IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 25, 2019, accepted October 17, 2019, date of publication October 31, 2019, date of current version November 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2950690

Object Tracking With Structured Metric Learning

XIAOLIN ZHAO[1], ZHUOFAN XU[2], BOXIN ZHAO[1], XIAOLONG CHEN[3], AND ZONGZHE LI[1]
[1]Equipment Management and UAV Engineering College, Air Force Engineering University, Xi'an, China
[2]Joint Operations College, National Defense University, Shijiazhuang, China
[3]Flight Automatic Control Research Institute, Xi'an, China

Corresponding author: Boxin Zhao (boxin.zhao@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61503405, Grant 61703428, Grant 61601021, and Grant U1533132, in part by the National Aeronautical Science Foundation of China under Grant 20160896007 and Grant 20160896008, and in part by the Fundamental Research Funds for the Central Universities under Grant 2016RC015.

ABSTRACT In this paper, we propose a novel tracking method based on structured metric learning, which takes the advantages of both structured learning and distance metric learning. In our method, tracking is formulated as a structured metric learning problem, which not only considers the importance of different samples, but also improves the discriminability by learning a specific distance metric for matching. Specifically, a concrete structured metric learning method is realized by making use of the constraints from the target and its neighbour training samples under the above framework. Besides, a closed-form solution is derived for the structured metric learning problem. To improve the matching robustness, the K-nearest neighbours (KNN) distance is employed to determine the final tracking result. Experimental results in the benchmark dataset demonstrate that the proposed structured metric learning based tracking method can achieve desirable performance.

INDEX TERMS Object tracking, structured metric learning, KNN distance.

I. INTRODUCTION

Object tracking is one of the research topics in computer vision, multimedia information processing, etc. Because it can be applied in many fields, such as motion analysis, intelligent surveillance, video editing and human computer interactive, it has achieved the attention of many researchers. However, realizing robust and accurate tracking is still a challenging task because there exist many complex factors. For example, the target may be occluded by some other objects, and the appearance of the target may change heavily. In addition, there may also be illumination changes and scale changes, and the background may be cluttered.

Researchers have proposed many kinds of tracking methods. According to whether using the background information to construct the appearance model, these tracking methods can be divided into two types: the generative model based methods and the discriminative model based methods. For the generative model [1]–[5], the similarity between the candidate samples and the target templates are often calculated and the tracking result is determined according to the similarity. However, only the target information is made use of while the background information is neglected in generative model. For the discriminative model [6]–[11], tracking is usually formulated as a binary classification problem and the confidence scores of the candidate samples are calculated to determine the tracking result based on the learned classifier. In the latter model, both the target and background information is applied, which often leads to better performance than the generative model. However, there still exist some issues in the discriminative model. For example, the objectives of tracking and classification are not completely consistent, and the samples from the background are obtained with down-sampling which cannot make full use of the background information, etc.

Deep learning has shown great potential and advantage in feature extraction and model fitting. It is a new thought to use deep learning for tracking problems. Deep learning is derived from the study of neural networks and can be understood as a deep neural network. Deep feature representation can be obtained through it, which avoids the complicated and cumbersome features of manual selection and the dimensional disaster of high-dimensional data. The main models of deep learning include Deep belief network (DBN) [12], [13] based on Restricted Boltzmann machine (RBM), Stacked auto encoders (SAE) based of Auto encoder (AE) [14], Convolutional neural networks (CNN) [15], and

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

Recurrent neural networks (RNN) [16]. Wu *et al.* proposed a new deep learning method based on the greedy deep weighted dictionary learning for mobile multimedia for medical diseases analysis, which provides guidance for the diagnosis of disease in wisdom medical [17]. Hu *et al.* proposed a new deep transfer metric learning (DTML) method to learn a set of hierarchical nonlinear transformations for cross-domain visual recognition, which can achieve better performance than existing linear metric learning methods [18]. Lu *et al.* presented a new discriminative deep metric learning(DDML) method for face and kinship verification in wild conditions, which achieves the acceptable results in both face and kinship verification [19]. In paper [20], deep learning is investigated in more detail, training deep artificial neural networks to represent the optimal control action during a pinpoint landing and assuming perfect state information. The results allow for the design of an onboard real-time optimal control system able to cope with large sets of possible initial states while still producing an optimal response. Reference [21] surveyed recent developments in the literature regarding deep reinforcement learning methods for building human-level agents and opened a discussion that potentially raises a range of future research directions in it.

Structured learning can well represent the output of some tasks in a unified framework with latent variables. An important property of structured learning is that it brings in the loss function to measure the importance of the training samples, which often improves the performance of the algorithm. To address the issues existing in the classification based discriminative model, structured learning is introduced into tracking by some researchers. Hare *et al.* [22] propose the Struck tracking method, which formulates tracking as a structured output problem and achieves great success. By introducing the structured learning strategy, the discriminability of the tracker becomes finer and the location accuracy can be further improved.

Metric learning aims at automatically learning a metric from data to better measure the distance or similarity between the data. It is also an important topic in machine learning, pattern recognition, etc. Since metric leaning can often capture the idiosyncrasies of the data of interest, it may perform better than the standard metrics (e.g., the Euclidean distance) for specific tasks. Metric learning has been successfully applied in many fields, such as image classification, object recognition and face recognition. Specifically, some researchers have introduced metric learning to tracking with good results.

Although both the structured learning and the metric learning have achieved great success in tracking, each of them focuses on only one aspect and does not pay enough attention to the other's advantages. To the best of our knowledge, there has not been a learning method combining both of them for tracking.

Therefore, in this paper, we propose a tracking method based on structured metric learning. First, we propose a structured metric learning based tracking framework is shown in Fig.1. On one hand, the structured learning is used to improve the discriminability of the model. On the other hand, the metric learning can improve the adaptivity to the appearance changes. Second, based on the formulation above, we present a concrete structured metric learning method for tracking. By dividing the training samples into target buffer and neighbour buffer, the fine spatial constraints are utilized to complete the metric learning. Third, the KNN distance is introduced to determine the final tracking results, which improves the robustness to challenging situations. We test the proposed tracking methods in the benchmark dataset [23] and the experimental results demonstrate that our method can achieve comparable performance to many state-of-the-art tracking methods.

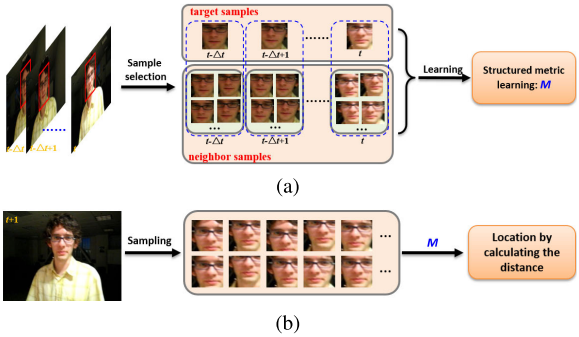


FIGURE 1. The overview of the proposed tracking framework, which includes two steps. (a) Learning the structured metric based on the selected target and neighbor samples. (b) Tracking based on the learned metric.

The main contributions of this paper are two folds:

(a) From the perspective of theory, a novel structured metric learning method is developed, which absorbs the advantages of both structured learning and metric learning methods. Concretely, the mathematical form of the structured metric learning is given and the corresponding optimization method is derived.

(b) From the perspective of practice, we apply the proposed structured metric learning in tracking and present a novel tracking method. Our method not only makes full use of the spatial constraint but also better measures the distance between the samples based on the learned metric, which can determine the tracking position more accurately.

