Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 787093, 12 pages
http://dx.doi.org/10.1155/2014/787093

Research Article

Visual Tracking via Feature Tensor Multimanifold Discriminate Analysis

Ting-quan Deng,[1] Jia-shu Dai,[2] Tian-zhen Dong,[2] and Ke-jia Yi[3]

[1] *College of Science, Harbin Engineering University, Harbin 150001, China*
[2] *College of Computer Science and Technology, Harbin Engineering University, Harbin 151001, China*
[3] *Science and Technology on Underwater Acoustic Antagonizing Laboratory, Systems Engineering Research Institute of CSSC, Beijing 100036, China*

Correspondence should be addressed to Jia-shu Dai; jiashu_dai@163.com

Received 25 June 2014; Accepted 25 August 2014; Published 9 November 2014

Academic Editor: Guoqiang Zhang

In the visual tracking scenarios, if there are multiple objects, due to the interference of similar objects, tracking may fail in the progress of occlusion to separation. To address this problem, this paper proposed a visual tracking algorithm with discrimination through multimanifold learning. Color-gradient-based feature tensor was used to describe object appearance for accommodation of partial occlusion. A prior multimanifold tensor dataset is established through the template matching tracking algorithm. For the purpose of discrimination, tensor distance was defined to determine the intramanifold and intermanifold neighborhood relationship in multimanifold space. Then multimanifold discriminate analysis was employed to construct multilinear projection matrices of submanifolds. Finally, object states were obtained by combining with sequence inference. Meanwhile, the multimanifold dataset and manifold learning embedded projection should be updated online. Experiments were conducted on two real visual surveillance sequences to evaluate the proposed algorithm with three state-of-the-art tracking methods qualitatively and quantitatively. Experimental results show that the proposed algorithm can achieve effective and robust effect in multi-similar-object mutual occlusion scenarios.

1. Introduction

Visual tracking is an important research area in computer vision and pattern recognition which can be applied to many domains, such as visual surveillance, traffic monitoring, human computer interaction, image compression, three-dimension reconstruction, and weapons automatically tracking combat. To make these applications viable, the results of visual tracking must be robust and precise.

Visual tracking is a challenging problem due to object appearance variations. Many issues can cause object appearance variations, including camera motions, camera viewpoint changes, environmental illumination changes, noise disturbance, background clutter, pose variation, and object shape deformation, and occlusions occur [1].

1.1. Related Works. In recent years, there are a wide range of tracking algorithms to deal with these object appearance variations. These algorithms can be roughly classified into two categories according to the model-construction mechanism, which are generative and discriminative methods.

The generative methods mainly focus on how to robustly describe the appearance model and then find the best matching appearance model of image patch with that of the object. The classical template matching tracking algorithm can be viewed as the generative model. The earliest template-based tracking method dates back to the Lucas-Kanade algorithm. The eigen-tracking [2] algorithm demonstrated that tracking can be considered as finding the minimum distance from the appearance model of tracked object to that of the subspace represented. Matthews et al. [3] show how to update the template which can avoid the "drifting" inherent in the naive method. The IVT [4] tracking algorithm utilizes subspace learning to generate a low-dimensional object appearance and incrementally update it. Hu et al. [5] proposed a visual object tracking algorithm which models appearance changes

by incrementally learning a tensor subspace representation. In the tracking procedure, the sample mean and an eigenbasis for each unfolding matrix of the tensor are adaptively updated. The classical mean-shift [6] tracker uses histogram as the appearance model; then the mean-shift procedure is achieved to locate the object. The Fragtrack [7] utilizes several fragments to design the appearance model which can handle pose change and partial occlusion. The ℓ_1-tracker [8] casts tracking problem as sparse approximation where the object is modeled by a sparse linear combination of target and a set of trivial templates. The sparse representation is obtained by solving an ℓ_1-regularized optimization least-squares problem, and the posteriori probability of candidate image patch belonging to the object class is inversely proportional to the residual between the candidate image patch and the reconstructed one. The ℓ_1-APG tracker [9] developed the ℓ_1-tracker that not only runs in real-time but also improves the tracking accuracy. The S-MTT [10] algorithm regularizes the appearance model representation problem employing sparsity-inducing $\ell_{p,q}$ mixed norms which can handle particles independently.

The discriminative methods treat visual tracking as a binary classification problem. It aims to separate the object from its surrounding complex background with a small local region. There are many newly proposed visual tracking algorithms based on boosting classifier because of its powerful discriminative learning capabilities. Online boosting algorithm has wide applications in object detection and visual tracking. Grabner et al. [11] proposed an online boosting tracker which is firstly given a discriminative evaluation of each feature from a candidate feature pool. Then online semisupervised boosting method [12] is proposed for the purpose of alleviating the object drifting problem in visual tracking. Ensemble tracking [13] uses weak classifiers to construct a confidence map by pixel classification to distinguish between the foreground and the background. The MIL tracker [14] represents an object by a set of samples; these samples corresponding to image patch are considered within positive and negative bags. Then, multiple instance boosting is used to overcome the problem that slight inaccuracies in labeled training examples can cause object drift. However, the tracking may fail when the training samples are imprecise. Pointing to this problem, the WMIL tracker [15] which integrates the sample important into the multiple instance learning is proposed. The SVM tracker [16] combined support vector machine into optical flow to achieve visual tracking. A visual tracking algorithm via an online feature selection mechanism for evaluating multiple object features is proposed in [17]. The VTD algorithm [18] designs the observation and motion model based on visual tracking composition scheme. The TLD tracker [19] explicitly decomposes the long-term tracking problem into three components which are tracking, learning, and detection. The CT tracker [20] extracted the sparse image feature combined with a naive classifier to separate the object from the background.

In the multiple moving objects scenarios, with the movement of one object, the reflected lights of other objects which reach to the camera lens may be hindered, making other objects' projection imaging incomplete or even completely invisible on the imaging plane. When the occlusion occurred, if the tracking object is similar to the occlusion object, the object is vulnerable to the similar objects influence in the progress of occlusion to separation which can cause drift. Thus, it is necessary to distinguish the tracking object with the potential similar objects in the scenarios. Meanwhile, when the object is partially occluded, the information from unoccluded part has a large reference value of determining the object state. Therefore, the object feature must maintain the structural relationship of the original space. This paper proposed a visual object tracking algorithm for multiple similar objects mutual occluded problem which combines these two ideas. First of all, a feature function is designed for the purpose of extracting the tensor feature which can maintain the spatial structure of the object. The multimanifold tensor data set is collected by template matching tracking algorithm in the initial few frames. A tensor distance is defined to determine the intramanifold and intermanifold neighborhood relationship. The object feature tensor is embedded into a low-dimensional space by multimanifold discriminate analysis. Then the object state in the next frame is obtained by Bayesian sequence inference. Considering the changes in the object appearance, an update strategy for the multimanifold set is needed to be set.

