Multimed Tools Appl (2017) 76:15839–15866
DOI 10.1007/s11042-016-3872-6

Robust visual tracking based on generative and discriminative model collaboration

Jianfang Dou[1] · Qin Qin[1] · Zimei Tu[1]

Received: 16 February 2016 / Revised: 11 July 2016 / Accepted: 15 August 2016 /
Published online: 3 September 2016
© Springer Science+Business Media New York 2016

Abstract Effective object appearance model is one of the key issues for the success of visual tracking. Since the appearance of a target and the environment changes dynamically, the majority of existed visual tracking algorithms tend to drift away from targets. To address this issue, we propose a robust tracking algorithm by integrating the generative and discriminative model. The object appearance model is made up of generative target model and a discriminative classifier. For the generative target model, we adopt the weighted structural local sparse appearance model combining patch based gray value and Histogram of Oriented Gradients feature as the patch dictionary. By sampling positives and negatives, alignment-pooling features are obtained based on the patch dictionary through local sparse coding, then we use support vector machine to train the discriminative classifier. The proposed method is embedded into a Bayesian inference framework for visual tracking. A combined matching method is adopted to improve the proposal distribution of the particle filter. Moreover, in order to adapt the situation change, the patch dictionary and discriminative classifier are updated by incremental learning every five frames. Experimental results on some publicly available benchmarks of video sequences demonstrate the accuracy and effectiveness of our tracker.

Keywords Visual tracking · Histogram of oriented gradients · Structural local sparse appearance model · Delaunay triangulation

1 Introduction

Visual tracking is an important and active research topic in computer vision community because of its wide range of applications, e.g., intelligent video surveillance, human computer

✉ Jianfang Dou
specialdays_2010@163.com

[1] Department of Automation and Mechanical and Electrical engineering, School of Intelligent Manufacturing and Control Engineering, Shanghai Second Polytechnic University, Shanghai 201209, China

interaction and robotics [43, 44]. The purpose of visual tracking is to estimate the state of the tracked target in a video. It is usually formulated as a search task where an appearance model is first used to represent the target and then a search strategy is utilized to infer the state of the target in the current frame. Although it has been extensively studied in the last two decades, it still remains to be a challenging problem due to many appearance variations caused by occlusion, pose, illumination, background clutter, and so on.

Broadly speaking, a tracking algorithm mainly includes two fundamental components: (1) a motion model (or called dynamic model), which relates the states of an object over time and predicts its likely state by supplying the tracker with a number of candidate states (e.g., Kalman filter [26], particle filter [25]); (2) an observation model (or called appearance model) [30], which represents the tracked object and verifies predictions by evaluating the likelihood of each candidate state in the current frame.

According to the different observation models used in existing object tracking algorithms, they can be categorized into methods based on template [1, 14], online classifiers [4, 28] and so on. In the template-based algorithms, the tracked object is described by one single template [14] or multiple templates. Then the tracking problem can be considered as searching for the regions which are the most similar to the tracked object. The trackers based on online classifiers aim to distinguish the tracked objects from its surrounding backgrounds by treating the tracking problem as a binary classification problem. Thus, both classic and recent machine learning algorithms could promote the progress of tracking algorithms or systems, including boosting [20, 21], support vector machine [23, 39], naive bayes [45], random forest [38], multiple instance learning [4], structured learning [22] and so on. Jia et al. [27] exploited both partial information and spatial information of the target based on a novel alignment-pooling method and proposed an efficient tracking algorithm based on structural local sparse appearance model and adaptive template update strategy. This algorithm made good use of the appearance and spatial structure information of the target and reduced the influence of the occluded target template. But when the target underwent heavy occlusions (such as for long term tracking videos when the target is disappeared or totally occluded), appearance changes, or interference of similar object and background, the robustness needed further improvement. So effective modeling of the object's appearance is one of the key issues for the success of a visual tracker.

In this paper, we propose a robust tracking algorithm by integrating the generative and discriminative model. The object appearance model is made up of generative target model and a discriminative classifier. For the generative target model, we adopt the weighted structural local sparse appearance model [27] combining patch based gray value and Histogram of Oriented Gradients feature as the patch dictionary. By sampling positives and negatives, alignment-pooling features are obtained based on the patch dictionary through local sparse coding, then we use a support vector machine to train the discriminative classifier. A robust inter-frame matching based on optical flow [24] and Delaunay triangulation [17, 18] accompanied with template matching is adopted to improve the proposal distribution of particle filter to enhance the performance of tracking. The proposed method is embedded into a Bayesian inference framework for visual tracking. Through alignment-pooling method across the local patches within one candidate region to obtain the similarity measure, at the same time using the trained discriminative classifier to get the classification score. The similarity measure and classification score are multiplied to obtain the particle confidences. Moreover, in order to adapt the situation change, the patch dictionary and discriminative classifier are updated by incremental learning every five frames. Experimental results on some publicly available benchmarks of video sequences demonstrate the accuracy and effectiveness of our tracker.

The main contributions of this paper are as follows:

1) Integrating inter-frame matching into the framework of visual tracking, the two parts can complement to another, thus improving the performance of tracking;
2) A Robust inter-frame matching based on Optical Flow and Delaunay Triangulation is adopted to enhance the robustness and accuracy of matching;
3) Considering the spatial configuration of each local patch of the target, weighting the structural local sparse appearance model;
4) Enhancing the object appearance model by integrating generative model and a discriminative classifier.