The remainder of this paper is organized as follows. Section II introduces the related work. Section III gives the details of the proposed tracking method, following which the experimental results are demonstrated in Section IV. The paper is concluded in Section V.

II. RELATED WORK

A. GENERATIVE TRACKING

Many famous trackers are realized based on the generative model, which is usually constructed based on the information of the target while ignoring the background. The basic generative model is template based tracking [24], [25], in which the target result is determined by template matching. Further, to improve the robustness to occlusions and pose

changes, Adam *et al.* [1] present the Frag tracking algorithm, which represents the template object by multiple image fragments or patches and combines the vote maps of the patches for the final results. To better deal with the appearance changes, some researchers make use of subspace learning and sparse representation to build the model. For examples, Ross *et al.* [3] present a tracking method based on incremental principal component analysis (PCA) subspace learning. Mei and Ling [26] propose a sparsity approximation based tracking method with ℓ_1-regularization. Besides, some researchers use different strategies to promote the running speed of the ℓ_1-regularized method [4], [27]. Moreover, Zhang *et al.* [9] also utilize the low rank algorithm to improve the tracking performance.

However, since most of the above methods only use the target information, it may have less discriminability and not work well in the condition of heavy occlusion, deformation, background clutter, etc. Moreover, it does not consider the adaption of the metric. In practice, our method follows the template matching strategy as well, but we develop a structured metric learning method to improve the method's discriminability and adaptivity.

B. DISCRIMINATIVE TRACKING

Discriminative model, which is also called tracking-by-detection, is constructed based on the information of both the target and the background. In the traditional discriminative model based methods, tracking is often formulated as a binary classification problem. Avidan [6] proposes a discriminative tracking method based on an off-line support vector machine (SVM). Grabner *et al.* [28] present an online boosting algorithm for tracking, which can adaptively select the features for representation. Tang *et al.* [29] introduce the co-training framework into tracking, which combines two views of SVMs to improve the robustness. Zhang *et al.* [30] make use of the compressive sensing to select features and train the naive Bayesian classifier as the tracker. In addition, some researchers also develop some methods based on discriminative subspace learning and sparse representation. For example, Li *et al.* [31] propose an incremental 3D discrete cosine transform based tracking method, which uses both the target and background to build the subspace. Sui *et al.* [32] develop the discriminative low rank tracking method which also makes use of the information of both the target and background. However, since the above methods are all implemented under the binary classification framework, there still exist some problems, such as the downsampling problem of the training samples, inconsistencies in the objectives of tracking and classification, etc.

To address the issues existing in the traditional discriminative model, many strategies, including multiple instance learning, structured learning, fuzzy learning, etc., have been introduced into tracking. For example, Babenko *et al.* [7] present an online multi-instance boosting method for tracking. In their method, the positive bags and negative bags are used to replace the positive and negative samples, which can reduce the label ambiguity of the samples. Hare *et al.* [22] develop a structured learning based tracking approach, which makes use of structured output to avoid the intermediate classification step. Henriques *et al.* [33] formulate tracking as a correlation filtering problem which makes more full use of the space information. Zhang *et al.* [34] introduce fuzzy learning into tracking and propose a fuzzy least squares SVM for tracking, which can effectively deal with the fuzzy boundary problem between the training samples. Although these discriminative methods address the issues in the traditional methods by different manners, the metrics hiding in these methods are still defined in advance or fixed, which lacks of the flexibility and accuracy to represent the appearances of different targets and scenes.

C. METRIC LEARNING BASED TRACKING

Researchers have proposed many different kinds of metric learning algorithms [35], [36] and hereby, we mainly have a short view about the Mahalanobis distance learning methods. The first Mahalanobis distance learning algorithm is proposed by Xing *et al.* [37]. It has no regularization, and aims to maximize the sum of distances between dissimilar points while keep the sum of distances between similar examples small. Schultz and Joachims [38] further add a diagonal matrix as the regularization, obtaining a new metric learning method. Moreover, Goldberger *et al.* [39] introduce the neighbourhood component analysis (NCA) method, Globerson and Roweis [40] propose maximally collapsing metric learning (MCML) method and Weinberger *et al.* [41] present the large margin nearest neighbors (LMNN) method.

Metric learning has been successfully applied in many computer vision fields, e.g. image classification [42], object recognition [43], image annotation [44], image retrieval [45], etc. Recently, some researchers have introduced metric learning into tracking and obtained some useful results. For example, Jiang *et al.* [46], [47] incorporate adaptive metric into differential tracking method and obtain a closed-form analytical solution to motion estimation. Li *et al.* [48] propose the non-sparse linear representations for visual tracking, where a Mahalanobis distance metric is learned online and incorporated for linear representation. Wu *et al.* [49] propose a metric learning based structural appearance model, which introduces online multiple instance metric learning algorithm to learn the metric and uses multiscale max pooling on the weighted local sparse codes. Yi *et al.* [50] develop the individual adaptive metric learning for visual tracking, which can improve the computational efficiency by adapting the distance from each individual sample point to a few anchor points instead of the distance between all pairs of samples. However, most of these methods are realized based on the binary classification model. In this paper, we focus on the Mahalanobis distance learning and propose a novel structured metric learning algorithm. It is driven by the tracking task and breaks through the limitation of the traditional classification framework.

Clustering analysis as a key step in object tracking plays an important role, which determines the final tracking result.

He *et al.* [51] suggest to use kernel k-means sampling for Nystr öm-based kernel matrix approximation to minimize the upper bound of a matrix approximation error. In order to get a lower complexity of spectral clustering and speed up eigenvector approximation for large-scale data, He *et al.* [52] propose an efficient spectral clustering method via explicit feature mapping. Considering to the performance of Non-negative matrix factorization that has been restricted due to its limited tolerance to data noise, as well as its inflexibility in setting regularization parameters, Leng *et al.* [53] propose a novel sparse matrix factorization method for data representation termed Adaptive Total-Variation Constrained based Non-Negative Matrix Factorization on Manifold (ATV-NMF). The beauty of this method is that the manifold graph regularization is also incorporated into NMF, which can discover intrinsic geometrical structure of data to enhance the discriminability. In our work, we propose a novel tracking method based on structured metric learning, which not only considers the importance of different samples, but also improves discriminability by learning a specific distance metric for matching. Specifically, a concrete structured metric learning method is realized by making use of the constraints from the target and its neighbouring training samples within the above framework.

III. TRACKING WITH STRUCTURED METRIC LEARNING

A. OVERVIEW

The overview of the proposed structured metric learning based tracking method, which is named *SML*, is shown in Fig. 1. The novel method has two main steps. The first step is to learn the structured metric. Hereby, we propose a novel structured metric learning method to learn the metric M, which considers both the structured learning and metric learning and can better represent the distance of the samples than the traditional Euclidean distance. As Fig. 1(a) shows, the target samples and some neighbor samples near the target are selected to learn the metric, which provides the spatial constraints and has fine discriminability. The second step is tracking and location. According to the learned M, we calculate the distances between the candidate samples and the target samples, respectively. Moreover, the minimum distance rule is used to determine the final tracking result. Further, the new tracking result is also used to relearn the metric for the tracking in next frames. In the following subsections, we will introduce the formulation based on structured metric learning, training process, tracking process and update scheme in details.

