Graph Based Discriminative Learning for Robust and Efficient Object Tracking

Xiaoqin Zhang[1],Weiming Hu[1],Steve Maybank[2],Xi Li[1]
[1]National Laboratory of Pattern Recognition, Institute of Automation, Beijing, China
{xqzhang,wmhu,lixi}@nlpr.ia.ac.cn
[2]School of Computer Science and Information Systems, Birkbeck College, London, UK
sjmaybank@dcs.bbk.ac.uk

Abstract

Object tracking is viewed as a two-class 'one-versus-rest' classification problem, in which the sample distribution of the target is approximately Gaussian while the background samples are often multimodal. Based on these special properties, we propose a graph embedding based discriminative learning method, in which the topology structures of graphs are carefully designed to reflect the properties of the sample distributions. This method can simultaneously learn the subspace of the target and its local discriminative structure against the background. Moreover, a heuristic negative sample selection scheme is adopted to make the classification more effective. In tracking procedure, the graph based learning is embedded into a Bayesian inference framework cascaded with hierarchical motion estimation, which significantly improves the accuracy and efficiency of the localization. Furthermore, an incremental updating technique for the graphs is developed to capture the changes in both appearance and illumination. Experimental results demonstrate that, compared with two state-of-the-art methods, the proposed tracking algorithm is more efficient and effective, especially in dynamically changing and clutter scenes.

1. Introduction

Object tracking has received significant attention due to its crucial value in visual applications including surveillance, human-computer interaction, intelligent transportation, augmented reality and video compression.

In the literature, there exists a variety of tracking algorithms from different perspectives, such as the snakes model [1], condensation [2], mean shift [3], appearance models [4], the probabilistic data association filter [5] and so on. These algorithms have achieved great successes in object tracking. However, it is still a great challenge to build a visual tracking system that is robust to a wide variety of conditions, especially if the system is based on a mobile camera. In this case, the tracker must deal simultaneously with the changes of both target and background. One traditional solution to this problem is to design a updating scheme based on a constant brightness constraint to accommodate the changes in appearance and illumination [4, 6, 7]. The underlying assumption is that the intensity of pixels inside the target region remain the same between two consecutive frames. However, the tracking errors accumulate, causing the template to drift away from the target. An alternative approach is to construct an appearance model which takes account of all possible variations in the appearance of the target [8, 9]. As in [8], a view-based eigenbasis representation of the object is learned off-line, and applied to form a two-view matching tracking algorithm. However, it is very difficult to collect training samples that cover all possible cases. Thus, this algorithm is only feasible in some specific conditions.

Recently incremental learning has provided an effective way to tackle the above problem. Specially, incremental subspace learning and its extensions have received more and more attention due to the following merits [10, 11, 12, 13]: (1) constant subspace assumption is more reasonable than constant brightness assumption; (2) it is easy to capture the changes of the appearance; (3) it is computation and storage efficiency. The pioneering work applying the incremental subspace learning to tracking is due to Lim *et al.* [12], where they extend the SKL (Sequential Karhunen-Loeve) [14] algorithm to effectively learn the variations of both appearance and illumination in an incremental way. However, their work only focuses on the matching between target subspace and candidates. The information for classification in the background is discarded. In [13], a two-class FDA (Fisher Discriminant Analysis) based model is proposed to learn the discriminative subspace to separate the target from the background. It has a more discriminative ability than PCA models, since it utilizes the background appearance as negative training data. Despite the success of FDA in the tracking literature, it still has the following limitations:1) the dimension of the embedding space is lower than the

class number due to the rank deficiency of the between-class scatter matrix, and that is rather restrictive in two-class classification problem; 2) it is optimal only in the case that the data for each class are approximately Gaussian distributed with equal covariance matrix. In fact, the sample distribution of background is usually *multimodal* and *irregular*, making FDA ineffective in this case.

In view of the forgoing discussions, we propose a graph embedding framework to combine ISL (incremental subspace learning) and FDA (Fisher discriminant analysis) for object tracking, and simultaneously compensate the limitations of both ISL and FDA. While maintaining a relatively low computational complexity, the proposed tracking algorithm performs quite robustly in dynamically changing and clutter environments. The main contributions of the proposed tracking algorithm are summarized as follows:

- The subspace of the target and its local discriminative structure against the background are learned simultaneously from a graph embedding based learning framework to effectively capture the variational appearance changes and reliably separate the target from the background.
- A heuristic negative sample selection scheme is proposed to make the classification between target and background more effective.
- The learning procedure is embedded into a Bayesian inference framework cascaded with a hierarchical motion estimation algorithm in order to improve the accuracy and efficiency of the localization.

