Robust Visual Tracking via a Collaborative Model Based on Locality-Constrained Sparse Coding

JIA HU1 AND XIAOPING FAN1,2, (Member, IEEE)
1School of Computer Science and Engineering, Central South University, Changsha 410083, China
2Laboratory of Networked Systems, Hunan University of Finance and Economics, Changsha 410205, China
Corresponding author: Jia Hu (hujia@csu.edu.cn)

This work was supported in part by the 2011 Collaborative Innovation Center for Development and Utilization of Financial and Economic Big Data Property of Hunan Province, in part by the National Natural Science Foundation of China under Grant 10.13039/501100001809, and in part by the Hunan Provincial Science and Technology Department under Grant 10.13039/501100002767.

ABSTRACT Target tracking is an important task in computer vision. Now many tracking algorithms have achieved great results. However, several challenges still hinder the development of tracking algorithms, such as abrupt motion, occlusion and so on. In order to use the feature information of the target more effectively and improve the accuracy and robustness of target tracking, a novel model is designed which is different from the previous discriminative component and generative component, and a novel discriminative-generative collaborative appearance model is presented to combine the two components in this paper. First, for the discriminative component, Locality-Constrained Sparse Coding Algorithm is proposed. In this algorithm, the objective function of the local feature of the target spatial information is determined by fusing the pyramid maximum pool and local feature histogram method. The objective function has three important parameters, which are solved by different optimization strategies. Second, for the generative component, the Histogram of Locality-Constrained Feature Algorithm is proposed. In this algorithm, the locality constraint is served to describe the spatial information of the target as a generative appearance model. Each image patch can be approximated by a linear combination of a local coordinate system formed by a dictionary whose elements are cluster centers that contain the most representative model of the target. Third, this paper designs a collaborative target tracking framework based on semi-supervised learning algorithm with locality constraint coding. The framework can quickly and robustly determine the feature information of the tracking region. The proposed algorithm is evaluated on the comprehensive test platform. The experimental results show that our method is more robust and efficient, and the precision and success rate of our algorithm are improved by 5.4% and 4.7%, respectively.

INDEX TERMS Locality-constrained, collaborative model, visual tracking, Bayesian framework.

I. INTRODUCTION

The last decade has witnessed the great success of visual tracking, especially in the last five years. As we know, visual tracking is the task of estimating the object motion trajectory and locating an object in a video, and it is one of the most fundamental problems in computer vision with a wide range of applications. A tracker is typically designed by a combination of appearance model, motion model, and update strategy. The appearance model plays a vital role in the tracking system and it is a key on how to design a robust appearance model. Meanwhile, despite recent progresses in visual tracking have yielded a steady increase in performance, visual tracking is still a challenging task due to the presence of occlusion, abrupt object motion, changing appearance, background clutter and so on [1]–[10]. One of the most crucial reasons is that the tracking algorithm has many difficulties in distinguishing between target and non-target.

All tracking algorithms can be classified into three distinct types: generative methods, discriminative methods, and collaborative methods. Generative methods [1]–[5] build a target representation and search for the region which is most similar to the target within an image frame. The methods are usually more robust to appearance variations of a target. In particular, generative methods can expediently introduce prior knowledge to adapt to the change of the target appearance. More importantly, the generative models train fast with less data. The generated model carries abundant target information,
and can easily meet the evaluation criteria and real-time requirements of target tracking when they deal with huge amounts of data information. The commonly used generating models are the Gaussian mixture model [11], the Bayesian network model [12], and the Markov model [13]. The Gaussian mixture model is a kind of parameterized probability statistical method, which represents the mixed signals such as background, small amplitude movement in background and shadow into the Gaussian mixture probability statistical model, establishes different Gaussian models for different states, uses the maximum likelihood probability to realize the background modeling, and updates the background Gaussian model in real-time by using learning factors. In theory, no matter how the observed data sets are distributed and what rules they present, they can be simulated by mixing multiple single Gauss models. In the target tracking algorithm, K Gaussian models are used to represent the characteristics of each pixel in the target. Each pixel in a new frame is matched with the Gaussian model. If the matching is successful, the point belonging to the target is judged. Literature [11] combines Gaussian mixture model (GMM) and optical flow method, in which GMM works on extracting regions of interest in complex environments while the optical flow method is used to determine tracking regions. This method has been successfully applied to continuous image sets, but due to the large computational complexity of the optical flow method, there is still much space for improvement in this paper. The algorithm based on Bayesian framework has been widely studied and achieved good results. Its general workflow is to detect different types of targets (such as background and prospect) by choosing appropriate features and judging by Bayesian rules. Literature [12] proposes a classification method based on sparse Bayesian regression and subspace clustering. Through a Bayesian statistical framework, the representation-based classification uses a more non-informative precision Hyperion approach. Experiments illustrate the effectiveness of the algorithm compared to state-of-the-art set-based methods. Markov random field is a kind of probability graph model, which is similar to the Bayesian framework in the form of dependence of variable relations. But the Markov model can be used to label a set of variables with specific dependencies, which Bayesian can not do (such as ring dependencies). Literature [13] proposes an uncertain Markov chain Monte Carlo sampler and uses it to detect the optimal distribution of samples to reduce the uncertainty of candidate targets. However, there are several problems remain to be solved. For example, when multiple samples are collected at the current target location, it is likely to lead to drift as the appearance model needs to adapt to these potentially misaligned examples; and these generative algorithms do not use the background information which is likely to improve tracking stability and accuracy.

On the other hand, discriminative methods [6]–[10], [14] pose the tracking problem as a binary classification task to find the decision boundary for separating the target object from the background. In general, such methods use information from both the target and the background for classification. So it can distinguish the target from its surroundings. And when the discriminative learning function is established, the prediction rate for classifying the new sample is very fast. A classifier is often used in target detection, that is, to detect the location of the target in the background. The target tracking can be achieved by getting the position of the target in the background. Various shape information of the target is included in the positive and negative samples needed for training the classifier. The tracking framework based on a discriminative model is mainly divided into two parts: off-line learning and on-line tracking. Off-line learning extracts features for the target region, forms sample samples, and obtains the appearance model of discriminative target-background through training classifier. In the tracking stage, when the new target area arrives, the feature is extracted, the test sample is formed, and the target-background confidence map of the sample area is identified by the classifier, so as to complete a tracking task. Common classifiers are support vector machine (SVM) and Adaboost. SVM classifier: The principle of SVM is based on linear partition. In linearly separable space, according to the principle of minimizing structural risk, we can find a hyperplane to distinguish two or more kinds of samples and find support vector according to the normal vector of the hyperplane. In linearly inseparable space, the sample space is mapped to a high-dimensional or infinite-dimensional complete space, so that the linearly inseparable problem in low-dimensional space can be transformed into a linearly separable problem in high-dimensional space. In literature [15], the method of fuzzy least squares support vector machine is developed, and the effective implementation of the closed solution is given by using the original form, dual form, and kernel form of the method. Besides, the least-squares regression model is constructed to control the adaptive template updating to maintain the robustness of the appearance model. Literature [16] proposes a structured support vector machine algorithm for graph rules by combining manifold learning and structure learning and constructs a robust appearance model using this algorithm. However, there are several problems too. First, the background varies continuously, hence making it difficult for adequately distinguishing a target from the background. Second, as a result of the need for a large number of sample data, the training speed of the discriminative models is relatively slow. So, when the limited data is available, discriminative models cannot perform better than generative models.