B. FORMULATION

Since we aim to track the target by calculating the distances between the candidate samples and the target samples, the metric plays an important role in calculating the distance and how to learn the distance metric is regarded as the kernel of our method. The proposed method is motivated by both the structured learning and the metric learning. As we have introduced before, the Struck method [22] formulates tracking as a structured regression problem. In Struck, the training samples obtained based on different transformations are assigned to different importance according to the overlap rate. In this paper, we utilize the structured learning strategy as well, and we bring the structured learning strategy into metric learning, by which we can better measure the distances of the training samples and the target samples.

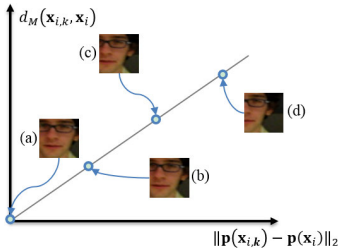


FIGURE 2. The spatial constraint from the neighbour samples. (a) denotes the target sample and (b)-(d) denote the samples with different distances to the target. It can be seen that the farther the position of the sample to the target, the larger the distance between it and the target in feature space.

The proposed method is based on the following structured constraints. The distances in the feature space between the target sample and the neighbour samples should be different according to the different spatial distances, the distance in the feature space increases with the increase of spatial distance. Moreover, the neighbour samples which are farther to the target should have larger distances to the target than the neighbour samples which are closer to the target. By considering the importance of different neighbour samples, the structured metric learning can provide finer discriminability than the traditional binary classification formulation. As Fig. 2 shows, (a) denotes the target sample and (b)-(c) denote the neighbor samples. Since the sample (d) is farther than (b) and (c) to (a), the distance between (d) and (a) should be larger.

Based on the above constraints, learning the structured metric M can be formulated as the following problem

$$\min_{M} \|M\|_F^2 - C \sum_{i,k} \Delta_{i,k} d_M^2(x_i, x_{i,k}), \tag{1}$$

where M is the metric to be optimized, $\|\cdot\|_F$ denotes the Frobenius norm, $d_M(\cdot,\cdot)$ denotes the distance between two samples with metric M, C is the trade-off parameter and $\Delta_{i,k}$ denotes the weight parameter which is used to measure the dissimilarity of different neighbor samples to the target. The first term is the regularization term, which is used to avoid overfitting. In the second term, x_i denotes the target sample and $x_{i,k}$ denotes the neighbor sample which is selected by transformation around x_i, where $k = 1, 2, \ldots, N_k$ and N_k denotes the number of transformations. Note that the second term is negative, which can make the structured constraints integrated in the unified framework with regularization.

C. OPTIMIZATION

Now we introduce how to solve the optimization problem in Eqn.1. According to the definition of distance metric,

Eqn.1 can be rewritten as follows

$$\min_{M} \|M\|_F^2 - C \sum_{i,k} \Delta_{i,k}(x_i - x_{i,k})^T M(x_i - x_{i,k}). \quad (2)$$

Let J denote the function to be optimized in Eqn.2, then the first-order derivative of J w.r.t M can be calculated by

$$\frac{\partial J}{\partial M} = 2M - C \sum_{i,k} \Delta_{i,k}(x_i - x_{i,k})(x_i - x_{i,k})^T. \quad (3)$$

Then the optimal solution to Eqn.1 can be obtained by setting $\partial J/\partial M = 0$. As a result,

$$M = \frac{1}{2} C \sum_{i,k} \Delta_{i,k}(x_i - x_{i,k})(x_i - x_{i,k})^T. \quad (4)$$

It can be seen that M can be obtained in closed form, which can be calculated efficiently.

D. TRAINING SAMPLES PREPARATION

The distance metric is learned by adding constraints to the training samples. As we have mentioned above, the training samples can be divided into target samples and neighbor samples. Hereby, we introduce how to select the training samples in details. In our method, we build two buffers, the target buffer T and the neighbor buffer N, to store these two types of samples respectively, and both of the two buffers have the fixed size. Since it is assumed that there is only a single optimal tracking result in each frame, the target buffer T is composed of the sequentially obtained tracking results in successive frames, i.e., the sample x_i corresponds to the tracking result in a frame. On the other hand, the neighbor buffer N contains a group of sample sets, and the elements in the neighbor buffer are also obtained in sequential frames. Note that the samples in N are selected by a sliding window around the position of the tracking result in that frame, and a target sample corresponds to a set of neighbor samples. Concretely, assume the tracking result in frame t is x_i, and its position is p_t. Then, x_i can be taken as a sample in the target buffer. Further, we slide a window with the same size as the target around p_t and get a set of samples $\{x_{i,k}\}$. The position of the sample $x_{i,k}$ is determined by the translation $p_t \circ y_k$, where $\circ$ denotes the translation operation, and y_k denotes the translation vector including both the height and the width, where $k = 1, 2, \ldots N_k$ and N_k is the number of the translation transformations. Because all the samples are normalized to the size of $N_s \times N_s$, the sliding step size is set as $d_x = aW/N_s$ in $x-$axis and $d_y = aH/N_s$ in $y-$axis, where W and H denote the width and height of the target, and a denotes a step coefficient. Then the transformation is set as $y = (\alpha d_x, \beta d_y)$ with $\alpha, \beta \in \{-N_s/a, \ldots, N_s/a\}$. Both the samples $\{x_i\}$ in the target buffer T and the sample sets $\{x_{i,k}\}$ in the neighbor buffer N are taken as the training samples to learn the metric.

In addition, the weight parameter $\Delta_{i,k}$ should be determined in advance as well. Hereby, the bounding box overlap rate [54] is employed to measure the importance of the samples. The parameter $\Delta_{i,k}$ corresponding to $x_{i,k}$ is defined as:

$$\Delta_{i,k} = 1 - \frac{area(R(x_i) \cap R(x_{i,k}))}{area(R(x_i) \cup R(x_{i,k}))}, \quad (5)$$

where $R(x_i)$ is the region of the target sample x_i, and $R(x_{i,k})$ is the region of the neighbor sample $x_{i,k}$.

E. TRACKING AND LOCATION

Based on the learned metric M, we can run the tracking process and determine the tracking results in the following frames. The tracking process includes two steps: select the candidate samples and determine the final tracking result.

Hereby, we adopt the sliding window search strategy to select the candidate samples. Assume the position of the tracked target in the last frame $t-1$ is p_{t-1}, and a candidate sample $\hat{x}_m$ in the current frame t is $p(\hat{x}_m)$, then we slide the searching window around p_{t-1} to select the candidate samples. Each candidate sample is obtained by cropping an image region centering at $p(\hat{x}_m)$ and with the same size as the target. If $p(\hat{x}_m)$ satisfies

$$\|p(\hat{x}_m) - p_{t-1}\|_2 < R_s, \quad (6)$$

then the corresponding sample $\hat{x}_m$ will be taken as the candidate sample, where $m = 1, 2, \ldots N_m$ and N_m is the number of the candidate samples. In Eqn.6, R_s denotes the searching radius.

With the selected samples $\{\hat{x}_m\}$, we introduce the KNN distance to calculate the distances between the candidate samples and the target subspace and determine the final tracking result. Specifically, the parameter K in the KNN distance is set as 10 and the distance metric with M is determined by Eqn. 4.

