent frames will be. If η is large, two neighboring superpixels will tend to possess two similar codes. Large λ and η are beneficial for preserving the locality on manifold by enforcing the temporal and spatial smoothness, but cannot reflect the appearance change. If λ and η are small, the superpixel code is capable of reflecting the appearance change. However, it impairs the similarity between neighboring superpixel codes. The parameters used in our experiments were set as: $\alpha = 0.85$, $\rho = 0.9$, $\lambda = 10$ and $\eta = 5$.

G. Implementation of CNC

The implementation flowcharts of the proposed method are described in Algorithm 1 and 2.

We implemented the proposed method in MATLAB on a machine with an Intel core i5-3470 CPU, 4 GB memory and Microsoft Windows 7 operating system. The running time is about 5 frames per second (FPS).

V. Experiments

A. Test Sequences and Competitive Methods

In experiments, we use ten challenging sequences in the benchmark presented in [41], which are listed in Table I. The tracking results are compared with the following ten state-of-the-art tracking methods: (1) MIL [18], [19], (2) VTD [17], (3) CT [20], (4) DF [21], (5) SCM [42], (6) CXT [43], (7) TLD [44], (8) Struck [45], (9) STC [46], (10) ONNDL [33]. Among the ten competitive methods, the results of the first eight methods are obtained from the benchmark. In general, the eight methods have obvious advantages and are the best current trackers in the benchmark. For the last two methods (STC and ONNDL), we obtain the results by executing the publicly-available source codes provided by the authors with well-tuned parameters.

B. Evaluation Measures

For the purpose of fairness, following [41], we use precision and success plots to evaluate the performance of various algorithms. A precision plot describes the percentage of frames whose estimated location is within a given threshold range based on the ground truth. A success plot depicts the ratio of successful frames at the overlap thresholds varying from 0 to 1 [41]. The one-pass evaluation (OPE) method is used to run all trackers throughout each test sequence with the initial object position in the first frame provided by the ground truth, and then the average precision and the success rate are computed and reported [41].

C. Quantitative Evaluation

Table II provides the tracking success rate on all the test sequences. The best, the second best and the third best results are shown using three different colors. Obviously, the proposed method achieves four best results, one second best results and

TABLE I
DESCRIPTION OF TEN CHALLENGING SEQUENCES

Sequences	Frame length	Frame size	Target size	Attribute description
Bolt	350	640×360	26×61	Drastic motion, shape deformation, moving camera, variation in scale
Basketball	725	576×432	34×81	Drastic motion, large variation in pose, half and full occlusion, illumination variation
David3	252	640×480	35×131	Partial occlusion, heavy appearance change, out-of-plane rotation, background clutters
Girl2	1500	640×480	44×171	Moving camera, cluttered background, blur, large variation in pose and scale, half and full occlusion
Woman	575	352×288	21×95	Lighting condition change, moving camera, half occlusion, deformation
Jogging	307	352×288	25×101	Large variation in pose, severe occlusion, in-plane rotation
Couple	140	320×240	25×62	Fast motion, deformation, scale variation, cluttered background
Crossing	120	320×240	17×50	Scale variation, deformation, out-of-plane rotation
Subway	175	352×288	19×51	Partial occlusion, deformation, background clutters
Football1	74	352×288	26×43	Fast motion, in-plane rotation, out-of-plane rotation, similar background

three third best results on the ten test sequences. Table III shows the results of the average center location errors in pixels. For seven out of ten video sequences, it is obvious that our method makes the top three compared with the ten competitive methods. As indicated in the two tables, our tracker performs better than the other ten state-of-the-art methods on the whole according to the two evaluation measures.

Fig. 5. provides the center distance in pixels between the tracking results and the given ground truth for each frame of the ten sequences. As shown in the figure, our method is superior to the other ten methods in most cases. Obviously, the proposed method successfully tracks the objects in the ten sequences. It is stable mostly due to the fact that the proposed method incorporates the consistency constraint into the non-negative coding.