The remainder of this paper is organized as follows: in Section 2 we summarize the previous works most related to our work. The proposed robust tracking method ASLA_DW (adaptive structural local sparse appearance model with discriminative weighted) is described in Section 3, respectively. Experiments and results are provided and analyzed in Section 4. Finally, our work is summarized and conclusions are drawn in Section 5.

2 Related work

Many promising approaches have been proposed to tackle object tracking. These methods can be roughly classified into two categories: generative methods and discriminative methods.

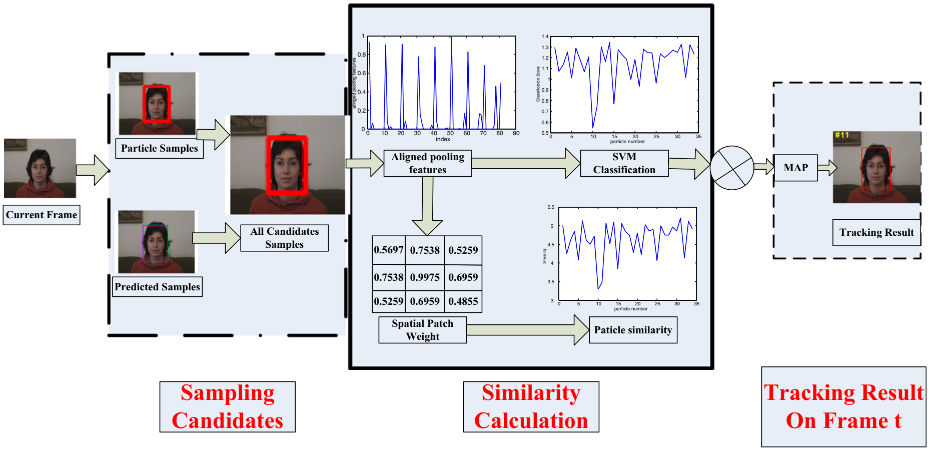
Generative methods represent objects with appearance models, and track targets by searching for the image region most similar to the models. Reference templates can be learned with a set of training data. Black et al. [8] learned a subspace model offline to represent targets and used parametric optical flow estimation simultaneously. Aside from static appearance models, online appearance models which are updated as the appearance of the target changes have also been presented. Wang et al. [40] constructed a dynamic multi-cue integration model for particle filter framework. Ross et al. [37] proposed a tracking framework based on the incremental image-as-vector subspace learning method with a sample mean update. Li et al. [29] modeled target appearance changes by incremental image-as-matrix subspace learning method through adaptively updating the sample mean and eigenbasis. Recently, sparse representation [12] has been successfully applied to visual tracking [32, 33]. In this case, the tracker represents each target candidate as a sparse linear combination of dictionary templates that can be dynamically updated to maintain an up-to-date target appearance model. This representation has been shown to be robust against partial occlusions, which leads to improved tracking performance. However, sparse coding based trackers perform computationally expensive l1 minimization at each frame.

Unlike generative methods, discriminative methods regard objects tracking as a binary classification problem to distinguish the target from background. These methods exploit the information of both the target and background. Avidan [3] combined optical flow representation with a support vector machine (SVM) classifier for objects tracking. Ozuysal et al. [36] proposed a random forest classifier which learned binary features of the target. These methods use a predefined and fixed feature sets for classifier learning. In addition, there are many other approaches that can select good object features online. Collins et al. [13] presented a two-class variance ratio to select best discriminate features online. Zhong et al. [46] proposed a weakly supervised online training data selection method for visual tracking. However, these methods may introduce an error from self-updating which can cause tracking drifting. Aiming at this problem, Babenko et al. [5] proposed an online multiple instance learning (MIL) method for object tracking. In [47], Zhou et al. improved the MIL by selecting the support instances adaptively and update the support instances by taking image data obtained previously and recently into account. However, in order to achieve accurate and robust tracking, most of the discriminative methods do not update their classifiers in which models are far away from the initial setting, as a result, drift occurs when the target appearance changes heavily.

In recent years, deep convolutional neural networks have improved state-of-the-art performance in many computer vision applications. Existing methods have also explored the usage of CNNs in online tracking. In [41], a three-layer CNN is trained on-line. Without pre-training and with limited training samples obtained online, CNN fails to capture object semantics and is not robust to deformation. In order to improve the proposal distribution of particle filter, we estimate the translation of the target by optical flow tracker and Delaunay triangulation through matching detected corners. This can be similar as the problem of image search. Nie L, Wang M et al. [34] proposed a content-based approach to automatically predict the search performance of image search. By fully exploring the information from simple visual concepts, Nie L, Yan S et al. [35] presented a scheme to enhance web image reranking for complex queries.

3 Proposed method

The flowchart of our proposed method is shown in Fig.1. It consists of three components: 1) sampling candidate; 2) similarity calculation; 3) MAP estimation. Given the current frame in time*t*, based on the patch dictionary and support vector machine classifier constructed in the first frame, we obtain the sample candidates combining particle sampling and predicted samples with optical flow tracker and surf matching and solve the local sparse coding by sparse representation to get the aligned pooling features. Then we use the discriminative classifier to calculate the classification score, at the same time spatially weight the aligned pooling features to get the particle similarity. Thirdly, we obtain the final confidence by multiplying classification score and particle similarity. The particle with the max confidence is chosen as the tracking result (shown as a red color rectangle).