First, we calculate the distances between $\hat{x}_m$ and the target samples in the target buffer, which follows the template matching strategy. For each candidate sample $\hat{x}_m$, the distance between $\hat{x}_m$ and a sample x_i in the target buffer can be calculated based on M:

$$d_M^2(\hat{x}_m, x_i) = (\hat{x}_m - x_i)^T M(\hat{x}_m - x_i). \quad (7)$$

Then we rank the distances $\{d(\hat{x}_m, x_i)\}$ in ascending order, and denote the ordered distances as $\{d_r(\hat{x}_m, x_i^r)\}$, where $d_r(\hat{x}_m, x_1^r) < d_r(\hat{x}_m, x_2^r) < \ldots < d_r(\hat{x}_m, x_T^r)$ and x_i^r denotes the reordered sample in the target buffer.

The KNN distance is defined as the average of the first K smallest $d_r(\hat{x}_m, x_i^r)$:

$$d_{knn}(\hat{x}_m, T) = \frac{1}{K} \sum_{i=1}^{K} d_r(\hat{x}_m, x_i^r). \quad (8)$$

After the KNN distances between all candidate samples and the target buffer have been calculated, the final tracking result can be determined by the minimum distance rule. By selecting the sample with the smallest KNN distance:

$$\hat{x}_{opt} = \arg\min_{\hat{x}_m} d_{knn}(\hat{x}_m, T), \quad (9)$$

the optimal tracking sample $\hat{x}_{opt}$ and its corresponding position representing the best state in the current frame can be obtained.

F. UPDATE SCHEME

Because there are many complex factors, such as occlusion and deformation, the appearance often changes with time going. Hereby, we propose an update model based on first-in and first-out (FIFO) rule and incremental strategy, by which we can learn a new metric to better calculate the distance and then adapt to the appearance changes.

First, we select the new training samples to replace some earlier ones based on the FIFO rule. Since the latest samples can better represent the changes of the appearance than the earlier ones, it is reasonable to use the FIFO rule. As we have shown in Section III-D, the training samples are stored in the target buffer T and neighbor buffer N, we update the samples in these two buffers respectively. For the target samples, we directly add the new obtained sample $\hat{x}_{opt}$ into the buffer T to replace the sample which came into the buffer in the earliest time. For the neighbor samples, we resample a set of new neighbor samples $\{\hat{x}_{opt,k}\}$ around the position of the tracking result $\hat{x}_{opt}$ in the current frame t, following the same sample selection strategy in Section III-D. Then we use the new set of samples to replace the earliest sample set in the neighbor buffer N.

Second, we recalculate M by calculating the term $\sum_{i,k} \Delta_{i,k}(x_i - x_{i,k})(x_i - x_{i,k})^T$ in Eqn. 4. Because we have used the FIFO rule to update the training samples, the above two terms can be obtained according to an incremental strategy and we do not need to calculate each $(x_i - x_{i,k})$. For the samples in the updated target buffer and its corresponding sample sets in the updated neighbor buffer, only the differences $\{(\hat{x}_{opt} - \hat{x}_{opt,k})\}$ need to be calculated based on the new sample set. It can be observed that, the FIFO and incremental strategy based update model can be realized with high efficiency, which greatly reduces the complexity of Eqn. 4.

With the new metric M, the tracking can be continued in the following frames and the new tracking results can be obtained. The complete algorithm is summarized in Algorithm. 1.

We further discuss the strengths and weakness of the proposed method. The main advantage of our tracking method is that the structured metric is online learned and can make better use of the spatial constraints and get finer discriminability. On one hand, it can adaptively calculate the distance between the samples of different targets, which is superior to the predefined metrics, e.g. Euclidean metric, cosine similarity. On the other hand, because the structured learning is introduced and the training samples are selected densely, the structured metric learning can achieve better discriminability than the other metric learning methods [47], [50] which follow the classification model. The weakness of the proposed method is that it has larger computation complexity than other methods with metric learning. In our method, the complexity for metric learning is $O(mnd)$, where m is the number of the samples in the target buffer, n is the number in the neighbor buffer, and d is the dimension of the features. For these parameters, n is much larger than that in the traditional binary classification model because the dense sampling brings more computation. For the comparison methods with metric learning, the computation complexity in methods [47], [50] is $O(d^3)$ and $O(d^2n)$ where n is the number of total samples, and d is the dimension of the features as well. For most sequences, considering the number of the samples and the feature dimension, our method needs to cost more time to learn the metric than many existing metric learning based methods.

Algorithm 1 The SML Tracking Algorithm

Require:

Current frame I_t; Previous object's position p_{t-1}; Target buffer T and neighbor buffer N; The learned metric M.

Ensure:

The object tracking result and its position p_t in I_t; Updated target buffer T and neighbor buffer N; The new learned metric M.

1: **IF** $t = 1$: **Initialization**.

(1) Set the tracking object manually.

(2) Select the target sample $\{x_1\}$ and the neighbor samples $\{x_{1,k}\}$ with $k = 1, 2, \ldots, N_k$ in the first frame, normalize them into the fixed size and extract the features for representation.

(3) Learn the structured metric M by the training samples.

2: **IF** $t > 1$:

2.1 Tracking.

(1) Choose the candidate samples $\{\hat{x}_m\}$ with the sliding window strategy in I_t, normalize them and extract the features.

(2) Calculate the distances between each candidate sample $\{\hat{x}_m\}$ and the target samples $\{x_i\}$ in the target buffer T with the learned metric M.

(3) Rank the distance and calculate the KNN distance according to Eqns. 7 and 8.

(4) Determine the tracking result and its position p_t with Eqn. 9.

2.2 Update.

(1) Take the tracking result $\hat{x}_{opt}$ as the new target sample to update the target buffer T.

(2) Select new neighboring samples $\{\hat{x}_{opt,k}\}$ around the target in frame I_t and use them to update the neighbor buffer N.

(3) Relearn M based on the updated T and N.

IV. EXPERIMENT

A. INITIALIZATION

The proposed SML tracker is implemented based on MATLAB and is initialized as follows. Because the histogram of orientation gradients (HOG) with 6-pixel-window size and 9 orientations has obtained great success and widely used

in object detection and tracking, we take it as the feature to represent the samples. Both the training samples and the candidate samples are normalized to the fixed size 30×30, which is set as the same size as in many other tracking methods [4], [9] and accords with the size of HOG. The tradeoff parameter for regularization is experimentally set as $C = 1$, which is used to balance the fitting errors and the regularization term. The buffer depth for the training samples is set as 25 to balance the robustness and computation complexity. A larger buffer depth will improve the robustness of the representation but increase the computation time, and vice versa. The sliding step coefficient for neighbor samples selection is set as 3, which can achieve a better tradeoff between the accuracy and computation complexity. The searching radius for tracking is experimentally set as 24 pixels which can satisfy the searching requirements on most testing sequences. A smaller radius may lose the target if it moves quickly while a larger searching radius will greatly increase the searching time.

We evaluate the performance of the proposed tracker in the benchmark dataset, which has 51 video sequences with different attributes. The sequences are captured in different conditions, such as occlusion, deformation, illumination changes, fast motion, scale variations and background clutter. Four criteria are adopted to evaluate the performance of the tracker. The first is the average center location error (CLE), which is defined as the average value of the errors of the position of the tracked results. The second is the average Pascal VOC overlap rate (VOR), where the overlap rate score in one frame is defined as $score = (R_t \cap R_g)/(R_t \cup R_g)$, and R_t, R_g denote the bounding boxes corresponding to the tracked result and the ground truth respectively. The third is precision, which is defined based on CLE. If the CLE in one frame is smaller than a predefined threshold Th_p, the tracking in that frame is taken as precise. Then precision is defined as the rate of the number of the precise frames and the total frames. The fourth is success rate (SR) which depends on the VOR. If the overlap rate is larger than a predefined threshold Th_s, the tracking is considered successful in that frame, and SR is defined as the rate of the number of the successful frames and the number of the total frames. By assigning different values to Th_p and Th_s, we can further obtain the precision plot and the success plot, which can be used to evaluate the overall performance of the tracker. Moreover, the area under the curve (AUC) score is also used for evaluation.