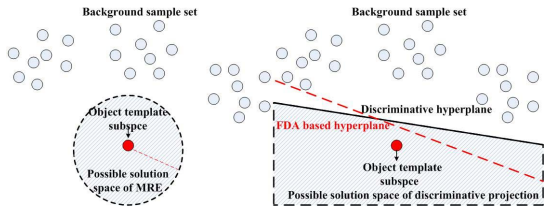
This paper is arranged as follows. Section 2 presents the graph embedding based discriminative learning method. The detail of the proposed tracking algorithm is described in Section 3. Experimental results are shown in Section 4, and Section 5 is devoted to conclusion.

2. Graph Embedding Based Learning

Graph embedding is a particular design of a graph with some special constraints[17, 18]. In [17], Yan *et al.* propose a graph embedding framework for dimension reduction, which reformulates the classic PCA and FDA in a graph embedding manner. Inspired by their work, we propose a graph embedding based method to effectively learn the variational appearance changes and the discriminative structure between the target and background.

2.1. Problem

As discussed in the introduction, ISL based tracker can gradually capture variational changes of the target, however discarding the information in the background makes it rather restrictive in cases when the object undergoes large appearance changes, because the minimization of reconstruction error provides a limited solution space (as illustrated in Fig.1(a)). The FDA based tracker takes account of background appearance as negative training samples, enhancing the tracking performance to some degree. However, as shown in Fig.1(b), it tends to give undesired results if training samples in a certain class are *multimodal* [15, 16], which is often observed in tracking applications. So the question is, despite the clutter environment, how to effectively learn the variational changes of the target while preserving the ability to discriminate the target from the background.



(a) (b)

Figure 1. The possible solution space in (a) the minimization of reconstruction error, and (b) the FDA projection (dashed line) and discriminative projection (solid line).

2.2. Graph Embedding for Dimension Reduction

Before introducing our work, let us review the graph embedding framework for dimension reduction [17].

Let $x_i \in \mathbb{R}^d (i = 1, 2, \cdots, n)$ be d-dimensional samples and $y_i \in \{1, 2, \cdots, C\}$ be associated class labels. Let n_c be the number of samples in the class c, where $\sum_{c=1}^{C} n_c = n$. The sample matrix is written as: $X = (x_1|x_2|\cdots|x_n)$. Let $G = \{\{x_i\}_{i=1}^n, W\}$ be an undirected weighted graph with vertex set $\{x_i\}_{i=1}^n$and the similarity matrix $W \in \mathbb{R}^{n\times n}$. The element w_{ij} of Wmeasures the similarity of the vertex pair i and j. The element of diagonal matrix D and the Laplacian matrix L of the graph G are defined as follows.

$$d_{ii} = \sum_{j\neq i} w_{ij},\ L = D - W \tag{1}$$

The graph embedding for dimension reduction is defined as the optimal low dimensional vector representations for the vertices of graph G that best characterize the similarity relationship between the data pairs. A general form is to minimize the *graph preserving criterion* as follows.

$$\begin{aligned} Z^* &= \arg\min_{Z^TBZ=I} \sum_{i,j} ||z_i - z_j||^2 w_{ij} \\ &= \arg\min_{Z^TBZ=I} 2tr(Z^TLZ) \end{aligned} \tag{2}$$

where z_i is the low dimension representation of x_i, Z is its data matrix, and B constrains the low dimensional representation. Suppose only linear projection as $z_i = P^T x_i$ is considered, and the constant factor in (2) is dropped for simplicity. Thus the objective function (2) becomes

$$P^* = \arg\min_{P^TXBX^TP=I} tr(P^TXLX^TP) \tag{3}$$

PCA pursues a subspace containing the maximum-variance directions in the original space, which can be obtained by solving the eigenstructure decomposition of covariance matrix S.

$$S = \sum_i (x_i - \mu)(x_i - \mu)^T = X(I - \frac{1}{n} ee^T)X^T \quad (4)$$

where e is an n-dimensional vector with $e = [1, 1, \cdots, 1]^T$, and μ is the mean of all samples. Thus PCA can be reformulated as

$$\begin{aligned} P^* &= \arg \min_{P^T P = I} - tr(P^T S P) \\ &= \arg \min_{P^T P = I} - tr(P^T X (I - \frac{1}{n} ee^T) X^T P) \end{aligned} \quad (5)$$

with the graph structure $\{w_{ij} = 1/n, i \neq j; B = I\}$.

