Pattern Recognition 48 (2015) 2459–2473

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Robust visual tracking via efficient manifold ranking with low-dimensional compressive features

Tao Zhou [a,b], Xiangjian He [c], Kai Xie [a,b], Keren Fu [a,b], Junhao Zhang [a,b], Jie Yang [a,b,*]

[a] Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240 China
[b] Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, China
[c] Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia

ARTICLE INFO

Article history:
Received 20 April 2014
Received in revised form
26 December 2014
Accepted 9 March 2015
Available online 19 March 2015

Keywords:
Visual tracking
Appearance model
Manifold ranking
Random projections
Low-dimensional compressive features
Spatial context

ABSTRACT

In this paper, a novel and robust tracking method based on efficient manifold ranking is proposed. For tracking, tracked results are taken as labeled nodes while candidate samples are taken as unlabeled nodes. The goal of tracking is to search the unlabeled sample that is the most relevant to the existing labeled nodes. Therefore, visual tracking is regarded as a ranking problem in which the relevance between an object appearance model and candidate samples is predicted by the manifold ranking algorithm. Due to the outstanding ability of the manifold ranking algorithm in discovering the underlying geometrical structure of a given image database, our tracker is more robust to overcome tracking drift. Meanwhile, we adopt non-adaptive random projections to preserve the structure of original image space, and a very sparse measurement matrix is used to efficiently extract low-dimensional compressive features for object representation. Furthermore, spatial context is used to improve the robustness to appearance variations. Experimental results on some challenging video sequences show that the proposed algorithm outperforms seven state-of-the-art methods in terms of accuracy and robustness.

1. Introduction

Visual tracking is a long standing research topic due to its wide range of applications such as behavior analysis, activity recognition, video surveillance, and human–computer interaction [1,2]. Although it has had a significant progress in the past decades, developing an efficient and robust tracking algorithm is still a challenging problem due to numerous factors such as partial occlusion, illumination variation, pose change, abrupt motion, and background clutter. These factors can lead to wrong association, and result in drift and even failure in tracking.

The main tracking algorithms can be categorized two classes: generative [3–6] and discriminative methods [7–13].

Generative methods focus on searching for the regions which are the most similar to the tracked targets with minimal reconstruction errors of tracking. Adaptive models including the WSL tracker [3] and IVT method [14] have been proposed to handle appearance variation. Recently, sparse representation methods have been used to represent an object by a set of trivial target templates and trivial templates [6,15] to deal with partial occlusion, pose variation and so on. Therefore, it is critical to construct an effective appearance model in order to handle various challenging factors. Furthermore, generative methods discard useful information surrounding target regions that can be exploited to better separate objects from backgrounds.

Discriminative methods treat tracking as a classification problem that distinguishes the tracked targets from the surrounding backgrounds. A tracking technique called tracking by detection has been shown to have promising results in real-time. This approach trains a discriminative classifier online to separate an object from its background. Collins et al. [7] selected discriminative features online to improve the tracking performance. Boosting method has been used for object tracking through combining weak classifiers to establish a strong classifier to select discriminative features, and some online boosting feature selection methods have been proposed for object tracking [8,16]. Babenko et al. [9] proposed a novel online MIL algorithm that achieved superior results with real-time performance for object tracking. An efficient tracking algorithm based on compressive sensing theories was proposed by Zhang et al. [10]. It uses low dimensional features randomly extracted from high dimensional multi-scale image features in the foreground and background, and it achieves better tracking performance than other methods in terms of robustness and speed. Moreover, although some efficient feature extraction techniques have been proposed for visual tracking [8,10,12], there

* Corresponding author at: Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, 800 Dongchuan road, Shanghai, 200240 China. Tel./fax: +86 21 3420 4033.
E-mail address: jieyang@sjtu.edu.cn (J. Yang).

http://dx.doi.org/10.1016/j.patcog.2015.03.008
0031-3203/

often exist a large number of samples from which features need to be extracted for classification, thereby entailing computationally expensive operations [9].

The above tracking methods have shown promising performance. However, they have some shortcomings. Firstly, although the goal of a generative method is to learn an object appearance model, an effective searching algorithm and an effective measuring method to match candidate samples to an object model are difficult to obtain. Secondly, background varies broadly during a tracking process, so it is difficult to achieve the aim of a discriminative method to distinguish a target region from a complicated background when the target looks similar to its background. Therefore, it is very difficult to construct a discriminative object representation. Thirdly, feature selection is of crucial importance for generating an effective appearance model. However, approaches using a large amount of features make the computational load very heavy. Therefore, the computational complexity of tracking methods is rather high, and this limits the real-time applications of these methods.

Graph-based ranking algorithms have been widely applied to information retrieval and have proved to have excellent performance and feasibility on a variety of data types [17–19]. The manifold ranking algorithm first constructs a weighted graph by considering each data node as a vertex. The ranking score of the query is iteratively propagated to nearby node via a weighted graph. Finally, nodes will be ranked according to the ranking scores, in which a larger score indicates higher relevance. In this paper, we develop a novel and robust tracking method based on manifold ranking, which regards tracking as a ranking problem. As shown in Fig. 1, we mark the tracked results as labeled nodes, while candidate samples are regarded as unlabeled nodes. The tracking objective is to estimate the corresponding likelihood that is determined by the relevance between the queries and all candidate samples. We use a manifold structure to measure the relevance between a model and samples, and in our method low-dimensional compressive features can efficiently compress features of foreground objects and their background. Experimental results on some challenging video sequences are demonstrated to show the effectiveness and robustness of the proposed model and algorithm in comparison with seven state-of-the-art tracking methods.

The main contributions of this paper are as follows:

1. A novel visual tracking method based on graph-manifold ranking is proposed.
2. An efficient manifold ranking algorithm is adopted. It can reconstruct graph efficiently in each tracking round and reduce the computation complexity.
3. Low-dimensional compressive features of an image are extracted by a very sparse measurement matrix for object representation. This matrix preserves the structure of the image and discriminates objects from their cluttered background effectively.
4. Our method exploits both temporal and spatial context information, and it is robust to appearance variations caused by abrupt motion, occlusion and background clutters.
5. Experimental results show that the proposed algorithm outperforms seven state-of-the-art methods in terms of accuracy and robustness.

This is an extension of our paper showing preliminary results in [20]. The rest of this paper is organized as follows. The graph-manifold ranking algorithm, the efficient manifold ranking algorithm and low-dimensional compressive features are described in Section 2. Details of our proposed method based on an efficient manifold ranking with low-dimensional compressive features are demonstrated in Section 3. Experimental results are shown and analyzed in Section 4. The conclusion is presented in Section 5.

2. Preliminaries

2.1. Graph-based manifold ranking

Manifold ranking (MR), a graph-based ranking algorithm, has been widely applied in information retrieval and shown to have excellent performance and feasibility on a variety of data types [17,18]. The manifold ranking method is described as follows: given a query node, the remaining unlabeled nodes are ranked based on their relevance to the given query. The goal is to learn a ranking function to define the relevance between unlabeled nodes and this query [18,19].

In [19,21], a ranking method that exploits the intrinsic manifold structure of data for graph labelling is proposed. Given a data set $X=\{x_1,x_2,\ldots,x_n\}\in\Re^{m\times n}$, where m is the dimension of feature vector and n is the number of data in the data set, some data points are labelled queries and the rest need to be ranked according to their relevance to the queries. Let $f: X\rightarrow\Re^n$ denote a ranking function which assigns a ranking value r_i to each point x_i, and r be a column vector defined by $r=[r_1,r_2,\ldots,r_n]^T$. Let $y=[y_1,y_2,\ldots,y_n]^T$ denote an indication vector, in which $y_i=1$ if x_i

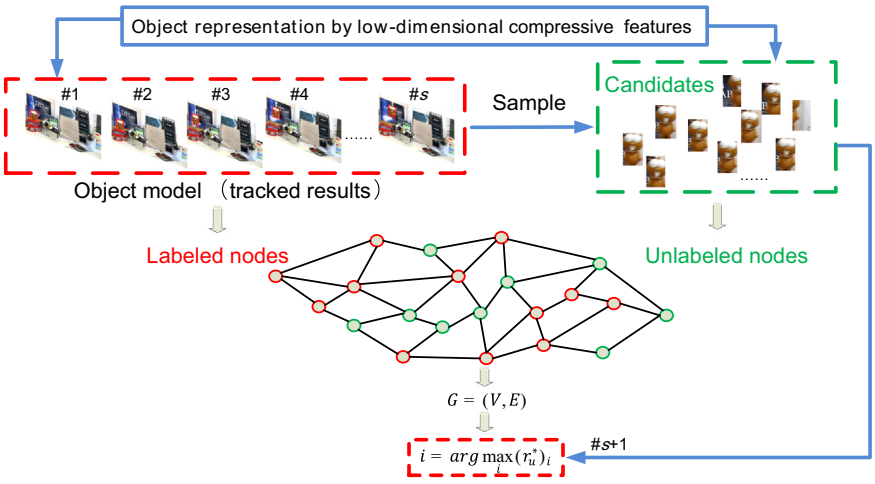

Fig. 1. Basic flow of our tracking algorithm. A graph is established combining labeled nodes (tracked results) and unlabeled nodes (candidate samples), and ranking scores represent the relevance between the object model and candidate samples.

is a query, and $y_i=0$ otherwise. Suppose all data points represent a graph $G=(V,E)$, where V represents vertex set, and edges E are weighted by an affinity matrix $W=[w_{ij}]_{n\times n}$. The strength of edge reflects the similarity between two vertices. To find the optimal ranking of queries, the cost function associated with r is defined as follows:

$$O(r)=\frac{1}{2}\left(\sum_{i,j=1}^{n}\|\frac{1}{\sqrt{D_{ii}}}r_i-\frac{1}{\sqrt{D_{jj}}}r_j\|^2+\mu\sum_{i=1}^{n}\|r_i-r_j\|^2\right) \tag{1}$$

where $\mu>0$ controls the balance of the smoothness constraint (the first term) and the fitting constraint (the second term), and D is a diagonal matrix with the element $D_{ii}=\sum_{j=1}^{n}w_{ij}$, for $i,j=1,2,\ldots,N$. To minimize the cost function, we can obtain the closed form solution as

$$r^*=(I-\alpha S)^{-1}y \tag{2}$$

where I is an identity matrix, $\alpha=1/(1+\mu)$ and $S=D^{-1/2}WD^{-1/2}$. Then, we use the iteration scheme to solve the following optimal problem:

$$r(t+1)=\alpha Sr(t)+(1-\alpha)y \tag{3}$$

where α is the control parameter, which balances each point's information between its original and neighbors' information. During each iteration, each point obtains information from its neighbors (first term), and retains its initial information (second term). The iteration process is repeated until convergence.