D. Qualitative Evaluation

Severe or partial occlusion: As shown in Fig. 6, MIL and CT are not satisfied when the object is heavily blocked, and the two methods lead to tracking failure after occlusion. On the one hand, Haar-like feature adopted in both methods has poor discriminative ability after occlusion. On the other hand, a number of samples may not contain the object after several frames of occlusion occur. Consequently, both of the methods fail due to occlusion such as Girl2 #329, #978, #1362, Woman #257, #479, Jogging #137, #210, #306. However, the two methods are valid for short-time and slight occlusion such as Subway #70, #127. Because the SCM and Struck methods can handle partial occlusion based on the part-based representation, both methods achieve the better tracking results on Subway and Woman sequences. VTD is not ideal either for dealing with heavy occlusion (e.g., Girl2 #329, #978, #1362, Jogging #137, #210, #306), since its design is apt to lose the object after serious occlusion.

The DF method builds an image descriptor, where the information of pixel values is smoothed. CXT exploits the context on-the-fly in distracters and supporters. STC employs the dense spatio-temporal context learning. The three methods yield the tracking drift when occlusion occurs several times (e.g., Girl2 #978, #1362, Basketball #717). In addition to the three methods, SCM and Struck also produce the tracking drift. ONNDL adopts the non-negative coding strategy and is able to handle occlusion and scale variation. However, it cannot correctly adjust the tracking when the object comes to severe occlusion such as Girl2 #329, #978, #1362, Jogging #306, and Subway #127.

It is observed that our performance is the best on the three videos with occlusion (e.g., Girl2 #978, #1362, David3 #130, #246, Jogging #210, #306) in comparison with other trackers. Thanks to the stability of our consistency-constrained coding and also to the update of the codebook and the classifier parameter after processing each frame, the proposed method improves the coding robustness and is also robust to occlusions.

Discrimination of similar objects and background clutter: As observed in our experience, the discrimination of similar objects is a difficult problem. Let us take Basketball and Subway for example, which are considered to be two challenging sequences. In the Basketball sequence, the object (No. 9 player) passed through the other players for many times and even overlapped with others, whose appearance is similar to a part of the background in the candidates. As we can see from the results, our tracker still successfully tracks the object (e.g., Basketball #110, #350, #717) and the similar situation occurs in Subway #70, #127.

In the sequences Bolt (#58, #214, #328), Subway (#70, #127) and Girl2 (#978, #1362), the quite complex background leads to tracking failure for many comparative algorithms. Our method adopts the consistency-constrained form based on the non-negative coding, and trains a stable classifier with real-time update, which can strongly discriminate those similar features between the object and the background. Thanks to this, the proposed method performs well in comparison with other comparative methods.

Rapid and abrupt motion: A fast and abrupt motion of the object or the camera can lead to blurred video frames and severe change of the object pose, which is one difficulty in object tracking. In order to test the ability in addressing this difficulty, we chose to test on Bolt, Basketball and Couple sequences. For the Bolt sequence, except for ONNDL and VTD, the other comparative methods fail when some fast motion happens (e.g., Bolt #58, #214). For the Couple sequence, except for TLD and Struck, the other comparative methods cannot correctly track the object (e.g., Couple #127). However, it is difficult for TLD to handle fast and abrupt object motions successfully (e.g., Basketball #110, #350, #717, Football #71), and the same situation also happens in Basketball for SCM and Struck methods.