The target tracking algorithm of the collaborative apparent model is based on the generative and discriminative model, which synthesizes the advantages of the two apparent models, and makes up for each other’s disadvantages by using their advantages, to achieve a more robust and efficient tracking algorithm. Dinh and Medioni [17] proposed a joint training framework of the discriminative model and generative model for partial occlusion processing and updated the parameters of the discriminative model and generative model online.
with the non-occlusive part. Zhong et al. [18] combined with the sparse theory, constructed the sparse based discriminative classifier (SDC) and the sparse based generative model (SGM) for the target tracking problem. In the particle evaluation part, the SDC score and SGM score of each particle were considered comprehensively. The algorithm can deal with the problem of plane rotation, non-plane rotation, illumination change, and occlusion. Although the hybrid target tracking technology combines the advantages of the two tracking algorithms, it is difficult to meet the real-time requirements in the actual scene due to the complexity model and the high computational cost of the algorithm.

Due to the good robustness to image damage and partial occlusion, sparse learning based tracking algorithm has become a hot research direction in the field of target tracking in recent years [19]–[22]. By combining particle filter theory and sparse learning [23], [24], the problem of target tracking is expressed as a process that under the framework of Bayesian theory, the prior probability of target state is known and the maximum posterior probability of target state is continuously solved after obtaining the new observation value of target. The target tracking algorithm based on sparse learning consists of four modules: particle filter, (joint) sparse learning, template updating, and occlusion detection. The rationale of sparse representation based methods is that a possible target candidate is a linear combination of sparse atoms in a dictionary. This type of method has been used in the \( \ell_1 \)-tracker [19] where an object is modeled by a sparse linear combination of target templates and trivial templates. And the templates are dynamically updated according to the similarity between the tracking result and the template set. Considering that the running speed of these methods is often limited by solving the \( \ell_1 \) norm optimization problem, many machine learning researchers have used locality-constrained linear coding (LLC) [25] to achieve a similar sparsity property as using a few anchor points [26] or the k-nearest neighbors (kNN) selection scheme with high computational efficiency. In addition, this coding method can also ensure that similar candidate samples are associated with similar coefficient vectors or share similar dictionaries so that the appearance information carried by their local dictionaries is used synthetically. Therefore, this encoding algorithm is a component of some visual tracking framework [27]–[29]. Although this framework has been successful, there is a typical shortcoming in this kind of trackers. The establishment of the frame appearance model is likely to be a nonlinear problem, and the change of appearance also presents a nonlinear distribution, and the candidate samples are expressed by the template by the linear representation scheme. Therefore, such a tracker framework is difficult to fully understand the appearance of the target, thereby aggravating the accumulation of errors.

According to the above discussion, we notice that the data of the first frame is labeled in the target tracking task, while the other data is not labeled, so we adopt the semi-supervised learning method [30]. Combining with the above analysis of locality-constrained coding, many trackers use boxes to represent the tracking results, and part of the structure of the tracking box is not the element of the target, so the discriminative model and the generative model are implemented by locality-constrained coding method. The main contributions of this paper are presented as follows:

1) For discriminative components, a locally constrained sparse coding algorithm is proposed. The algorithm combines the pyramid maximum pooling method with the local feature histogram to determine the target function of the local feature of the target spatial information. The objective function has three important parameters, which are solved by different optimization strategies.

2) For generative components, a histogram based local constraint feature extraction algorithm is proposed. The algorithm uses local constraints to describe the spatial information of the target as a generated appearance model. Each image patch can be approximated by a linear combination of local coordinate systems composed of a dictionary whose elements are the cluster centers of the most representative models of the target.

3) A collaborative target tracking framework based on local constraint coding semi supervised learning algorithm is designed. The framework can determine the feature information of tracking region quickly and stably.

II. APPEARANCE REPRESENTATION MODEL

A. LOCALITY-CONSTRAINED SPARSE CODING

1) CODING DESCRIPTORS IN LOCALITY-CONSTRAINED LINEAR CODING

Wang et al. [25] proposed the LLC method, which utilizes the locality constraints to project each descriptor in place of the sparsity regularization term in the sparse representation. Let \( X \) be a set of \( D \)-dimensional local descriptors extracted from an image, \( X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{D \times N} \). Given a dictionary with \( M \) entries, \( B = [b_1, b_2, \ldots, b_M] \in \mathbb{R}^{D \times M} \) and the set of codes for \( X \), \( C = [c_1, c_2, \ldots, c_N] \in \mathbb{R}^{M \times N} \), the LLC code uses the following criteria:

\[
\begin{align*}
\min_C \quad & \sum_{i=1}^{N} \| x_i - Bc_i \|_2^2 + \lambda \sum_{i=1}^{N} \| d_i \odot c_i \|_2^2 \\
\text{s.t.} \quad & 1^T c_i = 1, \quad \forall i
\end{align*}
\]

where \( \odot \) denotes element-wise multiplication operator (for vectors), \( \lambda \) is the regularization parameter, \( M \) is the number of atom in the dictionary, and \( d_i \) is the locality adapter used to measure the Euclidean distance between the data instance \( x_i \) and the dictionary atom. We define \( d_i = dist(x_i, B) \), where \( dist(x_i, B) = [dist(x_i, b_1), dist(x_i, b_2), \ldots, dist(x_i, b_M)]^T \). The constraint \( 1^T c_i = 1 \) follows the shift-invariant requirement of the LLC code.

Note that, firstly, the LLC code in Eq.(1) focuses on locality constrained, rather than sparsity. Locality can force the coding results to be sparse, but sparsity may not satisfy locality [25]. In this respect, locality constrained is more
significant than sparsity constrained. Due to locality constrained, the regularization term in LLC is smoother than that of $\ell_1$ norm in SC (SC: Sparse Coding). To meet the sparsity coding in SC, the results of the similar pixel blocks are likely to be different, while the over-completeness of the dictionary has exacerbated the difference [31]. LLC can ensure the similar pixel blocks to get similar coding, accordingly ensure the similarity of the reconstructed pixel blocks and locality smoothness. Second, sparse coding requires an optimization algorithm to iterate, which leads to high computational consumption, while the LLC algorithm has an analytical solution that can reduce computational consumption and speed up the operation.

2) LOCALITY-CONSTRAINED SPARSE CODING ALGORITHM

Since there is no label data involved in the dictionary learning process, the LLC algorithm belongs to the unsupervised learning algorithm. In the tracking process, the ground truth box of the labeled image patch always comprises discriminative information separating the target object from the background, while the rest of the image patches collected easily are unlabeled data instances which provide a means for the classifiers to improve the capability of generalization. So this section elaborates the Locality-Constrained Sparse Coding Algorithm (LCSC) using a semi-supervised learning approach in detail.