2.2. Efficient manifold ranking algorithm

In order to efficiently reconstruct graph, we use an efficient manifold ranking algorithm [19] to compute the ranking score. When tracking an object, the object has the same appearance in several consecutive frames, so we can use a small amount of data points to represent all data points. Moreover, the context of object tracking has the same appearance in several consecutive frames. Thus, we can use a small amount of anchor points to represent the whole data set, as each data point can be locally approximated by a linear combination of its nearby anchor points.

First, we briefly introduce how to use an anchor graph to model the data. Given a data set $X=\{x_1,x_2,\ldots,x_n\}\in\Re^{m\times n}$, $U=\{u_1,u_2,\ldots,u_d\}\in\Re^{m\times d}$ indicates a set of anchors sharing the same space with the data set. Then, we define a real value function $r:X\to R$, which assigns a semantic label for each point in X. The aim is to find a weight matrix that measures the relevance between the data points in X and the anchors in U. We obtain $r(x)$ for each point by a weighted average of these labels on anchors as follows:

$$r(x_i)=\sum_{k=1}^{d}z_{ki}r(u_k),\quad i=1,2,\ldots,n \tag{4}$$

where $\sum_{k=1}^{d}z_{ki}=1$ and $z_{ki}>0$, in which z_{ki} represents the weight representing the relevance between point x_i and an anchor u_k. The weights can be obtained through a Nadaraya–Watson kernel regression to increase smoothness. The graph construction process and the means to get the anchors can be found in [19] in detail.

We use a new approach to represent the adjacency matrix W. The weight matrix $Z\in\Re^{d*n}$ can be viewed as a d-dimensional representation of the data $X\in\Re^{m*n}$, in which d is the number of anchor points. It means that data points can be presented in a new space to replace the original feature space. We set the adjacency matrix as follows:

$$W=Z^TZ \tag{5}$$

where $W_{ij}>0$ if two points are correlative and they will share at least one common anchor point, otherwise $W_{ij}=0$. The new adjacency matrix is useful to explore relevance among data points. Let $H=ZD^{-1}$ and $S=H^TH$, Eq. (2) can be rewritten as follows:

$$r^*=(I_n-\alpha S)^{-1}y=(I_n-\alpha H^TH)^{-1}y=\left(I_n-H^T\left(HH^T-\frac{1}{\alpha}I_d\right)^{-1}H\right) \tag{6}$$

where I_n and I_d are the identity matrices, they are $n\times n$ and $d\times d$ respectively. It is easy to proof that $(I_1-\alpha H^TH)$ times $(I_1-H^T(HH^T-(1/\alpha)I_2)^{-1}H)$ obtains the identity matrix. By Eq. (6), the inversion computation part has been changed from an $n\times n$ matrix to a $d\times d$ matrix. Therefore, the change can efficiently reduce the computation load for $d\ll n$. As a result, the efficient manifold ranking algorithm has a complexity $O(dn+d^3)$.

2.3. Low-dimensional compressive features

The Haar-like features have been widely used for object representation and appearance modeling. They are typically designed for different tasks such as object detection, and objection tracking [9,10,22]. However, Harr-like features require high computational loads for feature extraction in training and tracking phases. Recently, Babenko et al. [9] adopted the generalized Haar-like features where each one was a linear combination of randomly generated rectangle features, and used online boosting to select a small set of them for object tracking. In our tracking framework, we use the low-dimensional compressive features proposed by Zhang et al. [10] for the appearance modelling. A large set of Haar-like features is significantly compressed using a very sparse measurement matrix. Object representation using the compressed features preserves the object structure represented in the original feature space, and these features in the compressed domain can be applied efficiently.

Given a random matrix $R\in\Re^{n*m}$ that projects a high-dimensional image feature $x\in\Re^m$ to a low-dimensional feature $v\in\Re^n$

$$v=Rx \tag{7}$$

where $n\ll m$. Ideally, the matrix R can provide a stable embedding to preserve the distances between all pairs of original signals. In other words, lower-dimensional features can recover original high-dimensional information. The Johnson–Lindenstrauss lemma [23] states that with high probability the distances between the points in a vector space are preserved if they are projected onto a randomly selected subspace with suitably high dimensions. Therefore, if the random matrix R in Eq. (7) meets the Johnson–Lindenstrauss lemma, we can reconstruct the original data x with minimum error from low-dimensional data v with high probability when x is compressive such as a video or an image. As such, a very sparse matrix is used for extracting compressive features, and it requests only to satisfy the Johnson–Lindenstrauss lemma in the real-time applications.

A typical measurement matrix satisfying the restricted isometry property is the random Gaussian matrix $R\in\Re^{n*m}$, $r_{ij}\sim N(0,1)$, so a very sparse random measurement matrix is defined as

$$r_{ij}=\sqrt{s}\times\begin{cases}1 & \text{with probability } 1/2s\\ 0 & \text{with probability } 1-1/s\\ -1 & \text{with probability } 1/2s\end{cases} \tag{8}$$

In order to satisfy the Johnson–Lindenstrauss lemma, the measurement matrix should be with $s=2$ or $s=3$ [23]. We can note that the measurement matrix is very easy to compute and it requires only a uniform random generator. In order to enhance the separability, distinguished ability and adaptability at a fixed scale, samples from this fixed scale are convolving with a set of rectangle

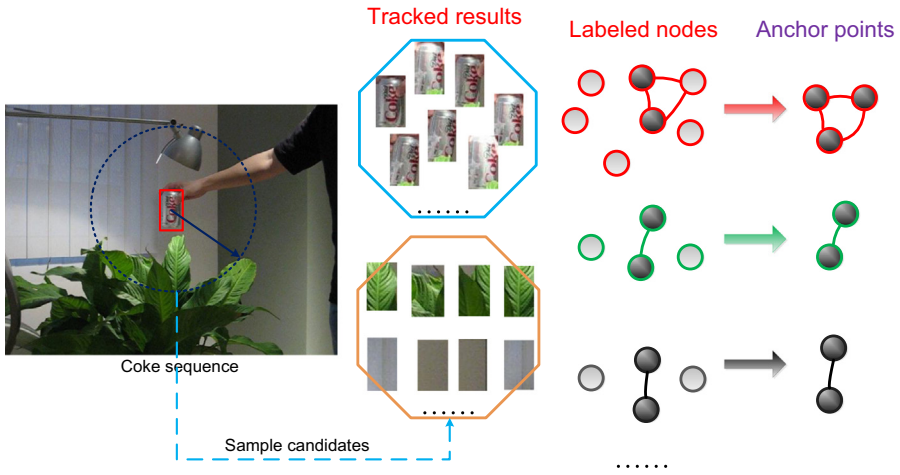

Fig. 2. Anchor points representation. For tracking, these tracked results (red circles) have the same appearance in several consecutive frames, so we can use a small amount of data points to represent all labeled points. Moreover, candidate points (green and black circles) have also same appearance and they are presented by a small amount of anchor points. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

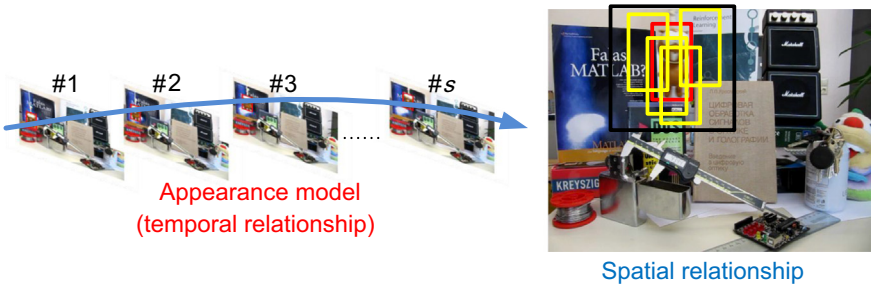

Fig. 3. Temporal and spatial relationships. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

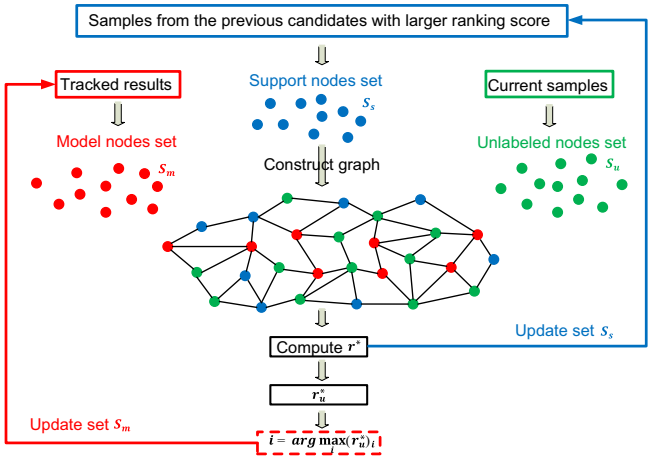

Fig. 4. The updating process of appearance model and the construction of support set.

filters at multiple scales. Each rectangular filter at a scale is defined

$$h_{p,q}(x,y)=\begin{cases}1 & 1\le x\le p, 1\le y\le q\\ 0 & \text{otherwise}\end{cases} \tag{9}$$

where p and q are the width and the height of a rectangle filter, respectively. Convolving an image patch with the rectangle filter at a fixed scale is equivalent to computing the internal image characters corresponding to this filter. Finally, we represent each filtered image as a column vector in $\Re^{wh}$ and then concatenate these vectors as a very high-dimensional multi-scale image feature vector x. Then, a very sparse matrix is adopted to project x onto a low-dimensional feature vector v. In tracking process, the sparse matrix remains fixed in the whole tracking process and it is computed once in the original stage. Therefore, a low-dimensional compressive features v can be efficiently computed and it is used to represent an object.