FDA embeds the training samples so that the ratio of within-class scatter matrix $S^{(w)}$ and between-class scatter matrix $S^{(b)}$ is minimized.

$$S^{(w)} = \sum_{c=1}^{C} \sum_{i:y_i=c} (x_i - \mu_c)(x_i - \mu_c)^T = X(I - \frac{1}{n_c} \sum_{c=1}^{n_c} e^c e^{cT}) X^T \quad (6)$$

$$S^{(b)} = \sum_{c=1}^{C} n_c (\mu_c - \mu)(\mu_c - \mu)^T = S - S^{(w)} \quad (7)$$

where $\sum_{i:y_i=c}$ denotes the summation over sample x_i such that $y_i = c$, μ_c is the mean of samples in class c and e^c is an n dimensional vector with $e^c(i) = 1$, if $y_i = c$. As a result, the object function of FDA can be described as follows.

$$P^* = \arg\min_P tr(\frac{P^T S^{(w)} P}{P^T S^{(b)} P}) = \arg\min_P tr(\frac{P^T S^{(w)} P}{P^T S P}) \quad (8)$$

with the graph structure $\{w_{ij} = \delta_{y_i,y_j}/n_{y_i}\}$, and the constraint $\{B = I - \frac{1}{n} ee^T\}$. Here δ_{y_i,y_j} is the function defined such that $y_i = y_j$, $\delta_{y_i,y_j} = 1$, otherwise $\delta_{y_i,y_j} = 0$.

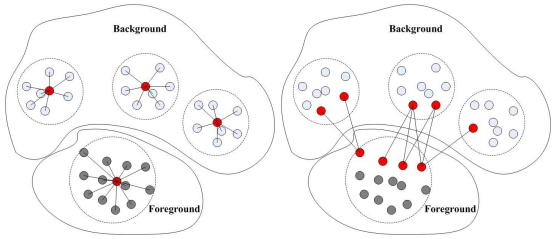
2.3. Graph Embedding Based Learning

An investigation [16] shows that the undesired behaviorz of FDA in *multimodal* case is caused by the *globality* when evaluating within-class compactness and between-class separability. Since FDA maximizes between-class separability under constraint of keeping within-class compactness to a certain level, when one of the classes is *multimodal*, this constraint is actually quite restrictive since these data samples should be typically evaluated as a single cluster. Therefore, the ability for maximizing the between-class separability is rather limited. A proper way to overcome the above limitation is to evaluate within-class compactness and between-class separability in a local manner to preserve the *multimodal* structure.

In the following part, we construct three novel graphs with topology structures designed to reflect the properties of the sample distributions.

2.3.1 Graph Structure

Suppose we have collected a series of positive and negative samples corresponding to the target and background in tracking applications. Recall that the data points $\{x_i\}_{i=1}^n$ are in $\mathbb{R}^d$, and each x_i is labeled by a class label $y_i \in \{1, 2\}$. The topology structures of graphs are designed as follows.



(a) Within-class Graph (b) Between-class Graph

Figure 2. The adjacency graphs for within-class compactness and between class separability (Note that the adjacency graphs only plot the connection edges for some typical samples for simplicity)

- Construct the target/foreground graph $\{G^f, W^f\}$.
  The PCA graph of foreground is constructed using the affinity matrix $\{w_{ij}^f = 1/n_f\}$, where n_f is the number of target samples, as illustrated in Fig.2(a), because the sample distribution is approximated by a Gaussian.
- Construct the background graph $\{G^b, W^b\}$.
  As shown in Fig.2(a), among the background samples, an edge is added between x_i and x_j, if x_j is one of x_i's k-nearest neighbors. Each element w_{ij}^b of the affinity matrix refers to the weight of the edge between x_i and x_j, and is determined by the local scaling method in [19]
  $$w_{ij}^b = exp\left(-\frac{||x_i - x_j||^2}{\sigma_i \sigma_j}\right) \quad (9)$$
  where σ_i represents the local scaling of the data samples around x_i, which is defined by
  $$\sigma_i = ||x_i - x_i^{(k)}|| \quad (10)$$
  where $x_i^{(k)}$ is kth nearest neighbor of x_i. In [19], $k = 7$ is a *universal* value, by which no tuning parameter remains, and it can effectively deal with data samples that are distributed of different scales. By default, $w_{ij}^b = 0$, if x_i and x_j are not connected.
- Construct the between-class graph $\{G', W'\}$.
  For G', we instead consider each pair of x_i and x_j with $y_i \neq y_j$, and likewise, connect x_i and x_j, if x_j is one of x_i's k'-nearest neighbors. The affinity matrix W' is also computed by the local scaling method. As shown in Fig.2(b), maximizing the between-class separability defined in this way has a similar meaning to maximize the margin between the two classes. And it is computationally efficient to only focus on the marginal samples.

2.3.2 Subspace Learning

Given the graph structures, the subspace of the target and the discriminative projection between two classes are obtained in the following steps.

Step 1. Learn the subspace P of the target by solving the Eq.(5) of the foreground graph.

