J. Vis. Commun. Image R. 31 (2015) 54–63

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Robust tracking based on local structural cell graph ☆

Heng Fan [a], Jinhai Xiang [b,*], Honghong Liao [c], Xiaoping Du [d]

[a] College of Engineering, Huazhong Agricultural University, Wuhan 430070, China
[b] College of Informatics, Huazhong Agricultural University, Wuhan 430070, China
[c] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[d] Key Laboratory of Digital Earth, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China

ARTICLE INFO

Article history:
Received 30 October 2014
Accepted 26 May 2015
Available online 5 June 2015

Keywords:
Visual tracking
Local structural cell (LSC)
Local structural cell graph (LSCG)
Bayesian framework
Graph matching
Superpixel
Local based representation
Appearance model

ABSTRACT

Structure information has been increasingly incorporated into computer vision, however most trackers have ignored the inner spatial structure of the object. In this paper, we develop a simple yet robust tracking algorithm based on local structural cell graph (LSCG). This approach exploits both partial and spatial information of the target via representing the object with local structural cells (LSCs) and constructing a graph to model the spatial structure between the inner parts of the object. The tracking is formulated as matching LSCG, whose nodes are target parts and edges are the interaction between two parts. Within the Bayesian framework, we achieve object tracking by matching graphs between the reference and candidates. Eventually, the candidate with the highest similarity is the target. In addition, an updating strategy is adopted to help our tracker adapt to the fast time-varying object appearance. Experimental results demonstrate that the proposed method outperforms several state-of-the-art trackers.

1. Introduction

Visual tracking is an essential component of many applications in computer vision, such as surveillance, human–computer interaction and robotics [1]. For robust object tracking, many different methods have been proposed. Despite reasonably good results from these approaches, some common challenges remain for tracking targets through complex scenes, e.g., when objects undergo significant pose variations or other severe deformations, i.e., object pose variations accompanied with long-term object partial occlusions or object intersections. In order to handle these problems, a wide range of appearance models for tracking have been presented by researchers [2]. Roughly speaking, these appearance models can be categorized into three types: based on visual representation such as global feature-based representations [3–5,12,15] and local feature-based representations [6–8,16]; based on statistical modeling containing generative models [3,9–11,17] and discriminative models [12,13,15,16]; based on structure information including [14,18–20]. Although structure information has drawn increasing interest in object recognition [20] and object detection [21,22], much less attention is paid to it in visual tracking.

In this paper, we exploit effective and efficient structure information for object tracking. Object is firstly segmented into superpixels. Then we construct the structural appearance model based on the superpixel map of object. The structure information is generated by superimposing a rectangular grid on top of the superpixel map as shown in Fig.1(d). In the rectangular grid, each grid is represented by its center point, which is annotated with the feature vector of the covered superpixel. Thus we obtain an undirected graph **G**, whose nodes are the grid center points and edges are the interactions between the grid points. Further, taking the local relationship of the inner object parts into consideration, a novel approach is proposed to construct the local structural cell (LSC) for each grid point. We employ these local structural cells (LSCs) to substitute the grid points in **G** and obtain a new local structural cell graph $\mathcal{G}$, whose nodes are the LSCs and edges are the interaction between LSCs. Therefore, the appearances of object parts and their relations are embedded into the local structural cell graph (LSCG), and the tracking is viewed as matching LSCG in the subsequent frames. Within the Bayesian inference framework, we can track the object by the similarities of the local structural cell graphs (LSCGs) between the reference and candidate targets, and select the candidate with maximal similarity as the object. Meanwhile, an online updating mechanism is used to adapt our tracker to occlusions and deformations.

The contributions of this work are summarized as follows. Firstly, we propose a novel appearance model LSCG to represent

☆ This paper has been recommended for acceptance by Yehoshua Zeevi.

* Corresponding author. Fax: +86 27 87286876.
E-mail addresses: hfan@webmail.hzau.edu.cn (H. Fan), jimmy_xiang@163.com (J. Xiang), hustliaohh@gmail.com (H. Liao), duxp@radi.ac.cn (X. Du).

http://dx.doi.org/10.1016/j.jvcir.2015.05.011
1047-3203/

the target, in which both the spatial and structural relationship is considered in the inner object parts. Secondly, a tracking method based on LSCG is proposed and implemented via the Bayesian framework. Finally, an intuitive and effective updating mechanism is introduced to improve robustness of the tracker in presence of appearance changes.

2. Related work

General tracking approaches can be categorized into either generative or discriminative models [2], however, only a few trackers take structure information into consideration. The discriminative methods regard tracking as a classification problem which aims to best separate the object from the ever-changing background. These methods employ both the foreground and background information. Avidan [23] proposes an ensemble tracker which treats tracking as a pixel-based binary classification problem. This method can distinguish target from background, however the pixel-based representation needs more computational resources and thereby limits its performance. In [7], Grabner et al. present an online boosting tracker to update discriminative features and further in [24] a semi-online method is proposed to handle drifting problem. Kalal et al. [25] introduce a P-N learning algorithm to learn effective features from positive and negative samples for object tracking. This tracking method nevertheless is prone to induce drifting problem when structure variations of object occur. Babenko et al. [12] utilize the multiple instance learning (MIL) method for visual tracking, which can alleviate drift to some extent. Whereas the MIL tracker may detect the positive sample that is less important because it does not consider the sample importance in its learning process. Further in [26], Zhang et al. propose the online weighted multiple instance learning (WMIL) by assigning weight to different samples in the process of training classifier. Nevertheless, the above methods undermine the robustness to occlusion and non-rigid distortion ignoring structure information.

The generative models formulate the tracking problem as searching for regions most similar to object. These methods are based on either subspace models [3] or templates [4,17]. To solve the problem of appearance variations caused by illumination or deformation, the appearance model is updated dynamically. In [3], the incremental visual tracking method suggests an online approach for efficiently learning and updating a low dimensional PCA subspace representation for the object. However, this PCA subspace based representation scheme is sensitive to partial occlusion. Adam et al. [4] present a fragment-based template model for visual tracking. Kwon et al. [17] decompose the appearance model into multiple basic observation models to cover a wide range of illumination and deformation. Similarly, the ignorance of structure information results in bad performance in deformation and occlusion.

Recently, several part-based tracking methods have been proposed [4,18,19]. In [4], the object is segmented into several fragments to construct appearance model, while this model does not consider the geometry relations between parts. Kwon et al. [18] use SIFT descriptor to generate parts, which is very unstable, and the tracking results usually consist of bad tracked parts. The tracker in [19] generates parts by oversegmenting the target into superpixels. This method formulates tracking task as figure/ground segmentation and models superpixels correspondence with CRF during the process of segmentation, whereas the high complexity of CRF constraints its further application in object tracking.

Another work similar to ours is [6], in which only a probability map of superpixels is constructed to distinguish the target from background without consideration of structure information, which is easy to cause tracking drift in color-similar background. In our work, we take structure information of target appearance into account, which enables our tracker to robustly track object in presence of occlusions and deformations.