3. Our proposed method

3.1. Framework

Fig. 1 shows the basic flow of our proposed tracking algorithm. The tracking problem is formulated as a ranking task. Firstly, we assume that the locations in the first t frames have been obtained by the CT tracker [10]. Let $l(x_i^*)$ denote the location of tracking result at the ith frame where x_i^* represents the sample. Then, we collect these tracked results to form the object appearance model set $S_m=\{x_1^*,x_2^*,\ldots,x_i^*\}, i=1,2,\ldots,t$, and the corresponding graph is taken as G_m. Secondly, for a new frame, we crop out a set of image patches x^r with N samples near the location $l(x_t^*)$ with a search radius at the current frame, i.e., $x^\beta=\{x:\|l(x)-l(x_t^*)\|<\beta\}$. These candidate image patches are collected to form a set of unlabeled nodes, represented by, $S_u=\{x_1^{s+1},x_2^{s+1},\ldots,x_i^{s+1}\}, i=1,2,\ldots,N$, and the corresponding graph is taken as G_u. Thirdly, the candidate G_u is combined with G_m to construct a graph $G=G_m\cup G_u$, in which the label $y_i=1$ if a node point is in G_m, and $y_i=0$ if a node point is in G_u. The ranking score vector $r^*=[r_m^*;r_u^*]$ can be obtained by the manifold ranking algorithm, where r_m^* is corresponding to G_m and r_u^* is corresponding to G_u. Then, the tracking result is added into S_m, while the other candidate samples are deleted. This procedure continues to sample candidates and constructs a new graph to obtain the largest ranking score as the tracking result until the end of the image sequence.

3.2. Anchor points representation in our method

The most time consuming part of the manifold ranking algorithms is to construct graph. In order to efficiently reconstruct the graph in the proposed tracking method, we use an efficient manifold ranking algorithm to compute the ranking score. In the efficient manifold ranking algorithm, each data point on the manifold can be locally approximated by a linear combination of its nearby anchor points. Thus, we only need to construct an anchor graph, and then ranking score can be estimated for each point as a weighted average of the labels on anchors' ranking values.

In our method, the object has the same appearance in several consecutive frames, so we can use a small amount of anchor points to represent all labeled nodes as shown in Fig. 2. Moreover, candidates sampled around the previous location also have the same appearance information, so a small number of candidate nodes can used to represent the most of candidate sample points as shown in Fig. 2.

3.3. Appearance model updating process

As shown in Fig. 1, we can obtain the locations in the first t frames by a CT tracker, and then to obtain the location of the $(t+1)$th frame by the manifold ranking algorithm. There exists an obvious problem that the size of S_m will be very large if all tracked results are added into the appearance model in each tracking round, so the computation complexity will be very heavy. In addition, the bad node impacts the performance of the appearance model. To track the next frame, we need to update the appearance model firstly. We compute the average ranking score of r_m^*

$$\mu_{r_m^*} = \sum_{i=1}^{t} (r_m^*)_i \tag{10}$$

where $(r_m^*)_i$ represents the score of the ith node in S_m. Then, we compute the displacement error e_i between the score of each node in S_m and the average score

$$e_i = \|(r_m^*)_i - \mu_{r_m^*}\|^2 \tag{11}$$

We delete the node that has the largest displacement error, and then add the current tracking result x_{t+1}^* into S_m. Thus, the number of S_m will be t constantly. It is worth noting that the average ranking score computed from the tracked results alleviates the noise effects.

3.4. Support set construction

In our method, object appearance model S_m only reflects the temporal relationship among consecutive frames, but it does not take into account the immediately surrounding background. In the tracking process, the context of a target in an image sequence consists of the spatial context including the local background and the temporal context including all appearances of the targets in the previous frames. As shown in Fig. 3 (left), our object appearance model S_m represents the temporal context in the previous frames. In Fig. 3 (right), note that the object can be influenced by its surrounding background, and there exists a correlation between the object (denoted by red rectangle) and its surrounding background (denoted by yellow rectangle). Therefore, in order to make use of surrounding background information and provide as much appearance information as possible for graph construction, we establish a support set to describe spatial context. The spatial context describes the relevance between the object and its surrounding background in a small neighborhood region.

Supposed that, in tracking the $(t+1)$th frame, we have obtained the object location $l(x_{t+1}^*)$ with ranking score, and the ranking score of the current candidate samples is denoted by r_u^*. We select s nodes from the candidate samples set S_u to construct the support set S_s. S_s is corresponding to the first $s+1$ largest ranking scores among all obtained r_u^*, and we then delete the largest one. The graph corresponding to the support set is denoted by G_s. The updating process of the appearance model and the construction of the support set construction are shown in Fig. 4.

Algorithm 1. The proposed tracking method.

Input: Video frame $f=1{:}F$

1. The first t frames are tracked by a CT tracker to construct an object appearance model set $S_m = \{x_1^*, x_2^*, \ldots, x_i^*\}$
2. for $f=t+1$ to F do
3. Crop out a set of candidate samples as unlabeled set S_u by $x^\beta = \{x : \|l(x) - l(x_t^*)\| < \beta\}$.
4. if $f==t+1$
5. Construct a graph $G = G_m \cup G_u$ and support set S_s.
6. Update model set S_m.
7. else
8. Construct a graph $G = G_m \cup G_s \cup G_u$.
9. Update model set S_m and support set S_s.
10. end if
11. The ith candidate sample that has the largest rank value in all r_u is taken as the object location in frame f and is denoted by $l_f(x^*)$, i.e., the i is selected by $i = argmax(r_u^*)_i$.
12. end for

Output: Tracking results $\{l_1(x^*), l_2(x^*), \cdots, l_F(x^*)\}$.

To track the $(t+2)$ th frame, a graph $G = G_m \cup G_s \cup G_u$ is constructed and the label $y_i=1$ if a node point is from S_m and S_s, while $y_i=0$ if a node point is from S_u. The ranking score matrix $r^* = [r_m^*; r_s^*; r_u^*]$ can be obtained by an efficient manifold ranking algorithm (see Section 2.2), where r_m^*, r_s^*, and r_u^* are corresponding to G_m, G_s, G_u respectively. The tracking scheme is summarized in Algorithm 1. Finally, the target in frame $t+2$ is the sample with the largest component in r_u^*, as the ith sample can be selected from S_u and computed by

$$i = \underset{i}{\operatorname{argmax}} (r_u^*)_i, \quad i = 1, 2, \ldots, N \tag{12}$$

where N is the number of candidate samples.

4. Experimental results and analysis

4.1. Experimental setup

We evaluate the proposed tracking method based on an efficient manifold ranking algorithm and an object representation with low-dimensional features using seven video sequences with impacted factors including abrupt motion, cluttered background,

Table 1
Evaluated video sequences.

Sequences	#Frames	Challenging factors
Deer	71	Abrupt motion, background clutter
Coke	291	Abrupt motion, partial occlusion
Bolt	293	Partial occlusion, scale change
Stone	593	Partial occlusion, background clutter
Couple	140	Partial occlusion, abrupt motion, background clutter
Lemming	1336	Partial occlusion, abrupt motion, background clutter
DavidIndoor	252	Partial occlusion, illumination variation

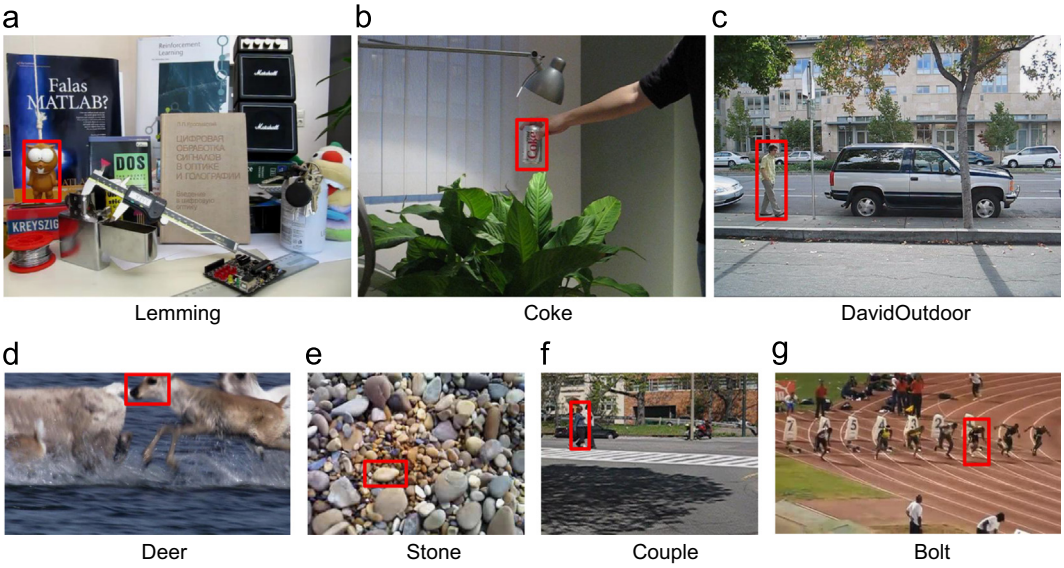

Fig. 5. The caption of tracking different objects in video sequences. (a) Lemming, (b) Coke, (c) DavidOutdoor, (d) Deer, (e) Stone, (f) Couple, and (g) Bolt.