Classification of the targets and backgrounds in sparse coding, it is necessary to consider both the labeled data and the underlying classifier in the objective function. To improve the underlying classifier, it is necessary to impose additional constraints on dictionary learning. Given a training dataset \((x_i, y_i)_{i=1}^N\), where \(x_i \in \mathbb{R}^N\), \(y_i\) denoting the label of \(x_i\), the goal is to learn a feature representation by exerting extra constraints such that the classifier can benefit from the new representation. We formulate the objective function as a joint problem of representation loss and discrimination loss as below:

\[
\min_{B, C, W} \left( \sum_{i=1}^N \|x_i - Bc_i \|^2_2 + \lambda \sum_{i=1}^N \|d_i \odot c_i \|^2_2 \right) + \gamma \left( \tau \sum_{i=1}^N \sum_{j=1}^M \max(0, 1 - y_i(w_j^\top c_i + a)) + \frac{1}{2} \sum_{k=1}^M \|w_k \|^2_2 \right)
\]

(2)

where the first term is the representation loss function and the second term is the discrimination loss function, and \(\tau\) is constantly controlling the strength of representation loss, and constant variable \(\gamma\) balance the cost function, and \(1^\top c_i = 1, \forall i\), and \(W = [w_1, w_2, \cdots, w_M]\). It is pointed out that the discrimination loss function is not a convex function and the number of solutions increases exponentially with the number of training samples, so is very difficult to solve. Note that the goal of the function of discrimination loss is to find out the classification error, so this paper uses the hinge loss function to model the error.

\[
\min_{B, C, W} \left( \sum_{i=1}^N \|x_i - Bc_i \|^2_2 + \lambda \sum_{i=1}^N \|d_i \odot c_i \|^2_2 \right) + \gamma \left( \tau \sum_{i=1}^N \sum_{j=1}^M \max(0, 1 - y_i(w_j^\top c_i + a)) + \frac{1}{2} \sum_{k=1}^M \|w_k \|^2_2 \right)
\]

(3)

The proposed objective function in Eq.(3) includes three variables: \(c\), \(B\) and \(W\). So the task now is to minimize Eq.(3) for the three parameters. It is obvious that the function is a non-convex problem in nature, but it is convex when only one variable while the other two variables are fixed.

3) LEARNING PARAMETERS

(1) Update Rule for \(B\): Updating dictionary \(B\) is only related to the representation term when fixing \(W\) and \(c\), so the function becomes Eq.(1). As can be seen from Eq.(1), the solution process is the implementation of feature selection actually: the feature descriptors to be reconstructed tends to select the nearest distance from the dictionary to form a local coordinate system. Therefore, there is an approximation algorithm for LLC fast encoding [25]. The approximation LLC encoding algorithm can not only preserve the local feature, but also guarantee the sparsity of the coding, but the final performance is not much different from the optimization model of Eq.(1). Instead of solving Eq.(1), we can use the K-Means clustering algorithm to obtain the K (K<D<M) nearest neighbors of \(x_i\) as the local bases \(B_i\) and generate a dictionary \(B\). Then we loop through all the training descriptors to update \(B\) incrementally. And the process consists of three stages: dictionary initialize, coding with bias and update dictionary.

1) Dictionary initialize: Some random partial samples of all feature descriptors are used as a set for dictionary learning. Then we can utilize the K-Means clustering algorithm to form a dictionary \(B\), and select the K (K<D<M) nearest neighbors as its local bases \(B_i\).

2) Coding with bias: Compute the Euclidean distance between \(x_i\) and \(b_j\), and coding with \(B\) by using Eq.(1) compute \(c_i\);

3) Update dictionary: Using local bases to restructure the feature descriptors \(x_i\), that is, optimizing the model of Eq.(4) to get the bias coding \(c_i \in \mathbb{R}^K\) (The solution using the Lagrange multiplier method can resolve.):\n
\[
\min_B \sum_{i=1}^K \|x_i - \bar{c}_iB_i \|^2 \quad s.t. \quad 1^\top \bar{c}_i = 1, \quad \forall i
\]

(4)

Then set the value of final coding \(c_i^* \in \mathbb{R}^M\) from 1 to K dimension to \((c_{i1}, c_{i2}, \cdots, c_{iK})\), and set the other dimension to 0. Then update the dictionary, and the above process is illustrated in Alg.(1).

(2) Update Rule of \(c\): The job of this section is to update \(c\) with fixing \(B\) and \(W\). Therefore, the objective function can
be rewritten as follows:

\[
\min_{c_i} \left( \sum_{i=1}^{N} \|x_i - Bc_i\|_2^2 + \lambda \sum_{i=1}^{N} \|d_i \odot c_i\|_2^2 \right) + \gamma \left( \tau \sum_{i=1}^{N} \sum_{j=1}^{M} \max(0, 1 - y_j(w_j^T c_i + a)) \right)
\]

where \(d_i \odot c_i = \text{diag}(d_i)c_i\) and \(\text{diag}(d_i)\) is a diagonal matrix in which the non-zero elements correspond to the entries of vector \(d_i\) and \(A = \text{diag}^2(d_i)\). Further,

\[
J(c_i) = \|x_i - Bc_i\|_2^2 + \lambda c_i^T \Lambda c_i \]

\[+ \gamma \tau \sum_{j=1}^{M} \max(0, 1 - y_j(w_j^T c_i + a)) \]

Since the objective function \(J(c_i)\) is convex, an optimal solution can be computed by setting the derivative of \(J(c_i)\) with respect to \(c_i\) to be zero.

\[
\frac{\partial J(c_i)}{\partial c_i} = B^T Bc_i - B^T x_i + \lambda \Lambda c_i - \gamma \tau \sum_{j=1}^{M} y_jw_j^T c_i = 0
\]

In this paper, we take into account both the labeled and unlabeled samples, which improves the generalization of dictionaries in the algorithm. Labels are unavailable for unlabeled samples, thus, an optimal value of \(c_i\) can be estimated using,

\[
\hat{c}_i = (B^T B + \lambda \Lambda)^{-1}(B^T x_i + Z)
\]

If there is a label for the sample, then \(Z = \gamma \tau \sum_{j=1}^{M} y_jw_j^T\), otherwise \(Z = 0\).

(3) Update Rule for \(W\): In the proposed framework, only the discrimination loss term involves \(W\). We assume the update quantity along the coordinate of component \(w_i\) and then the objective function can be rewritten as the form presented in Eq.(3).

\[
\min_{w} \gamma(\tau \sum_{i=1}^{N} \sum_{j=1}^{M} \max(0, 1 - y_j(w_j^T c_i + a)) + \frac{1}{2} \sum_{k=1}^{M} \|w_k\|_2^2)
\]

When updating \(w_i\) individually, the other vectors are fixed. The optimization of \(w\) can be achieved by using coordinate descent, in which only one coordinate is changed at a time. And the original problem can be decomposed into several subproblems. Thus, the optimization problem in Eq.(9) can be refined into,

\[
h_j(w_i) = h(w_i + se^T) = \gamma(\frac{1}{2}\|w_i + se^T\|^2 + \tau \sum_{r=1}^{N} \max(0, 1 - y_j(w_i + se^T)^T c_r))
\]

where \(h\) is a subproblem of the Eq.(9) with respect to \(w_i\), and \(s\) is a scale representing the update quantity along with the \(j\)-th component of \(w_i\), and \(e_j\) denote the \(j\)-th element is 1 and the other elements are 0.