Step 2. To obtain the discriminative projection V, we focus on the following constrained optimization problem.

$$\text{Maximize } J(V) = \sum_{i,j} ||V^T x_i - V^T x_j||^2 w'_{ij}$$

$$\text{subject to } \sum_{i,j} ||V^T x_i - V^T x_j||^2 w_{ij} = 1$$

where $$W = \begin{pmatrix} W^f & 0 \\ 0 & W^b \end{pmatrix}$$

the columns of the optimal solution V are the generalized eigenvectors corresponding to the l largest eigenvalues in

$$X(D' - W')X^T v = \lambda X(D - W)X^T v \qquad (11)$$

where D and D' are diagonal matrices defined in (1), and the discriminative projection is formed as $V = [v_1, v_2, \cdots, v_l]$. The proof is given in the Appendix.

As demonstrated in Fig.1(b), our approach can obtain a more discriminative projection than FDA in the *multimodal* case. In fact, it can also achieve comparable performance with FDA in the *Gaussian* case.

3. Proposed Tracking Algorithm

3.1. Overview of the Approach

Bayesian inference has provided a flexible and effective tracking framework. Therefore, we embed the graph based discriminative learning into Bayesian inference framework to form a robust tracking algorithm. The proposed tracking algorithm is schematically shown in Fig.3. First, the SSD (sum of squared differences) [20] iteration is applied to the current frame to estimate the motion of the object. The refined prediction of the state vector provides directional information to the particle generation process, and the number of particles as well as the region they covered are controlled by the residual error of the prediction. After the particle generation process, each particle is then evaluated by the discriminative observation model, which is learned via the graph embedding based method. A maximum a posterior (MAP) estimate of state is obtained as the output, and also is retained as a positive training sample. Meanwhile, some negative samples are carefully selected according to a heuristic strategy. Finally, the graph embedding structure and the SSD template are incremental updated when the training samples are ready.

Below we give a detailed description about each component in this framework, and the algorithm is summarized finally.

3.2. Hierarchical Motion Estimation

The motivation of cascading the SSD [20] algorithm with the Bayesian inference framework in our tracking algorithm is to provide a heuristic prediction to the particle generation process.

Suppose the target is well localized in frame $t-1$ as illustrated by the left column of Fig.4, and the corresponding state is denoted as s_{t-1}. We first apply SSD iterations to

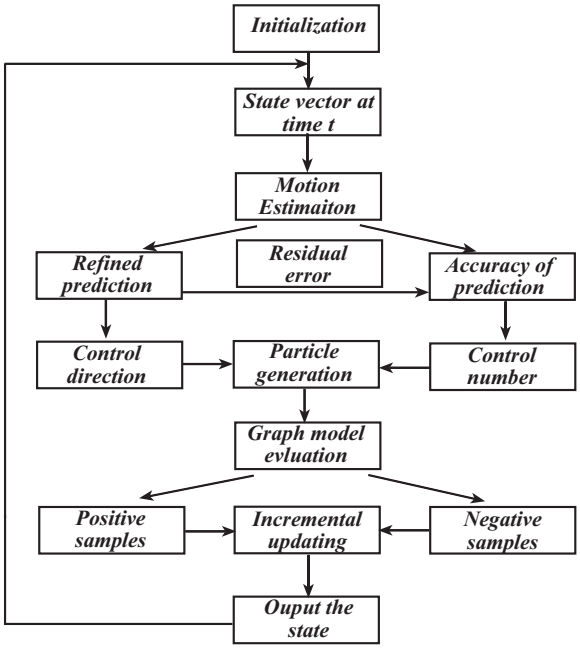


Figure 3. Overview of the proposed tracking algorithm

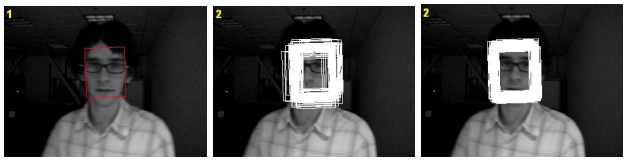


Figure 4. Particle configurations from zero-order transition model(middle column) and our transition model(right column)

frame t, and the convergent state is considered as predicted state $\hat{s}_t$. In order to accelerate this procedure, we adopt the multi-resolution scheme to form a hierarchical motion estimation. After it converges, we integrate the predicted information into a first-order state transition model to form an adaptive state transition model, which is described as

$$s_t = \hat{s}_t + \epsilon_t \qquad (12)$$

where ϵ_t is the system noise, and it is controlled by the residual error of $\hat{s}_t$.

As compared with the zero-order transition model illustrated in the middle column of Fig.4, the proposed transition model (the right column of Fig.4) generates particles more efficiently, since they are tightly centered around the object of interest so that the object can be accurately localized and tracked with less particles.