3. Structural appearance model

In this section, the structural appearance model for tracking is presented. Section 3.1 introduces the process of constructing LSCG and the LSCG method is illustrated in Section 3.2.

3.1. Local structural cell graph (LSCG)

For the rectangular region R (target region) in one frame (See Fig. 1(a)), we firstly extract the surrounding region[1] of the target and segment it into superpixels via SLIC [28] (see Fig. 1(b) and (c)). A rectangular grid $\mathbf{\Theta}$ is then utilized to superimpose on the superpixel map of the target area $\mathbf{M}$ (see Fig. 1(d)), which segments object into several uniform blocks. In the rectangle grid, each grid block is denoted by its center point, which is related to the feature vector of the covered superpixel. Assume that the rectangular grid $\mathbf{\Theta}$ consists of $m \times n$ grid points, and the collection of these points is represented by $\mathbf{\Omega} = \{g_{ij} | i = 1, 2, \ldots, m; j = 1, 2, \ldots, n\}$, where g_{ij} denotes grid point. Taking spatial relation between the grid points into account, we can construct an undirected graph $\mathbf{G} = (\mathbf{\Omega}, \mathbf{E})$, where $\mathbf{\Omega}$ is the node set and $\mathbf{E}$ represents the edge set between adjacent nodes respectively (see Fig. 1(e)). The $\mathbf{E}$ is defined as $\mathbf{E} = \left\{ e_{(ij)(kl)} | (i-k)^2 + (j-l)^2 \leqslant r^2 \right\}$, where r represents the distance (i.e., the radius of the cell node in local structural cell graph). For each node g_{ij} in $\mathbf{G}$, we select the associated superpixel from the superpixel map $\mathbf{M}$. Specifically, let $sp_{ij} = \mathbf{M}(p_{ij})$ represent the covered superpixel of the grid g_{ij} located at position p_{ij}. For g_{ij}, the feature vector f_{ij} of the superpixel sp_{ij} is used to represent its feature vector. The collection of these feature vectors is specified by $\mathbf{F} = \{f_{ij} | i = 1, 2, \ldots, m; j = 1, 2, \ldots, n\}$.

In order to further exploit local spatial relationship between inner object parts, we define the local structural cell (LSC). For each node g_{ij} in $\mathbf{G}$, its LSC c_{ij} is represented by its node members $g_{ij}^c = \{g_{ij}\} \cup \{g_{hk} | (h-i)^2 + (k-j)^2 \leqslant r^2\}$, node member features $f_{ij}^c = \{f_{ij}\} \cup \{f_{hk} | (h-i)^2 + (k-j)^2 \leqslant r^2\}$ and local interactions of the node members (see Fig. 1(f)), where r is radius of the cell node. We utilize LSCs to replace the nodes in $\mathbf{G}$, and a new LSCG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is obtained (see Fig. 1(g)), where $\mathcal{V} = \left\{ g_{ij}^c | i = 1, 2, \ldots, m; j = 1, 2, \ldots, n \right\}$ is the cell node set and $\mathcal{E}$ is the edge set representing the interactions between cell nodes in $\mathcal{G}$. The spatial structure information of the target is thus encoded in the proposed appearance model through $\mathcal{G}$.

3.2. Similarity of local structural cell graph

In graph matching, a key aspect is how to measure the similarity between two undirected graphs. Two crucial problems exist in this process: (1) The instability of undirected graph and (2) the criteria for evaluating the correspondence between undirected graphs. In general, the object parts are represented by nodes in graph. As the target is non-rigid and deformable, the number of nodes is changeable. Even worse, some nodes in the graph may disappear, other new nodes may be generated and the spatial structure relation between nodes are mutable due to object appearance variations, which greatly constrains the application of graph model in computer vision. In addition, there are lack of effective criterias and methods to measure the similarity between graphs.

To solve the problems, we exploit both partial and spatial

[1] The surrounding region is a square area centered at the location of target X_t^c, and its side length is equal to $\lambda_s [X_s^t]^{\frac{1}{2}}$, where X_t^c represents the center location of target region X_t and X_t^s denotes its size. The parameter λ_s is a constant variable, which determines the size of this surrounding region.

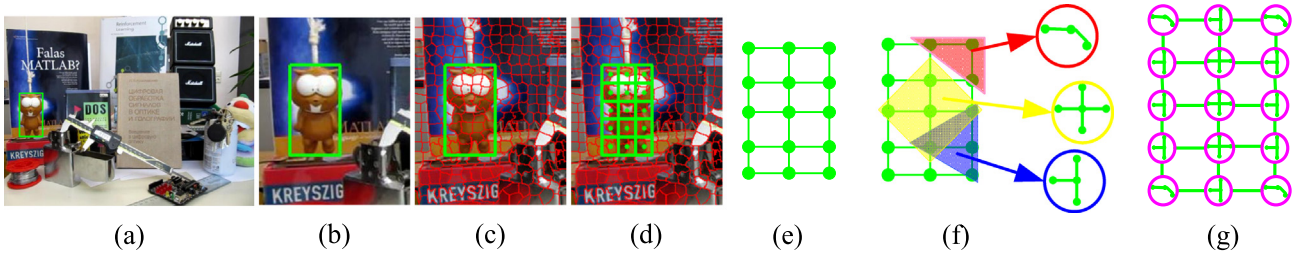


Fig. 1. Process of constructing LSCG appearance model. (a) Select target region in the frame. (b) Extract surrounding area of the object. (c) Segmentation results of (b). (d) Superimpose a rectangular grid on the superpixel map of the target. (e) Design the undirected graph. (f) Develop LSC for each grid node in (e). (g) Construct LSCG.

information of the target via representing the object with local structural cells (LSCs) and constructing a local structural cell graph (LSCG) to model the spatial structure between the inner parts of the object. The tracking task is formulated as LSCG graph matching in Bayesian framework. In LSCG, the relative position of each node and node number are fixed, which ensures the stability of graph structure. When computing the similarity between graphs, we compare the corresponding node pairs and edge pairs between the two graphs. When occlusions or deformations happen, some cells may be corrupted, which leads to failure match between certain node pairs or edge pairs. However, our local structural cell graph (LSCG) can handle these cases. Because of the stability structure of LSCG, the matching can be divided into corresponding node matching and edge matching as shown in Fig. 2(e). During matching, we firstly detect the matching degree of each node pair. In the presence of occlusions and deformations, although certain node pairs or edge pairs are invalided, others can be matched successfully, and the similarity between two graphs is obtained with the matched pairs. The process of handling occlusions or deformations can be illustrated in Fig. 2. However, other graph-based methods view object as an undirected graph. As the target is non-rigid and deformable, the number of nodes is changeable. Even worse, some nodes in the graph may disappear, other new nodes may be generated and the spatial structure relation between nodes are mutable due to object appearance variations. It is difficult to match two undirected graphs for object tracking.

