PAPER

Robust Visual Tracking Using Sparse Discriminative Graph Embedding

Jidong ZHAO[†], Nonmember, Jingjing LI^{†a)}, Member, and Ke LU[†], Nonmember

SUMMARY For robust visual tracking, the main challenges of a subspace representation model can be attributed to the difficulty in handling various appearances of the target object. Traditional subspace learning tracking algorithms neglected the discriminative correlation between different multi-view target samples and the effectiveness of sparse subspace learning. For learning a better subspace representation model, we designed a discriminative graph to model both the labeled target samples with various appearances and the updated foreground and background samples, which are selected using an incremental updating scheme. The proposed discriminative graph structure not only can explicitly capture multi-modal intraclass correlations within labeled samples but also can obtain a balance between within-class local manifold and global discriminative information from foreground and background samples. Based on the discriminative graph, we achieved a sparse embedding by using $L_{2,1}$ -norm, which is incorporated to select relevant features and learn transformation in a unified framework. In a tracking procedure, the subspace learning is embedded into a Bayesian inference framework using compound motion estimation and a discriminative observation model, which significantly makes localization effective and accurate. Experiments on several videos have demonstrated that the proposed algorithm is robust for dealing with various appearances, especially in dynamically changing and clutter situations, and has better performance than alternatives reported in the recent literature.

key words: visual tracking, sparse subspace, discriminative graph

1. Introduction

Visual tracking [1]–[5] has been one of the significant issue in computer vision, as it is of great value in such applications as surveillance, human-computer interaction, traffic monitoring and so on. In visual tracking, accurate tracking of targets in a dynamic environment faces so many challenges such as various appearance changes caused by illumination and scaling, 3D pose variations, partial occlusions and complex background clutter.

Among a variety of tracking algorithms, one popular class is model-based tracking, which ranges from view-based appearance models [6], integration of shape and color [7], 3D models [8], foreground/background models [9], templates with updating [10], kernel-based filters [11], support vector machines [12], subspace learning [13], [14] and local sparse appearance model [15]. These methods usually build or learn a model of the target object first and then use it for visual tracking, so the key problem is how to design or construct a suitable object model, which

should be robust to a wide variety of situations.

In fact, the visual tracking problem can be viewed as a two-class image classification problem, that is, it should recognize the tracked target from the similar background samples in tracking procedure. In image recognition, subspace representation can effectively improve both the problem of unclear sparse feature representation and the problem of overfull computation consumption, so it is particularly suitable for a visual tracking environment with realtime requirement [14], [16], [17]. In a two-class classification problem, a linear discriminant analysis (LDA) model had been proved to achieve the discriminative subspace by separating one class from the other. However, LDA has some limitations, such as the rank deficiency of the between-class scatter matrix and the neglect of local manifold structure. So some related work aims to solve these problems by introduce local information into LDA framework [18]–[20], which can be classified as graph embedding techniques by constructing two graphs. For handling various appearances of a target in visual tracking, according to the good performance in preserving the essential characteristics, such supervised graph embedding algorithms should be an effective way to obtain a good subspace model.

In recent years, sparse subspace learning has been proposed [21]–[24], aiming to combine two useful dimensionality reduction techniques: feature selection and subspace learning. As a novel dimensionality reduction technique, sparse subspace learning integrates feature selection and subspace transformation into a unified framework, for example, the row sparsity of the projection matrix can be obtained by l_2 -norm regularization [21]. For multi-modal target with various appearance changes, more original features should be used before projecting for achieving a more discriminative subspace model. At this time, the row sparsity of the projection matrix can exclude some irrelevant features from the subspace learning process to improve subspace-based tracking.

Furthermore, incremental learning has provided an effective way to tackle the changing target in tracking procedure. Particularly, incremental subspace learning and its variants have become more and more popular and helpful in visual tracking. In fact, the environment is not always constant and sometimes changing drastically. Therefore, incremental subspace learning could be an easy way to capture the various changing appearances, while needing less computation and storage cost. Some previous work [14], [16], [27] had proved that this method could effectively learn the

Manuscript received December 10, 2014.

Manuscript publicized January 19, 2015.

[†]The authors are with the School of Compute Science & Engineer, University of Electronic Science & Technology of China, Chengdu, 611731, PRC.

a) E-mail: lijing117@yeah.net

DOI: 10.1587/transinf.2014EDP7419

various changing appearances using an incremental updating scheme. However, such work neglected the multi-modal information of the target and the efficiency of sparse subspace.

In this paper, we propose a visual tracking algorithm called discriminative graph embedding tracking (DGET) to carry out sparse discriminative feature extraction in the LDA framework. Before tracking, there may be some multi-view labeled object samples (including shape, rotation, illumination, or partial occlusion changes) available. For supervised graph embedding, the natural way to make use of labeled information is to incorporate it into the graph structure by appropriate design. Moreover, inspired by sparse subspace learning, our algorithm can achieve a sparse embedding by using $L_{2,1}$ -norm in the LDA framework, which results in selecting relevant features and learning transformation simultaneously. At last, by updating the recent foreground subgraph and the latest background subgraph in tracking process, a reasonable incremental learning mechanism is designed to make the graph model more suitable for robust tracking and better handle *drift* problem during tracking procedure. The main contributions of our proposed tracking algorithm are summarized as follows:

- A sparse discriminative graph embedding is proposed to capture the multi-modal local information of the target, making the foreground and background samples more discriminative.
- An incremental updating scheme is proposed to make classifying the tracked target from the background samples more effective.
- The subspace learning procedure is well embedded into a Bayesian inference framework, which is implemented by a compound motion model and a discriminative observation model.