0.5697	0.7538	0.5259
0.7538	0.9975	0.6959
0.5259	0.6959	0.4855

Fig. 1 Flowchart of the proposed method

The novelty of our proposed method is listed as below:

1) A robust appearance model made up of generative target model and a discriminative classifier.
2) Integrating the inter-frame matching into the framework of visual tracking.
3) 3) Improving the proposal distribution of particle filter based on image matching.

3.1 Object appearance model

Our observation appearance model is shown in Fig.2. It based on generative and discriminative model. The generative model is made up of a patch dictionary and discriminative model with a svm (support vector machine) classifier. Given the initial target region in the first frame, we divide the target region into N patches. For each patch, the gray vector and HOG features are extracted and combined to form the patch dictionary. Through sampling positives and negatives, extracting features vectors (combined gray vector and Hog features), solving local sparse coding with the patch dictionary to get the aligned positive and negative pooling features, we use support vector machine to train the input training data to obtain the discriminative classifier. In the following subsections, we will describe each part of our appearance model in detail.

3.1.1 Histogram of oriented gradients (HOG)

The HOG representation is inspired by the SIFT descriptor proposed by Lowe [31]. It can be constructed by dividing the tracking regions into non-overlapping grids, and then computing the orientation histograms of the image gradient of each grid (Fig. 3).

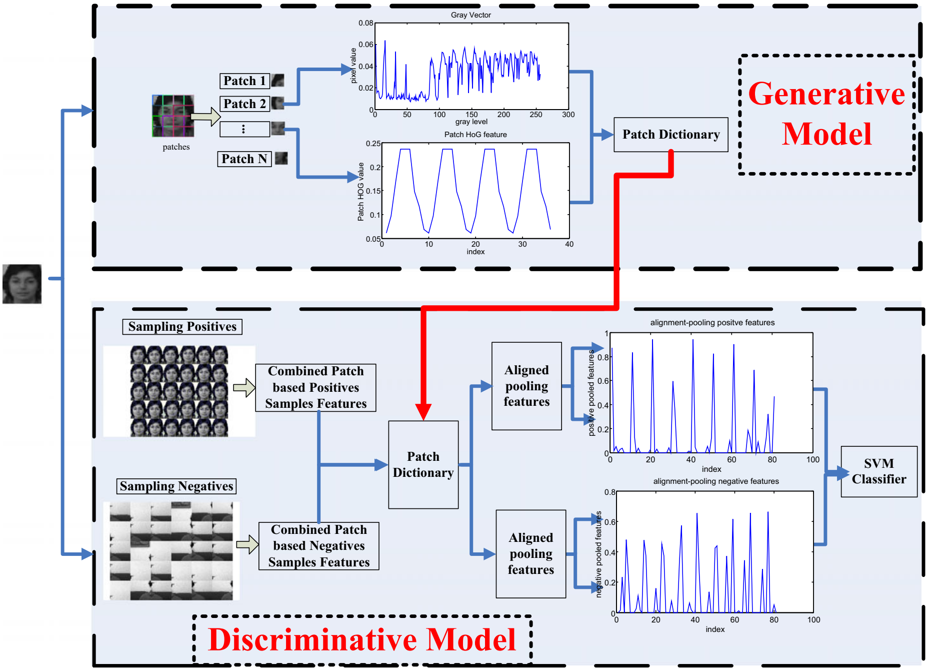
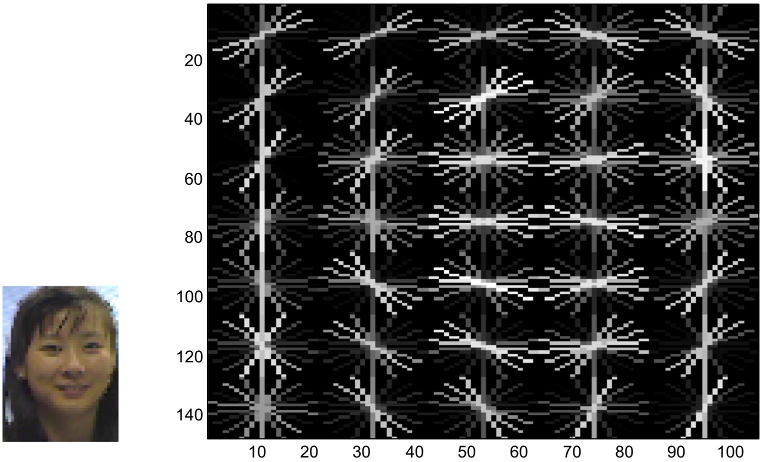


Fig. 2 Our Observation Appearance Model



(a) roi (b) image of hog feature

Fig. 3 Region of Interest and corresponding hog feature

Let$\theta(x, y)$and$m(x, y)$be the orientation and magnitude of the intensity gradient at image pixel(x, y). The image gradients can be computed via a finite difference mask [−1 0 1] and its transpose. The gradient orientation at each pixel is discretized into one of$p$ values by a contrast insensitive definition as follows:

$$B(x, y) = \text{round}\left(\frac{p.\theta(x, y)}{\pi}\right)\text{mod}p \tag{1}$$