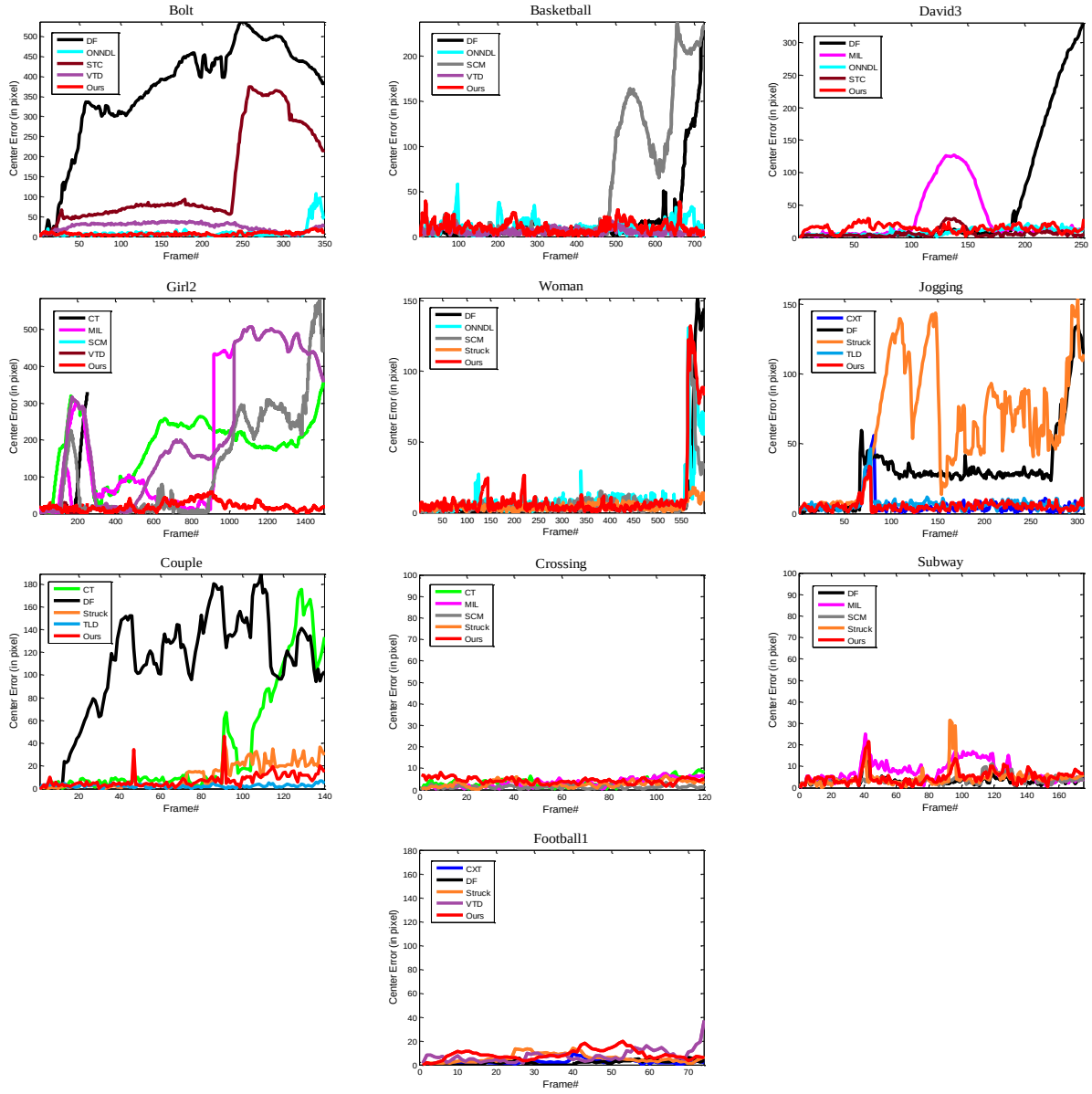

Fig. 5. Center distance (in pixels) of each frame between the tracking result and the ground truth for the top five trackers applied to ten video sequences.

TABLE II
RATE OF CORRECTLY TRACKED FRAMES (%). THE BEST, SECOND BEST AND THIRD BEST RESULTS ARE SHOWN IN RED, BLUE AND GREEN NUMBERS RESPECTIVELY.

Sequences	MIL	VTD	CT	DF	SCM	CXT	TLD	Struck	STC	ONNDL	Ours
Bolt	3.43	23.43	0.57	7.43	1.43	1.71	1.43	1.71	4.29	79.43	83.14
Basketball	28.14	92.41	25.93	71.59	60.28	2.48	2.48	10.21	23.59	84.13	81.10
David3	68.25	42.00	22.80	74.21	48.02	13.89	10.32	33.73	33.33	73.41	74.60
Girl2	37.33	15.33	22.25	7.20	38.53	15.33	4.33	10.47	6.93	0.80	80.73
Woman	18.76	27.00	14.11	93.47	84.92	20.60	16.58	93.47	22.78	60.47	86.26
Jogging	22.15	22.48	22.48	21.50	21.17	95.44	96.74	22.48	20.85	22.48	97.07
Couple	68.57	10.71	69.29	8.57	10.71	56.43	100.0	54.29	8.57	2.14	64.29
Crossing	98.33	58.33	96.67	68.33	100.0	34.17	51.67	94.17	17.50	25.00	96.67
Subway	79.43	21.71	77.71	100.0	98.85	22.86	22.86	90.86	22.29	22.29	91.43
Football1	72.97	68.92	8.11	100.0	41.89	97.30	39.19	87.84	35.14	50.00	75.73

It follows that those comparative methods have their limits in tracking object when the object motion is abrupt and rapid, and their performance is not stable enough. Our method faithfully models the object appearance and achieves the best and comparable tracking results when the motion blur and/or drastic movement of the object occur.