In order to match two graphs, we need to expand the undirected graph LSCG to the weighted undirected graph. Let graph $\mathcal{G} = \{\mathcal{V}, \mathcal{W}\}$ and graph $\mathcal{G}' = \{\mathcal{V}', \mathcal{W}'\}$, where $\mathcal{V} = \{v_i\}_{i=1}^N$ and $\mathcal{V}' = \{v_i'\}_{i=1}^N$ are the node sets, $\mathcal{W} = \{w_{ij}\}_{i,j=1}^N$ and $\mathcal{W}' = \{w_{ij}'\}_{i,j=1}^N$ denote the weighted edge sets in $\mathcal{G}$ and $\mathcal{G}'$ respectively. The element w_{ij} of $\mathcal{W}$ is used to indicate interaction between nodes v_i and v_j and can be defined via

$$w_{ij} = \begin{cases} exp(-\|f_i^c - f_j^c\|_2^2), & v_i \text{ and } v_j \text{ are adjacent cell nodes} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where f_i^c and f_j^c are the vector features of v_i and v_j in $\mathcal{G}$ respectively. Similarly, w_{ij}' can be gotten by Eq. (1).

For the arbitrary node pair (v_i, v_i') in $\mathcal{G}$ and $\mathcal{G}'$, we utilize d_i to indicate how well these two nodes are matched, which is obtained through

$$d_i = exp(-\|f_i^c - f_i^{c'}\|_2^2) \quad (2)$$

Virtually, some nodes in the graph may be invalid due to deformations and occlusions, which undermines graph matching (see Fig. 2(e)). We use $q_i \in \{0, 1\}$ to denote the matching degree of node pair (v_i, v_i'), and it can be determined by a threshold D_{th} through

$$q_i = \begin{cases} 1, & d_i \geqslant D_{th} \\ 0, & d_i < D_{th} \end{cases} \quad (3)$$

where if q_i equals to 1, the node pair (v_i, v_i') is valid, otherwise invalid. In the graph, the edges are much dependent on nodes. If the nodes are invalid, it is meaningless to match the edge pairs. Therefore, the match similarity z_{ij} for edge pair (w_{ij}, w_{ij}') can be obtained by the edge weight via

$$z_{ij} = q_i \cdot q_j \cdot exp\left(-\sqrt{(w_{ij} - w_{ij}')^2}\right) \quad (4)$$

In this work, the graph matching is divided into node matching and edge matching. Thus the similarity $\Gamma(\mathcal{G}, \mathcal{G}')$ between graphs $\mathcal{G}$ and $\mathcal{G}'$ can be represented via

$$\Gamma(\mathcal{G}, \mathcal{G}') = \sum_{i=1}^{N} q_i \cdot d_i + \sum_{i=1}^{N}\sum_{j=1}^{N} z_{ij} \quad (5)$$

Through LSCG appearance model, we can represent the object via a stable graph. With the help of LSCG matching method proposed in this paper, it is easy to locate object among the candidates under different scenes (e.g., partial occlusion, deformation and background clutter). For each candidate, we can get a similarity score via matching, and the one with the maximal similarity is chosen to be the target. Fig. 3 illustrates the whole process.

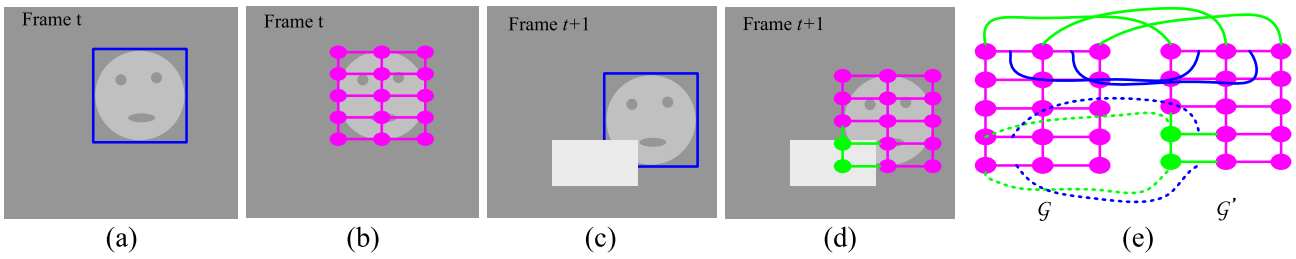


Fig. 2. The illustration of LSCG matching. Image (a) is the target in frame t, and (b) is the constructed LSCG of the target in (a). In (c), the object is occluded partially, and (d) is the LSCG of the object with some nodes being invalided. Image (e) illustrates the matching process in the presence of occlusions. The pink solid dots represent the nodes in LSCG, and the green solid dots are the occluded nodes. The green and blue solid curled lines are node pair match and edge pair match respectively, while green and blue dotted curled lines are the invalided pair match, which are not counted in computing match similarity. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

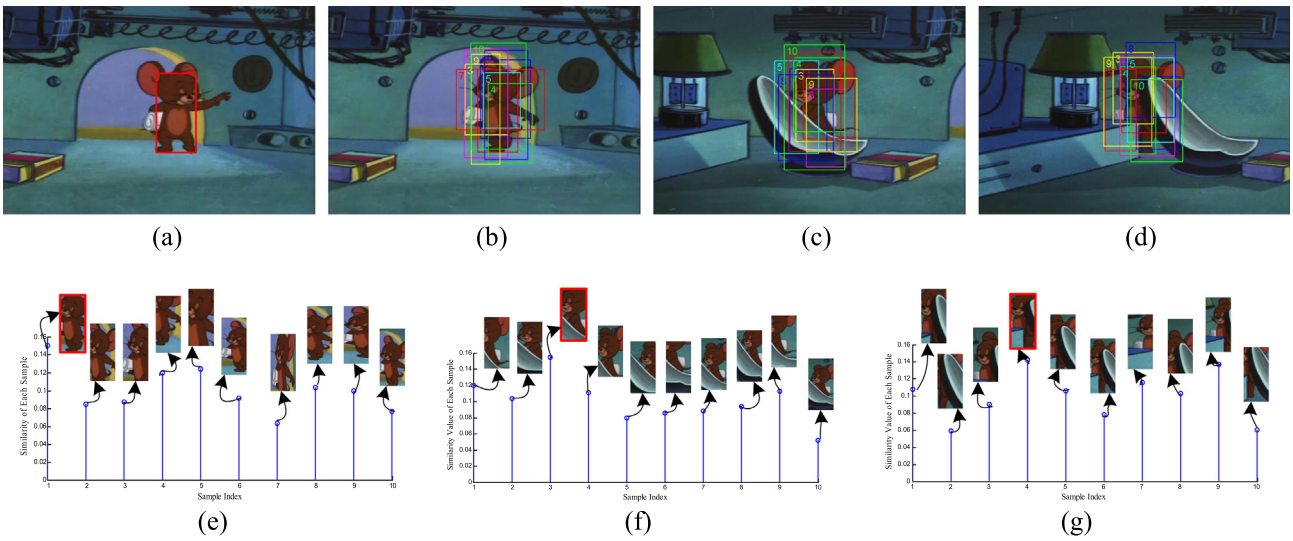


Fig. 3. Similarities between the reference and candidate objects. The reference is manually selected by a red rectangle in (a). In (b), the target undergoes deformation, and we sample some candidates by rectangles with different color. Through LSCG matching, the red frame with the maximal similarity value among all the candidates is determined to be the object as shown in (e). The target is severely occluded in (c) and (d) respectively. Likewise, we sample some candidates and compute their similarity values via matching with the reference. The red rectangles in (f) and (g) indicate the tracked object. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4. Proposed tracking algorithm

During the tracking process, the LSCG is a stable structure graph model (See Fig. 4), and thus the tracking task can be formulated as tracking LSCG. In this section, we will introduce our LSCG tracker. Section 4.1 describes the tracking formulation in this work, and an online dynamic updating mechanism is presented in Section 4.2.