In the set \(I(w_i + se_j)\), \(s\) does not change at any time. Thus, \(h(w_i)\) is quadratic [32]. Since the derivative of \(h\) of orders higher than two are zeros, the second-order Taylor expansion of \(h(w_i + se^T)\) can be represented as the form listed in Eq.(11),

\[
h(w_i + se^T) = h(w_i) + h'(w_i)s + \frac{1}{2}h''(w_i)s^2 \]

Therefore, we can solve the quadratic optimization with the Newton method [33] as in Eq.(12), in which \(h'(w_i)\) and \(h''(w_i)\) are presented in Eq.(13).

\[
s = \frac{-h'(w_i)}{h''(w_i)} \]

Although \(h'(w_i)\) is not differentiable at \([s(1-y_iw_i^T c_i)] = 0\) for some \(i\), the generalized second derivative can be defined as follows [34].

\[
\begin{cases}
  h'(w_i) = w_{ji} - 2\tau \sum_{r \in I(w_i + se_j)} y_r c_r (1 - y_iw_i^T c_i) \\
  h''(w_i) = 1 + 2\tau \sum_{r \in I(w_i + se_j)} c_r^2
\end{cases}
\]

\[
I(w_i) = \{i | (1 - y_iw_i^T c_i) > 0\}
\]

B. HISTOGRAM OF LOCALITY-CONSTRAINED FEATURE

1) HISTOGRAM GENERATIVE

For simplicity, this section also uses locality constraints to describe the spatial information of the target as a generative appearance model [35]. So this section elaborates a Histogram of Locality-Constrained Feature Algorithm (HLCF). We perform an overlapped sliding window scheme to sample \(M\) image patches for each candidate within a normalized region. Each image patch is converted into a vector \(p_i \in \mathbb{R}^{r \times 1}\), where \(r\) is the size of the patch, so \(P = [p_1, p_2, \cdots, p_M] \in \mathbb{R}^{r \times M}\), as shown in the Fig.1.

First, we generate a dictionary \(D = \{d_1, d_2, \cdots, d_J\} \in \mathbb{R}^{r \times J}\) in the paper, where \(J\) represents the number of cluster
centers obtained by clustering the patches sampled from the first frame. These dictionary elements are cluster centers obtained by the K-means clustering method, which contains the most representative model of the target. In fact, the dictionary forms a local coordinate system. In this way, each image patch can be approximated by a linear combination of the local coordinate system. Then, set \( \mathbf{b}_i = [\beta_i, \beta_2, \cdots, \beta_{ij}]^\top \) as a linear representation coefficient of \( p_i \), then the coefficient vector \( \mathbf{b}_i \) can be computed by the following formula,

\[
\beta_{ij} = \begin{cases} 
\frac{1}{Z} \exp(-\|p_i - d_j\|^2) & \text{if } d_j \in \mathcal{N}_k(p_i) \\
0 & \text{otherwise}
\end{cases} \quad (14)
\]

where \( Z \) is the normalization factor to ensure \( \sum_{j=1}^{I} \beta_{ij} = 1 \), and \( \mathcal{N}_k(p_i) \) denotes the set of \( k \) nearby dictionary elements of \( p_i \). Then, the coefficient vectors is connected into a matrix to form a histogram \( H = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_M] \in \mathbb{R}^{I \times M} \) to represent the candidate. However, this method is susceptible to deformation, noise, rotation, and translation. To solve this problem, we use the pyramid max-pooling method to modify the codes of final representation.

### 2) PYRAMID MAX POOLING

In the literature [36], the average pool mechanism is effective when histogram formation is used. However, this strategy may miss the spatial information of each image patch, and the maximum pool mechanism can ensure that the spatial information of the image patch is contained in the histogram. Spatial pyramid maximum pooling method has the following three advantages [37]: firstly, it can solve the defects caused by different sizes of input image patches; secondly, the features of feature map are extracted from different point view and fuse again, which shows the robustness of the algorithm; thirdly, it can improve the accuracy of histogram features.

The basic idea of the spatial pyramid is to carry out a series of fine grid divisions to feature space. In order to construct the space pyramid, firstly, the image is divided into \( 2^l \times 2^l \) subregions at different scales \( l = 0, 1, \cdots, L - 1 \), as shown in the Fig.2.

Then \( H \) is encoded into local features:

\[
\phi^l_c = \max(f^l_i) \quad c = 1, 2, \cdots, c(l) \quad (15)
\]

where \( c(l) \) indicates the number of subregions in the \( l \)-level spatial pyramid, and each \( f^l_i \) is composed of several elements in two-dimensional \( H \), as shown in Fig.2. Finally, we concatenate all the feature vectors to form a pyramid representation for each candidate image \( \phi = [\phi_0^1, \phi_1^1, \phi_2^1, \cdots, \phi_{c(1)}^1, \cdots, \phi_0^l, \phi_1^l, \phi_2^l, \cdots, \phi_{c(l)}^l] \). The maximum pooled feature discovers can recover the spatial information of the local appearance coding, and thus is more effective and robust than the coding histogram for appearance representation [35].

### III. TRACKING FRAMEWORK WITH COLLABORATIVE MODEL

#### A. BAYESIAN FORMULATION

In this work, the visual tracking problem is cast as a Bayesian inference task using a Markov model with hidden state variables [38]. Given the observed image patches \( o_{1:t} = \{o_1, o_2, \cdots, o_t\} \) at the \( t \)-th frame, the aim is to estimate the hidden state variables \( x_t \) of the target recursively,

\[
p(x_t | o_{1:t}) \propto p(o_t | x_t) \int p(x_t | x_{t-1}) p(x_{t-1} | o_{1:t-1}) dx_{t-1} \quad (16)
\]

where \( p(x_t | x_{t-1}) \) is called the motion model that describes the state transition between consecutive frames, and \( p(o_t | x_t) \) denotes the appearance model that evaluates the likelihood of an observed image patch. Finally, the state \( x_t \) is estimated by the maximum a posteriori estimation,

\[
\hat{x}_t = \arg \max_{x_t} p(o_t | x_t) p(x_t | o_{1:t}) \quad (17)
\]

where \( x^i_t \) is the \( i \)-th sample of the state \( x_t \).

**Motion Model:** In the Bayesian framework, an affine transformation with six parameters is used in the model.
\( p(\mathbf{x}_t | \mathbf{x}_{t-1}) \). Let \( \mathbf{x}_t = [x_t, l_t, \theta_t, s_t, \delta_t, \psi_t] \) is the hidden state variables, where \( x_t, l_t, \theta_t, s_t, \delta_t, \psi_t \) indicate translations, rotation angle, scale, aspect ratio, and skew at time \( t \) respectively. The state transition is formulated by random walk, i.e. \( p(\mathbf{x}_t | \mathbf{x}_{t-1}) = N(\mathbf{x}_t; \mathbf{x}_{t-1}, \Sigma) \), where \( \Sigma \) is a diagonal covariance matrix whose elements are the covariance of the affine parameters [39], [40].

**Appearance Model:** In this paper, our main work is to design an efficient likelihood function \( p(o_t | \mathbf{x}_t) \) which can find the most similar image patch with the appearance model. A robust appearance model can not only accurately describe the appearance of the target, but also can distinguish the target from the background. To achieve this goal, we employ a collaborative strategy appearance model: discriminative model \( D \), generative model \( G \) and collaborative strategy \( C \). And see section III-B, III-C, and III-D for details.

### B. THE DISCRIMINATIVE MODEL WITH LCSC

In the beginning, we sampled \( N \) image patches around the target location as a template set, and the target patch is the first template in set. The dictionary \( B^r \) of template set can be work out by Eq.(3). After learning the dictionary \( B^r \), it is possible to compute the coefficient vector \( c^r \) for each element of the template set over \( B^r \) as coding. Therefore, we get the corresponding dictionary and encoding of the target template. Then given a candidate sample \( x_t \), we first compute the reconstruction error between the candidate and the template via \( B^r \) and \( c^r \). If a candidate has a small reconstruction error on the target template set, the candidate is likely is to be a target; otherwise, it is likely to be the background. And then apply the following criterion to estimate its likelihood,

\[
\mathcal{L}_D(x) \propto \exp \left( -\| x^t - B^r c^r \|^2 \right) \quad (18)
\]

**Template Update:** The template set is composed of the target and image patches around it. Because the purpose of the discriminative model is to distinguish between the target and the background, the target must be sure that it is properly marked, and the target of the first frame is accurately marked. Thus the target template of the first frame remains unchanged, and the other templates of the template set are updated from low to high according to the likelihood value.