1.2. Plan of the Paper. This paper is organized as follows: in the next section, we first introduce the notation of tensor algebra and feature tensor. After that, multimanifold discriminate analysis is reviewed in Section 3. Section 4 details the visual tracking framework. In Section 5 comparative experimental results and analysis are showed, and conclusions are drawn in Section 6.

2. Feature Tensor

A tensor is a high-order array which can be maintained the original spatial structure of an object. Construct a feature tensor from an object appearance can increase tracking accuracy.

2.1. Tensor Algebra. Tensor can be viewed as multiorder array which exists in multiple vector spaces; the algebra corresponding to tensor is the mathematical foundation of multilinear analysis [21]. An N-order tensor is denoted as $\widetilde{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$; each elements in this tensor is represented as $x_{i_1,\ldots,i_n,\ldots,i_N}$ for $1 \le i_n \le I_n$.

The mode-n unfolding matrix $\widetilde{\mathbf{X}}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N)}$ of a tensor $\widetilde{\mathbf{X}}$ consists of all the mode-n column vectors.

The mode-n product of a tensor $\widetilde{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$ is $\widetilde{\mathbf{X}} \times_n \mathbf{U}$ which is a new tensor. The element of this tensor is

$$\left(\widetilde{\mathbf{X}} \times_n \mathbf{U}\right)_{i_1 \cdots i_{n-1} j_n i_{n+1} \cdots i_N} = \sum_{i_n} x_{i_1 i_2 \cdots i_N} u_{j_n i_n}, \tag{1}$$

where $x_{i_1 i_2 \cdots i_N}$, $u_{j_n i_n}$ are the elements of tensor $\widetilde{\mathbf{X}}$ and matrix $\mathbf{U}$.

The inner product of two tensors $\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is

$$\left\langle \widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}} \right\rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} x_{i_1 \cdots i_n \cdots i_N} y_{i_1 \cdots i_n \cdots i_N}. \quad (2)$$

The Frobenius norm of a tensor $\widetilde{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is

$$\left\| \widetilde{\mathbf{X}} \right\|_F = \sqrt{\left\langle \widetilde{\mathbf{X}}, \widetilde{\mathbf{X}} \right\rangle}. \quad (3)$$

2.2. Feature Tensor. The object appearance image from RGB color video sequence is a three-dimensional data, which formed a nature tensor structure. The color and edge information of the object have a better discrimination on the object class; the gradient feature can describe the object edge information. For a detailed description of object information, the feature function of an object appearance image is defined as follows:

$$f(i,j) = \left[R, R_x, R_y, \sqrt{R_x^2 + R_y^2}, G, G_x, G_y, \sqrt{G_x^2 + G_y^2}, B, B_x, B_y, \sqrt{B_x^2 + B_y^2} \right], \quad (4)$$

where R_x, R_y, G_x, G_y, B_x, B_y are the x-direction and y-direction gradients on the R, G, and B color channels.

Each pixel (i, j) on object appearance image corresponds to a twelve-dimensional feature vector; the size $a \times b \times 3$ object appearance image corresponds to a $\widetilde{\mathbf{X}} \in \mathbb{R}^{a \times b \times 12}$ feature tensor.

3. Multimanifold Discriminate Analysis

The basic assumption of the manifold learning is that high-dimensional datum can be considered as geometric correlation points which lie in low-dimensional smooth manifold. There is usually a submanifold structure corresponding to a single object class; different objects lie in different submanifolds. The multimanifold discriminate analysis can project the tensor data which is from a submanifold into a low-dimensional space.

3.1. Multimanifold Neighborhood Relationship of Feature Tensor. The appearance of each object under different poses is usually composed of a submanifold; the multiple different object appearance spaces formed the multimanifold. Each moving object appearance image in video sequence can extract a feature tensor $\widetilde{\mathbf{X}} \in \mathbb{R}^{a \times b \times 12}$. The set of feature tensor calculated by the appearance images from the first m frames is denoted as $M_i = \{\widetilde{\mathbf{X}}_{i1}, \widetilde{\mathbf{X}}_{i2}, \ldots, \widetilde{\mathbf{X}}_{im}\}$; then M_i can be seen as a submanifold. Because of the presence of multiple moving objects in the scenarios, the set of each submanifold $M = \{M_1, M_2, \ldots, M_n\}$ is a multimanifold dataset [22]. The entries $\widetilde{\mathbf{X}}_{i_1 i_2 \cdots i_N}$ $(1 \le i_j \le I_j, 1 \le j \le N)$ in $\widetilde{\mathbf{X}}$ are corresponding to the lth element in $\mathbf{x}$, where

$$l = i_1 + \sum_{j=2}^{N} \left(i_j - 1\right) \prod_{o=1}^{j-1} I_o \quad (2 \le j \le N). \quad (5)$$

The distance between two tensors $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ is (the order and dimension of $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ are the same)

$$d_T\left(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}\right) = \sqrt{\sum_{l,m=1}^{I_1 \times I_2 \times \cdots \times I_N} g_{lm} \left(\mathbf{x}_l - \mathbf{y}_l\right) \left(\mathbf{x}_m - \mathbf{y}_m\right)}, \quad (6)$$

where g_{lm} is the measurement coefficient. Since there are too many entries in the tensor data, the measurement coefficient is defined by the distance of points which have spatial neighborhood relationship. Consider

$$g_{lm} = \begin{cases} e^{-\|p_l - p_m\|_2^2 / 2\sigma^2} & \text{if } \mathbf{x}_m \in N_{k'}\left(\mathbf{x}_l\right) \\ 0 & \text{else,} \end{cases} \quad (7)$$

where σ is the regularization parameter and $\|p_l - p_m\|_2$ is the location distance between $\mathbf{x}_l$ and $\mathbf{x}_m$. If $\mathbf{x}_l$ and $\mathbf{x}_m$, respectively, correspond to the $\widetilde{\mathbf{X}}_{i_1 i_2 \cdots i_N}$ and $\widetilde{\mathbf{X}}_{i'_1 i'_2 \cdots i'_N}$ in tensor $\widetilde{\mathbf{X}}$, then

$$\|p_l - p_m\|_2 = \sqrt{\left(i_1 - i'_1\right)^2 + \left(i_2 - i'_2\right)^2 + \cdots + \left(i_N - i'_N\right)^2}. \quad (8)$$

The K_1 intramanifold neighborhood $N_{\text{intra}}^{K_1}(\widetilde{\mathbf{X}}_{ij})$ of the tensor $\widetilde{\mathbf{X}}_{ij}$ is as follows: calculate the tensor distance $d_{jl} = d_T(\widetilde{\mathbf{X}}_{ij}, \widetilde{\mathbf{X}}_{il})$, $j \ne l$ between the tensor $\widetilde{\mathbf{X}}_{ij}$ in submanifold M_i and another tensor $\widetilde{\mathbf{X}}_{il}$ in this submanifold; then the nearest K_1 intramanifold neighborhood of $\widetilde{\mathbf{X}}_{ij}$ can be obtained according to the tensor distance d_{jl}.