4.1. Tracking formulation

Our tracker is implemented within the Bayesian inference framework. Given the observation set of target $Y^t = \{y_1, y_2, \ldots, y_t\}$ up to the frame t, we can obtain estimation $\widehat{X}_t$ by computing the maximum a posterior via

$$\widehat{X}_t = \underset{X_t^i}{\operatorname{argmax}}\ p(X_t^i|Y^t) \tag{6}$$

where $\widehat{X}_t$ denotes the i-th sample at the state of X_t. The posterior probability $p(X_t^i|Y^t)$ can be obtained by the Bayesian theorem recursively via

$$p(X_t|Y^t) \propto p(y_t|X_t) \int p(X_t|X_{t-1}) p(X_{t-1}|Y^{t-1}) dX_{t-1} \tag{7}$$

where $p(X_t|X_{t-1})$ and $p(X_{t-1}|Y^{t-1})$ represent the dynamic model and observation model respectively.

The dynamic model indicates the temporal correlation of the target state between consecutive frames. We apply affine transformation to model the target motion between two consecutive frames within the particle filter framework. The state transition can be formulated as

$$p(X_t|X_{t-1}) = N(X_t; X_{t-1}, \Psi) \tag{8}$$

where Ψ is a diagonal covariance matrix whose elements are the variance of affine parameters. The observation model $p(y_t|X_t)$ represents the probability of the observation y_t as state X_t. In this paper, the observation is designed by

$$p(y_t|X_t) \propto \Gamma_t(\mathcal{G}, \mathcal{G}^t) \tag{9}$$

where the right side of the equation denotes the similarity between the candidate and the target based on the LSCG tracking. With the template updating, the observation model can robustly adapt to the appearance change of the target.

4.2. Online update

In this paper, a novel updating mechanism is proposed for LSCG to avoid the drift caused by occlusions and deformations. We update each LSC node alone to complete the updating of LSCG. An intuitive and effective strategy is adopted here. For each LSC node in LSCG, it is much dependent on its member grid points.

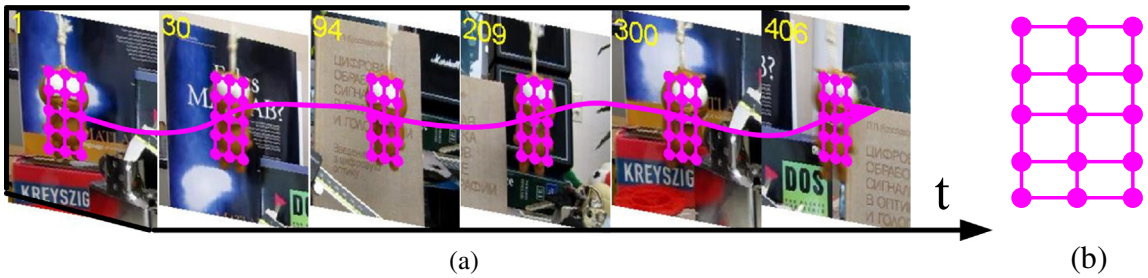


Fig. 4. During the tracking process, the structure of the LSCG appearance model keeps stable and unchangeable. (a) Shows the tracking process and (b) is the spatial structure of the LSCG. The pink solid dots denote LSC nodes in the LSCG.

For i-th node v_i in $\mathcal{G}$, we assume it consists of L grid nodes $\{g_j\}_{j=1}^{L}$. For each grid node g_j , f_{ij} denotes its feature vector (f_{ij} is initialized to f_{ij}^1) and f_{ij}^t represents its feature vector in frame t. Then we can compute their similarity $\mathcal{H}_{ij}^t$ as follows

$$\mathcal{H}_{ij}^t = exp\left(-\sqrt{\sum_{b=1}^{B}\left(f_{ij}(b)-f_{ij}^t(b)\right)^2}\right) \tag{10}$$

where B is the bin number of feature vector. We define two states for each grid node: ***live*** and ***dead***. For grid node g_j of v_i in frame t, its state is based on $\mathcal{H}_{ij}^t$. A threshold λ is set to determine its state via

$$\begin{cases} \mathcal{H}_{ij}^t \geqslant \lambda & \textbf{\textit{live}} \\ \mathcal{H}_{ij}^t < \lambda & \textbf{\textit{dead}} \end{cases} \tag{11}$$

Intuitively the more the live member grid points are, the more useful the LSC nodes are for LSCG matching. Based on the rate of live grid points, we use the weight w_i^t to measure the i-th LSC node v_i of LSCG in frame t. Let N_{il}^t and N_{id}^t be the number of live and dead grid points the LSC respectively, and the definition of weight w_i^t is given by

$$w_i^t = \frac{N_{il}^t}{N_{il}^t + N_{id}^t} \tag{12}$$

For each v_i in $\mathcal{G}$, let f_{i1}^c denote its initial feature vector, f_{it}^c denote its feature vector in frame t and f_i^c represent its feature vector. The feature vector f_i^c is constructed as a combination of the feature vector f_{i1}^c and f_{it}^c according to the weight w_i^t through

$$f_i^c = f_{i1}^c \times (1 - w_i^t) + f_{it}^c \times w_i^t \tag{13}$$

By taking this updating strategy, our LSCG can robustly adapt to the occlusions and deformations.

So far, we have introduced the overall procedure of the proposed tracking method, and we implement it as shown in Algorithm 1.

Algorithm 1. Tracking based on proposed method.

Initialization:
1 Extract the surrounding region of target and segment it into superpixels;
2 Establish a rectangular grid Θ superimposing on the target area;
3 Develop undirected graph **G** and feature vector collection **F** based on Section 3.1;
4 Construct initial LSCG $\mathcal{G}$ for the target according to Section 3.1;
5 Initial LSCG of object $\mathcal{G}_c = \mathcal{G}$;
Tracking:
6 ***for*** $t = 2$ to the end of the sequence ***do***
7 Generate K candidate targets within the framework of particle filter;
8 ***for*** $k = 1$ to K ***do***
9 Construct LSCG $\mathcal{G}_k^t$ for the k-th candidate object;
10 Compute the similarity Γ_k between $\mathcal{G}_c$ and $\mathcal{G}_k^t$ based on Section 3.2;
11 ***end for***
12 $\mathcal{K} = \underset{k}{\operatorname{argmax}}\, \Gamma_k$
13 The $\mathcal{K}$-th candidate target is chosen to be the object;
14 Update LSCG $\mathcal{G}_c$ of target with the tracking result according to Section 4.2;
15 ***end for***
End

5. Experimental results

In order to evaluate the performance of our tracking algorithm, we test our tracker on twelve challenging image sequences. These sequences cover most challenging situations in visual tracking as shown in Table 1. For comparison, we run eight state-of-the-art tracking algorithms with the same initial position of object. These algorithms are ℓ_1 tracking [9], Frag tracking [4], IVT tracking [3], MIL tracking [12], TLD tracking [16], CT tracking [15], OAB tracking [13] and SPT tracking [6] approaches. Some representative results are shown in this section.