B. COMPARISON WITH SOME STATE-OF-THE-ART METHODS

We compare the proposed tracking method with the top 10 trackers in Wu et al's benchmark [23], including Struck [22], SCM [55], TLD [56], ASLA [57], CXT [58], VTD [59], VTS [60], CSK [61], LSK [62] and DFT [63]. We first evaluate the overall performance of the trackers and then compare them in different conditions.

The comparison results of the overall performance are shown in Table. 1 and Fig. 3. From Table. 1 we can find that, the proposed SML tracker achieves the smallest average CLE and the largest average VOR. The precision of SML is 0.765, which outperforms the second best tracker, Struck, by about 10%. SML also gets the best SR result. Fig. 3 shows the precision plots and success plots of SML and the competing trackers. It can be seen that both plots generated by SML express better than the other trackers.

TABLE 1. The comparison results of average CLE (in pixel), average VOR, Precision ($Th_p = 20$) and SR ($Th_s = 0.5$) results of SML and the top 10 trackers in the benchmark. Best results are shown in bold.

Method	Average CLE	Average VOR	Precision (20)	SR (0.5)
SML	**31.2**	**0.5392**	**0.765**	**0.639**
Struck	50.5	0.4771	0.656	0.559
SCM	54.1	0.5052	0.649	0.616
TLD	48.1	0.4404	0.608	0.521
ASLA	73.0	0.4384	0.532	0.511
CXT	68.4	0.4292	0.575	0.492
VTD	47.4	0.4184	0.576	0.493
VTS	50.7	0.4189	0.575	0.496
CSK	88.8	0.4008	0.545	0.443
LSK	58.9	0.3974	0.505	0.456
DFT	69.2	0.3915	0.496	0.444

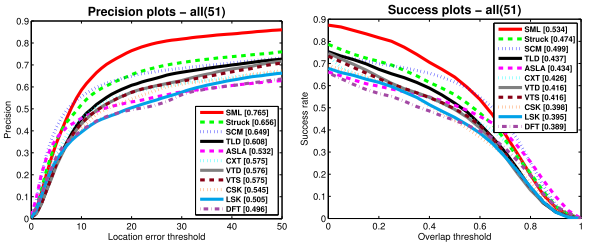


FIGURE 3. Precision plots and success plots obtained by SML and the top 10 trackers in the benchmark. The values in the square brackets represent the precision with $Th_p = 20$ pixels on precision plots and the area under the curve (AUC) on success plots, respectively.

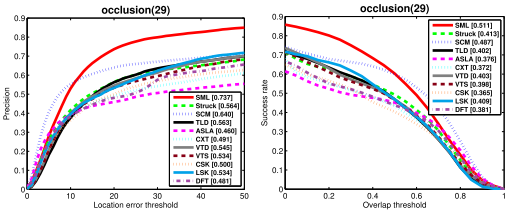
We also test the performance of the competing trackers in six representative conditions including occlusions, deformation, out-of-plane rotation, scale variation, background clutter and fast motion, and the precision and success plots are displayed in Fig. 4.

1) OCCLUSIONS

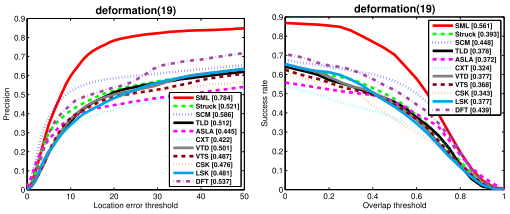
Fig. 4(a) shows the precision plots and the success plots of the competing trackers on the sequences with occlusions. It can be observed that the precision of SML at $Th_p = 20$ pixels is 0.737, which outperforms the second best tracker, SCM, by about 9.7%, and the AUC score of SML is similar to SCM. Because the learned metric has powerful discriminability and the KNN distance decision can improve the robustness to occlusion, SML can get good tracking results in the condition of occlusions.

2) DEFORMATION

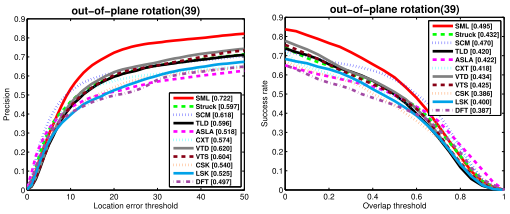
There are 19 sequences which have deformation to different degree and the comparison results are displayed in Fig. 4(b). We can see that SML gets both the best precision at $Th_p = 20$ pixels and the best AUC score, which outperforms SCM 20% and 11.3% respectively. Since our SML method intro-



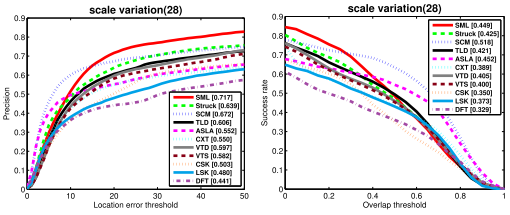
(a) Plots for occlusion attribute



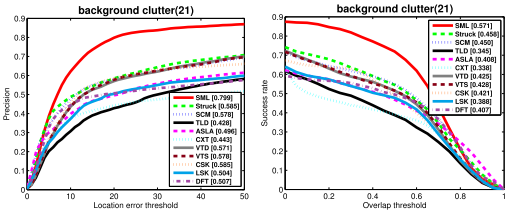
(b) Plots for deformation attribute



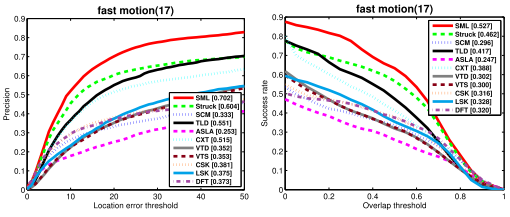
(c) Plots for out-of-plane rotation attribute



(d) Plots for scale variation attribute



(e) Plots for background clutter attribute



(f) Plots for fast motion attribute

FIGURE 4. Precision and success plots obtained by SML and the top 10 trackers in the benchmark for the sequences with different attributes. The value in the title represents the number of sequences with corresponding attribute.

duces the FIFO rule and incremental strategy to learn the metric online, it can well represent the deformation changes timely.

3) OUT-OF-PLANE ROTATION

Fig. 4(c) illustrates the precision plots and success plots of the competing trackers on 39 sequences with out-of-plane rotation. We can find that SML also acquires better plots than the competing trackers. On one hand, the metric is learned with high discriminability, which improves the robustness when matching in the tracking process. On the other hand, the online update scheme can ensure the metric adapts to changes in appearance.

4) SCALE VARIATIONS

Fig. 4(d) gives the comparison results on 28 sequences which has different scale changes. It can be seen that SML gets the largest precision at $Th_p = 20$ pixels. However, the AUC score of SML is not as good as SCM and ASLA, because the latter two methods use particle filter to control the scale changes while our method adopts the fixed-size bounding box for representation. But it should be noted that SML is better than Struck, TLD, etc, which use the fixed-size bounding box as well.