This paper is arranged as follows. Section 2 presents the related work, including operating locality preserving projection (LPP) [25] over LDA (which is simply called LPP-over-LDA), sparse subspace learning and tracking based on subspace representation model. The detail of the sparse discriminative graph embedding is described in Sect. 3. Section 4 gives the detail of our visual tracking algorithm. Experiments are shown in Sect. 5, and Sect. 6 is devoted to conclusion.

2. Related Work

In this section, we first introduce two related techniques, LPP-over-LDA framework and sparse subspace learning, used in designing our algorithms. Then, we review some tracking algorithms based on subspace model.

2.1 LPP-over-LDA

In recent years, one kind of supervised graph embedding framework has been proposed by introducing LPP into

the LDA framework, which can be simply called LPP-over-LDA. Several typical LPP-over-LDA algorithms are LFDA [18], LDE [19] and LSDA [20], which mainly aim to better preserve both discriminative and local intrinsic information.

In LPP-over-LDA, it aims to construct a between-class graph $\{G_b, W_b\}$ and a within-class graph $\{G_w, W_w\}$. According to Laplacian criteria, the objective functions can be achieved:

$$f_w = \sum_{i,j} (y_i - y_j)^2 \cdot w_{w,ij} = y^T L_w y \quad (1)$$

$$f_b = \sum_{i,j} (y_i - y_j)^2 \cdot w_{b,ij} = y^T L_b y \quad (2)$$

where $y = [y_1, \dots, y_m]$ is the low-dimensional subspace, $w_{w,ij}$ and $w_{b,ij}$ are the element of weight matrix W_w and W_b . L_w and L_b are the graph Laplacians of G_w and G_b .

LPP-over-LDA aims to maximize the between-class distances while keeping the within-class compactness at a constraint. Applying the idea of Fisher's criteria, and suppose $y = X^T A$. So the objective function can be obtained as:

$$\min \frac{f_w}{f_b} = \min \frac{y^T L_w y}{y^T L_b y} = \min \frac{A^T X L_w X^T A}{A^T X L_b X^T A} \quad (3)$$

So A can be achieved by:

$$X L_w X^T A = \lambda X L_b X^T A \quad (4)$$

LPP-over-LDA has been demonstrated to work well in supervised image recognition tasks, because the local manifold structures and the global discriminative information are considered simultaneously. However, the graphs of the existing LPP-over-LDA methods are too simple, may not be suitable for multi-view image recognition tasks. We design a novel supervised graph structure to overcome this problem.

2.2 Sparse Subspace Learning

In recent years, sparse subspace learning has been proposed and demonstrated to be effective in image recognition and data analysis. Cai et al. presented sparse subspace learning (SSL) [22], which introduces l_1 -norm regularization into every single column of the projection. However, each column of the transformation matrix is optimized one by one and their sparsity patterns are independent. Masaeli et al. provided linear discriminant feature selection (LDFS) [24], which modifies LDA with l_∞ -norm regularization. However, it needs expensive computation cost. Gu et al. proposed a framework for combining feature selection and subspace learning (FSSL) [21] and applied $L_{2,1}$ -norm on the subspace projection. The objective function of FSSL is shown as follows:

$$\begin{aligned} \min_A & \text{tr}(A^T X L X^T A) + \mu \|A\|_{2,1} \\ \text{s.t. } & A^T X D X^T A = I \end{aligned} \quad (5)$$

while D is a diagonal matrix whose entries are column sums of weight matrix W , $L(L = D - W)$ is the graph Laplacian, and μ is the balance parameter.

Inspired by FSSL, which properly joints feature selection and subspace embedding when achieving their row-sparse transformation matrix, our method will achieve sparse embedding by introducing $L_{2,1}$ -norm in LPP-over-LDA framework.

2.3 Tracking Based on Subspace Representation Model

Tracking based on subspace learning algorithms utilize a flattened vector to represent a target, resulting in a dimensionality reduction problem.

Classical subspace-based tracking approach [6] is enhanced by incremental subspace updating in Incremental Visual Tracking (IVT) [14]. IVT uses PCA to obtain subspace model, whose eigenbasis and mean are updated online. Thus, IVT will suffer a more serious loss when tracking is not exact, although it brings in a forgetting factor. Visual Tracking Decomposition (VTD) [16] applied sparse principal component analysis (SPCA) to construct multiple basic observation models of a set of feature templates. However, PCA and SPCA may fail when the intrinsic structure of data set is nonlinear.