3.3. Discriminative Observation Model

The observation model is a basic issue to be considered as the Bayesian inference is adopted for tracking. In this part, we propose a discriminative observation model, which utilizes both subspaces obtained by the graph embedding based learning to evaluate the observation candidates.

Suppose P and V represent the target subspace and the discriminant subspace respectively, then the observation

model can be defined as follow.

$$p(o_i|z^+, z^-, P, V) \propto$$
$$exp(-||o_i - PP^T o_i|| + \alpha(||z^- - V^T o_i|| - ||z^+ - V^T o_i||)) \quad (13)$$

where o_i denotes the observation candidate, z^+ and z^- represent the centers of positive samples and negative samples in the discriminant subspace, and α represents a weighting factor. The left part in formula (13) calculates the reconstruction error of the candidate in the target subspace, which evaluates the similarity between the candidate and the target subspace. while the right part evaluates the relative position of the candidate in the discriminative subspace that adds a constraint to pushes the tracker towards the positive sample group and pulls it away from the negative clusterings.

3.4. Heuristic Selection of Negative Sample

Negative samples play an important role in the discriminative learning process. If the negative sample lies too far from the target subspace, then negative sample may not help maximize the margin between two classes. On the other hand, if the negative sample lies too close to the target subspace, they may lie partly in the target subspace such that the estimated target subspace is pushed away from its true place.

In this paper, the negative samples are heuristically selected based on two subspace learned previously. First, the reconstruction error and discriminant constraint of all particles in formula (13) are retained after evaluation of the observations, which are denoted as $\{\pi_i^r, \pi_i^d\}_{i=1}^N$ and N is the number of particles. These two values describe how far the sample lies to the target subspace and its relative position in discriminant subspace. Then, the thresholds of both two values are carefully extracted as $\{T^r, T^d\}$ from $\{\pi_i^r, \pi_i^d\}_{i=1}^N$. Finally, each particle is evaluated to determine whether it is a negative sample as follows.

- **if** $(\pi_i^r > T^r)\&\&(\pi_i^d < T^d)$: denote that the sample lies too far from the target;
- **if** $(\pi_i^r < T^r)\&\&(\pi_i^d > T^d)$: denote that the sample lies too close to the target;
- **if**$(\pi_i^r < T^r)\&\&(\pi_i^d < T^d)$: denote that the sample is similar to the target but lies near to the cluster of background samples. Thus it is selected as negative training sample.

3.5. Incremental Updating

In most tracking applications, the tracker must simultaneously deal with the changes of both the target and the environment. As a result, it is necessary to update the graph structure and the SSD template incrementally to accommodate these changes.

In order to make the graph model depend more heavily on the most recent observations, we assume that the past data is gradually forgotten and new information is gradually added to the graph structure. Suppose that after tracking k frames, we have obtained k positive samples and m negative samples, and normally $k < m$. First we need to efficiently update the subspace of the target as well as its graph structure. In this paper, the strategy taken in [12] is adopted to incrementally learn the eigenbasis as new data arrive. Then the positive samples are added into the foreground graph, and the k most previous samples are gradually dropped to make the sample number balanced with negative ones. Due to the clutter essence of background, it is unnecessary to keep the background samples for long times. To make a tradeoff between accuracy and efficiency, a batch replacement strategy is adopted to construct the new graph structure of the background. For the between-class graph, we only focus on the negative samples lying relatively near to the target subspace. In addition, the SSD template is also updated at the $\frac{k}{2}th$ frame, which means the two different models are interleavedly updated in order not to possess the computational resources simultaneously (A more sophisticated updating strategy for incremental graph learning is prepared in later publication).

3.6. Summary of Tracking Algorithm

A summary of the graph based tracking algorithm is described as follows.

Algorithm 1 Graph Based Tracking Algorithm

Input: Given the available state information s_t and the learned subspaces $\{P, V, z^+, z^-\}$ of frame t;

1. Apply hierarchical SSD iteration to the observations of frame $t+1$ to obtain the predicted state of the target $\hat{s}_{t+1}$;
   $\hat{s}_{t+1} = SSD(I_{t+1}, s_t)$
   where I_{t+1} denotes the image matrix in frame $t+1$;
2. Retain the residual error of the refined state $\hat{s}_{t+1}$, then the particles are generated based on the adaptive transition model, in which the number N and generation region ϵ are controlled by the residual error:
   $s_{t+1}^{(n)} = \hat{s}_{t+1} + \epsilon_{t+1}, n = 1 \cdots N$;
3. Evaluate each particle by the graph based observation model
   $\pi^{(n)} = p(o_{t+1}|s_{t+1}^{(n)}, z^+, z^-, P, V), n = 1 \cdots N$;
   Also retain the reconstruction error and the discriminant constraint for each particle;
5. Get an MAP estimate of the state and keep it as a positive sample;
   $s_{t+1} = \arg\max_{s_{t+1}^{(n)}} p(s_{t+1}^{(n)}|o_{1:t+1}) \approx \arg\max_{s_{t+1}^{(n)}} \pi^{(n)}$;
6. Select the negative samples accordingly;
7. Check the frame number to make a decision: update the graph model or the SSD template;