**Dictionary Update:** While the template set is updated, dictionary \( B \) is expected to be updated online after obtaining the new tracking result at the \( t \)-th frame. For the current training template set \( X_{new} = \{x_1, x_2, \ldots, x_{new1}, x_{new2}, \ldots, x_{newL} \} \), the update of the dictionary \( B \) can be implemented by Alg.(1).

### C. THE GENERATIVE MODEL WITH HLCF

In this paper, the histogram model is used as the generative model. And the local information is represented by gray features. According to II-B, a dictionary \( D \) is generated in the first frame. And we sample \( M \) image patches in each candidate of the subsequent frame, then the coefficient vectors are worked out by Eq.(14). Then the coefficient vectors are connected into a matrix to form a histogram \( H \) to represent the candidate.

To represent the histogram of image patches more robustly, we adopt the spatial pyramid maximum pooling method. According to II-B.2, the image is divided into \( 2^l \times 2^l \) sub-regions at different scales, \( l = 0, 1, 2, 3 \) in experimental. The \( H \) in encoded into local features by Eq.(15), indicated \( \phi \).

Because of the effectiveness of the histogram intersection function, we use it to compute the similarity between the candidate and template [18]. And the likelihood function as follows,

\[
\mathcal{L}_G(x) \propto \exp \left( \sum_{j=1}^{J \times M} \min(\phi_j, \psi_j) \right) \quad (19)
\]

where \( \psi \) is the template histogram. Note that the two histograms need to be normalized in advance and normalized to \((0, 1]\).

**Update scheme:** To describe the change of appearance more accurately, a partial updating strategy is adopted in this paper. In the entire image sequence, the dictionary \( D \) is calculated by the first frame and remains unchanged during the tracking process, so the dictionary does not deteriorate by the result of the update failure. Therefore, we only need to update the pooled features of the template to capture changes in appearance. The update process is described below: In the calculation of \( \mathcal{L}_G(x) \), record every sum term, denoted as \( \kappa \).

If the value of \( \kappa \) is \( \phi_j \), then \( \psi_j \neq \phi_j \), the template histogram is updated, otherwise \( \psi_j \) remains unchanged.

### D. THE COLLABORATIVE MODEL

To take advantage of the discriminative model and the generative model and overcome their shortcomings, this paper proposes a collaborative appearance model effectively, which incorporates information of the discriminative model and generative model based on locality constraints. Thus, a more robust and efficient likelihood function is generated by the multiplication mechanism. For the candidate target \( c \), the joint likelihood function is defined as:

\[
p(o^c_t | s^c_t) = \mathcal{L}_D(x_c) \cdot \mathcal{L}_G(x_c) \quad (20)
\]

where \( \mathcal{L}_D(x_c) \) and \( \mathcal{L}_G(x_c) \) are formulated in Eq.(18) and Eq.(19) respectively. The target candidate corresponding to the maximum value of \( p(o^c_t | s^c_t) \) is selected as the result of the tracking.

### IV. EXPERIMENTAL RESULTS

#### A. EXPERIMENTAL SETUP

To verify the effectiveness of the proposed tracking algorithm thoroughly in this paper, we tested the performance of the proposed algorithm and the other eight state-of-the-art tracking algorithm of non deep learning method on Benchmark [41] (abbreviated as OTB). To be fair, this paper uses the source code and parameter settings provided by Benchmark. Since each algorithm has certain randomness, each algorithm runs 5 times on the test set and takes the average result as
the final data. The proposed algorithm is implemented in MATLAB and runs an average of 15 frames per second with a 3.5GHz CPU.

In the algorithm of this paper, the target region is normalized to 32 × 32. There are 81 local image patches (size: 16 × 16) in the normalized target region extracted through sliding windows overlapped, in which the step size is two pixels. And a scale-invariant feature transform (SIFT) feature descriptor is extracted from each image patch. In general, K value can be confirmed by soft clustering or hard clustering technique [42], [43]. In this paper, K value in K-means and SIFT is used to generate the dictionaries in the LCSC algorithm and spatial pyramid scales, e.g. L = 3, so \(2^L \times 2^L = 1 \times 2 \times 2 \times 4 \times 4\). And other parameters \(\lambda = 3, \gamma = 0.2, \tau = 5\). In addition, to achieve a balance between effectiveness and speed, the proposed tracker will be updated every 5 frames to learn a new dictionary. For the sake of comparing fairly, the values of the affine parameters are fixed to \([\omega/4, h/4, 0.04, 0.01, 0.0005, 0.0001]\), where \(\omega\) and \(h\) represent the width and height of the target in the first frame respectively.

Our tracker algorithm compared with eight other state-of-the-art trackers of non deep learning method: LCT [3], TGPR [4], SAMF [5], DSST [6], KCF [7], Struck [8], TLD [9], and SCM [18].

The following Table 1 shows the key differences between the different previous algorithms and our method.

### TABLE 1. The key differences between the different previous algorithms and our method.

| Tracker | Key Attributes | fps |
|---------|----------------|-----|
| LCT     | This method uses translation correlation filtering, scale cor. & relation filtering and correlation filtering of detection target confidence | 27.4 |
| TGPR    | This method directly analyze the probability of target appearance & asce exponentially related to the confidence of a classifier by using Gaussian Processes Regression | 0.7 |
| SAMF    | The method detects the target’s translation and scale transform. | |
| DSST    | The method first detects the best translation of the target, then detects the best scale, and detects 33 feature scales. | 25.4 |
| KCF     | The method extends the HOG of multichannel gradients, and processes data in frequency domain through Fourier transform with high speed. | 39.1 |
| Struck  | The method uses Haar feature and kernel structured output. | 10.0 |
| TLD     | The method uses the tracking by detection idea and it is a typical representative of long term. | 21.7 |
| SCM     | The method develop a sparsity-based discriminative classifier. | 0.4 |
| Ours    | Design a novel model which is different from the previous discriminative component and generative component, and presents a novel discriminative-generative collaborative appearance model to combine the two components. | |

B. QUANTITATIVE EVALUATION

1) PRECISION PLOT

One widely used evaluation metric on tracking precision is the center location error (CLE) which is defined as the average Euclidean distance between the center locations of the tracked targets and manually labeled ground truths [41],

\[
CLE = \frac{1}{N} \sum_{i=1}^{N} \| T_{center}^i - gt_{center}^i \|
\]

where \(\| \cdot \|\) represents Euclidean distance, \(N\) is the total number of frames, \(T_{center}^i\) is the center location of tracking result at \(i\)-th frame and \(gt_{center}^i\) is the true center location. The smaller the average center location error of a video sequence is, the more accurate the tracking is [44].

However, for a certain tracking algorithm, there may be one or several frames missing the tracked target, then the output location can be random and the average error value may not measure the tracking performance correctly. Therefore, the precision plot [45] has been adopted to measure an overall tracking performance. A different error threshold is set. For each error threshold, the number of frames in the video for which the center location error is less than the threshold are counted, and the percentage of this number of frames relative to the total number of frames is obtained as an evaluation value corresponding to the current threshold value, as

\[
\text{Precision}(\alpha) = \frac{1}{N} \sum_{i=1}^{N} \theta(\alpha)
\]

\[
\theta(\alpha) = \begin{cases} 
1 & \text{dis}(T_{center}^i, gt_{center}^i) \leq \alpha \\
0 & \text{dis}(T_{center}^i, gt_{center}^i) > \alpha 
\end{cases}
\]

where CLE threshold \(\alpha \in [0, 100]\), and \(\theta(\cdot)\) is an indicator function. As the representative precision score for each tracker, we use the score for the threshold \(\alpha = 20\) pixels [46].