5) BACKGROUND CLUTTER

There exists background clutter on 21 sequences and the comparison results of the competing trackers on these sequences are shown in Fig. 4(e). It can be found that SML greatly outperforms the rest trackers and has significant advantages, which implies that the learned structured metric has powerful and fine discriminability and can improve the robustness of SML against the background clutter.

6) FAST MOTION

The comparison results on the sequences with fast motion in Fig. 4(f) show that SML can get better precision and success plots than the competing trackers. For example, SML outperforms Struck by about 10% and 6.5% respectively on precision at $Th_p = 20$ pixels and AUC score. The good performance of SML in this condition may benefit from the learned metric which can improve the matching similarity between the candidate samples and the target samples.

C. COMPARISON WITH OTHER METRIC LEARNING BASED TRACKING METHODS

We also compare our SML tracking method with another metric learning based tracking methods, the MLSAM tracker [49] and the DML tracker [18]. The MLSAM tracker builds the sparse codes based structural appearance model, in which the adaptive metric is learned by introducing an online multiple instance learning algorithm. The DML tracker learns a nonlinear distance metric in a feed-forward neural network architecture to classify the target object and background regions for tracking. We compare SML with MLSAM and DML both qualitatively and quantitatively.

The comparison results of the overall performance in the benchmark dataset are shown in Fig. 5. It can be found that our SML method achieves the better precision and success

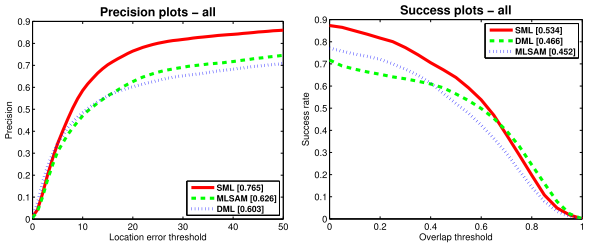


FIGURE 5. Precision and success plots of SML and other metric learning based trackers.

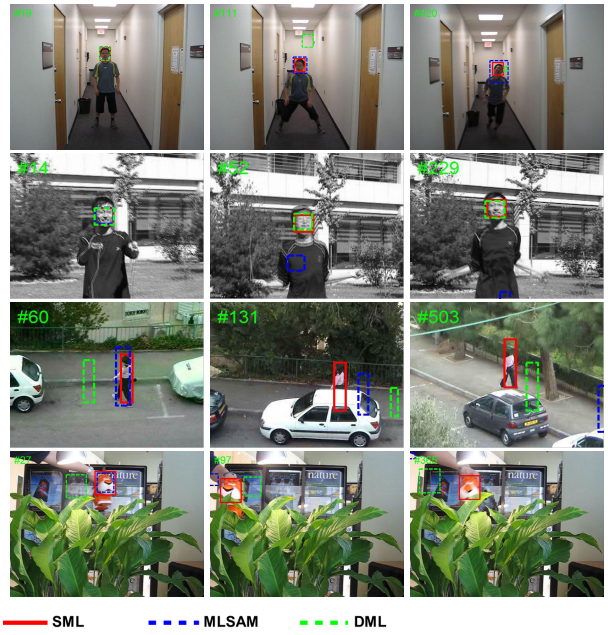


FIGURE 6. Examples of the tracking results obtained by SML and other metric learning based methods on some representative sequences. Top to down: *boy*, *jumping*, *woman* and *tiger2*.

plots than the other two trackers. Specifically, the precision at $Th_p = 20$ pixels and the AUC score of SML outperform the results of MLSAM and DML by more than 13% and 6% respectively. In addition, we also demonstrate some tracking result examples of these trackers on some representative sequences, shown in Fig. 6. On *boy* which has fast motion, out-of-view rotation and small scale changes, SML and MLSAM can complete the tracking while DML drifts. There also exists fast motion on *jumping*, but MLSAM loses the target on this sequence. Sequence *woman* has occlusion, deformation, background clutter and illumination changes, while *tiger2* contains fast motion, scale changes and in-plane rotation. It can be found that only our SML tracker successfully completes the tracking on these two sequences, while both MLSAM and DML fail. The comparison results indicate that our SML method which takes the advantage of metric learning performs better than the MLSAM and DML methods.

D. WITH VS. WITHOUT METRIC LEARNING

Our SML method takes the advantage of metric learning to improve the tracking performance. To demonstrate

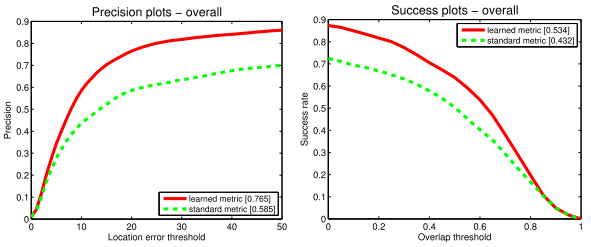


FIGURE 7. Precision plots and success plots obtained by the trackers with and without metric learning. The values in the square brackets represent the precision with $Th_p = 20$ pixels on precision plots and the area under the curve (AUC) on success plots, respectively.

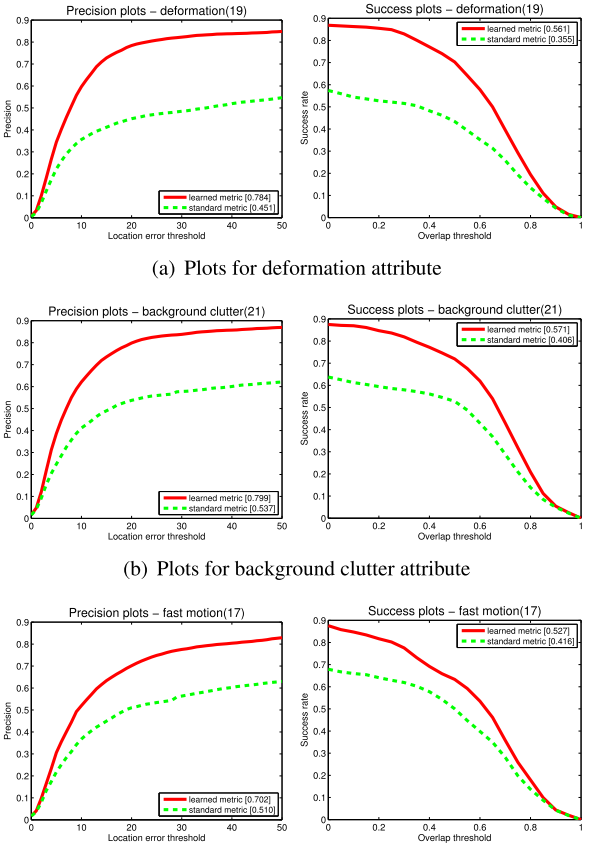


FIGURE 8. Precision and success plots obtained by the trackers with and without metric learning for the sequences with different attributes. The value in the title represents the number of sequences with corresponding attribute.

the effectiveness of metric learning, we implement another tracker which does not utilize metric learning for comparison. Hereby, the Euclidean metric is taken as the standard metric to implement the competing tracker, and our SML is denoted as the tracker with learned metric. The comparison results of the overall performance of these two trackers are shown in Fig. 7 and the results in three representative conditions are displayed in Fig. 8.