Output: MAP estimation: s_{t+1};

4. Experimental Results

In our experiment, the target is initialized manually and affine transformations is considered only. Specifically, the motion is characterized by $s = (t_x, t_y, a_1, a_2, a_3, a_4)$ where $\{t_x, t_y\}$ denote the 2-D translation parameters and

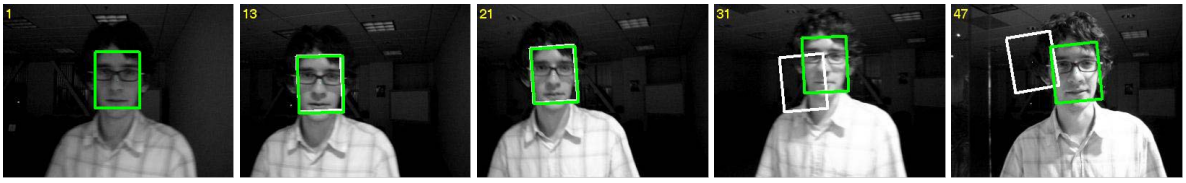


Figure 5. Tracking performances of our algorithms (white: without motion estimation, green: with motion estimation).

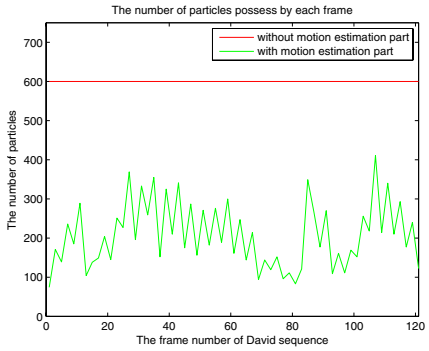


Figure 6. The number of particles possessed by two algorithms (red: without motion estimation, green: with motion estimation).

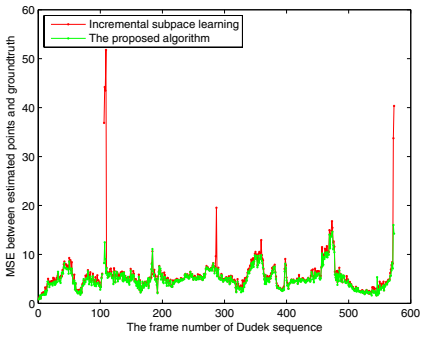


Figure 7. MSE between estimated points and groundtruth(red: incremental subspace learning, green: the proposed algorithm)

$\{a_1, a_2, a_3, a_4\}$ are deformation parameters. Each candidate image is rectified to a 20×20 patch, and the feature is a 400-dimension vector with zero-mean-unit-variance normalization. Several parts of experiments are presented to demonstrate the advantages of the proposed algorithm. All of the experiments are carried out on a dual-CPU Pentium IV 3.4GHz PC with 512M memory and run in real time.

We first test our algorithm to track a rapidly moving object. In order to demonstrate the importance of the motion estimation part, the David sequence[1] is sampled alternately to form a rapid motion testing sequence. The parameters are set to $\{N = 600, var(\epsilon) = [5^2, 5^2, 0.01^2, 0.02^2, 0.002^2, 0.001^2]\}$ in our model without the motion estimation part. As shown in Fig.5, it is clear that the algorithm without the motion estimation fails in frame 31, because it can't catch the rapid motion of the object. On the other hand, the model cascaded with motion estimation part can achieve a better performance with the same ϵ. Fig.6 displays the plot of actual number of particles possessed by our adaptive transition model in each frame. The average number of particle is 206.9, which means that in this case we actually saved nearly 400 particles.

The second part shows the experimental performance of our tracking algorithm, and a comparison to the ISL (incremental subspace learning) algorithm[12] in handling the abrupt changes of appearance and partial occlusion. The tracking result in Fig.8(a) witnesses that the ISL can successfully capture the slow changes of appearance, while it can't effectively adapt to the abrupt changes. Because the ISL only provides a restrictive solution space to accommodate to the appearance changes, as shown in Fig.1(a). On the contrary, the proposed algorithm can effectively capture these changes in both appearance and illumination, since the discriminative projection provides a larger solution space to absorb these variations. To further illustrate the superiority of our algorithm, we also test these two algorithms with the labeled Dudek sequence[2], and the MSE (mean square error) between the estimated points and the groundtruth is computed. The results in Fig.7 show that our approach achieve a more accurate performance in localization than ISL, especially around the frame 105 when the object is confronted with partial occlusion. The average MSE for our algorithm is 4.8194, while that for ISL is 5.7386. Fig.8(b) presents detail performances of both the algorithms in dealing with the partial occlusion. It is clear that discriminant constraint in our model can push the tracker back to the groudtruth position.