Let$b \in \{0, \dots, p-1\}$ranges over orientation bins. The feature vector at$(x, y)$is:

$$F(x, y)_b = \begin{cases} m(x, y) & \text{if } b = B(x, y) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Let$F$be a pixel-level feature map for an$w \times h$image and $k > 0$be a parameter indicates the side length of a square image region. A dense grid of rectangular "cells" [16] is defined and pixel-level features are aggregated to obtain a cell-based feature map$C$, with feature vectors$C(i, j)$for $0 \le i \le [(w-1)/k]$and$0 \le j \le [(h-1)/k]$. This aggregation can reduce the size of a feature map. After a bilinear interpolation to aggregate features, each feature can be normalized. The resulting feature vector is the HOG descriptor of the image region. Normally the parameters of HOG descriptor are set to be $p = 9$and$k = 8$, the size of the "cell" is2×2. This leads to a 36-dimensional feature vector.

3.1.2 Structural local sparse representation

The structural local sparse appearance model has been proposed in [27]. In this section, we review it briefly and draw out the spatial weight of candidate target used in our method. In [27], Jia et al. sampled a set of overlapping local image patches inside the target region with a spatial layout and used these local patches as the dictionary to encode the local image patches inside the candidate regions, ie. $\mathrm{M} = [\mathrm{m}_1, \mathrm{m}_2, \dots, \mathrm{m}_N, \mathrm{m}_{N+1}, \dots, \mathrm{m}_{N \times n}] \in R^{d \times (N \times n)}$, where $d$is the dimension of the local image patch vector, n is the number of target templates$\mathrm{T} = [\mathrm{T}_1,$

T_2, ... , T_n], $N$is the number of local patches sampled within one target region. Each column inMis obtained by $l_2$normalization on the vectorized local image patches. For a target candidate, the local image patches within it can be denoted by $Y = [y_1, y_2, \ldots, y_N] \in R^{d \times N}$. For the purpose of saving computing time, we adopt 8-bit gray scale image for analysis.

During tracking, the local patches within the target candidate region can be sparsely represented as a linear combination of the local patches dictionary by solving

$$\min_{b_i} \left\| y_i - Mb_i \right\|_2^2 + \lambda \left\| b_i \right\|_1, \quad \text{s.t.} b_i \geq 0 \tag{3}$$

where $\| \bullet \|_2$ and $\| \bullet \|_1$denote the$l_2$and $l_1$normalization respectively, $\lambda$is the regularization parameter,$b_i \in R^{(N \times n) \times 1}$ is the sparse code vector of the i-th local image patch, and $b_i \geq 0$ means all the elements of $b_i$are nonnegative. Note that $B = [b_1, b_2, \ldots, b_N]$represents the sparse coefficients of the local patches within one target candidate region. According to the target templates, the sparse coefficients of each local patch are divided into n(the number of target templates) segments, ie.$b_i^T = \left[b_i^{(1)T}, b_i^{(2)T}, \ldots, b_i^{(n)T} \right]$, where

$$b_i^{(k)T} = \left[b_{(k-1) \times N+1}, b_{(k-1) \times N+2}, \ldots, b_{(k-1) \times N+N} \right] \in R^{1 \times N} \tag{4}$$

denotes the k-th segment ofb_i. These coefficients are weighted to obtain$v_i$for the i-th patch,

$$v_i = \frac{1}{C} \sum_{k=1}^{n} b_i^{(k)} = \frac{1}{C} \begin{bmatrix} v_{i1} \\ v_{i2} \\ \cdots \\ v_{\mathrm{iN}} \end{bmatrix}, i = 1, 2, \ldots, N \tag{5}$$

where$v_i \in R^{N \times 1}$ is the sparse coefficients of the i-th local patch and $C$is a normalization term,$C = v_{i1} + v_{i2} + \ldots + v_{iN}$. Since one candidate target contains Nlocal image patches, all the vectors $v_i$can form a square matrix$V$,$V = [v_1, v_2, \ldots, v_N]$. According to the spatial layout of the target, the local patch can be best described by the block at the same positions of the template (i.e., using the sparse coefficients with the aligned positions). Therefore, we take the diagonal elements of the square matrix V as the final sparse coefficients of the local patches within the candidate region, ie.

$$f \in \mathrm{diag}(V) = \begin{bmatrix} f_1 \\ f_2 \\ \cdots \\ f_N \end{bmatrix} \tag{6}$$

where f is the sparse coefficients vector of all the local patches, i.e.,f_1 means the sparse code of the first patch and f_2 means the sparse code of the second patch.

3.1.3 Sampling positives and negatives

To initialize the classifier in the first frame, we draw positive and negative samples around the labeled target location. Suppose the location of the target object in the first frame is denoted by $l_1(x_1, y_1)$, we use a Gaussian perturbation to draw positive samples in a circular area which satisfies $\|l_{pos} - l_1\| < \gamma$, and draw negative samples in an annular area specified by$\gamma < \|l_{neg} - l_1\| < \eta$, where$\gamma$and are η thresholds defining the circle and annular areas, respectively. The sets,

l_{pos} and l_{neg}, denote the locations of positive and negative candidates, respectively. Without loss of generality, we set the scales of the positive and negative candidates the same as our labeled target object. We then crop the images specified by the set of samples l_{pos} and l_{neg} and compute the sparse code of each image patch to form the training data.