From Fig. 7 we can find that, the tracker with metric learning can get better precision plot and success plot than

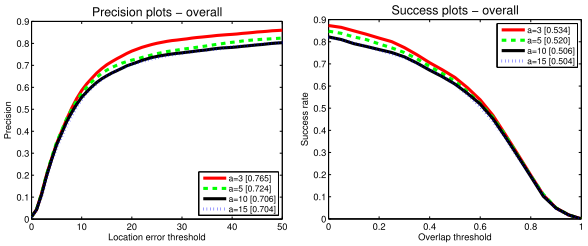


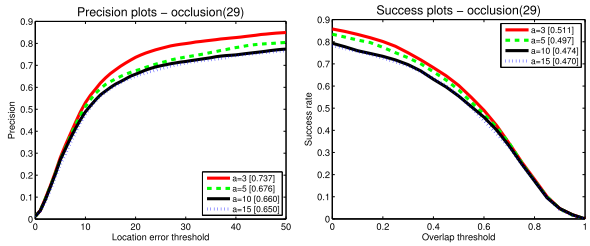
FIGURE 9. **Precision plots and success plots obtained by the trackers with different constraints. The values in the square brackets represent the precision with $Th_p = 20$ pixels on precision plots and the area under the curve (AUC) on success plots, respectively.**

the tracker without metric learning. Both the precision at $Th_p = 20$ pixels and the AUC score of SML outperform the tracker with the standard metric by about 18% and 10% respectively. Concretely, for the tracking results in the conditions of deformation, background and fast motion, the SML method with metric learning significantly outperforms the tracker with standard metric on both plots, as Fig. 8 shows. This indicates that the learned metric can greatly improve the tracking performance. Because the learned metric absorbs the advantages of both structured learning and metric learning, the learned metric can not only retain the high and fine discriminability, but also adapt to the the appearance changes caused by different conditions.

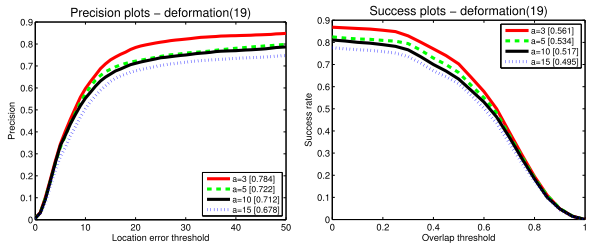
E. ANALYSIS OF STRUCTURED TERMS

In our method, the proposed structured metric learning framework makes use of the compact spatial relations as the structured constraints, which means that the samples closer to the target should have smaller distances to the target than the samples farther away. As mentioned before, the sliding step coefficient a is set as 3 to exploit the spatial constraints. To explore the effect of the sliding step size, we construct another three trackers by setting different values to a. In practice, the smaller the value of a, the more spatial information is exploited. Hereby, the competing trackers are with $a = 5$, $a = 10$ and $a = 15$, respectively. Note that the trackers with $a = 10$ and $a = 15$ can be considered as approximate realizations of the binary classification, because the step sizes have been $1/3$ or $1/2$ of the normalization size and the samples are selected far away from the target in this condition, which leads to weak spatial constraints. Because too small step size will greatly increase the computation complexity, the trackers with $a = 1$ and $a = 2$ are not taken into account.

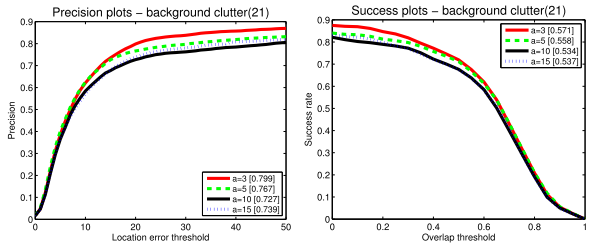
The comparison results of the overall performance of these two trackers are shown in Fig. 9 and the results in three representative conditions are displayed in Fig. 10. From these results, we can observe that, the tracker with $a = 3$ obtains better results than the others, and the performance of the trackers degrades with the value of a increasing. This indicates that the tracker with the smaller value of a makes more full use of the spatial structure constraints, which significantly improve the tracking performance. Specifically,



(a) Plots for occlusions attribute



(b) Plots for deformation attribute



(c) Plots for background clutter attribute

FIGURE 10. **Precision and success plots obtained by the trackers with different constraints for the sequences with different attributes. The value in the title represents the number of sequences with corresponding attribute.**

the comparison results between the trackers with $a = 3$ and $a = 15$ imply that the proposed structured terms with small step size can improve the discriminability of the tracker.

In summary, our SML method can achieve comparable performance to many famous trackers in the benchmark, which is verified by the results in Fig. 3 and Fig. 4. By comparing with other metric learning based tracking methods, we can find that the SML method can make better use of metric learning and achieve better tracking performance, which is shown in Fig. 5. By adding the metric learning term, it can be observed that SML gets better results than that without metric learning, which is illustrated in Fig. 7 and Fig. 8. We also explore the effect of the structured term in Fig. 9 to determine the best constraint. Although the proposed SML method obtains great success, there still exist many issues which decrease its performance. First, SML cannot work well in some complex conditions. For examples, there are complex deformations, huge illumination changes, fast motion and severe scale changes on sequences *Matrix* and *ironman*, our SML method drifts on both these two sequences. Second, because SML adopts the online metric learning, it costs more time to learn the metric than the traditional matching methods and cannot realize real time tracking in the current stage,

which also limits its application. We would like to address the above issues to further improve its performance and evaluate the tracking performance from the theoretical justification using methods proposed by [64], [65] in the future.

V. CONCLUSION

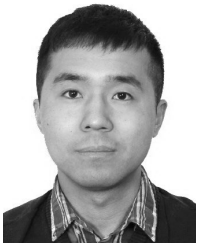
In this paper, we propose a structured distance metric learning based tracking method. After formulating tracking as a structured metric learning problem, a specific distance metric learning method is developed to adapt to the tracking task. By integrating structured learning and metric learning, the distance constraints of both the target samples and neighbor samples are taken into account, and the corresponding distance metric is learned. The learned metric can better measure the distances between the candidate samples and the target samples, as well as own high discriminability. The experimental results in the benchmark dataset demonstrate that the proposed tracking approach based on structured metric learning can acquire comparable performance to many state-of-the-art methods.