The discrimination analysis of the proposed algorithm and the traditional FDA based tracker is demonstrated in this part. As illustrated in Fig.9(a), both the two methods are applied to a video sequence with a clutter background that contains objects similar in appearance to the target. Moreover, these two methods are also tested with a low quality video sequence in a noisy and clutter environment in Fig.9(b). It is obvious the FDA based tracker gradually drift away from the groundtruth and finally loses the track completely. While the proposed algorithm follows the target well in both clutter and noisy background. To investi-

[1]We acknowledge to the author of the source data available at the URL: http://www.cs.toronto.edu/ dross/ivt/

[2]We acknowledge to the author of the source data available at the URL: http://www.cs.toronto.edu/vis/projects/dudekfaceSequence.html

(a) Object with abrupt changes

(b) Object with partial occlusion

Figure 8. Tracking performances of ISL and our algorithm (white: incremental subspace learning, green: the proposed algorithm)

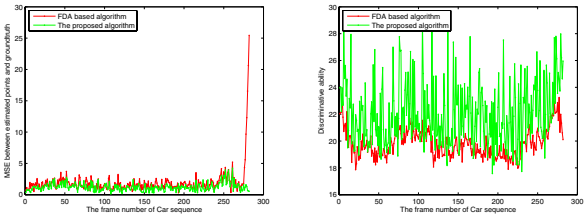


Figure 10. MSE between estimated points and groundtruth and the discriminative ability (red: FDA based algorithm, green: the proposed algorithm)

gate the reason, we have a quantitatively analysis of both algorithms in accuracy and discriminative ability. The discriminative ability of the projection in frame t is defined as follows.

$$Discr(t) = N_n^{-1} \sum_{i=1}^{N_n} ||z^+ - z_i^-||^2 \tag{14}$$

where z_i^- is the ith negative sample in the projective space and N_n is the number of negative samples. Fig.10 plots the results of MSE between estimated points and groundtruth and the discriminative ability of the projection in each frame. Obviously, our approach achieves a more accurate localization and has a more discriminative ability. Since FDA maximize between-class separability under constraint of keeping within-class compactness to a certain level. When the background class is clutter and irregular, this constraint is too restrictive. Therefore, the ability for maximizing the between-class separability is rather limited. On the other hand, the local evaluation employed in our algorithm is less restrictive and it encodes the sample distributions into the graph structures, while enhancing the discriminative ability.

5. Conclusion

This paper presents a graph embedding based object tracking algorithm in an unified framework, which can effectively learn the variational appearance changes and the discriminative structure between foreground and background simultaneously. In our implementation, the graph embedding structure inside background samples is evaluated in a local way for its irregular and multimodal properties. Also the conjunction between the vertex pair from different classes is defined on the margin of sample sets. Both of these two structures can greatly compensate the intrinsic drawbacks of traditional FDA when applied in tracking tasks. Meanwhile, this learning procedure is embedded into a Bayesian inference framework cascaded with a hierarchical motion estimation, which significantly improves the accuracy and efficiency of the object tracking. After carefully selecting data samples in several frames, an incremental updating technique for the models is proposed to accommodate for the changes in both appearance and illumination. Experimental results have demonstrated the efficiency and effectiveness of the proposed tracking algorithm.

Appendix

Lemma 1 *The solution of the constraint optimization problem:*

$$\text{Maximize } J(V) = \sum_{i,j} ||V^T x_i - V^T x_j||^2 w_{ij}^{'}$$

$$\text{subject to } \sum_{i,j} ||V^T x_i - V^T x_j||^2 w_{ij} = 1$$

where $V \in \mathbb{R}^{d \times l}$, is given by the generalized eigenvectors corresponds to the l largest eigenvalues of the following equation.

$$X(D' - W')X^T v = \lambda X(D - W)X^T v$$

where D', D are diagonal matrices defined in Eq.(1).