The K_2 intermanifold neighborhood $N_{\text{inter}}^{K_2}(\widetilde{\mathbf{X}}_{ij})$ of the tensor $\widetilde{\mathbf{X}}_{ij}$ is as follows: calculate the tensor distance $d_{js} = d_T(\widetilde{\mathbf{X}}_{ij}, \widetilde{\mathbf{X}}_{ls})$, $i \ne l$ between the tensor $\widetilde{\mathbf{X}}_{ij}$ in submanifold M_i and tensor $\widetilde{\mathbf{X}}_{ls}$ $(l \ne i)$ in another submanifold M_l $(l \ne i)$; then the nearest K_2 intermanifold neighborhood of $\widetilde{\mathbf{X}}_{ij}$ can be obtained according to the tensor distance d_{js}.

The multimanifold dataset and its neighborhood relationship are shown in Figure 1.

As can be seen from Figure 1, there are four initial moving objects in the scenarios, thus constructing four submanifolds which are M_1, M_2, M_3, M_4; these four submanifolds formed a multimanifold. The intramanifold neighborhood relationship of tensor $\widetilde{\mathbf{X}}_{13}$ in submanifold M_1 is $\widetilde{\mathbf{X}}_{12}, \widetilde{\mathbf{X}}_{14}, \widetilde{\mathbf{X}}_{17}$; the intermanifold neighborhood relationship of this tensor is $\widetilde{\mathbf{X}}_{22}, \widetilde{\mathbf{X}}_{24}, \widetilde{\mathbf{X}}_{41}, \widetilde{\mathbf{X}}_{43}, \widetilde{\mathbf{X}}_{44}$.

3.2. Multimanifold Discriminate Analysis. The objective of manifold learning is to recover the low-dimensional structure from the high-dimensional datum space and find a low-dimensional embedding map. In the multiple similar objects scenarios, it is hoped that the extracted object feature can distinguish the object and the potential similar objects in the scenarios. The objective of multimanifold learning is that the difference between a tensor and intramanifold neighborhood points decreases and the difference between the tensor and intermanifold neighborhood points increases

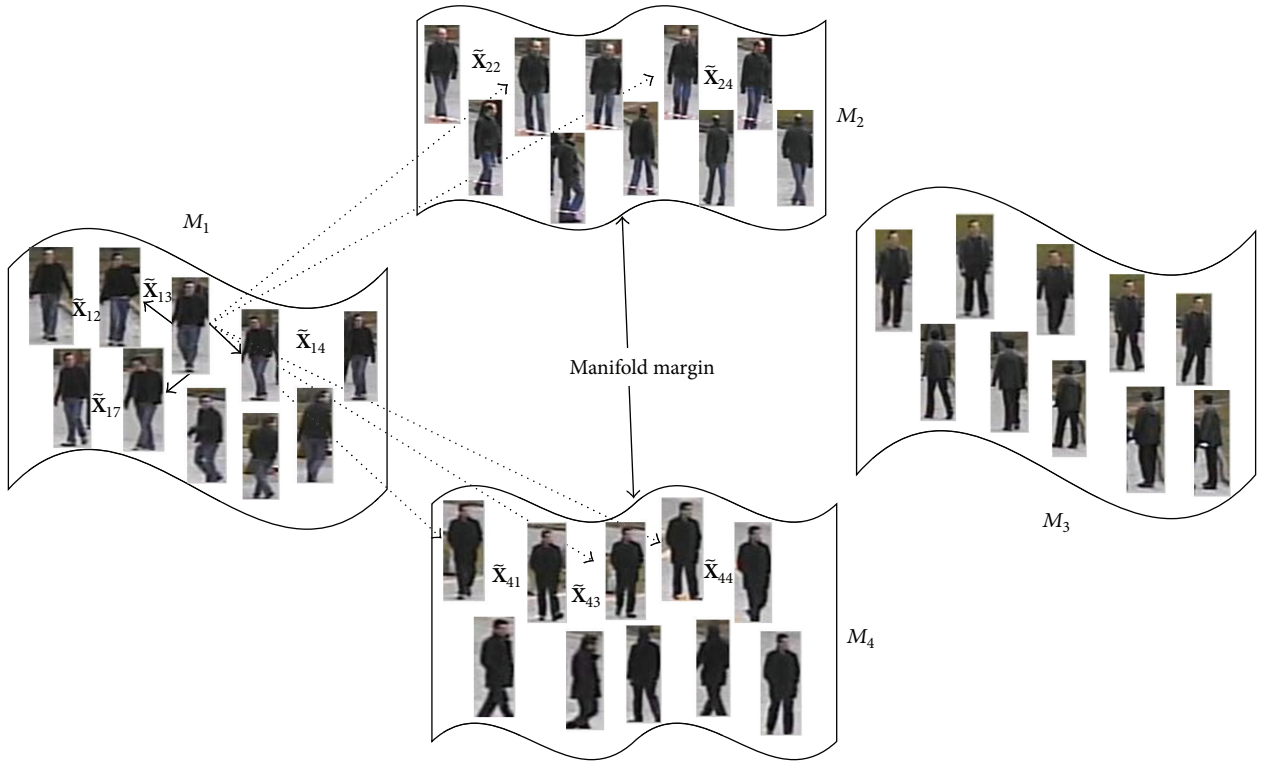

Figure 1: Multimanifold dataset and neighborhood relationships.

in the embedded space. Considering these, the objective function of multimanifold discriminate analysis is

$$\begin{aligned}&\arg\max_{\mathbf{U}_1,\mathbf{U}_2,\mathbf{U}_3} f\left(\mathbf{U}_1,\mathbf{U}_2,\mathbf{U}_3\right)\\&\quad= f_{\text{inter}}\left(\mathbf{U}_1,\mathbf{U}_2,\mathbf{U}_3\right)-f_{\text{intra}}\left(\mathbf{U}_1,\mathbf{U}_2,\mathbf{U}_3\right),\end{aligned}\tag{9}$$

where $\mathbf{U}_1$, $\mathbf{U}_2$, $\mathbf{U}_3$ are the multilinear projection matrices in the first-order, second-order, and third-order which are corresponding to the tensor in the submanifold M_i. Consider

$$\begin{aligned}&f_{\text{inter}}\left(\mathbf{U}_1,\mathbf{U}_2,\mathbf{U}_3\right)\\&\quad=\sum_{r=1}^{m}\sum_{s=1}^{m}w_{\text{inter}}^{rs}\left\|\left(\widetilde{\mathbf{X}}_{ir}-\widetilde{\mathbf{X}}_{js}\right)\times_1\mathbf{U}_1\times_2\mathbf{U}_2\times_3\mathbf{U}_3\right\|_F\quad(i\neq j),\\&f_{\text{intra}}\left(\mathbf{U}_1,\mathbf{U}_2,\mathbf{U}_3\right)\\&\quad=\sum_{r=1}^{m}\sum_{s=1}^{m}w_{\text{intra}}^{rs}\left\|\left(\widetilde{\mathbf{X}}_{ir}-\widetilde{\mathbf{X}}_{is}\right)\times_1\mathbf{U}_1\times_2\mathbf{U}_2\times_3\mathbf{U}_3\right\|_F,\end{aligned}\tag{10}$$