Graph embedding framework [26] can work well when data set is nonlinear. Zhang et al. proposed Tracking based on Graph (GT) [27], where a series of foreground samples and background samples are collected to construct within-class and between-class graph. It then aimed to evaluate within-class compactness and between-class separability. Qiao et al. focused on constructing a graph to preserve closed distance among some labeled head samples with accurate rotating angle for face tracking [29]. However, their graph structure are too simple for robust visual tracking task, and both neglected the multi-modal property in target set. Lu et al. used multi-modal target information to construct a locally connected graph (LCG) [28]. LCG could obtain a compact subspace model for visual tracking, but it did not take into account the effectiveness of including background samples and sparse subspace learning.

3. Sparse Discriminative Graph Embedding for Visual Tracking

In this section, a discriminative graph structure is designed to capture the multi-modal information of labeled data, and make the tracking more robust by separating the updated foreground samples from the around background ones. Then, we introduce our sparse embedding on the constructed graphs, which can make feature selection and subspace learning simultaneously. Such a sparse embedding is more suitable for tracked target with various appearances.

3.1 Discriminative Graph Structure

In visual tracking, the target would show various appear-

ances, due to changing illumination, partial occlusion, and so on. In such situations, it is necessary to use the intra-class information to discover the multi-modal information, and appropriate between-class structure to construct a suitable graph.

In this paper, we develop a discriminative graph structure, which aims to make the sub-clusters of the target close to each other and keep the nearby foreground and background samples more discriminative. The details of the graph structure are shown as follows.

Given three datasets $\{\mathcal{F}_s, \mathcal{F}_d, \mathcal{B}_d\}$, the initial foreground dataset $\mathcal{F}_s$, the dynamic foreground dataset $\mathcal{F}_d$ and the dynamic background dataset $\mathcal{B}_d$. Before tracking, $\mathcal{F}_d$ and $\mathcal{B}_d$ are both empty. We only can apply the $\mathcal{F}_s$, which is multi-modal, to learn the initial subspace. When tracking, we use incremental updating scheme to construct $\mathcal{F}_d$ and $\mathcal{B}_d$, which can be better deal with the current complex situation in tracking, except for some special occasions (e.g. the abrupt reverse change of background and target). The topology structures of discriminative graphs are modeled as a between-class graph $\{G_b, W_b\}$ and a within-class graph $\{G_w, W_w\}$ depicted in the following.

- Construct the within-class graph $\{G_w, W_w\}$

As shown above, the points of $\mathcal{F}_s$ can be classified into several sub-clusters according to the intraclass information. Suppose that the object contains γ sub-clusters $\mathcal{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_\gamma\}$. In Fig. 1(a), images of the object belonging to different modalities (sub-clusters) are represented by different colors. The intuition is that we should connect points within one sub-cluster together while keeping the ones between different sub-clusters near to each other. Firstly, for each x_j , an edge is added between x_k and x_j , if x_k is one of x_j 's k nearest neighbors, and they belong to one sub-cluster. For each pair of sub-clusters $\{(\mathcal{S}_k, \mathcal{S}_j), j \neq k\}$, we add a constrained link between the nearest pair of points from the two sub-clusters. The result graph is shown in Fig. 1.

When $\mathcal{F}_d$ contains samples, we can make use of these two foreground datasets to construct our within-class graph. To preserve more local information, we apply the k -nearest neighbor method to link the points within $\mathcal{F}_d$, which can help handle drift problem, as the samples achieved frames before and after show a high similarity. Meanwhile, the initial foreground dataset $\mathcal{F}_s$ is still useful in the tracking process. But $\mathcal{F}_d$ should be paid heavily dependance in tracking process. So we add constrained links between $\mathcal{F}_d$ and sub-clusters of $\mathcal{F}_s$, and clear away the constraints between sub-clusters of $\mathcal{F}_s$.

Besides, as the samples of $\mathcal{B}_d$ are collected around the foreground target, $\mathcal{B}_d$ also shows multi-modal structure. Therefore, we use the method, which is applied to $\mathcal{F}_s$, to construct the within-class graph of $\mathcal{B}_d$, shown in Fig. 2.

- Construct the between-class graph $\{G_b, W_b\}$

Unlike $\mathcal{F}_s$ which is constructed on a multi-modal train-

Fig. 1 The adjacency within-class graph. (a) The points of the different colors belong to different sub-clusters, according to the various appearances. (b) We set $k = 2$ in nearest neighbor searching to preserve the local structure. (c) We then connect these sub-clusters using constrained links. Therefore, it can well preserve the local geometrical structure of the data.