Table 2
Center location error (CLE). **Bold** fonts indicate the best performance while the *Italic* fonts indicate the second best ones.

Sequence	L1	CT	MIL	IVT	Frag	WMIL	LOT	Ours
Coke	85.3	42.0	*26.9*	70.5	124.8	76.9	42.5	**21.6**
Bolt	39.4	211.4	35.8	138.8	*18.8*	214.3	68.2	**7.6**
Deer	171.5	95.1	66.5	127.5	92.1	*25.1*	65.9	**23.0**
Stone	19.2	32.8	32.3	**2.5**	65.9	99.8	28.1	*6.4*
Couple	110.6	33.8	33.9	105.1	*32.6*	34.4	37.8	**9.3**
Lemming	184.9	26.3	25.9	93.4	149.1	96.9	**19.9**	*24.3*
DavidOutdoor	100.4	87.3	*38.4*	52.9	90.5	73.3	63.5	**29.5**
Average CLE	101.6	75.5	*37.1*	84.4	81.9	88.7	46.6	**17.7**

Table 3
Success rate (SR)(%). **Bold** fonts indicate the best performance while the *Italic* fonts indicate the second best ones.

Sequence	L1	CT	MIL	IVT	Frag	WMIL	LOT	Ours
Coke	13.1	50.2	*72.2*	15.8	3.5	44.8	13.7	**79.4**
Bolt	27.5	4.7	44.4	3.4	*54.6*	3.1	17.4	**81.7**
Deer	3.9	14.1	21.3	11.7	7.6	*83.5*	35.2	**85.9**
Stone	29.2	*35.2*	32.1	**65.2**	15.4	8.4	27.8	**65.2**
Couple	12.3	67.8	71.4	10.1	64.3	65.3	*69.7*	**92.8**
Lemming	3.9	74.8	53.5	17.8	13.4	24.4	**84.5**	*82.1*
DavidOutdoor	27.5	22.4	*64.8*	41.1	19.5	29.8	31.2	**72.3**
Average SR	16.8	38.5	*51.4*	23.9	25.5	37.1	40.0	**79.9**

severe occlusion and appearance change (see Table 1). Fig. 5 shows the caption of tracking different objects in video sequences. We compare our proposed tracker with seven other state-of-the-art methods including: L1 tracker (L1) [6], real-time compressive tracking (CT) [10], multiple instance learning tracker (MIL) [9], incremental visual tracking (IVT) [14], fragment tracker (Frag) [4], weighted multiple instance learning tracker (WMIL) [24] and locally orderless tracking (LOT) [25]. For fair comparison, we adopt the source codes or binary codes provided by the authors with tuned parameters for best performance. For some trackers involving randomness, we repeat the experimental results five times on each sequence and obtain the averaged results.

In our experiments, the parameters are used in our algorithm as follows. The dimensionality of the compressive feature is set to 200. The first t frames are tracked by the CT method and t is set to 30. In the CT method, we randomly select 45 positive samples and 50 negative samples. The number of nodes in the support set is set $s=10$.

4.2. Quantitative analysis

We perform experiments on seven publicly available standard video sequences. As the ground truth, the center position of a target in a sequence is labeled manually. This ground truth is provided in Wu's work [26]. For quantitative analysis, we use average center location errors as evaluation criteria to compare the performance, and the pixel error in every frame is defined as follows:

$$\mathrm{CLE}=\sqrt{(x'-x)^2+(y'-y)^2} \tag{13}$$

where (x',y') represents the object position obtained by different tracking methods, and (x,y) is the ground truth. The second evaluated metric is the success rate [27], and the score in every frame is defined as follows:

$$\mathrm{score}=\frac{\mathrm{area}(\mathrm{ROI}_T\cap\mathrm{ROI}_G)}{\mathrm{area}(\mathrm{ROI}_T\cup\mathrm{ROI}_G)} \tag{14}$$

where ROI_T is the tracking bounding box and ROI_G is the ground truth bounding box. If the score is larger than 0.5 in one frame, the tracking result is considered as a success. Table 2 reports the center location error, where smaller CLE means more accurate tracking results. In Table 2, each row represents the average center location errors of the eight algorithms testing on a certain video sequence. The number marked in bold indicates the best performance in a certain testing sequence, and the number in italics refers to the second best result. Table 3 reports the success rates, where larger average scores mean more accurate results. From Tables 2 and 3, we can see that our method achieves the best or second best performance compared with L1, CT, MIL, WMIL, Frag, IVT and LOT for most of the sequences. Moreover, we draw the error curve according to Eq. (13) for each video sequence (Fig. 6). In addition, Figs. 7–9 show the screen captures for some of the video clips. More details of experiments are analyzed and discussed in the following subsections.

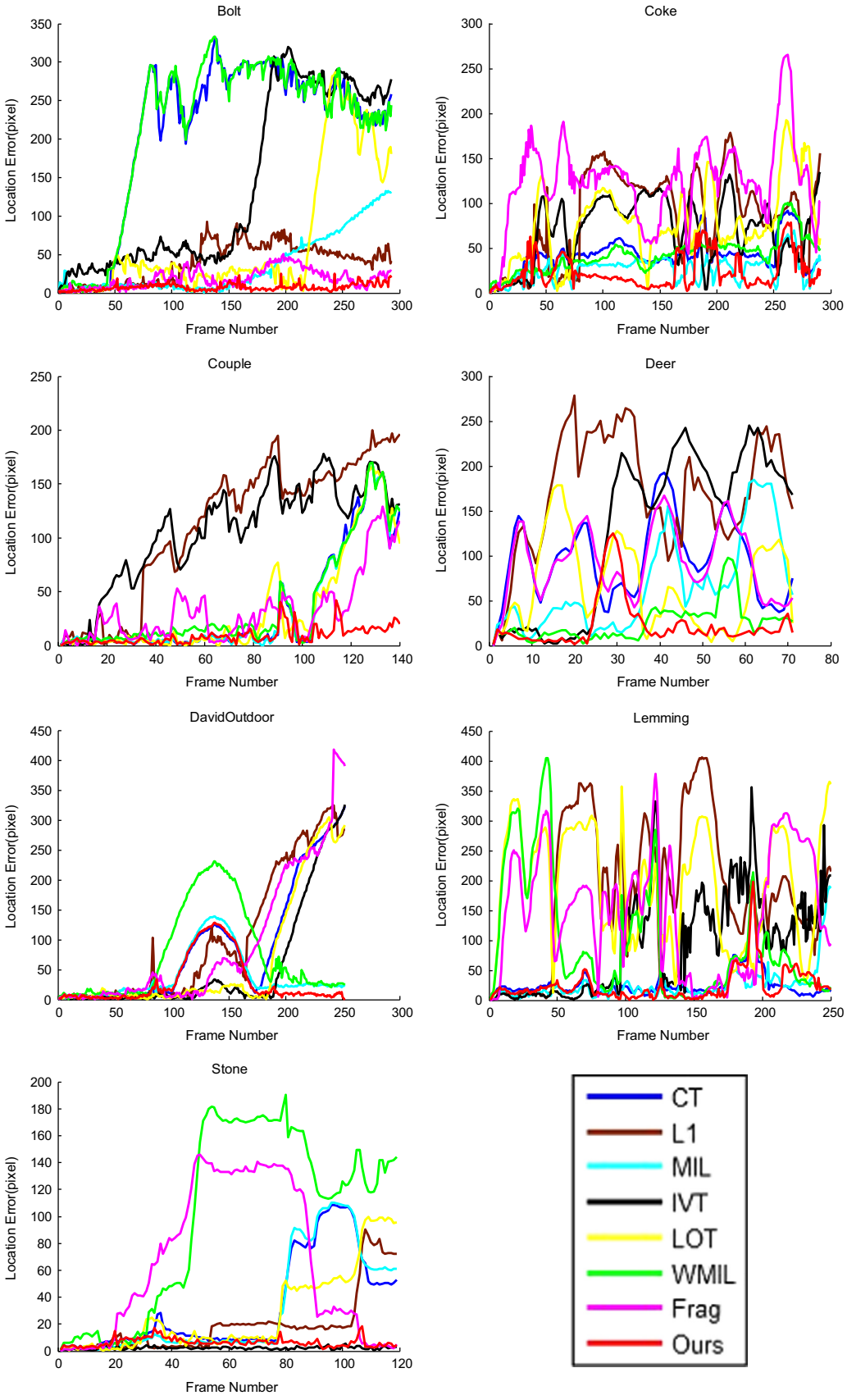

Fig. 6. Error plots of all tested sequences for different tracking methods.