where m is the number of submanifold points; K_1, K_2 are the number of intramanifold and intermanifold neighborhood. $\mathbf{W}_{\text{intra}}$ and $\mathbf{W}_{\text{inter}}$ are the intramanifold and intermanifold weight matrices; the size is $m\times m$; the elements are separately as follows:

$$\begin{aligned}w_{\text{intra}}^{rs}&=\begin{cases}e^{(-d_T(\widetilde{\mathbf{X}}_{ir}-\widetilde{\mathbf{X}}_{is})/\sigma)} & \text{if } \widetilde{\mathbf{X}}_{is}\in N_{\text{intra}}^{K_1}\left(\widetilde{\mathbf{X}}_{ij}\right)\\0 & \text{else,}\end{cases}\\w_{\text{inter}}^{rs}&=\begin{cases}e^{(-d_T(\widetilde{\mathbf{X}}_{ir}-\widetilde{\mathbf{X}}_{js})/\sigma)} & \text{if } \widetilde{\mathbf{X}}_{js}\in N_{\text{inter}}^{K_2}\left(\widetilde{\mathbf{X}}_{ir}\right)\\0 & \text{else,}\end{cases}\end{aligned}\tag{11}$$

where d_T is the tensor distance; σ is bandwidth, which is the weighted coefficient of tensor $\widetilde{\mathbf{X}}_{ij}$ in the submanifold M_i. Consider

$$\begin{aligned}q_{ij}&=\sum_{l=1}^{m}w_{\text{intra}}^{jl},\\C_i&=\frac{\sum_{j=1}^{m}q_{ij}*\widetilde{\mathbf{X}}_{ij}}{\sum_{j=1}^{m}q_{ij}}.\end{aligned}\tag{12}$$

Then C_i can be viewed as the weighted center of submanifold M_i.

Due to the fact that there is no closed optimal solution of the optimization problem in (9), for the purpose of computing $\mathbf{U}_p$ $(p=1,2,3)$, recursively solve the projection matrix in every order of the tensor feature. Consider

$$\arg\max_{U_p} f\left(\mathbf{U}_p\right)=f_{\text{inter}}\left(\mathbf{U}_p\right)-f_{\text{intra}}\left(\mathbf{U}_p\right),\tag{13}$$

where

$$
\begin{aligned}
f_{\text{inter}}\left(\mathbf{U}_p\right) &= \sum_{r=1}^{m}\sum_{s=1}^{m} w_{\text{inter}}^{r,s}\left\|\left(\left(\widetilde{\mathbf{X}}_{ir}-\widetilde{\mathbf{X}}_{js}\right)\times_1\cdots\times_{p-1}\right)\times_p \mathbf{U}_p\right\|_F \\
&= \sum_{r=1}^{m}\sum_{s=1}^{m} w_{\text{inter}}^{r,s}\left\|\mathbf{U}_p^T\left(\left(\widetilde{\mathbf{X}}_{ir}-\widetilde{\mathbf{X}}_{js}\right)\times_1\cdots\times_{p-1}\right)_{(p)}\mathbf{U}_p\right\|_F \\
&= tr\left(\mathbf{U}_p^T\mathbf{A}_{\text{inter}}\mathbf{U}_p\right), \\
f_{\text{intra}}\left(\mathbf{U}_p\right) &= tr\left(\mathbf{U}_p^T\mathbf{A}_{\text{intra}}\mathbf{U}_p\right), \\
\mathbf{A}_{\text{inter}} &= \sum_{i=1}^{m}\sum_{j=1}^{m} w_{\text{inter}}^{r,s}\left(\left(\widetilde{\mathbf{X}}_{ir}-\widetilde{\mathbf{X}}_{js}\right)\times_1\cdots\times_{p-1}\right)_{(p)} \\
&\quad\times\left(\left(\widetilde{\mathbf{X}}_{ir}-\widetilde{\mathbf{X}}_{js}\right)\times_1\cdots\times_{p-1}\right)_{(p)}^T, \\
\mathbf{A}_{\text{intra}} &= \sum_{r=1}^{m}\sum_{s=1}^{m} w_{\text{intra}}^{r,s}\left(\left(\widetilde{\mathbf{X}}_{ir}-\widetilde{\mathbf{X}}_{is}\right)\times_1\cdots\times_{p-1}\right)_{(p)} \\
&\quad\times\left(\left(\widetilde{\mathbf{X}}_{ir}-\widetilde{\mathbf{X}}_{is}\right)\times_1\cdots\times_{p-1}\right)_{(p)}^T.
\end{aligned}
\tag{14}
$$

Then

$$
f\left(\mathbf{U}_p\right) = tr\left(\mathbf{U}_p^T\left(\mathbf{A}_{\text{inter}}-\mathbf{A}_{\text{intra}}\right)\mathbf{U}_p\right). \tag{15}
$$

To maximize the $f(\mathbf{U}_p)$ by solving the eigen-value equation,

$$
\left(\mathbf{A}_{\text{inter}}-\mathbf{A}_{\text{intra}}\right)u_p = \lambda u_p, \tag{16}
$$

obtain $\mathbf{U}_p$.

The eigen-values are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{d'} \geq 0 \geq \lambda_{d'+1} \geq \cdots \geq \lambda_d$; the corresponding eigen-vector of eigen-value λ_p is $[u_{p_1}, u_{p_2}, \ldots, u_{p_d}]$, where d is the dimension pth order in the original feature tensor from submanifold M_i. The directional projection positive along the eigen-vector u_{p_l} which is corresponding to the eigen-value λ_l of $(\mathbf{A}_{\text{inter}}-\mathbf{A}_{\text{intra}})$ is positive; that is, intermanifold neighborhood distance of tensors is bigger than the intramanifold neighborhood distance which are projected along this direction. Therefore, the projection matrix $\mathbf{U}_p = [u_{p_1}, u_{p_2}, \ldots, u_{p_{d'}}]$ consists of all of the eigen-vectors which are corresponding to the positive eigen-values. Thus, the tensor data which are in submanifold M_i can be embedded in a low-dimensional space via multilinear projection matrix $\mathbf{U}_1$, $\mathbf{U}_2$, $\mathbf{U}_3$. In this lower-dimensional space, the difference between tensor data and its intramanifold neighborhood points decreases and the difference between it and its intermanifold neighborhood points increases, so that the distinguishing ability between the object and the similar ones is greater.

4. Visual Tracking Framework

In order to achieve tracking of an object in scenarios, Bayesian sequence inference is used to obtain the object final state. Meanwhile, the multi-manifold datasets and the multilinear projection matrice which are calculated from multi-manifold discriminate analysis should be updated.