TABLE III
AVERAGE CENTER LOCATION ERROR. THE BEST, SECOND BEST AND THIRD BEST RESULTS ARE SHOWN IN RED, BLUE AND GREEN NUMBERS RESPECTIVELY.

Sequences	MIL	VTD	CT	DF	SCM	CXT	TLD	Struck	STC	ONNDL	Ours
Bolt	380	25	364	245	456	385	453	399	140	10	6
Basketball	104	6	70	18	53	215	269	118	75	10	10
David3	55	62	66	51	73	222	281	106	6	9	14
Girl2	55	57	113	240	126	231	382	233	371	321	23
Woman	120	109	107	9	8	73	187	4	65	9	15
Jogging	88	85	116	31	133	6	7	62	150	152	10
Couple	38	105	36	36	109	42	3	11	143	150	13
Crossing	5	21	6	22	2	24	24	3	34	39	6
Subway	7	141	12	3	4	139	160	4	159	163	6
Football1	14	8	21	2	20	3	113	5	73	10	11

E. Precision Plot and Success Plot

We select six typical video sequences (Basketball, David3, Girl2, Woman, Jogging and Couple) from the ten considered video sequences. Fig. 7 shows the precision plots for the six sequences. Obviously, our method performs the best in David3, Girl2 and Jogging, and it is still satisfactory in Basketball and Woman although it is not the best. VTD performs the best in Basketball which exhibits fast motion, but it is not favorable in the other five sequences. The MIL, CT and TLD methods work well in Couple, however, they are less effective in the other four sequences. Fig. 8 depicts the success plots on the selected six sequences, from which we can obtain a similar conclusion. It is noteworthy that the success indices of the CXT, TLD and STC methods on Basketball are not as desirable as Fig. 7, mainly because the tracking box size cannot successfully change with the object and becomes smaller and smaller, which reduces the success rate on the whole.

VI. CONCLUSION

The CNC method is proposed in this paper, in which the locality constraint on manifold is incorporated into the non-negative coding. By combining the proposed online update scheme of the codebook and classifier parameter, our tracker reduces drifts and enhances the ability to adaptively deal with appearance change in dynamic scenes. The qualitative results and the quantitative comparison with those ten state-of-the-art methods based on a set of challenging video sequences demonstrate that the proposed tracker achieves an impressive object tracking performance.

APPENDIX

The convergence proof of the proposed CNC is provided in the section. We exploit the property that the residual of the objective function in Eq. (4) is monotonically decreasing during the iterative update as shown in Eq. (10). We firstly introduce an auxiliary function $Z(h,h')$ [7], [32].

Definition 1. $Z(h,h')$ is an auxiliary function for $J(h)$, if

$$Z(h,h') \geq J(h), Z(h,h) = J(h) \tag{A.1}$$

is satisfied [7], [32].

Lemma 1. If $Z(h,h')$ is an auxiliary function for $J(h)$, $J(h)$ is non-increasing under the following update rule [7], [32]:

$$h^{t+1} = \underset{h}{\operatorname{argmin}}\, Z(h,h^t) \tag{A.2}$$

Proof. $J(h^{t+1}) \leq Z(h^{t+1},h^t) \leq Z(h^t,h^t) = J(h^t)$

Since $\mathbf{B}^{t-1}$ is formed in frame $t-1$, it is constant for the superpixel coding in frame t, we can get the following proposition.