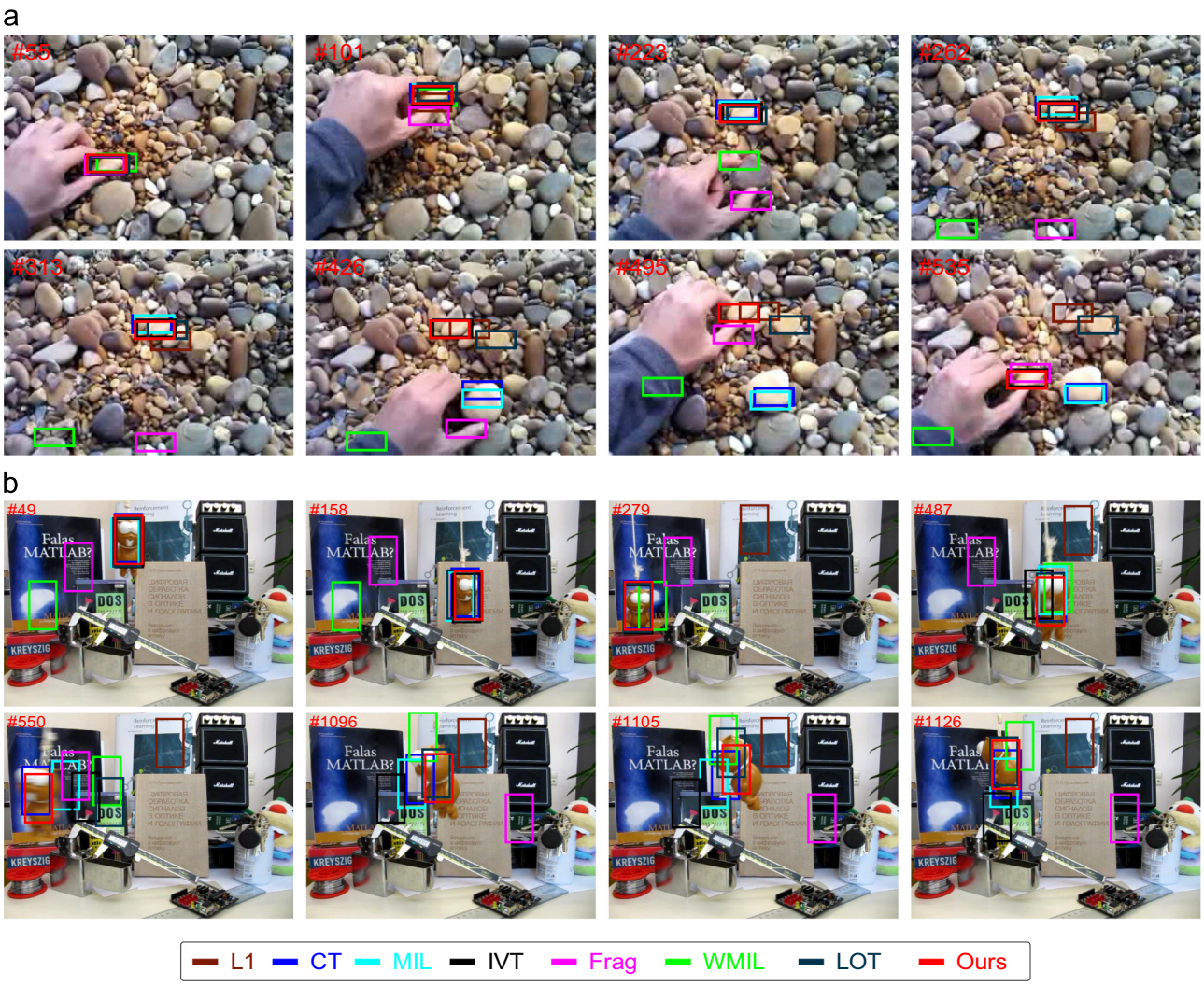

Fig. 7. Sampled tracking results for tested sequences of (a) Stone and (b) Lemming.

4.3. *Qualitative analysis*

Partial occlusion: The objects suffer heavy or longtime partial occlusion, scale change, deformation and rotation in sequences: Bolt (Fig. 9(a)), Lemming (Fig. 8(b)), DavidOutdoor (Fig. 7(c)), Coke (Fig. 7(a)). In the Bolt sequence, Fig. 9(a) demonstrates that our tracking method performs well in terms of position and scale when the objects undergo severe occlusion and deformation at frames #112, #157 and #167, while the other methods including IVT, CT, WMIL and L1 completely fail to track the objects in these frames. This can be attributed to some reasons: (1) we can extract discriminative features based on a very sparse matrix to separate an object well from its background, and also object representation with low-dimensional compressive features can preserve the structure of original image space; and (2) the outstanding ability of the manifold ranking algorithm is to discover the underlying geometrical structure and the relevance between object appearance and candidate samples. Thus, our tracker is more robust to overcome tracking drift and abrupt motion. In the DavidOutdoor sequence, our method and MIL perform better than other methods at frames #193, #206 and #252. The other methods suffer from sever drift and some of these methods completely fail to track. Furthermore, our method performs more accurately than MIL at frames #230 and #252. Thus, our method can handle occlusion and it is not sensitive to partial occlusion since the manifold ranking algorithm can measure the relevance between object appearance and candidate samples. Furthermore, we can also see the advantages of our tracking method in the Lemming and Coke sequences (see Figs. 8(b) and 7(a)).

Abrupt motion and blur: The objects in Deer (Fig. 9(b)), Coke (Fig. 7(a)), Couple (Fig. 7(b)) and Lemming (Fig. 8(b)) sequences have abrupt motions. It is difficult to predict the location of a tracked object when it undergoes an abrupt motion. As illustrated in Fig. 9(b), when an object undergoes an in-plane rotation, all evaluated algorithms except the proposed tracker do not track the object well. We also see that the WMIL method can track the object well except in frames #43 and #56. The CT method suffers completely from drifts to the background at frames #7, #17, #39, #43, #56, #60 and #68. In the Coke sequence, we can see that our method perform better than all other evaluated algorithms (see all shown frames in Fig. 8(a)). For the Couple sequence, our tracker performs better than other methods whereas the WMIL, LOT and MIL algorithms are able to track the objects in some frames. In the Lemming sequence, only the CT and our method perform well at frame #550, while the other algorithms fail to track the target objects well. What is more, the Frag method suffers completely from drift in the shown frames and it verifies that the Frag method cannot adaptively adjust these changes, resulting in serious drifts. We can also see that the LOT method can track the object well except that there are few drifts in a couple of frames see frames #550 and #1105). Blurry images exist in the Deer sequence (see Fig. 9(b)), because a fast motion makes it difficult to track the

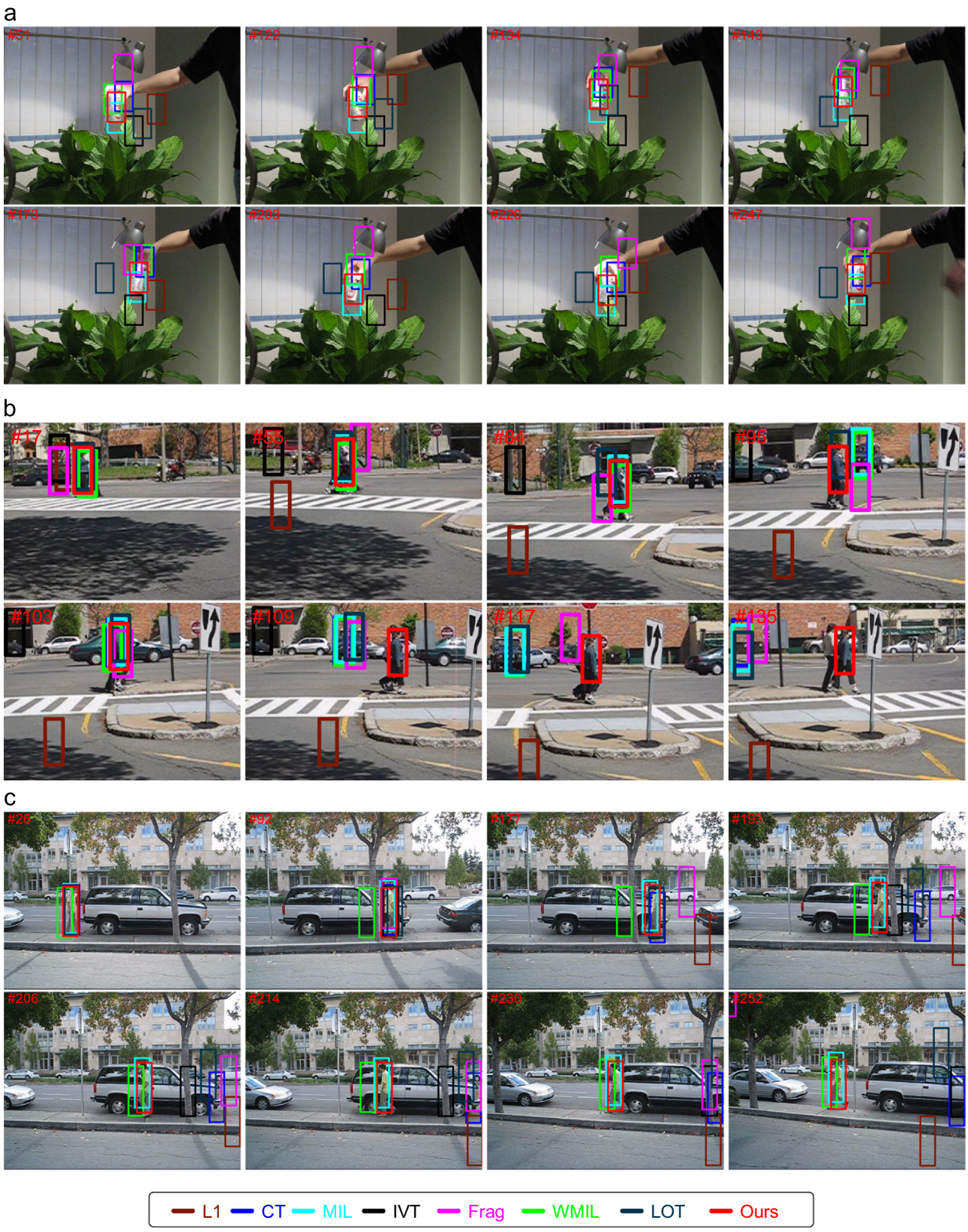

Fig. 8. Sampled tracking results for tested sequences of (a) Coke, (b) Couple and (c) DavidOutdoor.

target object. As shown in frames #56 and #71 of Fig. 9(a), the proposed method can still track the object well compared with other methods.