Fig. 2 The adjacency within-class and between-class graph. (a) To preserve the local intrinsic structure, k -nearest neighbor method is used within $\mathcal{F}_d$ and $\mathcal{B}_d$ ($k = 2$). (b) More utilize the initial samples to handle the drift problem in tracking, we apply constrained link between $\mathcal{F}_d$ and sub-clusters of $\mathcal{F}_s$. To preserve the multi-modal information of background samples, we add constraints between its each sub-clusters. (c) To achieve a better and more discriminative subspace, we should separate foreground and background samples far away. Here, we set the default number as 3.

ing set before tracking, both $\mathcal{F}_d$ and $\mathcal{B}_d$ are synchronously updated in tracking process, so the samples between $\mathcal{F}_s$ and $\mathcal{B}_d$ usually show a lower similarity compared with the ones between $\mathcal{F}_d$ and $\mathcal{B}_d$. In addition, recent tracking environment is very important for target recognition, so the discrimination between $\mathcal{F}_d$ and $\mathcal{B}_d$ is far more essential than that between $\mathcal{F}_s$ and $\mathcal{B}_d$. Therefore, in the between-class graph, we do not connect $\mathcal{B}_d$ and $\mathcal{F}_s$. We connect each pair between $\mathcal{F}_d$ and sub-clusters of $\mathcal{B}_d$, where the distance between two class is defined as the smallest distance between two respective member points of the two classes. We connect the corresponding nearest pair of points until the default number of connections is reached, shown in Fig. 2(c).

Our discriminative graph structure have several advantages: (1) The graph of $\mathcal{F}_s$ aims to better handle drift problem, especially when facing the abrupt transformation of the states defined by $\mathcal{F}_d$; (2) The within-class and between-class graph are current target-centered, that is, $\mathcal{B}_d$ is selected around the current target and $\mathcal{F}_d$ consist of the recent samples which are most relevant to the current one. This means these two graphs can preserve more information of the current target.

3.2 Sparse Embedding on Discriminative Graphs

After the discriminative graph is constructed, we can directly obtain the discriminative subspace. In the beginning, we apply LPP to initialize the multi-view low-dimensional subspace as we only achieve the labeled samples. When tracking, we obtain foreground and background samples, then use Eq. (3) to learn the discriminative low-dimensional subspace for tracking. Inspired by sparse subspace learning framework, the transformation matrix A which satisfies $y = X^T A$ should hold the sparse property for dimensionality reduction. A possible way is to find A which can best fit the low-dimensional feature y in the least squares sense:

$$\min \|y - X^T A\|_2^2 \quad (6)$$

If only a few elements of A are nonzero, then the learned subspace can preserve more important information. In order to make A sparse, $L_{2,1}$ regularization term is added, and it can make the transformation matrix row sparse, which results in feature selection. The $L_{2,1}$ -norm is defined as

$$\|A\|_{2,1} = \sum_{i=1}^n \sqrt{\sum_{j=1}^p ([A]_{ij})^2}.$$

Therefore, we obtain the final objective function as follows:

Algorithm 1: Sparse Embedding on Discriminative Graphs**Step 1:** Construct the Discriminative Graph Structure**Step 2:** Alternative Optimization**Initialize** $\Delta_0 = I, t = 0, \varpi, \sigma$;**repeat** Compute y_{t+1} using Eq. (11) Update A_{t+1} using Eq. (8) $t = t + 1$;**until** convergence**Step 3:** Subspace Embedding

$$\min(y^T L_w y + \varpi \|y - X^T A\|_2^2 + \sigma \|A\|_{2,1}) \quad (7)$$

$$s.t. \quad y^T L_b y = I$$

where ϖ and σ are two balance parameters.

We define $\mathcal{F} = y^T L_w y + \varpi \|y - X^T A\|_2^2 + \sigma \|A\|_{2,1}$. By differentiating $\mathcal{F}$ with respect to A and setting it to zero, we achieve:

$$A = (XX^T + \varpi \Delta / \sigma)^{-1} Xy = \Psi y \quad (8)$$

where Δ is a diagonal matrix whose i -th diagonal element $\Delta_{ii} = (\|\alpha_i\|_2)^{-1}$ if $\alpha_i \neq 0$. (α_i is the i -th row vector of A).

By substituting Eq. (8) into $\mathcal{F}$, we get:

$$\begin{aligned} \mathcal{F} &= y^T L_w y + \varpi \|y - X^T A\|_2^2 + \sigma \|A\|_{2,1} \\ &= y^T L_w y + \varpi (A^T X X^T A - 2A^T X y + y^T y) \\ &\quad + \sigma A^T \Delta A \\ &= y^T [L_w + \varpi (\Psi^T X X^T \Psi - 2\Psi^T X + I) \\ &\quad + \sigma \Psi^T \Delta \Psi] y \end{aligned} \quad (9)$$

Let $\mathcal{L} = L_w + \varpi (\Psi^T X X^T \Psi - 2\Psi^T X + I) + \sigma \Psi^T \Delta \Psi$ and Eq. (7) can be represented as

$$\min y^T \mathcal{L} y \quad (10)$$

$$s.t. \quad y^T L_b y = I$$

According to the Lagrangian method, the optimization problem can be solved by computing the following generalized eigenvector problem:

$$\mathcal{L} y = \lambda L_b y \quad (11)$$

In summary, the sparse subspace can be achieved in an alternative way, shown in the Algorithm 1.

4. Proposed Tracking Algorithm

4.1 Framework of Algorithm

Bayesian framework has provided a robust and effective framework in many tracking algorithms [14], [16], [27]. We introduce our sparse discriminative graph embedding into Bayesian framework to learn a flexible and effective tracking algorithm. The goal is to find the best configuration of the target with a given observation.