4.1. Sequence Inference. In the visual tracking problem, the movement of the object is unable to predict, the object state in the current frame only related to that in the prior frame; then the visual tracking process satisfies the Markov process [23]. A bounding box $o_t = (x_t, y_t, w_t, h_t)$ is used to describe the object state at the tth frame, where (x_t, y_t), w_t, h_t denote the upper left corner coordinate, the width, and height of the bounding box.

Given a set of observed object appearance images $S_t = \{s_1, s_2, \ldots, s_t\}$, the objective of visual tracking is to obtain the optimal estimate value of the hidden state variables o_t. There is a similar result as that of the object state which is obtained according to Bayes' theorem. Consider

$$
P\left(o_t \mid S_t\right) \propto P\left(s_t \mid o_t\right)\int P\left(o_t \mid o_{t-1}\right)P\left(o_{t-1} \mid S_{t-1}\right)do_{t-1}, \tag{17}
$$

where $P(o_t \mid o_{t-1})$ refers to the state transition model and $P(s_t \mid o_t)$ refers to the observation model. According the observation model $P(s_t \mid o_t)$, we can obtain the tracking results.

State Transition Model. This was used to model the movement of object between consecutive frames. Because of the irregular movement of object, the object state is difficult to predict and the moving speed of the object is not very fast. It is considered that the object state in the current frame is near to that in the prior frame. Then, the object state o_t is modeled by independent Gaussian distribution around its counterpart in state o_{t-1}, described as

$$
P\left(o_t \mid o_{t-1}\right) = N\left(o_t; o_{t-1}, \Sigma\right), \tag{18}
$$

where Σ means the diagonal covariance matrix corresponding to the variables x_t, y_t, w_t, h_t, and the elements are σ_x^2, σ_y^2, σ_w^2, σ_h^2. N particles can be randomly generated pointing to Gaussian distribution. Each particle corresponds to an object state; then N particles can obtain multiple states $\{o_t^i, i = 1, 2, \ldots, N\}$. During the visual tracking process, the more the particles we generated are, the more accurate the object state estimate was, but at the same time, the computational efficiency was low. For the purpose of efficient and effective of the visual tracking algorithm, there is a balance sought between these factors.

Observation Model. This was used to measure the difference between the appearance observation and the object appearance model. Given a drawn particle state o_t^i and the corresponding cropped image patch z_t^i in the frame image I_t, the probability of an image patch being generated from the submanifold space is inversely proportional to the difference between image patch and the appearance model and could be calculated between the negative exponential distance of the projected data and the weighted center of submanifold. Consider

$$
p\left(\mathbf{z}_t^j \mid o_t\right) = \exp\left\{-\frac{\left(\left\|\left(\mathbf{z}_t^j - C_i\right)\times_1 \mathbf{U}_1\times_2 \mathbf{U}_2\times_3 \mathbf{U}_3\right\|_F\right)}{\sigma^2}\right\}, \tag{19}
$$

where σ indicates the bandwidth, $\|\cdot\|_F$ is the Frobenius norm, and $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3$ are the multilinear projection matrix of the ith object in submanifold M_i.

The state o_t^i corresponding to the maximum $p(z_t^i \mid o_t)$ is the optimal object state at the tth frame. Let $\varepsilon = \|(\mathbf{z}_t^j - C_i) \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3\|_F$ represent the error between feature tensor which is calculated by observation $\mathbf{z}_t^j$ and the weighted center C_i of submanifold M_i.

4.2. Multimanifold Data Sets Update. The appearance image of the object changes with the movement of it in the scenarios; the submanifold of the object should have different posture object appearance feature tensors. Therefore, the multimanifold data set should be updated in the tracking process. Because of the factors, such as occlusion and so forth which influence the object appearance, the appearance images of the tracked object have the non-object information; then obtained object feature tensor will not be in the submanifold. Therefore, the update strategy is necessary. From the perspective of the human sensory vision, the appearance information of object changes in the process of occlusion; the changes of object between consecutive frames are bigger or the object feature tensor is far away with the center of submanifold in the embedded space, while the changing information between consecutive frames is small or the object feature tensor is near the center of submanifold in the embedded space; that is, the object state is well determined.

The image first-order entropy is used to describe the gray value distribution of the object image, but not to consider it spatial distribution, while the image second-order entropy uses the 2-tuple feature (i, j) which is calculated by spatial distribution. The image second-order entropy could describe the changes of the object, where i is the gray value ($0 \le i \le 255$) and j is the neighborhood gray value ($0 \le j \le 255$). $p_{ij} = f(i, j)/ab$ denotes the gray value and neighborhood gray distribution, where $f(i, j)$ is the counts of the occurrence of the 2-tuple feature and ab is the size of image. The second-order entropy is defined as

$$H = \sum_{i=0}^{255} E_i = \sum_{i=0}^{255} p_{ij} \ln p_{ij}. \tag{20}$$

The difference of the object in consecutive frames is described by the second-order entropy. When the second-order entropy difference of the object image in consecutive frames is bigger, the object maybe occluded. Simultaneously, the feature tensor of appearance image would be far away from the weighted center of submanifold; namely, the error is bigger. As shown in Figure 2, the object is largely occluded at the frames 33–46 and 48–63, and small part occluded at the frames 69–77.

For a best state o_t of object i which is newly obtained, when the difference of second-order entropy with the prior frame $H_d < \delta\overline{H}_d$ and the error in low-dimensional tensor space embedded $\varepsilon > \delta\overline{\varepsilon}_{M_i}$, the feature tensor calculated by the newly obtained object state o_t should add into the submanifold M_i, where $\overline{H}_d$ is mean of the difference of second-order entropy, $\overline{\varepsilon}_{M_i}$ is the mean of the errors, and δ is the adjustment factor which takes 1.2 in this experiment.

When the tensor number in a submanifold M_i is the multiples of the initial number, the multimanifold discriminate analysis is computed on the new multimanifold datasets; then the weighted center of submanifold and multilinear projection matrices are updated. There will be a small portion of the determined object data abandoned, but the tensors which added into the data set are essentially the feature tensors of object appearance.

The whole tracking algorithm is working as follows.

(1) Locate the object state in the first frame, either manually or by using an automated detector.

(2) Tracking objects use template matching tracking algorithm in the initial m frames.

(3) Extract the feature tensors $\widetilde{\mathbf{X}}_{ij}$ ($i = 1 \cdots N_o$, $j = 1 \cdots m$) from each object appearance images which are cropped according to the obtained objects states.

(4) Construct the multimanifold dataset M using the obtained feature tensors $\widetilde{\mathbf{X}}_{ij}$ ($i = 1 \cdots N_o$, $j = 1 \cdots m$).

(5) Determine the neighborhood relationship using tensor distance in the multimanifold dataset.