3.1.4 One-class support vector machine

Support vector machines (SVMs) are a family of classification algorithms, developed under the statistical learning theory, originally formulated for binary classification [11, 15]. SVMs offer a solution to optimizing the generalization performance of a decision function, inferred from a given set of training data. Given training data and its corresponding labels:

$$(\mathrm{x}_n, y_n), n = 1, 2, \ldots, N, \mathrm{x}_n \in R^D, y_n \in \{-1, +1\} \tag{7}$$

SVMs learning consists of the following constrained optimization:

$$\begin{aligned} &\min_{\mathrm{w},\xi_n} = \frac{1}{2}\mathrm{w}^{\mathrm{T}}\mathrm{w} + C\sum_{n=1}^{N} \xi_n \\ &s.t.\ \ \mathrm{w}^{\mathrm{T}}\mathrm{x}_n y_n \geq 1 - \xi_n \forall n \\ &\qquad \xi_n \geq 0 \forall n \end{aligned} \tag{8}$$

$\xi_n(n = 1, 2, \ldots, N)$are slack variables which penalize data points which violate the margin requirements. Note that we can include the bias by augment all data vectors x_n with a scalar value of 1. The corresponding unconstrained optimization problem is the following:

$$\min_{\mathrm{w},\xi_n} = \frac{1}{2}\mathrm{w}^{\mathrm{T}}\mathrm{w} + C\sum_{n=1}^{N} \max\left(1 - \mathrm{w}^{\mathrm{T}}\mathrm{x}_n y_n, 0\right) \tag{9}$$

The objective of Eq.(8) is known as the primal form problem of L1-SVM, with the standard hinge loss. This optimization problem can be formed by a Lagrange multiplier, and solved by applying quadratic programming to its dual form.

3.2 Predicted sample candidates

In order to improve the proposal distribution of the particle filter and at the same time when the tracked target is totally lost or disappeared and reappeared after a while, we predict the target location respectively by optical flow tracker, template matching and SURF Keypoints Matching to obtain the extra candidates samples. Based on this operation, we can recapture the target and track it robustly. This operation is shown in Fig. 4. In the following subsections, we describe each part in detail.

3.2.1 Translation obtained by optical flow tracker and Delaunay triangulation

1) Optical Flow Tracker

Optical flow or optic flow [10] is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera)

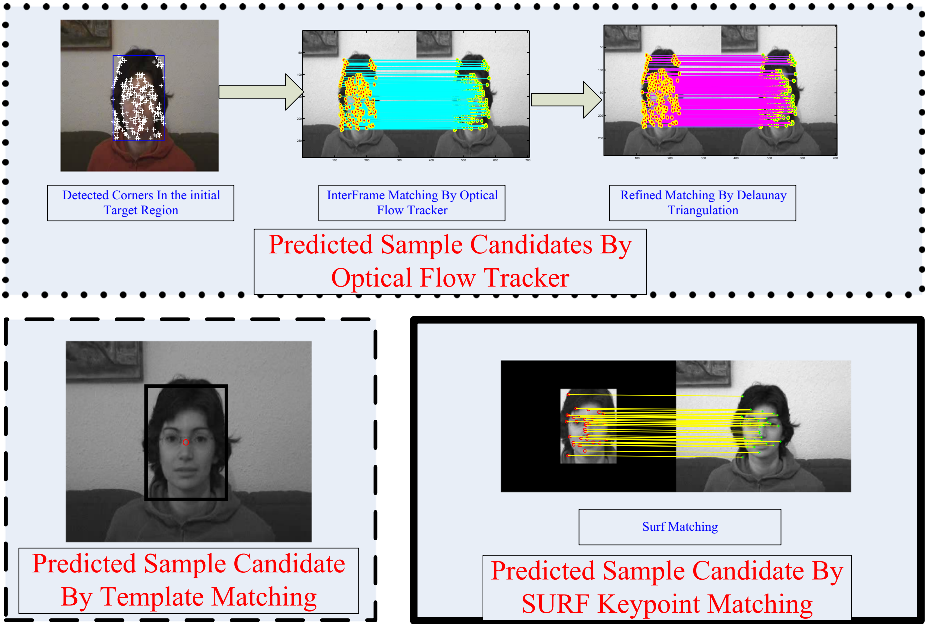

Detected Corners In the initial Target Region
InterFrame Matching By Optical Flow Tracker
Refined Matching By Delaunay Triangulation
Predicted Sample Candidates By Optical Flow Tracker
Predicted Sample Candidate By Template Matching
Surf Matching
Predicted Sample Candidate By SURF Keypoint Matching

Fig. 4 Predicted Sample Candidates Obtained respectively Optical Flow Tracker, Template Matching and SURF Keypoint Matching

and the scene. It is the displacement field for each of the pixels in an image sequence. It is the distribution of the apparent velocities of objects in an image. By estimating optical flow between video frames, one can measure the velocities of objects in the video. In general, moving objects that are closer to the camera will display more apparent motion than distant objects that are moving at the same speed. Optical flow estimation is used in computer vision to characterize and quantify the motion of objects in a video stream, often for motion-based object detection and tracking systems.

The experimental brightness of any object point is constant over time. Close to points in the image plane move in a similar manner (the velocity smoothness constraint). Suppose we have a continuous image, $f(x, y, t)$ refers to the gray-level of (x, y) at time t. Representing a dynamic image as a function of position and time permits it to be expressed.