Let $c_t = (x_t, y_t)$ be the coordinates of the center of the detection window with width w_t , height h_t and scale s_t for the face in frame t . $\mathcal{X}_t = \{c_t, h_t, w_t, s_t\}$ is the state at time t , and $\mathcal{I}_{1:t}$ is the observation up to time t . The Bayesian tracking framework is then shown as follows:

$$p(\mathcal{X}_t | \mathcal{I}_{1:t}) \propto p(\mathcal{I}_t | \mathcal{X}_t) \int p(\mathcal{X}_t | \mathcal{X}_{t-1}) p(\mathcal{X}_{t-1} | \mathcal{I}_{1:t-1}) d\mathcal{X}_{t-1} \quad (12)$$

where $p(\mathcal{I}_t | \mathcal{X}_t)$ denotes the observation model that measures how much the target and observation at the proposed state coincide, and $p(\mathcal{X}_t | \mathcal{X}_{t-1})$ represents the motion model that proposes the next state $\mathcal{X}_t$ based on the previous state.

In the following, we will describe the compound motion model and discriminative observation model.

4.1.1 Compound Motion Model

In Eq. (12), the combined dynamic model $p(\mathcal{X}_t | \mathcal{X}_{t-1})$ describes the temporal correlation of the target states between consecutive frames. We can deploy the affine transformation to define motion model, that is, $p(\mathcal{X}_t | \mathcal{X}_{t-1}) = N(\mathcal{X}_t; \mathcal{X}_{t-1}, \Sigma)$, where Σ is a diagonal covariance matrix whose elements are the variances of the x , y position, height, width and scale of the object. The general case only defines one motion model, which is named as without estimation motion model in this paper.

In tracking process, the motion of the target can mainly be smoothly or drastically changing. In VTD, Kwon et al. separated the motion model into a combination of two basic motion models [16]. One is the smooth model, which aims to simulate the smaller motion changing. The other is the drastic model, which represents the abrupt motion. However, two motion models are a little simple in a complex environment, where the target may change in many motions. Therefore, in our algorithm, we update and combine the two basic motion models as:

$$\begin{aligned} p(\mathcal{X}_t | \mathcal{X}_{t-1}) &= \omega p_s(\mathcal{X}_t | \mathcal{X}_{t-1}) + (1 - \omega) p_a(\mathcal{X}_t | \mathcal{X}_{t-1}) \end{aligned} \quad (13)$$

where $\omega \in [0, 1]$ is the balance parameter. $p_s(\mathcal{X}_t | \mathcal{X}_{t-1})$ and $p_a(\mathcal{X}_t | \mathcal{X}_{t-1})$ describe two basic types of motions made by a Gaussian perturbation with a different variance. This case is referred as with estimation motion model in this paper. This practice can prevent from heavily depending on one model (smooth $p_s(\mathcal{X}_t | \mathcal{X}_{t-1})$ or abrupt $p_a(\mathcal{X}_t | \mathcal{X}_{t-1})$). And it can deal with multiple motions by changing the balance parameter ω .

4.1.2 Discriminative Observation Model

The observation model is a basic issue in tracking algorithms based on Bayesian framework. Many subspace learning algorithms [14], [27] have been embedded into the Bayesian framework to evaluate the observation model. In

our algorithm, we also introduce our sparse discriminative graph embedding into the Bayesian framework to achieve a discriminative observation model.

Consider an image patch $\mathcal{I}_t$ predicated by $\mathcal{X}_t$, and $\mathcal{I}_t$ was generated from a subspace of the target spanned by A and centered at $\bar{o}$. The probability of a sample being generated from this subspace is inversely proportional to the distance from the sample to the reference point of the subspace. The probability can be computed by [14]:

$$p(\mathcal{I}_t|\mathcal{X}_t) = \mathcal{N}(\mathcal{I}_t; \bar{o}, AA^T + \varepsilon I). \quad (14)$$

The discriminative observation model should be updated real-time to adapt to the changing target. How to update this model attributes to our incremental updating scheme in our graph. Next part, we will introduce the incremental updating scheme.

4.2 Incremental Updating

In most tracking tasks, it is necessary to cope with the changing of both the foreground and background. Therefore, it is important to update the graph structure to make it more suitable for the current situation. We apply a local manifold criterion for updating foreground samples, and a Heuristic criterion for selecting background samples.

To keep the within-class graph of the foreground data set more compact, a local manifold criterion is introduced to remove the sample most irrelevant to the foreground data set, whose size is limited to n . If the size is larger than n , when a new foreground sample comes, we first apply k -nearest neighbor method to find the intrinsic structure of the foreground data set. Then, we search and delete the point, which has the fewest degree in the graph. This practice not only can make the foreground samples more compact during tracking as it could help preserve the intrinsic feature of the foreground target, but also can handle the drift problem in the beginning, since the drift one has different intrinsic feature with other foreground samples. If more than one point has the same degree, the earliest one would be removed, and this is simply called time-forget-factor, which is similar to the scheme in IVT. This practice can remove the effectiveness of the older ones, as generally the older ones contribute less to the current target and more irrelative to the current one.