Proposition 1. According to Eq. (6),

$$\begin{aligned}\underset{\mathbf{c}^t>0}{\operatorname{argmin}}\, J_{CNC}(\mathbf{C}^t) = \underset{\mathbf{c}^t>0}{\operatorname{argmin}} \sum_i (-\mathbf{c}_i^{t^T}(2\widehat{\mathbf{B}^{t-1^T}}\widehat{\mathbf{x}_i^t} \\ + 2\lambda\widehat{\mathbf{c}_i^{t-1}} + 2\eta\sum_{k\in\mathcal{S}_i}\mathbf{c}_k^t) + \mathbf{c}_i^{t^T}\widehat{\mathbf{B}^{t-1^T}}\widehat{\mathbf{B}^{t-1}}\mathbf{c}_i^t)\end{aligned} \tag{A.3}$$

Then we obtain the auxiliary function for $J_{CNC}(\mathbf{C}^t)$ as follow:

$$\begin{aligned}Z(\mathbf{C}^t,\mathbf{C}^{t'}) = \sum_{p,q} -2(\widehat{\mathbf{B}^{t-1^T}}\widehat{\mathbf{x}_p^t} + \lambda\widehat{\mathbf{c}_p^{t-1}} + \eta\sum_{k\in\mathcal{S}_i}\mathbf{c}_k^t)_q c_{p,q}^{t'} \\ \left(1+\log\frac{c_{p,q}^t}{c_{p,q}^{t'}}\right) + \sum_{p,q}(\widehat{\mathbf{B}^{t-1^T}}\widehat{\mathbf{B}^{t-1}}\mathbf{c}_p^{t'})_q\frac{c_{p,q}^{t^2}}{c_{p,q}^{t'}}\end{aligned} \tag{A.4}$$

Proof. For obtaining the lower bound for Eq. (A.3), we use the inequality $z \geq 1+\log z, \forall z>0$, then

$$Q = 2\widehat{\mathbf{B}^{t-1^T}}\widehat{\mathbf{x}_p^t} + 2\lambda\widehat{\mathbf{c}_p^{t-1}} + 2\eta\sum_{k\in\mathcal{S}_i}\mathbf{c}_k^t \tag{A.5}$$

$$\sum_p \mathbf{c}_p^{t^T}Q = \sum_{p,q}Q_q c_{p,q}^t \geq \sum_{p,q}Q_q c_{p,q}^{t'}\left(1+\log\frac{c_{p,q}^t}{c_{p,q}^{t'}}\right) \tag{A.6}$$

$$\sum_p \mathbf{c}_p^{t^T}\widehat{\mathbf{B}^{t-1^T}}\widehat{\mathbf{B}^{t-1}}\mathbf{c}_p^t \leq \sum_{p,q}\left(\widehat{\mathbf{B}^{t-1^T}}\widehat{\mathbf{B}^{t-1}}\mathbf{c}_p^{t'}\right)_q\frac{c_{p,q}^{t^2}}{c_{p,q}^{t'}} \tag{A.7}$$

By summing over all the bounds, we can get $Z(\mathbf{C}^t,\mathbf{C}^{t'})$, which satisfies: (1) $Z(\mathbf{C}^t,\mathbf{C}^{t'}) \geq J_{CNC}(\mathbf{C}^t)$, and (2) $Z(\mathbf{C}^t,\mathbf{C}^t) = J_{CNC}(\mathbf{C}^t)$.

Proposition 2. The iterative update of Eq. (10) is convergent.

To find the minimum of $Z(\mathbf{C}^t,\mathbf{C}^{t'})$, we take the Hessian matrix of $Z(\mathbf{C}^t,\mathbf{C}^{t'})$

$$\begin{aligned}\frac{\partial^2 Z(\mathbf{C}^t,\mathbf{C}^{t'})}{\partial c_{m,n}^t\partial c_{p,q}^t} = \delta_{m,p}\delta_{n,q}(2(\widehat{\mathbf{B}^{t-1^T}}\widehat{\mathbf{x}_m^t} + \lambda\widehat{\mathbf{c}_m^{t-1}} \\ + \eta\sum_{k=1}^S\mathbf{c}_k^t)_n\frac{c_{m,n}^{t'}}{c_{m,n}^{t^2}} + 2\left(\widehat{\mathbf{B}^{t-1^T}}\widehat{\mathbf{B}^{t-1}}\mathbf{c}_m^{t'}\right)_n\frac{1}{c_{m,n}^{t'}})\end{aligned} \tag{A.8}$$

which is a diagonal matrix with positive diagonal entries. So $Z(\mathbf{C}^t,\mathbf{C}^{t'})$ is a convex function of $\mathbf{c}_i^t$, and we can obtain the global minimum of $Z(\mathbf{C}^t,\mathbf{C}^{t'})$ by setting $\frac{\partial Z(\mathbf{C}^t,\mathbf{C}^{t'})}{\partial c_{m,n}^t} = 0$ and solving for $c_{m,n}^t$, from which we can get Eq. (10).