- Assume each pixel moves but does not change intensity.
- Pixel at location(x, y)in frame $t-1$ is pixel at $(x+\Delta x, y+\Delta y)$in frame t.
- Optic flow associates displacement vector with each pixel.

The optical flow describes the direction and time pixels in a time sequence of two consequent dimensional velocity vectors, carrying direction and the velocity of motion is assigned to each pixel in a given place of the picture. For making computation simpler and quicker we transfer the real world three dimensional (3-D + time) objects to a (2-D + time) case. Then we can describe the image by of the 2-D dynamic brightness function of $I(x, y, t)$. Provided that in the neighborhood of pixel, change of brightness intensity does not generate motion field, we can use the following expression

$$I(x, y, t) = I(x+\delta x, y+\delta y, t+\delta t) \tag{10}$$

Using Taylor series for the right hand part of Eq. (10), we obtain

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + H.O.T \tag{11}$$

From Eq. (10, 11), with neglecting higher order terms (H.O.T.) and after modifications we get

$$I_x {v^x}_x {+^x} I_y {v^y}_y {+^y} I_t {v^t}_t {=^t} 0 \tag{12}$$

or in formal vector representation

$$\nabla I \cdot \overrightarrow{v} = -I_t \tag{13}$$

where ∇I is so-called the spatial gradient of brightness intensity and $\overrightarrow{v}$ is the optical flow (velocity vector) of the image pixel and I_t is the time derivative of the brightness intensity [6]. Thus optical flow can give significant information about the spatial arrangement of the objects viewed and the rate of change of this arrangement.

2) Fine Matching Based On Delaunay Triangulation [17, 18]

There may be some wrong matches in the initial surf matches. In order to filter out the wrong matches, we adopt the Delaunay triangulation to refine the matches. The Delaunay triangulation has many remarkable properties that make it the most widely-used triangulation.

Let $P = \{p_1, \cdots, p_n\}$ be a set of points in R^d. The Voronoi cell associated to a point p_i, denoted by $V(p_i)$, is the region of space that is closer from p_i than from other points in$P$ [9]:

$$V(p_i) = \left\{ p \in R^d : \forall j \neq i, \|p - p_i\| \leq \|p - p_j\| \right\} \tag{14}$$

$V(p_i)$is the intersection of $n-1$half-spaces bounded by the bisector planes of segments $[p_i p_j]$,$j \neq i.V(p_i)$is therefore a convex polytype, possibly unbounded. The Voronoi diagram of P, denoted by $Vor(P)$, is the partition of space induced by the Voronoi cells $V(p_i)$.

Triangulation is a process that takes a region of space and divides it into sub-regions. The space may be of any dimension, however, a 2D space is considered here since we are dealing with 2D points. In this case, the sub-regions are simply triangles. Euler formula of Triangulation is:

$$f - e + v = 1 \tag{15}$$

where f is the number of facet; e is the number of edges, v is the number of vertex. The complexity of n points P constructed triangulation has N_{tri} triangles and N_{edge} edges. In this case, $e = N_{edge}$.

$$N_{tri} = 2n - 2 - k \tag{16}$$

$$N_{\mathrm{edge}} = 3n - 3 - k \tag{17}$$

where k is the number of points P in on the convex hull of P.

The Delaunay triangulation $Del(P)$ of P is defined as the geometric dual of the Voronoi diagram: there is an edge between two points p_i and p_j in the Delaunay triangulation if and only if their Voronoi cells $V(p_i)$and$V(p_j)$ have a non-empty intersection. It yields a triangulation of P, that is to say a partition of the convex hull of P into d-dimensional vertexes ($i\,.\,e.$ into triangles in 2D, into tetrahedra in 3D, and so on). The formula of $Del(P)$ is Eq. (18). Figure 5a displays an example of a Voronoi diagram and its associated Delaunay triangulation in the plane.

$$Del(p) = \left\{ T\left(p_i, p_j, p_k\right) \middle| p_i \in P, p_j \in P, p_k \in P \atop C\left(p_i, p_j, p_k\right) \cap P \backslash \left(p_i, p_j, p_k\right) = \phi \right\} \tag{18}$$

where $C(p_i, p_j, p_k)$ is the circle circumscribed by three vertices $p_i\,, p_j\,, p_k$, which form a Delaunay Triangle $T(p_i, p_j, p_k)$.

The algorithmic complexity of the Delaunay triangulation of n points is $O(n\log n)$in 2D [2]. Figure 5b and c Show the created Delaunay Triangulations using 20 discrete points.

3.2.2 Gravity Center of Corners

After inter frame matching by optical flow with corners, Delaunay Triangulation is adopted to refine the matching corners (to remove outliers). Based on the refined corners in the current frame, the gravity of these corners is used as a predicted sample candidate.