Besides, as depicted in [27], if the background sample lies too far from the foreground target, then it may be of no use to achieve a discriminative subspace. On the other hand, if the background sample lies too close to the foreground target, it may lie partly in the foreground data set. Then the estimated foreground target is pushed away from its true place. So we first apply the heuristic selection method [27] to choose the background samples. In subspace, let D_f is the distance between background candidate and the center of foreground sample set, and D_b is the distance between background candidate and the center of background sample set. We define two thresholds T_f and T_b , which are dynamic according to the required number of background samples.

Algorithm 2: Tracking Algorithm

Input: The state information $\mathcal{X}_t = \{c_t, h_t, w_t, s_t\}$ and the learned subspaces $\mathcal{O}_t = \{A, \lambda\}$ of frame t

1. Compound Motion Model

$p(\mathcal{X}_t|\mathcal{X}_{t-1}) = \omega p_s(\mathcal{X}_t|\mathcal{X}_{t-1}) + (1 - \omega)p_a(\mathcal{X}_t|\mathcal{X}_{t-1})$
 ω is chosen as $[0, 0.5, 1]$, resulting in three motion models.

2. Discriminative Observation Model

$p(\mathcal{I}_t|\mathcal{X}_t) = \mathcal{N}(\mathcal{I}_t; \bar{o}, AA^T + \varepsilon I)$

3. State Estimation

$p(\mathcal{X}_t|\mathcal{I}_{1:t}) \propto p(\mathcal{I}_t|\mathcal{X}_t) \oint p(\mathcal{X}_t|\mathcal{X}_{t-1})p(\mathcal{X}_{t-1}|\mathcal{I}_{1:t-1})d\mathcal{X}_{t-1}$

4. Incrementally Update foreground and background

5. Update Discriminative Subspace

Output: $\mathcal{X}_{t+1}, \mathcal{O}_{t+1} = \{A, \lambda\}$

The criterion of evaluating background samples is shown as follows:

$$D_f < T_f \text{ and } D_b < T_b \quad (15)$$

Second, we just select the background samples, which are the latest ones. This practice can make sure that the background samples are heavily depend on the current foreground target.

After updating the foreground and background data set, the graph structure can be reconstructed to learn a new discriminative observation model for tracking. In our paper, the strategy taken in [14] is adopted to incrementally learn the eigenbasis when new samples arrive.

4.3 Summary of Algorithm

A summary of the sparse discriminative graph embedding tracking algorithm is described in Algorithm 2. And a diagram of the proposed algorithm is depicted in Fig. 3.

Fig. 3 Diagram of the proposed tracking algorithm.

5. Experimental Results

In the section, we conduct several experiments to demonstrate the effectiveness of our algorithms. At the beginning, we manually initialize the related parameters. Also we collect several samples for constructing the static foreground graph. In the experiments, we choose ten target images consisting of several appearances. Several representative video sequences are used in our experiments, and they are Dudek[†] and David^{††}, faceocc2^{†††}, woman^{††††} and singer^{††††}. Table 1 shows the characteristics of these sequences.

We first test the importance of motion estimation model. Then, we compare our proposed tracking algorithm with several subspace based tracking algorithms, which are IVT, GT and superpixel tracking (ST) [30] and LCG [28]. In order to achieve the optimal subspace, LCG need enough training set with various appearance before tracking. Because of including background samples and using incremental updating scheme, DGET only need a small training set before tracking. For fair comparison, we use a similar small sample set (three samples in each sub-cluster) for DGET and LCG in experiments. The mean ratio of a target center's offset over the triangle area defined by two eyes and mouth is used to measure the performance.

In the first section, we test our proposed algorithm with and without motion estimation model. So we use David sequence, which contains a rapidly moving object. As shown in Fig. 4, we can see clearly the algorithm with the motion estimation can work well in frame 32, while the algorithm without motion model cannot handle the rapid motion of the object.

Table 1 The challenges of the experimental sequences.

Sequence	3D Pose	Illumination	Occlusion	Scaling
Dudek	large	small	none	small
David	large	large	small	small
faceocc2	none	middle	large	none

In this face tracking section, we compare DGET, GT, IVT, ST and LCG on two face sequences. We show center error of 5 algorithms on Dudek sequence in Fig. 5. Comparative results for selected frames are presented in Fig. 6, 7 and 8.

From the results, we get the following conclusions:

- IVT applies PCA to achieve the subspace which is updated online. With the number of the targets slowly increases, the eigenbasis may be optimized or not, as the added one can be exact-located and can be also false-located. Though bringing in the forgetting factor, IVT may abandon the good targets while adding bad ones. Besides, IVT can successfully capture the smooth changes of appearance, while it fails to effectively adapt to the abrupt changes and would enlarge this error tracking, such as David sequence (Fig. 8(b)). Meanwhile, IVT cannot work well in partial occlusion, such as faceocc2 sequence (Fig. 7(b)).
- GT introduces a motion estimation, which can better

Fig. 5 Center error result of 5 algorithms on Dudek sequence.

Fig. 4 Tracking results on David sequence of our algorithm, (a) shows the results without motion model, (b) shows the results with motion model.