REFERENCES

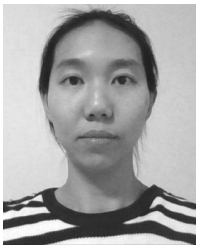
[1] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 798–805.
[2] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sep./Oct. 2009, pp. 1436–1443.
[3] 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.
[4] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1305–1312.
[5] Y. Yuan, H. Yang, Y. Fang, and W. Lin, "Visual object tracking by structure complexity coefficients," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1125–1136, Aug. 2015.
[6] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, Aug. 2004.
[7] 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.
[8] Y. Xie, W. Zhang, Y. Qu, and Y. Zhang, "Discriminative subspace learning with sparse representation view-based model for robust visual tracking," *Pattern Recognit.*, vol. 47, no. 3, pp. 1383–1394, Mar. 2014.
[9] S. Zhang, X. Yu, Y. Sui, S. Zhao, and L. Zhang, "Object tracking with multi-view support vector machines," *IEEE Trans. Multimedia*, vol. 17, no. 3, pp. 265–278, Mar. 2015.
[10] B. Ma, J. Shen, Y. Liu, H. Hu, L. Shao, and X. Li, "Visual tracking using strong classifier and structural local sparse descriptors," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1818–1828, Oct. 2015.
[11] R. Yao, S. Xia, Z. Zhang, and Y. Zhang, "Real-time correlation filter tracking by efficient dense belief propagation with structure preserving," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 772–784, Apr. 2017.
[12] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[13] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
[14] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT, 2007, pp. 153–160.
[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[16] I. Sutskever, "Training recurrent neural networks," Ph.D. dissertation, Univ. Toronto, Toronto, ON, Canada, 2013.
[17] C. Wu, C. Luo, N. Xiong, W. Zhang, and T.-H. Kim, "A greedy deep learning method for medical disease analysis," *IEEE Access*, vol. 6, pp. 20021–20029, 2018.
[18] J. Hu, J. Lu, Y.-P. Tan, and J. Zhou, "Deep transfer metric learning," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5576–5588, Dec. 2016.
[19] J. Lu, J. Hu, and Y.-P. Tan, "Discriminative deep metric learning for face and kinship verification," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4269–4282, Sep. 2017.
[20] C. Sánchez-Sánchez and D. Izzo, "Real-time optimal control via deep neural networks: Study on landing problems," *J. Guid., Control, Dyn.*, vol. 41, no. 5, pp. 1122–1135, Feb. 2018.
[21] N. D. Nguyen, T. Nguyen, and S. Nahavandi, "System design perspective for human-level agents using deep reinforcement learning: A survey," *IEEE Access*, vol. 5, pp. 27091–27102, 2017.
[22] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 263–270.
[23] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2411–2418.
[24] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 10, pp. 1025–1039, Oct. 1998.
[25] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, Jun. 2004.
[26] 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.
[27] C. Bao, Y. Wu, H. Ling, 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.
[28] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. Brit. Mach. Vis. Conf.*, vol. 1, Sep. 2006, pp. 47–56.
[29] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-tracking using semi-supervised support vector machines," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.
[30] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer, 2012, pp. 864–877.
[31] X. Li, A. Dick, C. Shen, A. van den Hengel, and H. Wang, "Incremental learning of 3D-DCT compact representations for robust visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 863–881, Apr. 2013.
[32] Y. Sui, Y. Tang, and L. Zhang, "Discriminative low-rank tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3002–3010.
[33] 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.
[34] S. Zhang, S. Zhao, Y. Sui, and L. Zhang, "Single object tracking with fuzzy least squares support vector machine," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5723–5738, Dec. 2015.
[35] A. Bellet, A. Habrard, and M. Sebban, "A survey on metric learning for feature vectors and structured data," Jun. 2013, *arXiv:1306.6709*. [Online]. Available: https://arxiv.org/abs/1306.6709
[36] B. Kulis, "Metric learning: A survey," *Found. Trends Mach. Learn.*, vol. 5, no. 4, pp. 287–364, 2013.
[37] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with sideinformation," in *Advances in Neural Information Processing Systems*, vol. 15. Cambridge, MA, USA: MIT Press, 2003, pp. 521–528.
[38] M. Schultz and T. Joachims, "Learning a distance metric from relative comparisons," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2004, pp. 41–48.
[39] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2004, pp. 513–520.
[40] A. Globerson and S. T. Roweis, "Metric learning by collapsing classes," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2005, pp. 451–458.
[41] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.

[42] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, *Learning for Large Scale Image Classification: Generalizing to New Classes at Near-Zero Cost*. Berlin, Germany: Springer, 2012, pp. 488–501.
[43] N. Verma, D. Mahajan, S. Sellamanickam, and V. Nair, "Learning hierarchical similarity metrics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2280–2287.
[44] H. Wang, L. Feng, J. Zhang, and Y. Liu, "Semantic discriminative metric learning for image similarity measurement," *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1579–1589, Aug. 2016.
[45] J. Liang, Q. Hu, W. Wang, and Y. Han, "Semisupervised Online multi-kernel similarity learning for image retrieval," *IEEE Trans. Multimedia*, vol. 19, no. 99, pp. 1077–1089, May 2016.
[46] N. Jiang, W. Liu, and Y. Wu, "Learning adaptive metric for robust visual tracking," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2288–2300, Aug. 2011.
[47] N. Jiang, W. Liu, and Y. Wu, "Adaptive and discriminative metric differential tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1161–1168.
[48] X. Li, C. Shen, Q. Shi, A. Dick, and A. van den Hengel, "Non-sparse linear representations for visual tracking with Online reservoir metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1760–1767.
[49] Y. Wu, B. Ma, M. Yang, J. Zhang, and Y. Jia, "Metric learning based structural appearance model for robust visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 5, pp. 865–877, May 2014.
[50] S. Yi, N. Jiang, X. Wang, and W. Liu, "Individual adaptive metric learning for visual tracking," *Neurocomputing*, vol. 191, pp. 273–285, May 2016.
[51] L. He and H. Zhang, "Kernel K-means sampling for Nyström approximation," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2108–2120, May 2018.
[52] L. He, N. Ray, Y. Guan, and H. Zhang, "Fast large-scale spectral clustering via explicit feature mapping," *IEEE Trans. Cybern.*, vol. 99, no. 3, pp. 1058–1071, Mar. 2018.
[53] C. Leng, G. Cai, D. Yu, and Z. Wang, "Adaptive total-variation for non-negative matrix factorization on manifold," *Pattern Recognit. Lett.*, vol. 98, pp. 68–74, Oct. 2017.
[54] M. Everingham, L. Gool, C. Williams, and A. Zisserman. (2005). *Pascal Visual Object Classes Challenge Results*. [Online]. Available: http.www.pascal-network.org
[55] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1838–1845.
[56] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.
[57] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1822–1829.
[58] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1177–1184.
[59] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1269–1276.
[60] K. Junseok and K. M. Lee, "Tracking by sampling trackers," in *Proc. Int. Conf. Comput. Vis.(ICCV)*, Nov. 2011, pp. 1195–1202.
[61] J. A. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer-Verlag, 2012, pp. 702–715.
[62] 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.
[63] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1910–1917.
[64] J. Wen and X.-W. Chang, "On the KZ reduction," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1921–1935, Mar. 2019.
[65] J. Wen, K. Wu, C. Tellambura, and P. Fan, "Closed-form word error rate analysis for successive interference cancellation decoders," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 8256–8267, Dec. 2018.

XIAOLIN ZHAO received the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2011. He is currently with the Equipment Management and UAV Engineering College, Air Force Engineering University, Xi'an, China. His current research interests include computer vision, image/video processing, and pattern cognition.



ZHUOFAN XU received the Ph.D. degree in control science and engineering from Air Force Engineering University, China, in 2018. He is currently with the Joint Operations College, National Defense University, Shijiazhuang, China. His current research interests include UAV intelligent control and airspace control.



BOXIN ZHAO received the Ph.D. degree from the College of Mechatronics and Automation, National University of Defense Technology, Changsha, China, in 2016. She is currently with the Equipment Management and UAV Engineering College, Air Force Engineering University, Xi'an, China. Her current research interests include computer vision and UAV localization.

XIAOLONG CHEN received the bachelor's degree in automation from Tsinghua University, in 2009, and the master's degree in embedded systems from ISAE, France, in 2014. He is currently with the Flight Automatic Control Research Institute, Xi'an, China. His current research interests include unmanned aircraft control and computer vision.

ZONGZHE LI received the Ph.D. degree from the Institute of Software, School of Computer Science, National University of Defense Technology, Changsha, Hunan, in 2012. He is currently with the Equipment Management and UAV Engineering College, Air Force Engineering University, Xi'an, China. His current research interests include computer vision, parallel computing, and machine learning.