Fig. 3 shows the results under one-pass evaluation (OPE), spatial robustness evaluation (SRE) and temporal robustness evaluation (TRE) using distance precision. Overall, the proposed algorithm performs favorably against state-of-the-art methods in SRE and TRE. We present the quantitative comparisons of distance precision rate at 20 pixels.

2) SUCCESS PLOT

Another evaluation criterion is the overlap rate of the tracking bounding box. Give the tracked bounding box \(r_t\) and the ground truth bounding box \(r_a\), the overlap rate is defined as,

\[
\text{overlap}_\text{region} = \frac{|r_t \cap r_a|}{|r_t \cup r_a|}
\]

where \(\cap\) and \(\cup\) denotes the intersection and union of two regions respectively, and \(|\cdot|\) denotes the number of pixels in the region. To measure the performance on a sequence of frames, we count the number of successful frames whose overlap rate is larger than the threshold \(\lambda\). The higher the overlap rate is, the better the tracking effect is.

Similar to the precision, to more accurately assess the tracking algorithm in the video sequence, the threshold \(\beta \in [0, 1]\) is set, and the percentage of frames in which the overlap...
rate is greater than the threshold is calculated, as

$$\text{Success}(\beta) = \frac{1}{N} \sum_{i=1}^{N} \eta(\beta)$$

$$\eta(\beta) = \begin{cases} 1 & \text{overlap}_{\text{region}} \geq \beta \\ 0 & \text{overlap}_{\text{region}} < \beta \end{cases}$$

The success rate is more comprehensive and accurate than the overlap rate because it evaluates the tracking results for the entire video sequence at different thresholds. However, using one success rate value at a specific threshold (e.g., $\beta = 0.5$) for tracker evaluation may not be fair or representative. Thus we use the area under the curve (AUC) of each success plot, which can also be seen as the area above the X-axis and under the success plot.

The performance comparisons of the success rate are shown in Fig.4 shows the results under one-pass evaluation (OPE), spatial robustness evaluation (SRE), and temporal robustness evaluation (TRE) using the overlap success rate. Overall, the proposed algorithm performs favorably against the state-of-the-art methods of non deep learning in all three metrics: OPE, SRE, and TRE.

3) ATTRIBUTE-BASED EVALUATION

We further analyze the tracker performance under different video attributes (as following Table 2) [41]. Fig.5 shows the OPE for eleven main video attributes with distance precision rate is greater than the threshold is calculated, as

$$\text{Success}(\beta) = \frac{1}{N} \sum_{i=1}^{N} \eta(\beta)$$

$$\eta(\beta) = \begin{cases} 1 & \text{overlap}_{\text{region}} \geq \beta \\ 0 & \text{overlap}_{\text{region}} < \beta \end{cases}$$

The performance comparisons of the success rate are shown in Fig.4 shows the results under one-pass evaluation (OPE), spatial robustness evaluation (SRE), and temporal robustness evaluation (TRE) using the overlap success rate. Overall, the proposed algorithm performs favorably against the state-of-the-art methods of non deep learning in all three metrics: OPE, SRE, and TRE.

3) ATTRIBUTE-BASED EVALUATION

We further analyze the tracker performance under different video attributes (as following Table 2) [41]. Fig.5 shows the OPE for eleven main video attributes with distance precision rate is greater than the threshold is calculated, as

$$\text{Success}(\beta) = \frac{1}{N} \sum_{i=1}^{N} \eta(\beta)$$

$$\eta(\beta) = \begin{cases} 1 & \text{overlap}_{\text{region}} \geq \beta \\ 0 & \text{overlap}_{\text{region}} < \beta \end{cases}$$

The performance comparisons of the success rate are shown in Fig.4 shows the results under one-pass evaluation (OPE), spatial robustness evaluation (SRE), and temporal robustness evaluation (TRE) using the overlap success rate. Overall, the proposed algorithm performs favorably against the state-of-the-art methods of non deep learning in all three metrics: OPE, SRE, and TRE.

C. QUALITATIVE EVALUATION

In this section, we qualitatively compare our method with other state-of-the-art trackers without deep learning. For simplicity, we only discuss background clutter, illumination variations, occlusion, abrupt motion, motion blur,
deformation, rotation, and scale variations. And these representative image sequences are selected from the dataset of the benchmark [41].

1) BACKGROUND CLUTTER
Firstly, we evaluate these trackers on two sequences with background clutter, which are basketball and couple. In the basketball sequence, the appearance changes of objects are mainly caused by the running athletes, especially when they are close to each other. In addition, the complexion is similar between teammates. As can be seen from Fig.6, TLD and Struck algorithms drift to other players because of the similar appearance of the players in the same team in the sequence basketball. And DSST algorithm drifts to
FIGURE 6. From top to bottom, visual illustration of the trackers over the sequences basketball and couple with background clutter.

FIGURE 7. From top to bottom, visual illustration of the trackers over the sequences sylvester and trellis with illumination variations.

FIGURE 8. From top to bottom, visual illustration of the trackers over the sequences faceocc2 and suv with occlusion.

other non-player regions. However, KCF, SAMF, LCT and proposed algorithm successfully tracked the entire image sequence.

In the couple sequence, the low resolution and the visual angle changes make backgrounds and target blur and clutter. As you can see from Fig. 6, KCF, SCM, and DSST drift completely, while our method can track the target successfully with small errors, mainly because it exploits the collaborative model to accurately distinguish the background and the target.

2) ILLUMINATION VARIATIONS
Fig. 7 shows the tracking results from two challenging sequences to evaluate whether our algorithm is able to handle illumination variations. As a whole, the nine trackers, including the proposed tracker, can track the target correctly on the sequence sylvester.

For the sequence trellis, the sequence, with the properties of local illumination variations, deformation and scale change, is taken by a mobile camera in an outdoor complex background, and tracking the target is a moving human face. Thus it is difficult to track the target. The algorithm can complete the tracking the target in the whole sequence. In the 415-th frame, a cluster of sunlight shines on the target, causing intense illumination changes. However, the proposed algorithm does not deviate the target, while the other algorithms deviate more or less.

3) OCCLUSION
Occlusion is the most common and critical problem in visual tracking. The proposed algorithm and the other eight algorithms have designed the occlusion handing method. Even if the target is lost, the target will be retrieved in the subsequent frames with the above algorithms in the sequence faceocc2. It can be confirmed the Fig.8.

However, it can also be seen from the Fig.8: when the target is occlusive, TLD, Struck, KCF and TGPR trackers all deviate from the target in the sequence suv, and the rest of the
trackers is drifting except proposed tracker, and the proposed algorithm can achieve excellently.

4) ABRUPT MOTION AND MOTION BLUR
We evaluate these trackers on two sequences with abrupt motion and motion blur, which are deer and jumping. In the sequence deer, a deer runs in the river, and the target we follow is the head of the deer. As we can see from Fig.9, all trackers are still able to track effectively during deer running except SCM tracker in the deer, even though there is a bias in some frames.

In the sequence jumping, a person is skipping, and we are tracking the person’s face. Because of the blurred appearance changes, SCM, DSST and LCT trackers lose their target, while the proposed algorithm can successfully track the target because the proposed algorithm can capture the essential characteristics of the fuzzy target.