[†]<http://www.cs.toronto.edu/~dross/ivt/>

^{††}http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml

^{†††}http://ice.dlut.edu.cn/lu/Project/iccv_spt_webpage/iccv_spt.htm

^{††††}<http://cv.snu.ac.kr/vtd>

Fig. 6 Comparative tracking results of (a) GT, (b) IVT, (c) ST, (d) LCG and (e) DGET on Dudek sequence at frame #54, #280, #394, #464, #512, #573.

Fig. 7 Comparative tracking results of (a) GT, (b) IVT, (c) ST, (d) LCG and (e) DGET on faceocc2 sequence at frame #51, #144, #423, #479, #501, #551.

adapt to abrupt changes in both appearance and illumination from a more complex situation. However, when the target is located with minor mistakes and benchmark of the next frame, the error will be enlarged. As GT aims to keeping within-class compactness, so the current target will be most falsely located when com-

paring with former error target.

- ST is provided recently which is based on Mean shift. It needs a larger cost of computation and storage when tracking for a relative long time. ST needs to generate the super-pixel regions in each tracking. Generally, it would take more than 20 minutes to tracking a 200-

Fig. 8 Comparative tracking results of (a) GT, (b) IVT, (c) ST, (d) LCG and (e) DGET on David sequence at frame #108, #184, #222, #227, #237, #246.

frame video, while other benchmarks only cost less than 1 minute. Besides, when tracking a long sequence, ST more easily obtained error sample than DGET, especially tracking the severe occluded sequence.

- Both of DGET and LCG combine the targets with a great variability to construct a stable subspace which uncovers the intrinsic multi-modal structure, so they have a high self-adaption to states transformation. But there is three differences between DGET and LCG. First, DGET includes background samples in graph structure, and obtains subspace in the LDA framework, so it can recognize the target much better. Second, the subspace model of DGET is optimized by sparse representation, so it can select important features for recognizing the target sample. Besides, DGET uses an incremental updating scheme for foreground and background samples, so it need less training samples before tracking and can handle the drifting much better.
- Our DGET can work better than IVT as DGET mainly introduces a compound motion model to well adapt to changing appearance. DGET can work better than LCG as DGET includes background samples in graph structure, and obtains subspace in the LDA framework. Besides, DGET performs better than GT as DGET introduces the sparse representation into the graph structure, which can better deal with the occluded situation.

From the above face tracking results, we can find that DGET can track the target much better and achieve a more

exact location than benchmarks. It is necessary to point out that DGET need some labeled samples with various appearances before tracking process. This condition could be difficult to meet at some time. In addition, tracking algorithms based on graph embedding, including DGET, generally cannot outperform other types of tracking algorithm (e.g. ST) in non-face tracking, perhaps because manifold structure of non-face target is unsuitable to be simulated by nearest neighbor graph. How to overcome these limitations still need further research.

6. Conclusion

For visual tracking, because the proposed discriminative graph can model the evolving processing of the target variation, our algorithm can obtain a good subspace mapping efficiently. Furthermore, our algorithm can improve the sparse feature representation problem by a natural combination of graph embedding and sparse subspace learning, so it can obtain an optimal subspace model. With the help of incremental learning and Bayesian based tracking framework, our algorithm can also effectively handle the *drift* problem. Thus our algorithm is adaptive, accurate, and robust. Our experiments demonstrate the effectiveness of the proposed tracker in various environments where the targets undergo large pose and lighting changes.

In the future, we aim to design a more sophisticated updating strategy for incremental graph learning, which uses less labeled samples, if possible just one known sample.

Acknowledgments

This work was supported in part by the National Science Foundation of China under Grant 61273254 and 61371183.