5.1. Parameter

The proposed algorithm is implemented in MATLAB on a 3.2 GHz Intel E3-1225 v3 Core PC with 8 GB memory. The parameters of the proposed tracker are fixed in all experiments. The number of particles in Bayesian framework is set to 300–500. The resolution of superpixels is empirically fixed to 300 in our work. The parameters m and n are based on initial rectangle of the target. In the rectangle, the size of grid block is fixed to 10×10. Thus m and n can be gotten by $m = width/10$ and $n = height/10$, where *width* and *height* are the initial size of the object. The parameter λ_s and radius r in Section 3.1 are set to 1.5 and 1 respectively in all the experiments. The bin number B in Eq. (11) is set to 8. The threshold D_{th} in Eq. (4) is set within [0.3, 0.5]. The threshold λ in Eq. (12) is fixed to 0.4.

5.2. Discussions

In order to study the influence of the parameters in our tracking approach, we test two parameters most concerning the performance of the tracking method, which are grid block size and cell radius r. We change the parameters to obtain different tracking results and give detailed analysis based on these results in the following.

5.2.1. Influence of grid block size

We test four kinds of grid block size: 5×5, 10×10 and 15×15. If the grid block size is too small, the number of grid points is too many and they are close to each other. In consequence, the local structural cell constructed is less discriminative. If the grid block size is too large, the number of grid points is too less and they cannot extract enough information. Therefore using middle-size is

Table 1
The tracking sequences used in our experiments.

Sequences	Frames	Main challenges
Basketball	725	Occlusion, Scale changes, Background clutter
Bicycle	271	Occlusion, Scale changes, Scale changes
Bolt	350	Occlusion, Deformation, Background clutter, Scale changes
Cup	303	Background clutter, Scale changes
Deer	71	Fast motion, Motion blur
Face	415	Occlusion
Gymnastics	767	Deformation, Scale changes
Iceskater	445	Deformation, Scale changes
Jogging	307	Deformation, Occlusion, Scale changes
Lemming	1336	Motion blur, Occlusion, Illumination Variations, Scale changes
Liquor	1500	Occlusion, Background clutter
Woman	551	Occlusion, Scale variation, Pose change, Scale changes

Table 2
Influence of grid block size (average center location errors).

	5×5	10×10	15×15
Basketball	12.36	4.93	9.54
Bicycle	7.6	4.40	9.2
Bolt	21.1	6.70	30.5
Cup	5.6	2.60	6.4
Deer	45.6	7.43	13.5
Face	5.61	4.38	6.87
Gymnastics	7.5	8.11	15.4
Iceskater	23.4	18.63	21.54
Jogging	10.5	9.08	24.2
Lemming	21.2	5.44	35.6
Liquor	7.86	5.26	10.54
Woman	9.8	6.13	45.6
Average CLE	15	7.01	19.07

Table 3
Influence of cell radius (average center location errors).

	$r = 0$	$r = 1$	$r = 2$
Basketball	15.4	4.93	11.5
Bicycle	3.8	4.40	17.9
Bolt	16.5	6.70	24.21
Cup	4.61	2.60	10.54
Deer	9.54	7.43	21.14
Face	8.69	4.38	11.6
Gymnastics	19.87	8.11	30.4
Iceskater	25.64	18.63	40.62
Jogging	14.21	9.08	13.6
Lemming	10.56	5.44	14.6
Liquor	12.4	5.26	8.6
Woman	8.9	6.13	28.6
Average CLE	12.51	7.01	19.44

reasonable. Table 2 shows a comparison among results of different sizes of the grid block (from column 2 to column 4).

5.2.2. Influence of cell radius

We test three kinds of cell radius: 0, 1 and 2. If the cell radius r is 0, each LSC contains only one grid node and there will be no edge match similarity in LSCG match. If r is 1, each LSC contains a center grid node and its adjacent nodes. In this case, LSC is stable. If r is 2, each LSC consists of too many grid nodes, which leads to mutable structure of LSC. Thus $r = 1$ is suitable. Table 3 shows a comparison among results of different values of the cell radius (from column 2 to column 4).

Table 5
Average overlapping rate (OR). Red fonts indicate the best performance while the blue fonts indicate the second best.

	ℓ_1	Frag	IVT	MIL	TLD	CT	OAB	SPT	Ours
Basketball	0.03	0.55	0.41	0.21	0.09	0.61	0.16	0.83	0.82
Bicycle	0.31	0.25	0.33	0.43	0.39	0.29	0.24	0.55	0.68
Bolt	0.02	0.20	0.01	0.01	0.14	0.01	–	0.73	0.65
Cup	0.74	0.67	0.71	0.39	0.72	0.53	0.76	–	0.77
Deer	0.13	0.10	0.03	0.03	0.49	0.04	0.59	0.10	0.71
Face	0.85	0.86	0.36	0.55	0.77	0.38	0.79	0.74	0.87
Gymnastics	0.07	0.49	0.39	0.08	0.37	0.05	0.12	–	0.48
Iceskater	0.10	0.55	0.32	0.49	0.46	0.39	0.65	0.60	0.61
Jogging	0.57	0.65	0.13	0.01	0.73	0.13	–	–	0.66
Lemming	0.14	0.13	0.12	0.12	0.30	0.33	0.61	0.65	0.81
Liquor	0.60	0.24	–	–	0.58	0.25	–	0.83	0.85
Woman	0.06	0.19	0.16	0.13	0.29	0.16	–	0.60	0.61
Average OR	0.30	0.41	0.27	0.22	0.44	0.26	0.49	0.63	0.71

5.3. Quantitative comparison

We evaluate the above-mentioned trackers via overlapping rate [27] as well as center location error, and the comparing results are shown in Tables 4 and 5.

Fig. 5 shows the center location error of utilized tracker on twelve test sequences. Overall, the tracker proposed in this paper outperforms the state-of-the-art algorithms.

5.4. Qualitative comparison

5.4.1. Heavy occlusion

The targets in the sequences *Face*, *Bicycle*, *Woman* and *Liquor* undergo severe occlusion as shown in Fig. 6(a). It is difficult for IVT and CT to locate the object because they do not have any mechanism to resist heavy occlusion. Although MIL, ℓ_1, OAB, TLD and Frag are robust under occlusion, they still do not have a good performances in these sequences because they are prone to update background into target and drift away. SPT can basically track the target in these sequences, however, it shrinks to the non-occluded parts of the target since the inner structure information of the object has not been appropriately taken into account. LSCG can keep the inner structure of the target, thereby our tracker robustly locate the target even in heavy occlusion, as depicted in Fig. 6(a).