Proof: Since $||A||^2 = tr(AA^T)$, we see that:

$$J = \sum_{i,j} tr\{(V^T x_i - V^T x_j)(V^T x_i - V^T x_j)^T\} w_{ij}^{'}$$
$$= \sum_{i,j} tr\{V^T (x_i - x_j)(x_i - x_j)^T V\} w_{ij}^{'}$$

The operation of trace is linear and w_{ij}' is a scalar, we can move the summation and w_{ij}' inside the trace:

$$J = tr\{V^T \sum_{i,j} ((x_i - x_j) w_{ij}^{'} (x_i - x_j)^T) V\}$$
$$= tr\{V^T (2XD'X^T - 2XW'X^T) V\}$$

(a) Object in clutter background

(b) Object in noisy environment

Figure 9. Tracking performances of FDA and our algorithm (white: FDA based algorithm, green: the proposed algorithm)

$$= 2tr\{V^T(X(D' - W')X^T)V\}$$

Thus the optimization problem can be reformulated as

$$\text{Maximize } J(V) = 2tr\{V^T(X(D' - W')X^T)V\}$$
$$\text{subject to } 2tr\{V^T(X(D - W)X^T)V\} = 1$$

The Lagrangian is given by:

$$L = 2tr\{V^T(X(D' - W')X^T)V\} + \lambda\{1 - 2tr\{V^T(X(D - W)X^T)V\})\}$$

Let $V = [v_1, \cdots, v_l]$, and we have:

$$\frac{\partial L}{\partial v} = 4X(D' - W')X^T v - 4\lambda X(D - W)X^T v$$

Thus, the optimization problem is solved by finding the l generalized eigenvectors that correspond to the l largest eigenvalues of the given equation.

$$X(D' - W')X^T v = \lambda X(D - W)X^T v$$

Acknowledgment

This work is partly supported by NSFC (Grant No. 60520120099 and 60672040) and the National 863 High-Tech R&D Program of China (Grant No. 2006AA01Z453).

References

[1] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4): 321-332, 1988.

[2] M. Isard, and A. Blake. Condensation: conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5-28, 1998.

[3] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5): 234-240, 2003.

[4] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10): 1296-1311, 2003.

[5] C. Rasmussen and G. D. Hager. Probabilistic Data Association Methods for Tracking Complex Visual Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6): 560-576, 2001.

[6] I. Matthews, T. Ishikawa and S. Baker. The Template Update Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4): 810-815, 2004.

[7] J. Vermaak, P. Perez, M. Gangnet, and A. Blake. Towards improved observation models for visual tracking: Selective adaptation. *In Proceeding of European Conference on Computer Vision*, vol.1, pp.645-660, 2002.

[8] M. J. Black and A. D. Jepson. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *International Journal of Computer Vision*, 26(1): 63-84, 2004.

[9] S. Avidan. Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1064-1072, 2004.

[10] J. Ho, K. C. Lee, M. H. Yang, D. Kriegman. Visual Tracking Using Learned Linear Subspaces. *In Proceeding of International Conference on Computer Vision and Pattern Recognition*, vol.1, pp. 782-789, 2004.

[11] Y. Li, L. Xu, J.Morphett and R. Jacobs. On Incremental and Robust Subspace Learning. *Pattern recognition* 37(77): 1509-1518, 2004.

[12] J. Lim, D. Ross, R. S. Lin, and M. H. Yang. Incremental learning for visual tracking. *In Advances in Neural Information Processing Systems*, pp.793-800, 2004, The MIT Press.

[13] R. S. Lin, D. Ross, J. Lim, and M. H. Yang. Adaptive discriminative generative model and its applications. *In Advances in Neural Information Processing Systems*, pp.801-808, 2004, The MIT Press.

[14] A. Levy and M. Lindenbaum. Sequential Karhunen-Loeve Basis Extraction and its Application to Images. *IEEE Transactions on Image Processing*, 9(8): 1371-1374, 2000.

[15] T. K. Kim, J. Kittler. Locally Linear Discriminant Analysis for Multimodally Distributed Classes for Face Recognition with a Single Model Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3): 318 -327, 2005.

[16] M. Sugiyama. Local Fisher Discriminant Analysis for Supervised Dimensionality Reduction. *Proceedings of International Conference on Machine Learning*, pp.905-912, 2006.

[17] S. Yan, D. Xu, B. Zhang and H. Zhang. Graph Embedding: A General Framework for Dimensionality Reduction. *In Proceeding of International Conference on Computer Vision and Pattern Recognition*, vol.2, pp.830-837, 2005.

[18] H. T. Chen, H. W. Chang, and T. L. Liu. Local Discriminant Embedding and Its Variants. *In Proceeding of International Conference on Computer Vision and Pattern Recognition*, vol.2, pp.846-853, 2005

[19] Z. Manor and P. Perona. Self-tuning spectral clustering. *In Advances in Neural Information Processing Systems*, pp.1601-1608, 2004, The MIT Press.

[20] G. D. Hager, P. N. Hager. Efficient region tracking with parametric models of geometry andillumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10): 1025-1039, 1998.