Background clutters: The trackers are easily confused when an object is very similar to its background. Figs. 9(b), 7(b), 8(b) and 8(a) demonstrate the tracking results in the Deer, Couple, Lemming and Stone sequences with background clutters. Fig. 8(a) shows that different trackers track a yellow cobblestone located among a lot of similar stones. Thus, it is very difficult to distinguish the object from its background and to keep tracking the object

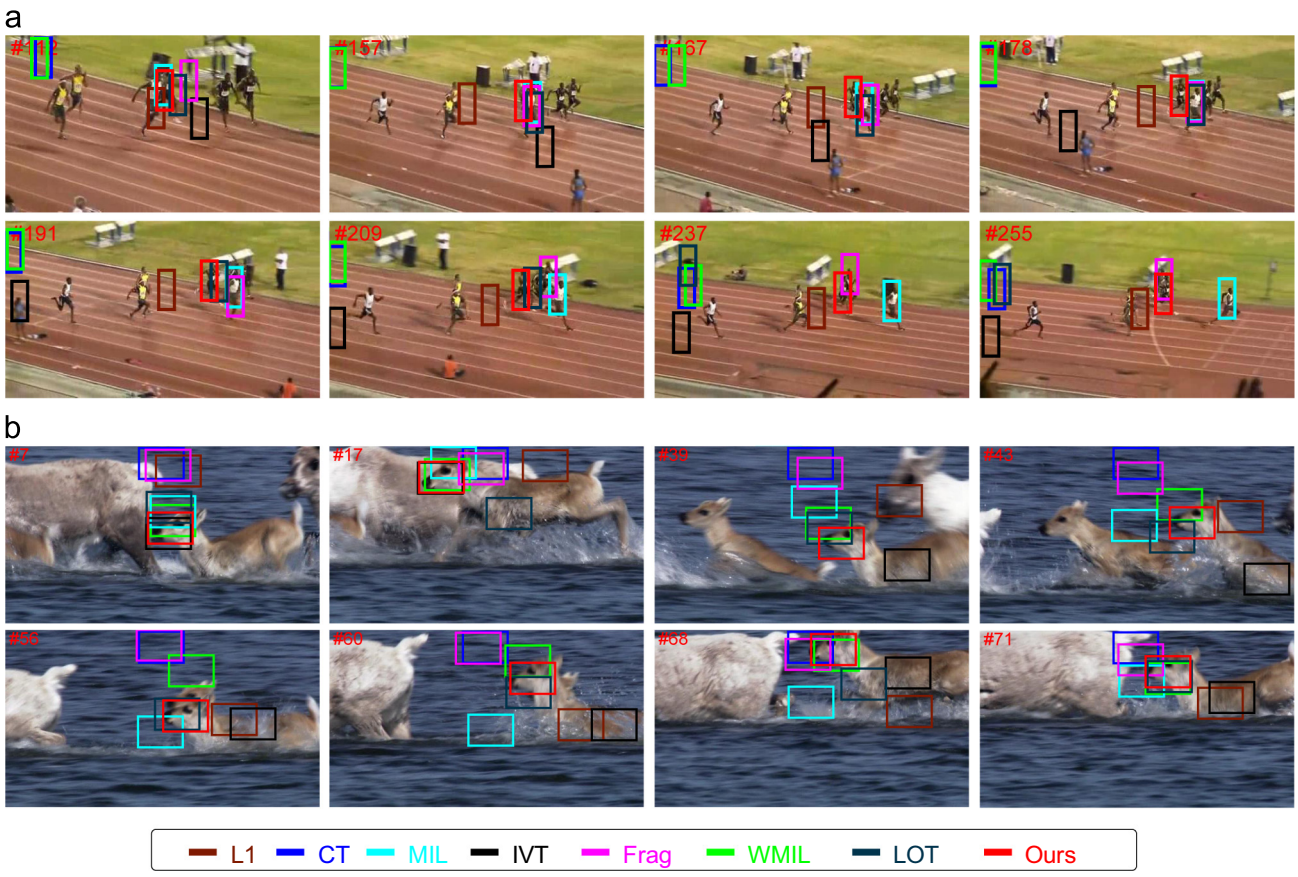

Fig. 9. Sampled tracking results for tested sequences of (a) Bolt and (b) Deer.

correctly. Comparatively, our method and IVT exhibit a better discriminative ability and outperform other methods at frames #495 and #535. The MIL and WMIL trackers completely drift to the background at frames #426, #495 and #535 and this verifies that the selected features by the MIL and WMIL trackers are less informative than our method. The Frag tracker has severe drifts at all frames except frames #55 and #535 and is unable to handle a large background clutter because its template is not updated online. The CT method has severe drifts at frames #426, #495 and #535 because it only uses compressive features and the Bayesian classifier is sensitive to background clutter.

In the Deer sequence, our method outperforms all other methods in all given frames. In the Lemming sequence, the L1 tracker completely drifts to the background at all given frames and this verifies that sparsity is not useful for tracking. The Frag, IVT, L1 methods suffer from severe drifts at frames #1105 and #1126 as shown in Fig. 9(b). Meanwhile, CT performs well too at frames #1105 and #1126, but creates a drift at frame #1096, because it is sensitive to the background clutter and the abrupt motion. In the Couple sequence, the IVT and L1 methods completely fail to track the object at all shown frames, while the MIL, WMIL and CT methods can track well in the first frame. However, they completely fail to track at other frames because they cannot effectively distinguish the object from the background clutters.

4.4. Comparison our method with other classifiers

It should be noted that the proposed tracking algorithm is significantly different from other classifiers such as support vector machine method (SVM). The outstanding ability of the manifold learning algorithm is to discover the underlying geometrical structure and the relevance between different data in a data set. To verify that the performance of our tracker outperforms the performance of methods using a SVM classifier, we construct two tracking methods using SVM. In Fig. 10(1), we assume the locations in the first t frames have been obtained by the CT tracker shown in [10]. Then these tracked results are selected as positive samples, while many image patches away from the current location are selected as negative samples (see Fig. 10(1) for details). In Fig. 10(2), we collect these image patches around the current location as positive samples, and the image patches away from the current location as negative samples.

In these experiments, we use the Haar features to represent the object and the dimensionality of the compressive features is set to 200. The first t frames are tracked by the CT method and t is set to 30. Table 4 reports the center location error, where smaller CLE means more accurate tracking results. From Table 4, we can see that our method achieves the best performance compared with SVM classifiers. Fig. 11 shows the screen captures for some of the video clips. In the Bolt sequence, we can see that the two SVM based methods completely fail to track the target object in frame #200 and there are some tracking error in frames #130 and #130. In the DavidOutdoor and Lemming sequences, our tracker performs better than other methods.

4.5. Tracking with different numbers of labeled and unlabeled nodes

We discuss the effect of the proposed tracking method against different numbers of labeled and unlabeled nodes. In our method, the tracked results and the support set are regarded as labeled nodes, while all candidates sampled around the location in the previous frame are regarded as unlabeled nodes. To sample candidates, we crop out a set of image patches x^r with N samples near the location in previous frame with a search radius β at the current frame, i.e., $x^\beta = \{x : \| l(x) - l(x_t^*) \| < \beta\}$. The parameter β is related to target's motion speed and represents the radius of

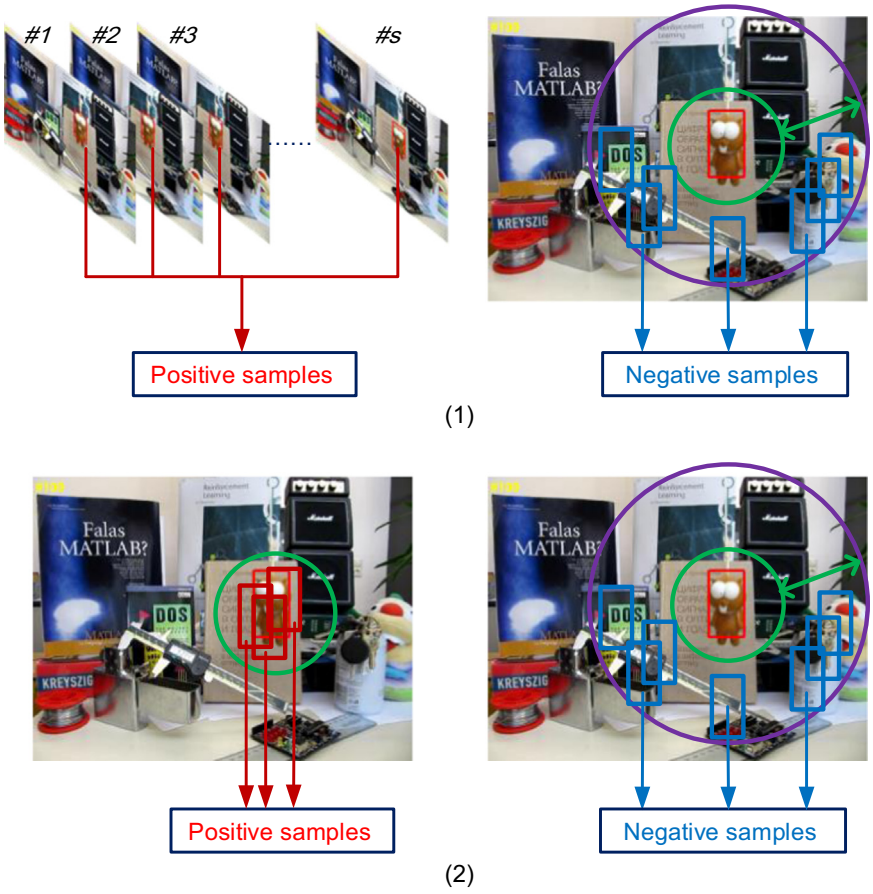

Fig. 10. Two sampling methods using SVM classifier. *Top*: the tracked results are selected as positive samples, while many image patches away from the current location are selected as negative samples (SVM(a)); *bottom*: the image patches around the current location are collected as positive samples, and the patches away from the current location as negative samples (SVM(b)).