5.4.2. Deformation

Deformation is a challenge for tracker, because the template features have completely changed when deformation occurs. As shown in Fig. 6(b), MIL, CT, IVT, OAB, TLD and ℓ_1 do not have good

Table 4
Average center location errors (CLE) (in pixels) and average frame per second (FPS). The best result is shown in red and the second best in blue fonts.

	ℓ_1	Frag	IVT	MIL	TLD	CT	OAB	SPT	Ours
Basketball	131.25	16.24	68.42	94.83	129.37	19.02	86.40	4.82	4.93
Bicycle	49.03	55.77	7.82	10.50	10.63	51.50	60.90	5.44	4.40
Bolt	361.79	100.60	374.95	365.40	87.85	348.00	–	6.85	6.70
Cup	2.92	7.07	1.81	40.60	3.12	25.13	4.45	–	2.60
Deer	91.75	93.48	222.55	214.00	47.82	235.97	27.60	97.04	7.43
Face	5.44	4.86	62.55	36.22	6.91	57.07	10.50	18.23	4.38
Gymnastics	72.84	9.44	15.24	115.88	12.46	131.90	87.52	–	8.11
Iceskater	137.74	23.90	56.94	36.66	31.86	64.45	13.90	19.01	18.63
Jogging	14.59	9.30	130.00	146.18	7.20	124.76	–	–	9.08
Lemming	179.59	143.75	182.91	135.38	17.04	82.47	16.22	7.38	5.44
Liquor	35.23	103.77	–	–	33.92	154.73	–	8.28	5.26
Woman	134.15	106.21	138.64	116.35	72.63	108.88	–	12.22	6.13
Average CLE	101.36	56.20	114.71	119.27	38.40	116.99	34.17	19.92	7.01
Average FPS	0.5	4	32	32	18	90	22	0.4	0.9

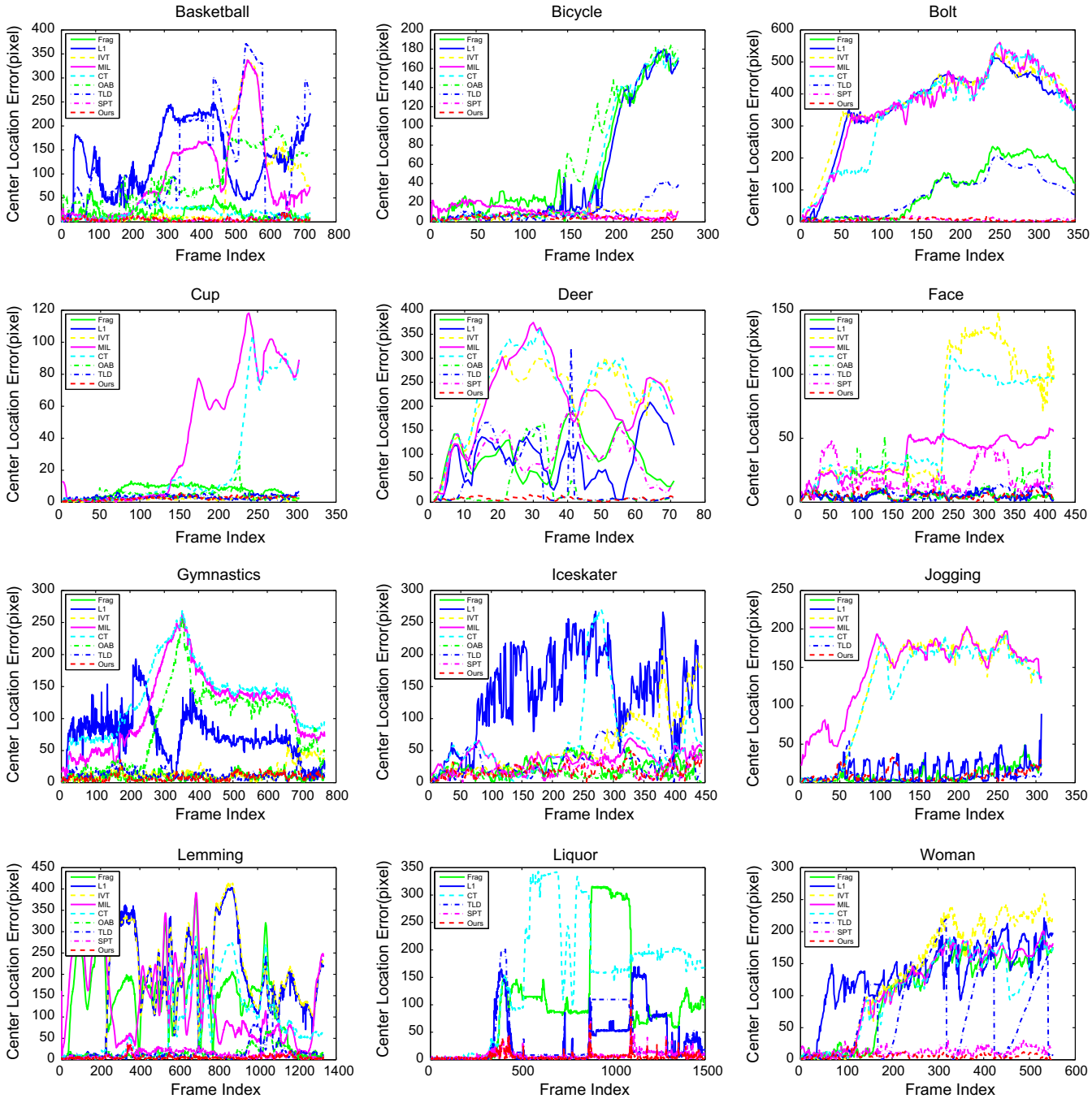
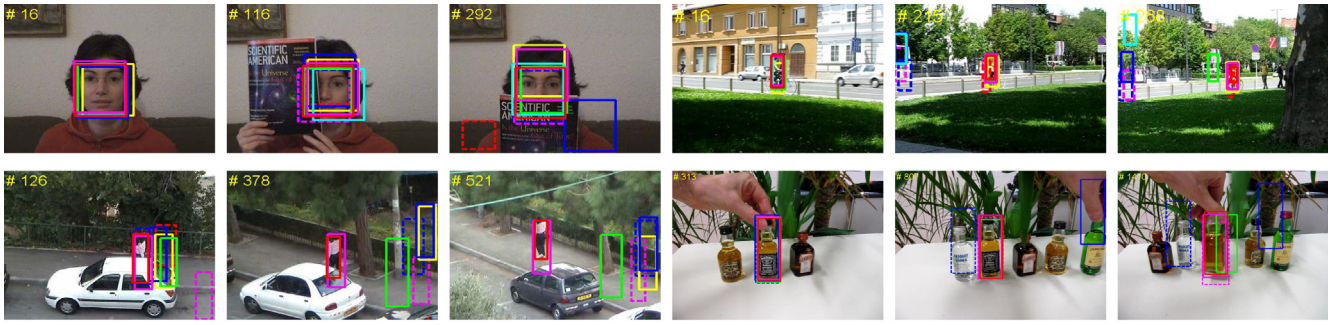


Fig. 5. Quantitative evaluation in terms of center location error (in pixel). The proposed method is compared with eight state-of-the-art algorithms on twelve challenging test sequences.