(6) Calculate the weighted centers of each submanifold and the multilinear embedded matrices through multimanifold discriminate analysis.

(7) Advance to the next frame t. Draw particles according to the object prior state o_{t-1} and crop the appearance images corresponding to each of the particles. Extract the feature tensors of each of the appearance images. The best object state in current frame is calculated by Bayesian sequence inference.

(8) Calculate the difference of second-order entropy with the prior frame and the error in low-dimensional tensor space embedded; if $H_d < \delta\overline{H}_d$ and $\varepsilon > \delta\overline{\varepsilon}_{M_i}$, the feature tensor calculated by the newly obtained object state o_t should add into the submanifold M_i.

(9) When the tensor number in a submanifold M_i is the multiples of the initial number m, go to step (3).

5. Comparative Experiments and Analysis

In order to verify the effectiveness of the proposed algorithm, CAVIAR data sets and PETS outdoor multiperson data sets are used to be verified. The initial state of a moving object is determined by automatically tracking detectors [24] or artificial markers. The initial multimanifold data set is calculated by the object states which come from template matching tracking algorithm. The proposed algorithm is compared with three state-of-the-art trackers which are IVT [4], L1-APG [9], and MIL [14]. The Bayesian sequence inference needs to consider the particle number which impacts on the overall efficiency of the algorithm; the particle number is chosen to be 200 for comprehensive consideration. Each object appearance image is resized to a $64 \times 32 \times 3$ patch.

5.1. CAVIAR Data Sets. In this experiment, the experiment scenarios come from the Portugal Mall surveillance video

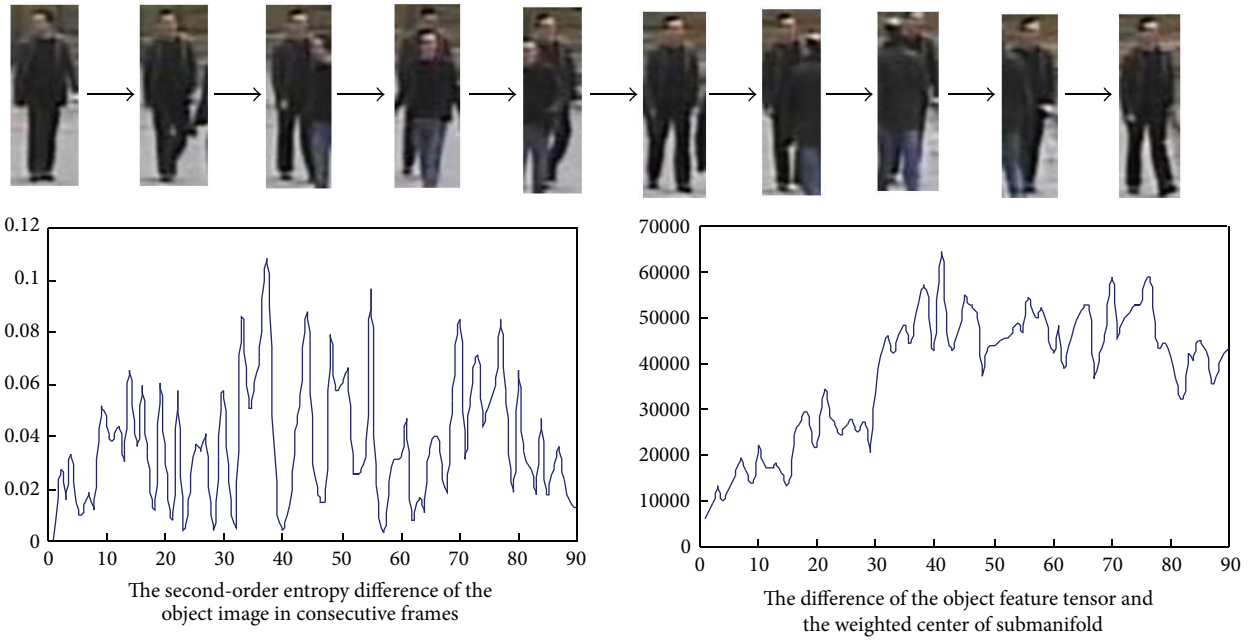

Figure 2: The change of the object in consecutive frames.

data sets. There are object scale change, pose variation and occlusion during the three objects walking away from the camera. Testing video sequences are color images of 388 × 284 resolutions. The Gaussian variances of the three objects are (8, 8, 0.5, 0.5), (4, 4, 0.5, 0.5), (2, 2, 0.5, 0.5). The results are shown in Figure 3.

As can be seen from the results, the three main objects did not occlude before the initial 57 frames; the three comparison algorithms can achieve tracking. Since the 57th frame, object 2 gradually occludes object 3 until object 3 is unable to be seen, while the IVT and L1-APG algorithms are all missing object 3 and offset to object 2 which led to the wrong tracking. Since the 87th frame, object 1 gradually occludes object 3 while the IVT tracker could not distinguish them due to the fact that object 1 is similar to object 3 and then object 3 is mistaken as object 1 which carried the wrong tracking. Meanwhile, the color of object 2 is largely different from object 2 and object 3; the IVT and L1-APG trackers can achieve the better results in tracking object 2. The MIL tracker did not achieve the accurate tracking on the three objects due to the interference of the background. The proposed algorithm achieved complete tracking on the three objects which was not subject to the interference of similar object in the tracking process.

5.2. PETS Outdoor Multiperson Data Sets. In this experiment, the experiment scenarios come from the PETS2009 surveillance video data sets. There are multiple human objects that move around in multiple directions in the scenarios which are similar to each other. The objects cross occlusion and the objects scale pose variation during the walking. Testing video sequences are color images of 768 × 576 resolutions. The Gaussian variances of the four objects are (4, 3, 0.5, 0.5), (4, 4, 0.5, 0.5), (2, 2, 0.5, 0.5), (6, 6, 0.5, 0.5). The results are in Figure 4.

As can be seen from the results, object 2 gradually completely occludes object 1 since the 26th frame which makes object 1 lost most of its information. Then, the IVT and L1-APG trackers lost object 1 while they achieved tracking object 2 which is not occluded. The MIL tracker roughly achieves tracking of objects 1 and 2. Object 1 occludes object 3 in the 36th frame; then the IVT, L1-APG, and MIL trackers are disturbed by object 1 when tracking object 3; the three algorithms are all offset to object 1 because object 1 and object 3 are very similar. Object 1 is occluded by object 4, since the 56th frame, the IVT, and L1-APG trackers are disturbed by object 1 when tracking object 1. The two trackers lost object 4 and offset to object 1 while the MIL tracker achieved tracking object 4. Object 4 and object 2 mutual occluded since the 64th frame; MIL tracker failed to track object 4 while the IVT and L1-APG are completely wrong tracking. This video sequence often occurs an object occluded another one which made the tracking very difficult, the proposed algorithm tracking successfully without excessive interference with similar objects, and achieved a complete tracking of the four objects.