References

- [1] R. Yao, Q. Shi, C. Shen, et al, "Robust tracking with weighted on-line structured learning," European Conference on Computer Vision, Springer Berlin Heidelberg, pp.158–172, 2012.
- [2] J.H. Yoon and K.J. Yoon, "Visual tracking via adaptive tracker selection with multiple features," European Conference on Computer Vision, pp.28–41, Springer Berlin Heidelberg, 2012.
- [3] T. Zhang, B. Ghanem, S. Liu, et al, "Low-rank sparse learning for robust visual tracking," European Conference on Computer Vision, pp.470–484, Springer Berlin Heidelberg, 2012.
- [4] S. He, Q. Yang, R.W.H. Lau, et al, "Visual tracking via locality sensitive histograms," Computer Vision and Pattern Recognition, pp.2427–2434, 2013.
- [5] N. Wang, J. Wang, and D.-Y. Yeung, "Online robust non-negative dictionary learning for visual tracking," International Conference on Computer Vision, pp.657–664, 2013.
- [6] M.J. Black and A.D. Jepson, "Eigenttracking: Robust matching and tracking of articulated objects using viewbased representation," European Conference on Computer Vision, pp.329–342, 1996.
- [7] S. Birchfield, "Elliptical head tracking using intensity gradient and color histograms," Comput. Vision and Pattern Recognition, pp.232–237, 1998.
- [8] M.L. Cascia and S. Sclaroff, "Fast, reliable head tracking under varying illumination," Computer Vision and Pattern Recognition, pp.604–608, 1999.
- [9] M. Harville, "A framework for high-level feedback to adaptive, per-pixel mixture of Gaussian background models," European Conference on Computer Vision, pp.531–542, 2002.
- [10] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," IEEE Trans. Pattern Anal. Mach. Intell., vol.26, no.6, pp.810–815, 2004.
- [11] B. Georgescu, D. Comaniciu, T.X. Han, et al, "Multi-model component-based tracking using robust information fusion," 2nd Workshop on Statistical Methods in Video Processing, vol.3247, pp.61–70, 2004.
- [12] S. Avidan, "Support vector tracking," Computer Vision and Pattern Recognition, vol.1, pp.184–191, 2001.
- [13] J. Ho, K. Lee, M. Yang, et al, "Visual tracking using learned linear subspaces," Computer Vision and Pattern Recognition, vol.1, pp.782–789, 2004.
- [14] D. Ross, J. Lim, R.Lin, et al, "Incremental learning for robust visual tracking," International Journal of Computer Vision, Springer, vol.77, no.1, pp.125–141, 2008.
- [15] B.Y. Liu, J.Z. Huang, L. Yang, et al, "Robust tracking using local sparse appearance model and K-selection," Computer Vision and Pattern Recognition, pp.1313–1320, 2011.
- [16] J. Kwon and K.M. Lee, "Visual tracking decomposition," Computer Vision and Pattern Recognition, pp.1269–1276, 2010.
- [17] W.M. Hu, X. Li, X.Q. Zhang, et al, "Incremental tensor subspace learning and its applications to foreground segmentation and tracking," International Journal of Computer Vision, vol.91, no.3, pp.303–327, 2011.
- [18] M. Sugiyama, "Local fisher discriminant analysis for supervised dimensionality reduction," International Conference on Machine Learning, vol.148, pp.905–912, 2006.
- [19] H.T. Chen, H.W. Chang, and T.L. Liu, "Local discriminant embedding and its variants," Computer Vision and Pattern Recognition, pp.846–853, 2005.
- [20] D. Cai, X.F. He, K. Zhou, et al, "Locality sensitive discriminant analysis," International Joint Conference on Artificial Intelligence, AAAI, pp.708–713, 2007.
- [21] Q.Q. Gu, Z.H. Li, and J.W. Han, "Joint feature selection and subspace learning," International Joint Conference on Artificial Intelligence, AAAI, pp.1294–1299, 2011.
- [22] D. Cai, X.F. He, and J.W. Han, "Spectral regression: A unified approach for sparse subspace learning," International Conference on Data Mining, pp.73–82, 2007.
- [23] C.P. Hou, F.P. Nie, D.Y. Yi, et al, "Feature Selection via Joint Embedding Learning and Sparse Regression," International Joint Conference on Artificial Intelligence, AAAI, pp.1324–1329, 2011.
- [24] M. Masaeli, G. Fung, and J.G. Dy, "From transformation based dimensionality reduction to feature selection," International Conference on Machine Learning, pp.751–758, 2010.
- [25] X.F. He and P. Niyogi, "Locality preserving projections," Advance in Neural Information Processing Systems, pp.153–162, 2003.
- [26] S.C. Yan, D. Xu, B.Y. Zhang, et al, "Graph embedding and extensions: A general framework for dimensionality reduction," IEEE Trans. Pattern Anal. Mach. Intell., vol.29, no.1, pp.40–51, 2007.
- [27] X. Zhang, W. Hu, S. Maybank, et al, "Graph based discriminative learning for robust and efficient object tracking," International Conference on Computer Vision, pp.1–8, 2005.
- [28] K. Lu, Z. Ding, and S. Ge, "Locally connected graph for visual tracking," Neurocomputing, Elsevier, vol.120, pp.45–53, 2013.
- [29] H. Qiao, P. Zhang, B. Zhang, et al, "Learning an intrinsic variable preserving manifold for dynamic visual tracking," IEEE Trans. Syst., Man Cybern. B, vol.40, no.3, pp.868–880, 2010.
- [30] S. Wang, H.C. Lu, F. Yang, et al, "Superpixel Tracking," IEEE International Conference on Computer Vision, pp.1323–1330, 2011.

Jidong Zhao received his B.S. degree, M.Sc. and Ph.D. degrees in Computer Application Technology from the University of Electronic Science and Technology of China, in 2003, 2006, and 2009 respectively. He is currently an associate professor in School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include pattern recognition and computer vision.

Jingjing Li received his M.Sc. degree in Information Security from University of Electronic Science and Technology of China in 2013. Now he is a Ph.D. candidate in School of Computer Science and Technology, University of Electronic Science and Technology of China. He has a great interest in computer vision, especially video-based data analysis, and visual object tracking.

Ke Lu received his B.S. degree in thermal power engineering from Chongqing University, China in 1996. He obtained his M.Sc. and Ph.D. degrees in Computer Application Technology from the University of Electronic Science and Technology of China, in 2003 and 2006, respectively. He is currently a professor in School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include pattern recognition, multimedia and computer vision.