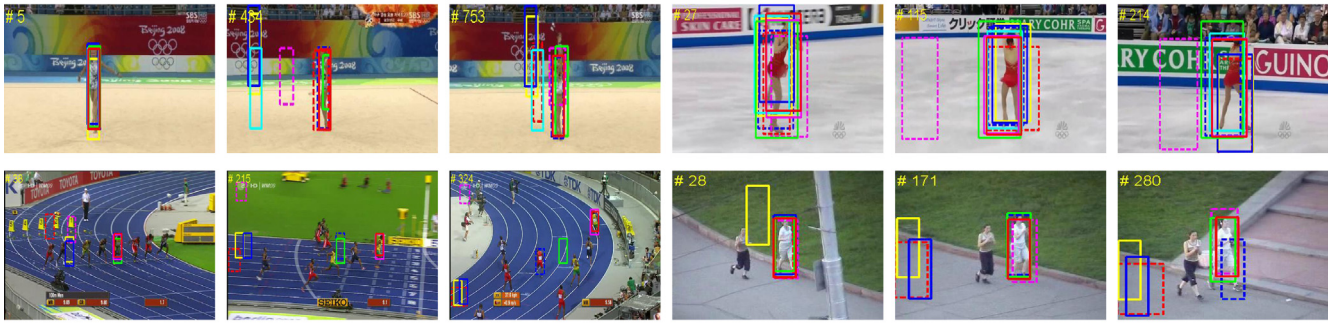
performances in the sequences *Gymnastics*, *iceskater*, *Bolt* and *Jogging*. Differently, Frag and SPT have relatively better tracking results in these sequences, because part-based trackers are less sensitive to structure variation than holistic appearance. Whereas, the lack of effective updating strategy still easily cause drifting away even failure. Our tracker have obvious advantage in handling structure deformation even in high frequency, since the inner structure of the target is exploited carefully and with the help of effective updating mechanism, our tracking method robustly adapts to the structure deformation.

5.4.3. Fast motion and motion blur

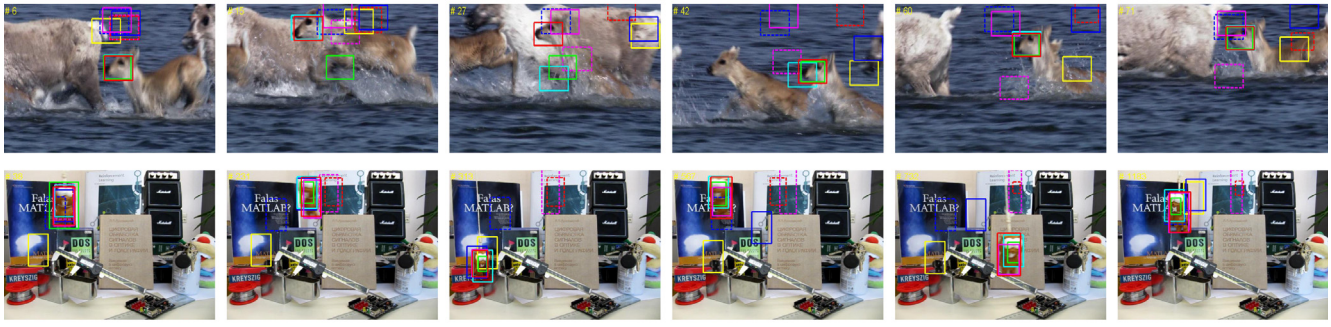
Fig. 6(c) demonstrates experimental results on two challenging sequences (*Deer* and *Lemming*). Because the target undergoes fast and abrupt motion, it is more prone to cause blur, which causes drifting problem. It is worth noticing that the suggested approach in this paper performs better than other algorithms. When motion blur happens, our LSCG can still effectively represent the target appearance at the level of superpixel. Besides, our updating mechanism can resist motion blur to some degree. Hence our tracker will not be undermined by the abrupt movement.



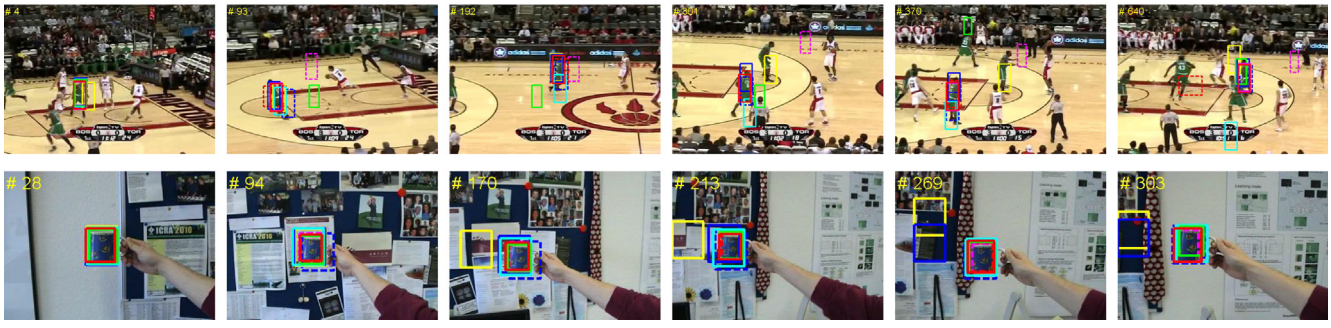
(a) *Face, Bicycle, Woman* and *Liquor* with heavy occlusion



(b) *Gymnastics, Iceskater, Bolt* and *Jogging* with deformation



(c) *Deer* and *Lemming* with fast motion and motion blur



(d) *Basketball* and *Cup* with background clutter

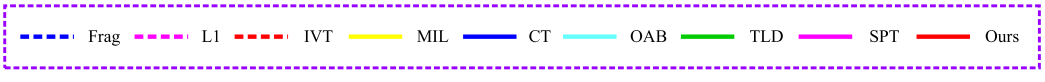


Fig. 6. Tracking results on various challenging sequences.

Table 6
Full comparison between SPT and the proposed tracker with average center location errors (CLE) and frame per second (FPS).