3.2.3 Predicted location by template matching

A normalized cross correlation based template matching technique has been used as the measurement scheme. The object's template and the rectangular window in the image centered

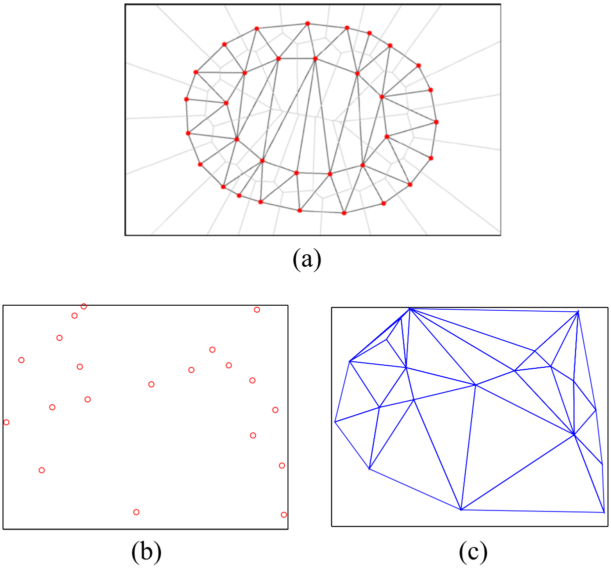


Fig. 5 **a** The Voronoi diagram (*gray edges*) of a set of 2D points(red dots) and its associated Delaunay triangulation (*black edges*). **b**The Delaunay Triangulation Discrete Points(20). **c** Triangulated Meshes

at the predicted position form the two inputs to the correlation system. The equations for normalized cross correlation are as follows.

$$R_{\mathrm{ccoeff}}(x,y)=\frac{\sum_{x'}\sum_{y'}\left[T'\left(x',y'\right)*I'\left(x+x',y+y'\right)\right]}{\sqrt{\sum_{x'}\sum_{y'}\left[T'\left(x',y'\right)\right]^2*\sum_{x'}\sum_{y'}\left[I'\left(x+x',y+y'\right)\right]^2}} \tag{19}$$

$$T'\left(x',y'\right)=T'\left(x',y'\right)-\frac{\sum_{x''}\sum_{y''}\left[T'\left(x'',y''\right)\right]}{(w\cdot h)} \tag{20}$$

$$I'\left(x+x',y+y'\right)=I'\left(x+x',y+y'\right)-\frac{\sum_{x''}\sum_{y''}\left[I'\left(x'',y''\right)\right]}{(w\cdot h)} \tag{21}$$

where, *T*: Template of the object to be matched; T': Zero mean Template; *I*: Search Window in the image; I': Zero mean Search window R : Resultant Matrix.

The position of the maximum of the correlation output R_{ccoeff} is taken as the position of the object. The search window is a rectangular window centered at the predicted position of the object. Its dimensions are taken as twice the dimensions of the object to make a trade-off between the computational cost and the probability of correct measurement. Figure 6 shows the template image and matching result of template matching.result.

3.2.4 Predicted location by SURF matching

SURF, also known as approximate SIFT, employs integral images and efficient scale space construction to generate keypoints and descriptors very efficiently. SURF uses two stages namely keypoint detection and keypoint description [7]. In the first stage, rather than using DoGs as in SIFT, integral images allow the fast computation of approximate Laplacian of Gaussian images using a box filter. The computational cost of applying the box filter is independent of the size of the filter because of the integral image representation. Determinants of the Hessian matrix are then used to detect the keypoints. So SURF builds its scale space by keeping the image size the same and varies the filter size only.

The first stage results in invariance to scale and location. In the final stage, each detected keypoint is first assigned a reproducible orientation. For orientation, Haar wavelet responses

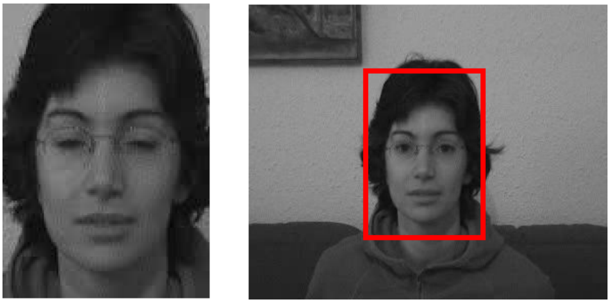

(a) Template Image (b) Template Matching Result

Fig. 6 Template Matching Example. **a** Template Image; **b** Template Matching Result

and directions are calculated for a set of pixels within a radius of where refers to the detected keypoint scale. The SURF descriptor is then computed by constructing a square window centered around the keypoint and oriented along the orientation obtained before. This window is divided into regular sub-regions and Haar wavelets of size are calculated within each sub-region. Each sub-region contributes 4 values thus resulting in 64D descriptor vectors which are then normalized to unit length. The resulting SURF descriptor is invariant to rotation, scale, contrast and partially invariant to other transformations. Shorter SURF descriptors can also be computed however best results are reported with 64D SURF descriptors [7]. The result of SURF matching is illustrated in Fig.7, while Fig.8 denotes all the candidate samples obtained by the method above, different predictions with different colors.

3.3 Motion model

Particle filter [25] is a Bayesian sequential importance sampling technique that aims to estimate the posterior distribution of state variables for a given dynamic system. It uses a set of weighted particles to approximate the probability distribution of the state regardless of the underlying distribution, which is very effective for dealing with nonlinear and non-Gaussian systems. As a typical dynamic state inference problem, online visual tracking can be modeled by particle filter.