Table 4
Center location error (CLE) for comparing our method with SVM classifiers.

Methods	Deer	Stone	Coke	Bolt	Couple	Lemming	DavidOutdoor
SVM(a)	97.1	82.4	56.5	81.8	33.4	162.1	68.7
SVM(b)	60.2	68.9	86.1	33.4	18.9	165.6	67.9
Ours	23.0	6.4	23.9	7.6	9.3	24.3	29.5

search window. The radius value must be large if the object moves quickly between the consecutive frames. Moreover, the larger the radius is, the bigger the number of candidates is.

Tables 5 and 6 report that the center location errors against different numbers of labeled and unlabeled nodes on Coke and DavidOutdoor sequences, respectively. In Tables 5 and 6, L is the number of labeled nodes and β is the search radius. As indicated in Table 5, our tracker can obtain better performance when 40 or 50 labeled nodes are selected and β is set to around 20. These labeled nodes can construct the appearance model adequately, and the manifold ranking algorithm can discover the underlying geometrical structure and the relevance between object appearance and candidate samples.

However, tracking performance is a bit sensitive to the number of labeled nodes, as labeled nodes include these tracked results of the CT tracker and the support set. If there are tracking drifts in the first t frames using the CT tracker, these labeled nodes cannot effectively model the appearance information. As a result, it will bring the accumulated error as shown in Tables 5 and 6.

In order to further analyze the effect of the number of labeled nodes, we set a fixed search radius to compare against different numbers of labeled nodes. Fig. 12 shows the center location errors with different numbers of labeled nodes under a fixed search radius. As shown in Fig. 12, our tracker has better performance with 40 labeled nodes in all the shown sequences. There are some big tracking errors for some numbers of labeled nodes because of the original tracked results.

Overall, we usually select 40 labeled nodes including the tracked results in the first 30 frames and the support set. On one hand, the labeled node set can effectively construct an object appearance model. On the other hand, it can the accumulated error when using only the tracked results for the first 30 frames. In our future work, we will use other robust tracking methods to obtain the locations in the first t frames. Moreover, we improve to reduce the sensitiveness to the labeled nodes.

4.6. Complexity analysis

The most time consuming part of the proposed tracking algorithms is to construct the graph. In the original manifold ranking algorithm, it usually uses kNN graph with its good ability to capture the local structure of the data. But the construction cost for kNN graph is $O(kn^2)$, which is very complex in large scale situations. In our method, the inversion computation part has been changed from an $n \times n$ matrix to a $d \times d$ matrix. If $d \ll n$, this change can significantly speed up the calculation of manifold ranking, which is very important for real-time object tracking. As a result, the efficient manifold ranking algorithm has a complexity $O(dn+d^3)$. Due to a low complexity for computing the ranking function r^*, we can reconstruct the graph in each tracking round efficiently.

Fig. 11. Comparison our tracking method with SVM classifiers (a) Bolt (b) DavidOutdoor (c) Lemming.

Table 5
Comparison of center location errors against different numbers of labeled and unlabeled nodes on Coke sequence.

L	β												
	13	14	15	16	17	18	19	20	21	22	23	24	25
20	72.7	81.4	80.1	71.4	80.8	79.8	75.0	81.5	135.4	131.6	76.8	76	146.6
30	42.2	65.3	37.5	37.9	40.3	53.3	39.5	43.0	38.5	37.5	60.5	43.6	129.2
40	162.1	36.6	32.7	33.2	33.1	29.4	37.0	21.6	34.3	33.3	36.4	44.1	35.6
50	41.0	27.1	20.2	42.1	37.4	21.5	35.3	31.8	31.8	39.2	38.1	39.5	40.2
60	136.1	49.3	38.1	37.0	33,9	39.5	33.5	32.8	36.7	36.3	34.5	38.4	34.9
70	27.4	31.7	28.6	32.8	132.7	29.9	34.0	35.2	36.3	34.6	37.9	36.1	35.2
80	21.9	33.5	37.8	35.2	33.9	35.2	33.1	33.8	34.7	38.9	35.8	34.9	35.3
90	40.3	56.2	35.6	33.2	21.7	32.9	36.4	33.4	32.7	32.0	32.1	31.0	36.1
100	27.5	31.4	34.3	48.0	28.1	34.3	32.6	33.9	31.5	32.3	29.6	37.2	36.0

Table 6
Comparison of center location errors against different numbers of labeled and unlabeled nodes on DavidOutdoor sequence.

L	β												
	13	14	15	16	17	18	19	20	21	22	23	24	25
20	110.0	68.7	87.1	108.7	108.6	32.6	77.1	67.7	35.8	88.1	88.9	71.6	90.5
30	30.1	30.9	79.9	86.4	73.7	84.9	91.1	79.6	85.2	85.8	67.5	89.5	32.7
40	66.8	105.8	88.4	32.6	83.8	31.2	33.2	29.5	32.7	41.4	33.0	92.4	86.9
50	65.7	68.3	67.6	67.7	92.8	31.5	85.9	33.0	85.3	33.5	32.7	32.0	32.9
60	103.7	66.4	65.6	66.7	30.9	32.8	33.0	34.1	32.2	32.6	31.9	31.8	32.5
70	57.2	85.4	97.3	93.1	91.7	33.9	32.3	33.5	33.3	32.0	32.3	32.6	32.9
80	66.9	68.9	82.8	82.9	92.1	74.4	75.2	33.6	32.8	31.9	32.9	32.8	32.5
90	66.4	69.6	68.3	69.4	105.6	71.8	70.2	32.5	93.0	72.3	33.0	31.6	32.1
100	68.7	70.7	72.9	68.4	74.6	94.7	79.5	31.3	31.5	92.8	91.5	93.1	31.8

For IVT method, the computation involves matrix–vector multiplication and the computation complexity is $O(dk)$. The computation complexity of L1 tracker using LASSO algorithm is $O(d^2+dk)$. The computation complexity of the CT tracker using random projection to extract features is $O(cn)$. The computational load of our method is mainly to extract features and construct a graph, and the complexity is $O(cn+dn+d^3)$, where c is the number of nonzero entries in each row of projection matrix R.

In order to compare the detailed computational time of our tracker with other tracking methods, we test different trackers using MATLAB on an i3 3.20 GHz machine with 4 GB RAM. Then, all trackers are implemented on different video sequences, the whole running time is stored on each sequence, and then we can obtain the frames per second (FPS) at the current sequence. Finally, we report the average FPS from the all test sequences in Table 7.

Table 7
Comparison with average FPS.

Algorithm	L1	CT	MIL	IVT	WMIL	LOT	Ours
Average FPS	1.5	54.3	15.2	26.1	32.4	2.7	9.4

4.7. Discussion

As shown in our experiments, our method can address these factors including abrupt motion, cluttered background and occlusion more effectively. The reasons are as follows. (1) We can extract discriminative features based on a very sparse matrix to separate an object well from its background, and the object representation with low-dimensional compressive features can preserve the structure of original image space. (2) The outstanding ability of the manifold ranking algorithm is to discover the underlying geometrical structure and the relevance between object appearance and candidate samples. (3) Our method combines temporal and spatial context information for tracking, and it is very insensitive to multiple factors. Thus, our tracker can obtain favorable performance.

However, our proposed method may fail when an out-of-plane rotation and an abrupt motion occur in the current sequence (see Fig. 13). Fig. 13(a) shows an out-of-plane rotation and an abrupt motion after #75. Our method drifts away the ground truth because the appearance model cannot match well between the object model and the candidates, and it cannot distinguish the object from the changed background. Moreover, our method is sensitive when there exists a complex background and when there exists similar appearance information between the object and the non-objects in a sequence (Fig. 13(b)). Therefore, our method