5.3. Quantitative Evaluation. Aside from the qualitative comparison, we used two metrics to quantitatively compare the experimental results of the tracking algorithms which are tracking success ratio and center location error [20]. We initially manually labeled "ground truth" locations in each experimental scenario.

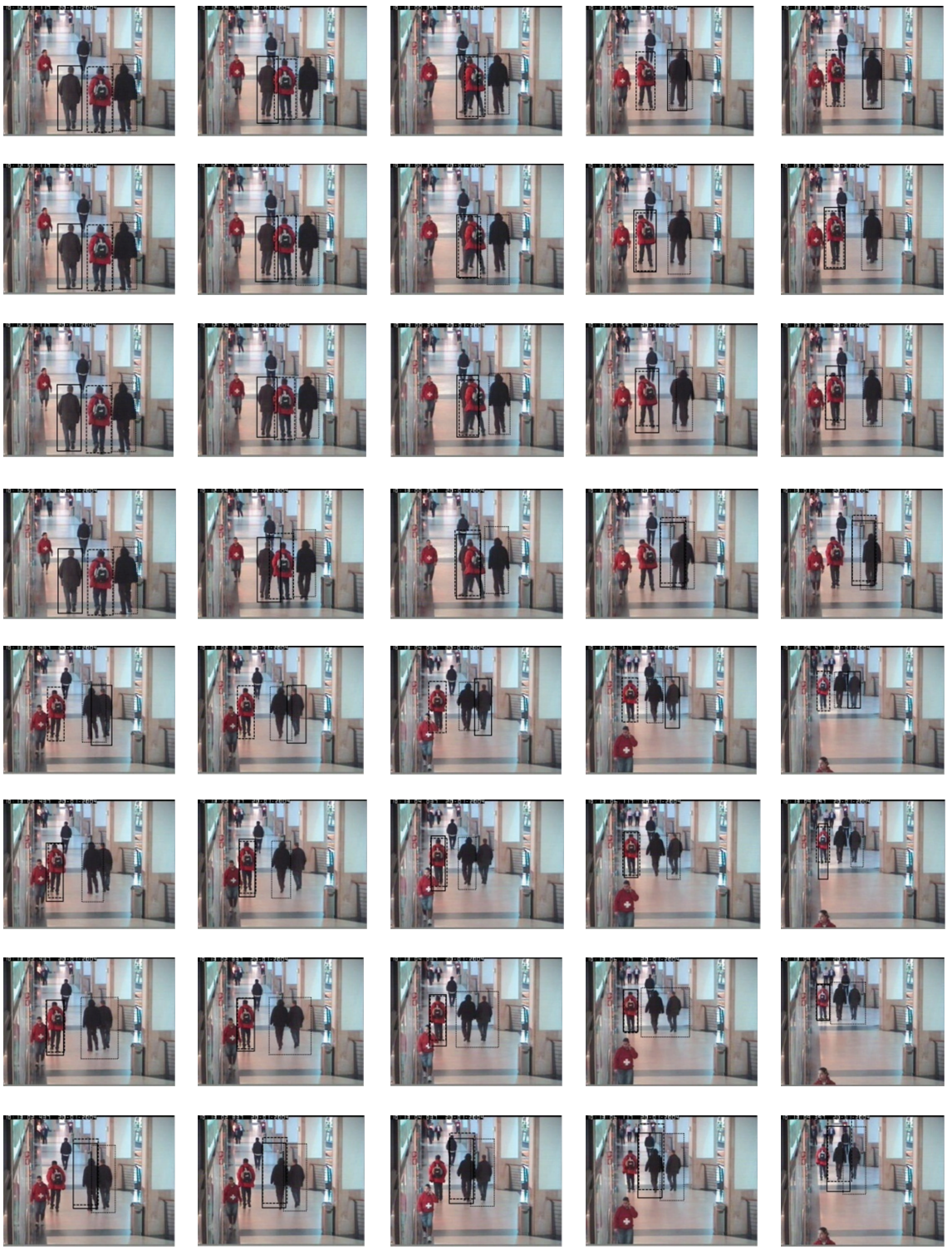

Figure 3: Some experiments results on CAVIAR data sets (proposed algorithm results: 1st, 5th row; IVT algorithm results: 2nd, 6th row; L1-APG algorithm results: 3rd, 7th row; MIL algorithm results: 4th, 8th row; frames: 1, 42, 57, 87, 93, 108, 118, 148, 200, and 282).

Figure 4: Some experiments results on PETS outdoor multiperson data sets (proposed algorithm results: 1st, 5th row; IVT algorithm results: 2nd, 6th row; L1-APG algorithm results: 3rd, 7th row; MIL algorithm results: 4th, 8th row; frames: 1, 26, 31, 36, 48, 56, 59, 64, 68, and 90).