Fig. 6. Representative frames from ten sequences. The results obtained by those ten state-of-the-art algorithms and ours are shown in different colors: MIL in pink, VTD in purple, CT in green, DF in black, SCM in gray, CXT in blue, TLD in turquoise, Struck in orange, STC in dark red, ONNDL in cyan, and Ours in red.

REFERENCES

[1] S. Zhang, H. Yao, X. Sun, and X. Lu, "Sparse coding based visual tracking: Review and experimental comparison," *Pattern Recognit.*, vol. 46, no. 7, pp. 1772-1788, 2013.

[2] S. Zhang, H. Zhou, F. Jiang, and X. Li, "Robust visual tracking using structurally random projection and weighted least squares," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. NIPS*, 2001, pp. 585-591.

[4] X. Zhang, Y. Yang, L. C. Jiao, and F. Dong, "Manifold-constrained coding and sparse representation for human action recognition," *Pattern Recognit.*, vol. 46, no. 7, pp. 1819-1831, 2013.

[5] M. Belkin, "Problems of learning on manifolds," Ph.D. dissertation, Dept. Math., University of Chicago, Chicago, IL, 2003.

[6] S. Zhang, S. Kasiviswanathan, P. C. Yuen, and M. T. Harandi, "Online dictionary learning on symmetric positive definite manifolds with vision applications," in *Proc. AAAI*, 2015, pp. 3165-3173.

[7] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788-791, Oct. 1999.

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

[9] H. Zhang, A. Berg, M. Maire, and J. Malik, "Svm-knn: Discriminative nearest neighbor classification for visual category recognition," in *Proc. CVPR*, 2006, pp. 2126-2136.

[10] L. Bottou and V. N. Vapnik, "Local learning algorithms," *Neural Comput.*, vol. 4, no. 6, pp. 888-900, Nov. 1992.

[11] C. Cortes and V. Vapnik, "Support vector networks," *Mach. Learn.*, vol. 20, pp. 273-297, Sep. 1995.

[12] J. Lim, D. Ross, R. -S. Lin, and M. -H. Yang, "Incremental learning for visual tracking," in *Proc. NIPS*, 2005, pp. 793-800.

[13] S. Zhang, H. Zhou, H. Yao, Y. Zhang, K. Wang, and J. Zhang, "Adaptive NormalHedge for robust visual tracking," *Signal Process.*, vol. 110, pp. 132-142, May 2015.

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

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

[16] H. Grabner, C. Leistner, and H. Bischof, "Semisupervised on-line boosting for robust tracking," in *Proc. ECCV*, 2008, pp. 234-247.

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

[18] B. Babenko, M. CH. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. CVPR*, 2009, pp. 983-990.

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

[20] K. Zhang, L. Zhang, and M. H. Yang, "Real-time compressive tracking," in *Proc. ECCV*, 2012, pp. 864-877.

[21] E. G. Learned-Miller and L. S. Lara, "Distribution fields for tracking," in *Proc. CVPR*, 2012, pp. 1910-1917.

[22] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pur. Appl. Math.*, vol. 59, no. 8, pp. 1207-1223, Aug. 2006.