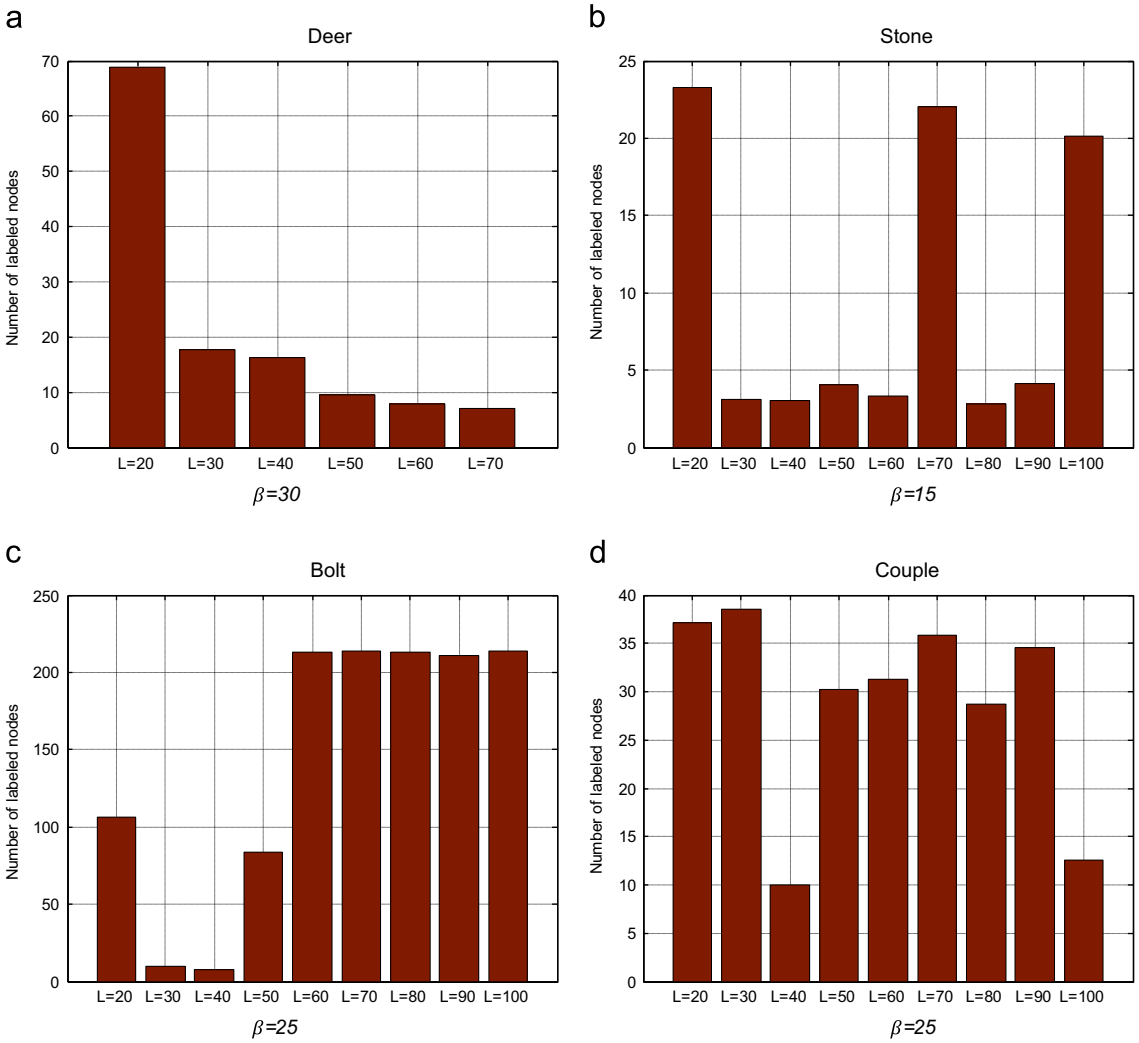

Fig. 12. Comparison of center location errors against different numbers of labeled nodes under a fixed search radius.

Fig. 13. Two failed tracking cases: (a) out of plane rotation and abrupt motion; (b) similar appearance information between object and non-objects.

cannot distinguish an object from background clutters in the above mentioned two cases.

Overall, our method performs favorably against the other state-of-the-art tracking methods in the challenge sequences.

5. Conclusions

This paper has proposed a novel framework named manifold ranking based visual tracking. The algorithm was initially proposed to rank data along their manifold, and has been widely applied in information retrieval and shown to have excellent performance and feasibility on a variety of data types. In order to address the shortcomings of original manifold ranking in graph reconstruction and heavy computation load, we adopt the efficient manifold ranking algorithm. The ability for efficiently constructing a graph is more applicable for tracking problem. What is more, we adopt non-adaptive random projections to preserve the structure of original image space, and a very sparse measurement matrix is used to efficiently extract compressive features for object representation. Furthermore, our method exploits both temporal and spatial context information for tracking, and is very insensitive to background clutters and appearance change. Experiments on some challenging video sequences have demonstrated the superiority of our proposed method to seven state-of-the-art ones in terms of accuracy and robustness.

Conflict of interest

No conflict of interest.

Acknowledgments

This research is partly supported by National Natural Science Foundation of China (NSFC) (No. 61273258) and 973 Plan, China (No. 2015CB856004).

References

[1] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, Acm Comput. Surv. (CSUR) 38 (4) (2006) 13.
[2] W. Hu, T. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object motion and behaviors, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 34 (3) (2004) 334–352.
[3] A.D. Jepson, D.J. Fleet, T.F. El-Maraghi, Robust online appearance models for visual tracking, IEEE Trans. Pattern Anal. Mach. Intell. 25 (10) (2003) 1296–1311.
[4] A. Adam, E. Rivlin I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, IEEE, New York, NY, 2006, pp. 798–805.
[5] J. Kwon, K.M. Lee, Visual tracking decomposition, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, San Francisco, CA, 2010, pp. 1269–1276.
[6] X. Mei, H. Ling, Robust visual tracking using l1 minimization, in: 2009 IEEE 12th International Conference on Computer Vision, IEEE, Miami, FL, 2009, pp. 1436–1443.
[7] R.T. Collins, Y. Liu, M. Leordeanu, Online selection of discriminative tracking features, IEEE Trans. Pattern Anal. Mach. Intell. 27 (10) (2005) 1631–1643.
[8] H. Grabner, M. Grabner, H. Bischof, Real-time tracking via on-line boosting, in: BMVC, vol. 1, 2006, pp. 6–15.
[9] B. Babenko, M.-H. Yang, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1619–1632.
[10] K. Zhang, L. Zhang, M.-H. Yang, Real-time compressive tracking, in: Computer Vision—ECCV 2012, Springer, Firenze, Italy, 2012, pp. 864–877.
[11] Z. Kalal, J. Matas, K. Mikolajczyk, Pn learning: Bootstrapping binary classifiers by structural constraints, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, San Francisco, CA, 2010, pp. 49–56.
[12] K. Zhang, H. Song, Real-time visual tracking via online weighted multiple instance learning, Pattern Recognit. 46 (1) (2013) 397–411.
[13] K. Fu, C. Gong, Y. Qiao, J. Yang, I.Y.-H. Gu, One-class support vector machine-assisted robust tracking, J. Electron. Imaging 22 (2) (2013) 023002.
[14] D.A. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, Int. J. Comput. Vis. 77 (1–3) (2008) 125–141.
[15] T. Bai, Y. Li, Robust visual tracking with structured sparse representation appearance model, Pattern Recognit. 45 (6) (2012) 2390–2404.
[16] H. Grabner, C. Leistner, H. Bischof, Semi-supervised on-line boosting for robust tracking, in: Computer Vision—ECCV 2008, Springer, Marseille, France, 2008, pp. 234–247.
[17] J. He, M. Li, H.-J. Zhang, H. Tong, C. Zhang, Manifold-ranking based image retrieval, in: Proceedings of the 12th annual ACM international conference on Multimedia, ACM, New York, USA, 2004, pp. 9–16.
[18] D. Zhou, J. Weston, A. Gretton, O. Bousquet, B. Schölkopf, Ranking on data manifolds, in: NIPS, vol. 3, 2003.
[19] B. Xu, J. Bu, C. Chen, D. Cai, X. He, W. Liu, J. Luo, Efficient manifold ranking for image retrieval, in: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, ACM, Beijing, China 2011, pp. 525–534.
[20] T. Zhou, X. He, K. Xie, K. Fu, J. Zhang, J. Yang, Visual tracking via graph-based efficient manifold ranking with low-dimensional compressive features, in: proceedings of the IEEE International Conference on Multimedia and Expo (ICME), IEEE, Chengdu, China, 2014, pp. 1-6.
[21] C. Yang, L. Zhang, H. Lu, X. Ruan, M.-H. Yang, Saliency detection via graph-based manifold ranking, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Portland, OR, 2013, pp. 3166–3173.
[22] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 1, IEEE, Kauai, HI, 2001, pp. 511–518.
[23] D. Achlioptas, Database-friendly random projections: Johnson–Lindenstrauss with binary coins, J. Comput. Syst. Sci. 66 (4) (2003) 671–687.
[24] K. Zhang, H. Song, Real-time visual tracking via online weighted multiple instance learning, Pattern Recognit. 46 (1) (2013) 397–411.
[25] S. Oron, A. Bar-Hillel, D. Levi, S. Avidan, Locally orderless tracking, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 1940–1947.

[26] Y. Wu, J. Lim, M.-H. Yang, Online object tracking: A benchmark, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Portland, OR, 2013, pp. 2411–2418.

[27] M. Everingham, L. van Gool, C.K. Williams, J. Winn, A. Zisserman, The Pascal visual object classes (voc) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.

Tao Zhou received the M.S. degree in computer application technology from Jiangnan University, Wuxi, China, in 2012. Currently, he is pursuing the Ph.D. degree at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. His current research interests include small object detection, visual tracking and machine learning.

Xiangjian He received a B.S. in mathematics from Xiamen University in 1982, an MS in applied mathematics from Fuzhou University in 1986, and a Ph.D. in computing sciences from the University of Technology, Sydney, Australia, in 1999. From 1982 to 1985, he was with Fuzhou University. From 1991 to 1996, he was with the University of New England. Since 1999, he has been with the University of Technology, Sydney. He is the director of Computer Vision and Recognition Laboratory and deputy director of the Research Centre for Innovation in IT Services and Applications at the University of Technology, Sydney.

Kai Xie received the B.S. degree in computer science from Huazhong Agricultural University, Wuhan, China, in 2008, the M.S. degree in computer science from Central South University, Changsha, China, in 2011. Currently, he is pursuing the Ph.D. degree at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. His current research interests include small object detection, object detection, visual tracking.

Keren Fu received the B.Sc. degree in automation from Huazhong University of Science and Technology, Hubei, China, in 2011. Currently, he is pursuing the Ph.D. degree at the Institute of Image Processing and Pattern Recognition, Shanghai JiaoTong University, Shanghai, China. His current research interests include object detection, saliency detection, visual tracking, and machine learning.

Junhao Zhang received the M.S. degree in School of Software from Shanghai Jiao Tong University, Shanghai, China, in 2012. Currently, he is pursuing the Ph.D. degree at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. His current research interests include image registration, image matching, feature descriptor, object detection, visual tracking.

Jie Yang received the Ph.D. degree from the Department of Computer Science, Hamburg University,Hamburg, Germany, in 1994. Currently, he is a professor at the Institute of Image Processing and Pattern Recognition, ShanghaiJiao Tong University, Shanghai, China. He has led many research projects (e.g., National Science Foundation, 863 National High Tech. Plan), had one book published in Germany, and authored more than 200 journal papers. His current research interests include object detection and recognition, data fusion and data mining, and medical image processing.