5) DEFORMATION AND ROTATION
Target deformation and rotation are also challenging factors in the field of visual tracking. In the david sequence, the target undergoes two obvious changes in illumination, the pose changes caused by the glasses, and the rotation of the target caused by the swinging motion. It brings a great challenge to visual tracking. As we can see from Fig.10, all trackers can perform better tasks except Struck in sequence david. But the proposed algorithm has higher precision.

In the sequence singer2, there are several attributes: the low contrast ratio between background and target, the pose change, the scale change, and the illumination change. Therefore, the task is very difficult to complete. Only the proposed algorithm, TGPR, DSST and LCT do a good job of tracking tasks.

6) SCALE VARIATIONS
Scale change is also one of the most common challenges in visual tracking, and there is no very good way to solve this challenge. As a result, most trackers perform poorly on this factor. It can be confirmed in Fig.11. In the sequence carscale, the target has the challenge of occlusion as well as the scale variation. Overall, all trackers can capture part of the target, however, the scales are all incorrect.
The sequence woman can be divided into two parts: occlusion and scale variation, and the color feature of the target is similar to the background. Therefore, the tracking algorithm is easy to fail. In the first half, the target is partially occluded, and all the algorithms can find the target accurately. In the second half, all the algorithms deviate from the target because of the scale variation of the target except the proposed algorithm.

V. DISCUSS AND ANALYSIS

A. VALIDATION OF KEY COMPONENTS AND PARAMETERS

Effectiveness of dictionary update: It is important to update the dictionary B online to effectively deal with appearance variations. Fig. 12 illustrates a comparison of the proposed tracker with the dictionary update scheme and without it. The comparison in Fig. 12 clearly demonstrates that the proposed tracker performs better with the dictionary update scheme than without it.

Effectiveness of critical parameters: In addition to validating key components of the proposed tracker, the effect of several critical parameters on the tracking performance is also studied. The proposed algorithm requires the precise control of three essential parameters, λ, γ and τ that have a critical impact on the tracking performance: λ is constantly controlling the sparsity of α, γ balance the cost function, and τ is constant controlling the strength of representation loss. The comparison in Fig. 13.

From the results illustrated in Fig. 13, the tracking performance is better when λ = 3, γ = 0.2 and τ = 5.

B. COMPUTATIONAL COMPLEXITY ANALYSIS

The computational load of the proposed method is mainly in solving the LCSC algorithm iteratively. We use N to represent the total number of all data instances, and use K to represent the number of nearest neighbors. For updating B, we use an approximation algorithm for LLC fast encoding. This reduces the computation complexity from $O(M^2)$ to $O(M + K^2)$, where $K \ll M$. For updating c, the method of seeking a closed solution is given in this paper, and the computation complexity is $O(MDM)$. Finally, the time complexity for updating W is $O(MNM)$. Thus the complexity of the total operations is approximate $O(MDM + MNM)$. 

**FIGURE 12.** Effectiveness of dictionary update: OPE performance of the proposed tracker with and without the proposed dictionary update scheme.

**FIGURE 13.** Effect of $\lambda$, $\gamma$ and $\tau$: OPE performance of the proposed tracker for different values.
Fig. 5 shows that our algorithm performs well against state-of-the-art trackers without deep learning in Benchmark [41]. Among the state-of-the-art trackers without deep learning, the Struck method achieves the second-best results. Our tracker runs at around 15 frames per second. The main computational load of our tracker is the process of learning Parameters and pyramid max pooling.

C. ANALYSIS AND FUTURE WORK

One of the limitations of this algorithm is the optimization of K value and the optimization of pooling. If the target scale of the tracking area changes greatly, the accuracy also declines. Although the scale change was also considered in this paper, it is still an important problem to be solved in target tracking, which is another limitation of this algorithm. In addition, the research object of this paper is only limited to the target area, and the context space information of the target area is not effectively used. However, the algorithm proposed can achieve better results in the category of a non-deep learning algorithm, but it can only be used when the target area has been provided, and it is not suitable for other cases. More positively, various kinds of deep neural networks are now widely studied and have illustrated outstanding accuracy and efficiency in existing tracking benchmarks [47]–[50]. As is reported in literature [51], [52], a multi-mode framework based on a neural network is proposed, which can better capture effective targets. However, it is undeniable that this paper has made some contribution to the theoretical knowledge framework in the field of computer vision. As for us, target tracking can also be regarded as a classification of foreground and background, which can also be rethought the problem of target tracking in the field of deep learning. Next, we plan to design methods to find the global optimal K-value, and methods to integrate scale change and local feature extraction technology, which will make a breakthrough in the accuracy of this algorithm. To sum up, we have improved the local constraint method in the field of non-deep learning, but greater advantages in efficiency and effect had been showed in deep learning method, which is quite a hot topic worthwhile to do researches.

VI. CONCLUSION

In this paper, a collaborative model based on a locality-constrained coding framework is proposed for accurate and robust visual tracking. The collaborative model is capable of building a robust representation dictionary, which can capture the intrinsic geometrical attributes of the target. Thus, the coefficient vector can accurately characterize the target appearance, which leads to accurate and robust tracking results. The experimental results on the OTB dataset show the rationality of the two schemes with 5.4% and 4.7% improvement on the precision plot and success plot, respectively, and demonstrate the effectiveness and robustness of the proposed tracker when compared with other state-of-the-art methods without deep learning.