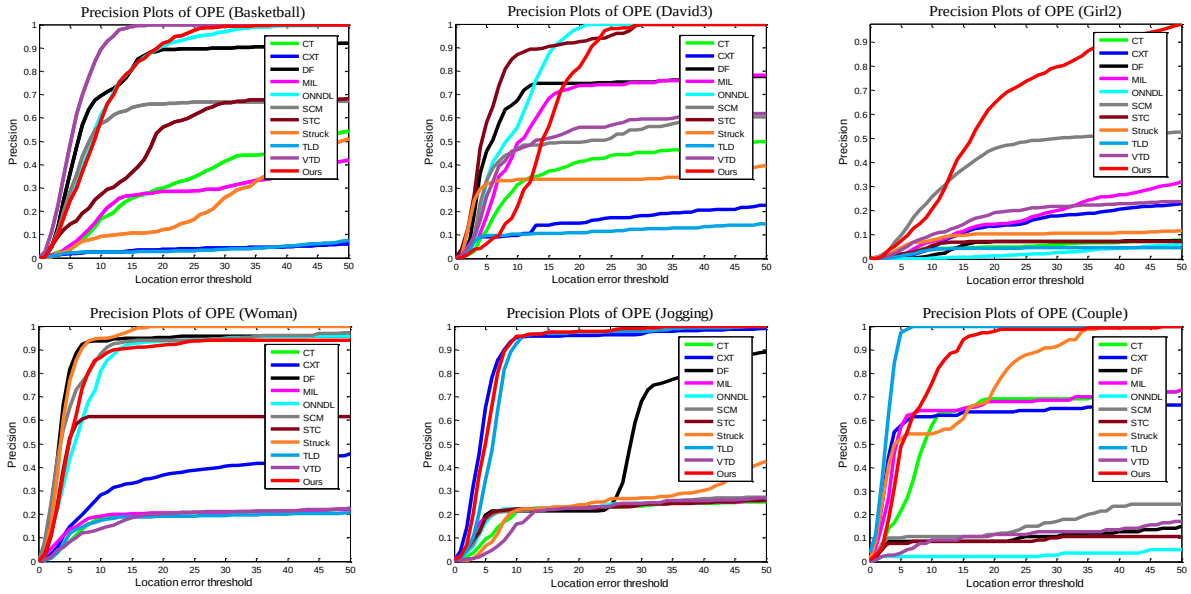

Fig. 7. Precision plots of OPE on six typical sequences.

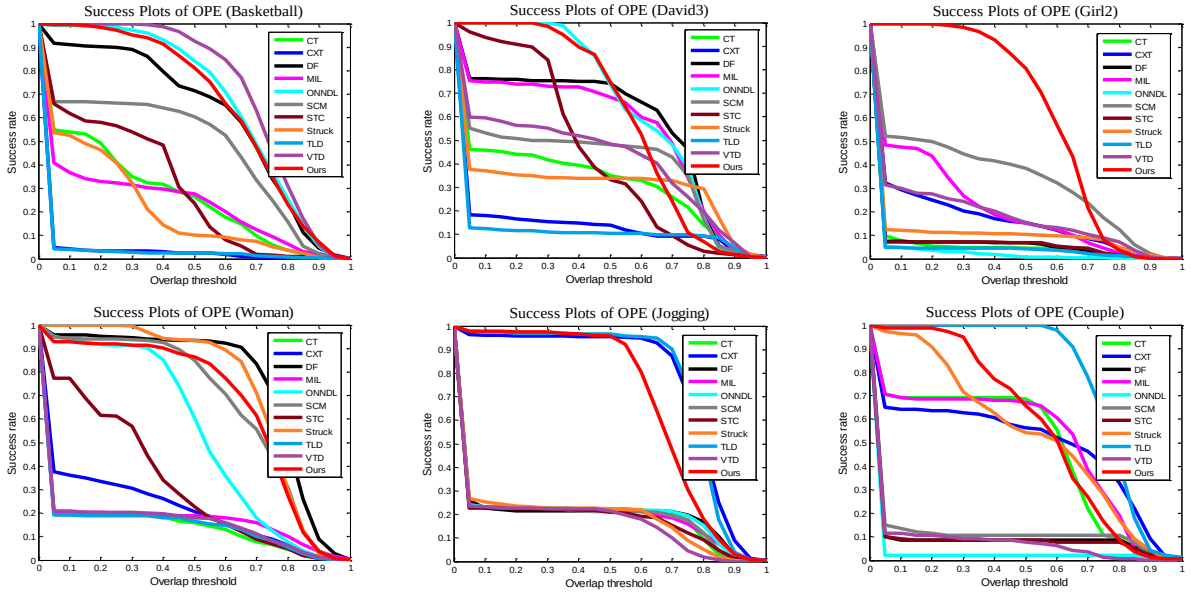

Fig. 8. Sucess plots of OPE on six typical sequences.