	SPT in [6]	Ours
Lemming	7.38	5.44
Liquor	8.28	5.26
Singer1	5.64	10.15
Baskerball	4.82	4.93
Woman	12.22	6.13
Transformer	14.23	–
Bolt	6.85	6.7
Bird1	47.24	52.4
Bird2	17.54	27.1
Girl	10.34	8.11
Surfing1	48.12	–
Racecar	4.24	6.8
Average CLE	15.34	13.3
Average FPS	0.4	0.9

5.4.4. Background clutter

The sequences *Cup* and *Basketball* in Fig. 6(d) are challenging due to the background clutter and the target undergoes the scale variation. Our tracker performs well in this sequence as the target can be differentiated from the cluttered background with the use of our LSCG model. In addition, the updating scheme is also robust to the complex background.

5.5. Full comparison with robust superpixel tracking

In [6], Yang et al. propose a superpixel based tracking method (SPT), which is similar to ours. Different from [6], however, the proposed tracker takes advantage of local information of the object and its spatial structure relationship to track the whole object. Therefore the proposed tracker can still locate target in the presence of local occlusions or deformations. In order to fully compare our method with the robust superpixel tracking in [6], we run our tracker with the challenging sequences presented in [6] and compare the tracking results between these two methods. The comparison results are shown in Table 6.

5.6. Limitations

The proposed tracking method usually performs well in handling challenging cases, however there exist some scenarios where our tracker may fail to locate the object, as shown in Fig. 7. In Fig. 7(a), our tracker can locate the object when it suffers from partial occlusion (frame 23), nevertheless it is not able to deal with long time full occlusion situation (frame 30 and 44). Although the target reappears later (frame 56), the proposed tracker cannot locate it for lack of a re-initialization mechanism. In Fig. 7(b), the object undergoes serious deformation and scale change. In the beginning, the appearance and scale of the object varies slightly, our tracker can locate it accurately (frame 1, 11 and 31). However, as the scale of the target thoroughly changes, which leads to serious deformation (frame 68 and 104), our tracker fails to keep the track of the object.

6. Conclusion

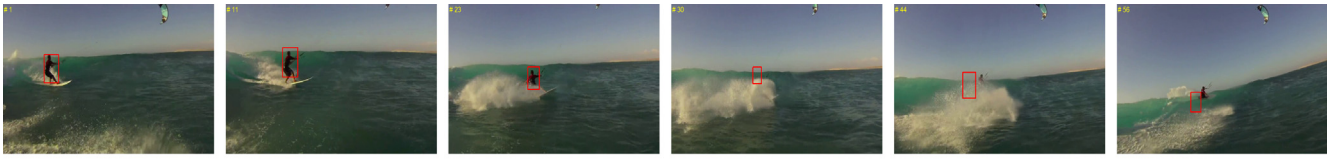
In this paper, we propose and demonstrate an effective and robust tracking method based local structural cell graph at the level of superpixel. In our tracker, structure information is taken into consideration to construct LSCG appearance model. The target tracking is interpreted as matching the candidate particle to the target. Moreover, the online updating mechanism is presented to robustly adapt to the occlusions and deformations. Quantitative and qualitative comparisons with eight state-of-the-art methods on twelve challenging image sequences demonstrate the robustness of our tracker.

Acknowledgment

This work was supported by the Fundamental Research Funds for the Central Universities (Program No. 2014BQ083).

References

[1] Y. Wu, J. Lim, M.H. Yang, Online object tracking: a benchmark, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2411–2418.
[2] X. Li, W. Hu, C. Shen, et al., A survey of appearance models in visual object tracking, ACM Trans. Intell. Syst. Technol. 4 (4) (2013) 2411–2418.
[3] D.A. Ross, J. Lim, R.S. Lin, et al., Incremental learning for robust visual tracking, Int. J. Comput. Vision 77 (1-3) (2008) 125–141.
[4] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2006, pp. 798–805.
[5] H. Wang, D. Suter, K. Schindler, et al., Adaptive object tracking based on an effective appearance filter, IEEE Trans. Pattern Anal. Mach. Intell. 29 (9) (2007) 1661–1667.
[6] F. Yang, H.C. Lu, M.H. Yang, Robust superpixel tracking, IEEE Trans. Image Process. 23 (4) (2014) 1639–1651.
[7] H. Grabner, H. Bischof, On-line boosting and vision, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2006, pp. 260–267.
[8] J. Fan, Y. Wu, S. Dai, Discriminative spatial attention for robust tracking, in: European Conference on Computer Vision (ECCV), 2010, pp. 480–493.
[9] X. Mei, H. Ling, Robust visual tracking and vehicle classification via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (11) (2011) 2259–2272.
[10] Z.Y. Xiao, H.C. Lu, D. Wang, L2-RLS based object tracking, IEEE Trans. Circ. Syst. Video Technol. 24 (8) (2014) 1301–1308.



(a) Failure case 1 (video *Surfing1*)

(b) Failure case 2 (video *Transformer*)

Fig. 7. Two failure cases.

[11] X. Jia, H. Lu, M.H. Yang, Visual tracking via adaptive structural local sparse appearance model, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 1822–1829.
[12] 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.
[13] H. Grabner, M. Grabner, H. Bischof, Real-time tracking via on-line boosting, in: British Machine Vision Conference (BMVC), 2006, pp. 47–56.
[14] F. Yang, H.C. Lu, M.H. Yang, Learning structured visual dictionary for object tracking, Image Vis. Comput. 31 (12) (2013) 992–999.
[15] K.H. Zhang, L. Zhang, M.H. Yang, Fast compressive tracking, IEEE Trans. Pattern Anal. Mach. Intell. 36 (10) (2014) 2002–2015.
[16] Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-learning-detection, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1409–1422.
[17] J. Kwon, K.M. Lee, Visual tracking decomposition, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 1269–1276.
[18] J. Kwon, K.M. Lee, Highly non-rigid object tracking via patch-based dynamic appearance modeling, IEEE Trans. Pattern Anal. Mach. Intell. 35 (10) (2013) 2427–2441.
[19] X. Ren, J. Malik, Tracking as repeated figure/ground segmentation, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
[20] J. Choi M, A. Torralba, A.S. Willsky, A tree-based context model for object recognition, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2) (2012) 240–252.
[21] A. Toshev, B. Taskar, K. Daniilidis, Object detection via boundary structure segmentation, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 950–957.
[22] L. Zhu, Y. Chen, A. Yuille, et al., Latent hierarchical structural learning for object detection, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 1062–1069.
[23] S. Avidan, Ensemble tracking, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 261–271.
[24] H. Grabner, C. Leistner, H. Bischof, Semi-supervised on-line boosting for robust tracking, in: European Conference on Computer Vision (ECCV), 2008, pp. 234–247.
[25] Z. Kalal, J. Matas, K. Mikolajczyk, P-N learning: bootstrapping binary classifiers by structural constraints, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 49–56.
[26] K.H. Zhang, H. Song, Real-time visual tracking via online weighted multiple instance learning, Pattern Recogn. 46 (1) (2013) 397–411.
[27] M. Everingham, L.V. Gool, C. Williams, et al., Partbased visual tracking with online latent structural learning, in: PASCAL Visual Object Classes Challenge, 2010.
[28] R. Achanta, A. Shaji, K. Smith, et al., SLIC superpixels compared to state-of-the-art superpixel methods, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2274–2282.