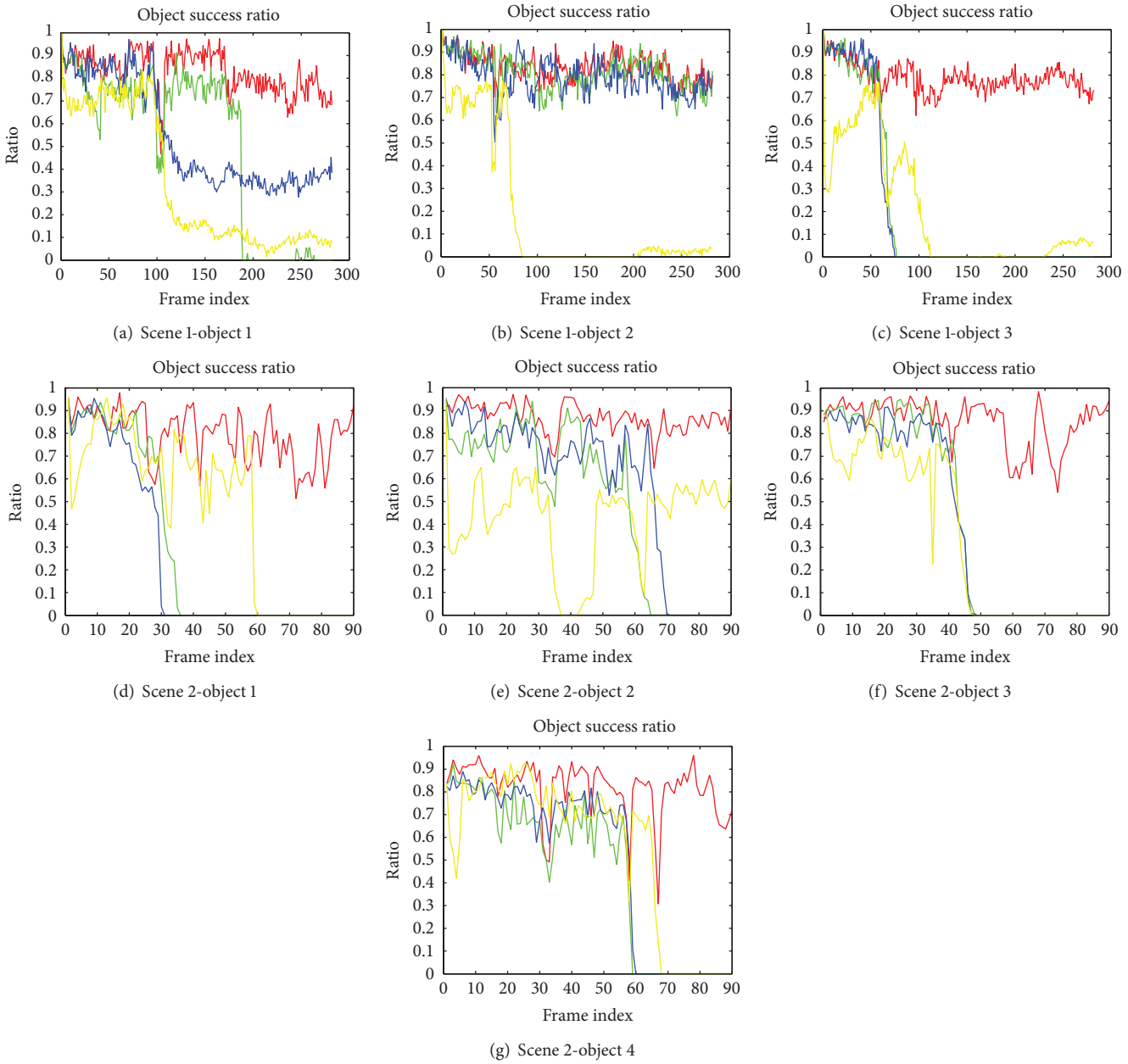

FIGURE 5: Tracking success ratio (the red line is the proposed method results, the green line is the IVT results, the blue line is the L1-APG results, and the yellow line is MIL results).

The tracking success ratio is

$$\text{ratio} = \frac{\text{area}\left(R_e \cap R_g\right)}{\text{area}\left(R_e \cup R_g\right)}, \tag{21}$$

where R_e is the experiment tracking bounding box, R_g is the ground truth bounding box, and area() means the area of the region. The tracking result in one frame is considered as a success when the tracking success ratio is above 0.5. The tracking success ratios of four trackers in two scenarios are shown in Figure 5.

As can be seen from Figure 5, the IVT and L1-APG trackers achieve tracking of object 2 in the first scenarios; the three comparison trackers do not achieve completely tracking of other objects in both scenarios due to the disturbance of background information or the similar objects. The tracking success ratios of the proposed algorithm with seven objects in two scenarios are all greater than 0.5 which means that the algorithm achieved accurate tracking and is essentially better than the other three trackers.

The center location error between experiment bounding box and ground truth bounding box is

$$e_c = \sqrt{\left(x_e - x_g\right)^2 + \left(y_e - y_g\right)^2}, \tag{22}$$

Table 1: Center point errors.

Algorithm	S1-O1-err	S1-O2-err	S1-O3-err	S2-O1-err	S2-O2-err	S2-O3-err	S2-O4-err
Proposed	**3.6782**	**2.3003**	**7.7059**	**3.2667**	**2.3803**	**2.5869**	**2.3028**
IVT	19.5312	3.6100	69.6434	101.9247	34.9553	37.5040	71.2216
L1-APG	15.1778	2.4146	68.5690	115.1737	18.7706	5.6723	32.8672
MIL	28.1737	47.1390	35.3870	56.2570	25.1693	89.4335	89.4335

where x_e, x_g, y_e, y_g are the x-axis and y-axis coordinates of the center of the experiment tracking bounding box and the ground truth bounding box.

The errors of four trackers in two scenarios are shown in Table 1. S2-O2-err represents the center location error of the second object in scenarios 2. The data in bold refer to optimal results.

As can be seen from Table 1, the other three trackers rarely achieve a complete tracking, so the tracking center point errors is large. The errors in the proposed method are significantly better than the other three trackers, and the errors are within the acceptable range.

Our tracker is implemented in MATLAB 2012a and runs at 1.1 frames and 0.8 frames per second on an Inter Xeon 2.4 GHz CPU with 8 GB RAM, which is lacking in real-time.

6. Conclusions

In this paper, we proposed a visual object tracking algorithm via feature tensor multimanifold discriminate analysis which considers the tracking is vulnerable to the interference of similar objects. The object appearance model described by feature tensor can maintain the object spatial structural which helps to deal with the partial occlusion problem and helps better to distinguish the object with similar ones in the embedded low-dimensional subspace through multimanifold discriminate analysis. In addition, the update strategy is designed from the perspective of object appearance change which is used to determine if it is needed to update the multimanifold datasets. As can be seen from the comparison experiments, the proposed algorithm is able to adapt to the object pose variation, scale change, and undisturbed tracking of similar objects in scenarios and also can achieve complete tracking even if the object was completely occluded. The proposed algorithm exist some defects, and when the object is continuously occluded in the dense moving objects scenarios, the object appearance will be incomplete which cannot construct an accurate multimanifold datasets that caused tracking failure.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (10771043) and the National Natural Science Foundation of Inner-Mongolia Autonomous Region China under Grant (2012MS0931).

References

[1] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: a review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, 2011.

[2] 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*, vol. 26, no. 1, pp. 63–84, 1998.

[3] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810–815, 2004.

[4] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 125–141, 2008.

[5] W. Hu, X. Li, X. Zhang, X. Shi, S. Maybank, and Z. Zhang, "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.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[7] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 798–805, June 2006.

[8] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011.

[9] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1830–1837, June 2012.

[10] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via structured multi-task sparse learning," *International Journal of Computer Vision*, vol. 101, no. 2, pp. 367–383, 2013.

[11] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proceedings of the British Machine Vision Conference (BMVC '06)*, pp. 47–56, September 2006.

[12] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 234–247, Marseille, France, October 2008.

[13] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007.

[14] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[15] K. Zhang and H. Song, "Real-time visual tracking via online weighted multiple instance learning," *Pattern Recognition*, vol. 46, no. 1, pp. 397–411, 2013.

[16] S. Avidan, "Support vector tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004.

[17] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631–1643, 2005.

[18] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1269–1276, San Francisco, Calif, USA, June 2010.

[19] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[20] K. Zhang, L. Zhang, and M. H. Yang, "Real-time compressive tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 864–877, 2012.

[21] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognition*, vol. 44, no. 7, pp. 1540–1551, 2011.

[22] W. Yang, C. Sun, and L. Zhang, "A multi-manifold discriminant analysis method for image feature extraction," *Pattern Recognition*, vol. 44, no. 8, pp. 1649–1657, 2011.

[23] J. Sherrah, B. Ristic, and N. J. Redding, "Particle filter to track multiple people for visual surveillance," *IET Computer Vision*, vol. 5, no. 4, pp. 192–200, 2011.

[24] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: an evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.