[23] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289-1306, Apr. 2006.

[24] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, "Robust and fast collaborative tracking with two stage sparse optimization," in *Proc. ECCV*, 2010, pp. 624-637.

[25] S. Zhang, H. Yao, X. Sun, and X. Lu, "Sparse coding based visual tracking: Review and experimental comparison," *Pattern Recognit.*, vol. 46, no. 7, pp. 1772-1788, 2013.

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

[27] S. Zhang, H. Yao, X. Sun, and S. Liu, "Robust visual tracking using an effective appearance model based on sparse coding," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 1-18, 2012.

[28] X. Mei and H. Ling, "Robust visual tracking using L1 minimization," in *Proc. ICCV*, 2009, pp. 1436-1443.

[29] 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.

[30] D. Wang, H. Lu, and M. H. Yang, "Least soft-threshold squares tracking," in *Proc. CVPR*, 2013, pp. 2371-2378.

[31] H. Liu, M. Yuan, F. Sun, and J. Zhang, "Spatial neighborhood-constrained linear coding for visual object tracking," *IEEE Trans. Ind. Inform.*, vol. 10, no. 1, pp. 469-480, Feb. 2014.

[32] C. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 45-55, Jan. 2010.

[33] N. Wang, J. Wang, and D. CY. Yeung, "Online robust non-negative dictionary learning for visual tracking," in *Proc. ICCV*, 2013, pp. 657-664.

[34] 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.

[35] H. Zhang, S. Hu, X. Zhang, and L. Luo, "Visual tracking via constrained

incremental non-negative matrix factorization," *IEEE Signal Proc. Let.*, vol. 22, no. 9, pp. 1350-1353, Sep. 2015.
[36] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290-2297, Dec. 2009.
[37] D. P. Bertsekas, *Nonlinear Programming*, Nashua, NH, USA: Athena Scientific, 1999, pp. 307-310.
[38] R. Krishnapuram and J. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98-110, May 1993.
[39] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. CVPR*, 2010, pp. 3360-3367.
[40] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. CVPR*, 2006, pp. 2169-2178.
[41] Y. Wu, J. Lim, and M. CH. Yang, "Online object tracking: A benchmark," in *Proc. CVPR*, 2013, pp. 2411-2418.
[42] W. Zhong, H. Lu, and M. -H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. CVPR*, 2012, pp. 1838-1845.
[43] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. CVPR*, 2011, pp. 1177-1184.
[44] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. CVPR*, 2010, pp. 49-56.
[45] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. ICCV*, 2011, pp. 263-270.
[46] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M. CH. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proc. ECCV*, 2014, pp. 127-141.

Xiaolin Tian is currently an Associate Professor in the Electronic Engineering School, Xidian University, Xi'an, China. He received PhD degree from Xidian University in 2008. During 2011 and 2012, he was a visiting scholar at Vision Lab, University of California, Los Angeles, USA. His current research interests are in the areas of image and video processing.

Licheng Jiao received the B.S. degree from Shanghai Jiaotong University, Shanghai, China, in 1982, the M.S. and PhD degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

Since 1992, he has been a Professor with the School of Electronic Engineering, Xidian University, Xi'an, China. He was in charge of about 40 important scientific research projects, and published more than 20 monographs and 100 papers in international journals and conferences. His research interests include image processing, natural computation, machine learning, and intelligent information processing.

Zhipeng Gan received the B.S. degree in electronic engineering from the Northwest A&F University, Yangling, China, in 2013. He is currently pursuing the M. S. degree in Xidian University, Xi'an, China.

Chaohui Wang received his PhD degree in applied mathematics and computer vision from Ecole Centrale Paris in 2011. After that he was a postdoctoral researcher at the Vision Lab of University of California, Los Angeles (Jan. 2012 $\sim$ Mar. 2013) and at the Perceiving Systems department, Max Planck Institute for Intelligent Systems (Mar. 2013 $\sim$ Aug. 2014). He is currently an assistant professor at Université Paris-Est Marne-la-Vallée, France. His research interests include computer vision, machine learning and medical image analysis.

Xiaoli Zheng received the B.S. degree in electronic engineering from the Henan University, Kaifeng, China, in 2012. She is currently pursuing the M. S. degree in Xidian University, Xi'an, China.