REFERENCES

[1] X. You, X. Li, Z. He, and X. F. Zhang, “A robust local sparse tracker with global consistency constraint,” Signal Process., vol. 111, pp. 308–318, Jun. 2015.
[2] Z. He, S. Yi, Y.-M. Cheung, X. You, and Y. Y. Tang, “Robust object tracking via key patch sparse representation,” IEEE Trans. Cybern., vol. 47, no. 2, pp. 354–364, Feb. 2017.
[3] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, “Long-term correlation tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 5388–5396.
[4] J. Gao, H. Ling, W. Hu, and J. Xing, “Transfer Learning Based Visual Tracking with Gaussian Processes Regression,” in Computer Vision—ECCV Lecture Notes in Computer Science, no. 8691, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer, 2014, pp. 188–203.
[5] Y. Li and J. Zhu, “A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration,” in Computer Vision—ECCV Lecture Notes in Computer Science, no. 8926, L. Agapito, M. M. Bronstein, and C. Rother, Eds. Springer, 2014, pp. 254–265.
[6] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, “Discriminative scale space tracking,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 8, pp. 1561–1575, Aug. 2017.
[7] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 3, pp. 583–596, Mar. 2015.
[8] S. Hare, A. Saffari, and P. Torr, “Struck: Structured output tracking with kernels,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2011, pp. 263–270.
[9] Z. Kalal, J. Matas, and K. Mikolajczyk, “P-N learning: Bootstrapping binary classifiers by structural constraints,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. 2010, pp. 49–56.
[10] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, “Hierarchical convolutional features for visual tracking,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 3074–3082.
[11] A. K. Chauhan and P. Krishan, “Moving object tracking using Gaussian mixture model and optical flow,” Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 3, no. 4, Apr. 2013.
[12] M. E. Fathy and R. Chellappa, “Uncertainty calibrated Markov chain Monte Carlo sampler for visual tracking based on multi-shape posterior,” J. Math. Imag. Vis., vol. 60, no. 5, pp. 681–691, Jan. 2018.
[13] Z. Liu, J. Wang, G. Liu, and L. Zhang, “Discriminative low-rank preserving projection for dimensionality reduction,” Appl. Soft Comput., vol. 85, Dec. 2019, Art. no. 105768.
[14] S. Zhan, S. Zhao, Y. Sui, and L. Zhang, “Single object tracking with fully least squares support vector machine,” IEEE Trans. Image Process., vol. 24, no. 12, pp. 5723–5738, Dec. 2015.
[15] S. Zhang, Y. Sui, S. Zhao, and L. Zhang, “Graph-regularized structured support vector machine for object tracking,” IEEE Trans. Circuits Syst. Video Technol., vol. 27, no. 6, pp. 1249–1262, Jun. 2017.
[16] T. B. Dinh and G. Medioni, “Co-training framework of generative and discriminative trackers with partial occlusion handling,” in Proc. IEEE Workshop Appl. Comput. Vis. (WACV), Jan. 2011, pp. 642–649.
[17] W. Zhong, H. Lu, and M.-H. Yang, “Robust object tracking via sparsity-based collaborative model,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 1838–1845.
[18] X. Mei and H. Ling, “Robust visual tracking using $l_1$ minimization,” in Proc. IEEE Int. Conf. Comput. Vis., Sep. 2009, pp. 1436–1443.
[19] C. Bao, Y. Wu, H. Li, and H. Ji, “Real time robust l1 tracker using accelerated proximal gradient approach,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 1830–1837.
[20] T. Zhang, S. Liu, C. Xu, S. Yan, B. Ghanem, N. Ahuja, and M.-H. Yang, “Structural sparse tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 150–158.
[21] Z. Lai, W. K. Wong, Y. Xu, J. Yang, and D. Zhang, “Approximate orthogonal sparse embedding for dimensionality reduction,” IEEE Trans. Neural Netw. Learn. Syst., vol. 27, no. 4, pp. 723–735, Apr. 2016.
[22] L. Nie, X. Song, and T.-S. Chua, “Learning from multiple social networks,” Synth. Lectures Inf. Concepts, Retr. Services, vol. 8, no. 2, pp. 1–118, Apr. 2016.
[23] L. Nie, L. Zhang, M. Wang, R. Hong, A. Farseev, and T.-S. Chua, “Learning user attributes via mobile social multimedia analytics,” ACM Trans. Intell. Syst. Technol., vol. 8, no. 3, pp. 1–19, Apr. 2017.
[25] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained linear coding for image classification,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. 2010, pp. 3360–3367.

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

[27] B. Liu, J. Huang, C. Kulikowski, and L. Yang, “Robust visual tracking using local sparse appearance model and K-Selection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 12, pp. 2968–2981, Dec. 2013.

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

[29] Z. Liu, Z. Lai, W. Ou, K. Zhang, and R. Zheng, “Structured optimal graph based sparse feature extraction for semi-supervised learning,” Signal Process., vol. 170, May 2020, Art. no. 107456.

[30] J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” IEEE Trans. Image Process., vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[31] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, “Coordinate descent method for large-scale L2-loss linear support vector machines,” J. Mach. Learn. Res., vol. 9, pp. 1369–1398, Jun. 2008.

[32] D. Bertsekas and D. Bertsekas, Nonlinear Programing. Belmont, MA, USA: Athena Scientific, 1999.

[33] O. L. Mangasarian, “A finite Newton method for classification,” Optim. Methods Softw., vol. 17, no. 5, pp. 913–929, Jan. 2002.

[34] T. Zhou, Y. Lu, and H. Di, “Locality-constrained collaborative model for robust visual tracking,” IEEE Trans. Circuits Syst. Video Technol., vol. 27, no. 2, pp. 313–325, Feb. 2017.

[35] B. Liu, J. Huang, L. Yang, and C. Kulikowski, “Robust tracking using local sparse appearance model and K-selection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2011, pp. 1313–1320.

[36] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,” in Computing Vision—ECCV. Cham, Switzerland: Springer, 2014, pp. 346–361.

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

[38] N. R. Bhubaneswari and V. G. Sivakumar, “Novel image classification technique using particle filter framework optimised by multikernel sparse representation,” Brazilian Arch. Biol. Technol., vol. 59, 2016.

[39] W. Li, P. Wang, and H. Qiao, “Top-down visual attention integrated particle filter for robust object tracking,” Signal Process., Image Commun., vol. 43, pp. 28–41, Apr. 2016.

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

[41] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to State-of-the-Art superpixel methods,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[42] S. Chan, X. Zhou, and S. Chen, “Online classification for object tracking based on superpixel,” Neurocomputing, vol. 286, pp. 88–108, Apr. 2018.

[43] F. Li and S. Liu, “Object tracking via a cooperative appearance model,” Knowl.-Based Syst., vol. 129, pp. 61–78, Aug. 2017.

[44] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “Exploiting the circulant structure of tracking-by-detection with kernels,” in Computing Vision—ECCV (Lecture Notes in Computer Science), no. 7575, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Germany: Springer, 2012, pp. 702–715.

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

[46] D. Huang, L. Kong, J. Zhu, and L. Zheng, “Improved action-decision network for visual tracking with meta-learning,” IEEE Access, vol. 7, pp. 117206–117218, 2019.

[47] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, “Action-decision networks for visual tracking with deep reinforcement learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2711–2720.

[48] H. Nam and B. Han, “Learning multi-domain convolutional neural networks for visual tracking,” 2015, arXiv:1510.07945. [Online]. Available: http://arxiv.org/abs/1510.07945

[49] Y. Song, C. Ma, L. Gong, J. Zhang, R. Lau, and M.-H. Yang, “CREST: Convolutional residual learning for visual tracking,” 2017, arXiv:1708.00225. [Online]. Available: http://arxiv.org/abs/1708.00225

[50] Y. Wei, X. Wang, W. Guan, L. Nie, Z. Lin, and B. Chen, “Neural multimodal cooperative learning toward micro-video understanding,” IEEE Trans. Image Process., vol. 29, pp. 1–14, 2020.

[51] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, and T.-S. Chua, “MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video,” in Proc. 27th ACM Int. Conf. Multimedia (MM), Nice, France: Association for Computing Machinery, 2019, pp. 1437–1445.

---

JIA HU received the B.Sc. degree in computer science and technology from Dezhou University, Dezhou, Shandong, China, in 2008, and the M.Sc. degree in computer software and theory from Chang’an University, Xi’an, Shaanxi, China, in 2011. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Central South University, Changsha, Hunan, China. His main research interests include machine learning, computer vision, and digital image processing.

XIAOPING FAN (Member, IEEE) was born in Nanchang, Jiangxi, China, in 1961. He received the B.S. degree in electrical engineering from the Jiangxi University of Technology, Nanchang, in 1981, the M.S. degree in traffic information engineering and control from Changsha Railway University (Central South University), Changsha, Hunan, in 1984, and the Ph.D. degree in control science and engineering jointly from the South China University of Technology, Guangzhou, Guangdong, and The Hong Kong Polytechnic University, Hong Kong, in 1998. He was an Assistant Professor with the School of Information Engineering, Changsha Railway University, from 1984 to 1994. He was an Associate Professor, from 1994 to 1999. From 1999 to 2010, he was a Professor with the School of Information Science and Engineering, Central South University. Since 2010, he has been a Professor with the School of Information Technology and Management, Hunan University of Finance and Economics. He has authored two books and more than 300 journal articles and conference papers. He holds over 15 inventions. His research interests include robot control, wireless sensor networks, data mining, and intelligent transportation systems. He serves as an Associate Editor for Journal of Systems Engineering.