There exist two fundamental steps in the particle filter method: 1) prediction and 2) update. Let x_t denote the state variable describing the affine motion parameters of an object and y_t denote its corresponding observation vector (the subscript t indicates the frame index). The two steps recursively estimate the posterior probability based on the following two rules:

$$p(\mathrm{x}_t|\mathrm{y}_{1:t-1}) = \int p(\mathrm{x}_t|\mathrm{x}_{t-1})p(\mathrm{x}_{t-1}|\mathrm{y}_{1:t-1})d\mathrm{x}_{t-1} \tag{22}$$

$$p(\mathrm{x}_t|\mathrm{y}_{1:t}) = \frac{p(\mathrm{y}_t|\mathrm{x}_t)p(\mathrm{x}_t|\mathrm{y}_{1:t-1})}{p(\mathrm{y}_t|\mathrm{y}_{1:t-1})} \tag{23}$$

where$\mathrm{x}_{1\,:\,t} = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_t\}$stand for all available state vectors up to time $t$and $\mathrm{y}_{1\,:\,t} = \{\mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_t\}$denote their corresponding observations. $p(\mathrm{x}_t|\,\mathrm{x}_{t-1})$is called the motion model that describes the state transition between consecutive frames, and $p(\mathrm{y}_t|\,\mathrm{x}_t)$denotes the observation model that evaluates the likelihood of an observed image patch belonging to the object class.

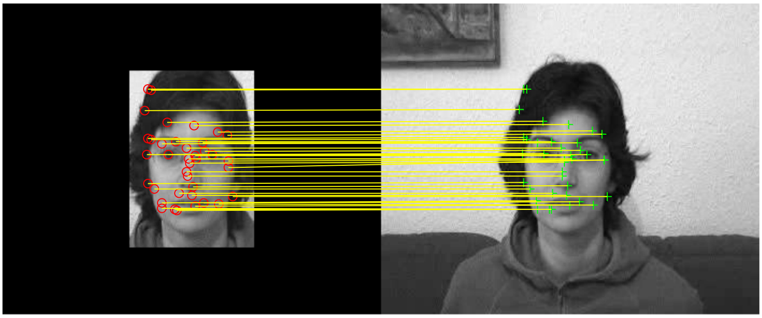


Fig. 7 Matching result By SURF

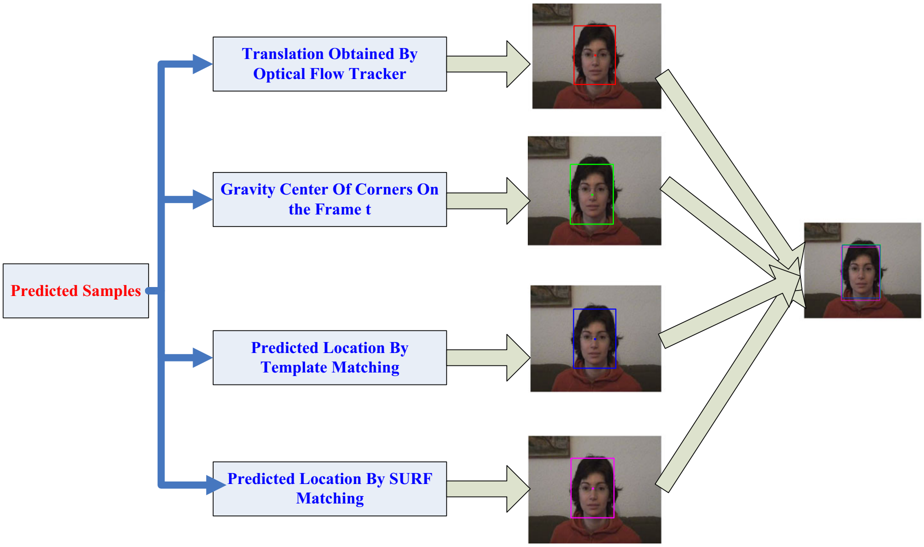

Translation Obtained By Optical Flow Tracker
Gravity Center Of Corners On the Frame t
Predicted Samples
Predicted Location By Template Matching
Predicted Location By SURF Matching

Fig. 8 All the Candidate Samples Obtained by the method above. For the Optical Tracker, the predicted location in the current frame is denoted as red rectangle; Gravity Centers of Corners as green rectangle; Template Matching as blue rectangle; Surf Matching as Magenta rectangle

In the particle filter framework, the posterior $p(\mathrm{x}_t|\,\mathrm{y}_{1\,:\,t})$is approximated by$N$weighted particles$\left\{\mathrm{x}_t^i, w_t^i\right\}_{i=1,2,\ldots,N}$, which are drawn from an importance distribution$q(\mathrm{x}_t|\,\mathrm{x}_{1\,:\,t-1}, \mathrm{y}_{1\,:\,t})$, and the weights of the particles are updated as

$$w_t^i = w_{t-1}^i \frac{p\left(\mathrm{y}_t|\mathrm{x}_t^i\right)p\left(\mathrm{x}_t^i|\mathrm{x}_{t-1}^i\right)}{q\left(\mathrm{x}_t|\mathrm{x}_{1:t-1}, \mathrm{y}_{1:t}\right)} \tag{24}$$

Fig. 9 Examples from the dataset: girl, faceocc1, jogging

Table 1 Tracking sequences used in our experiments

Sequences	Main Challenging Factors
Girl	Scale Variation, Occlusion, In-Plane Rotation, Out-of-Plane Rotation
FaceOcc1	Occlusion
Seq_sb	Scale Variation, Occlusion, In-Plane Rotation, Out-of-Plane Rotation
Deer	Motion Blur, Fast Motion, In-Plane Rotation, Background Clutters, Low Resolution
Jogging	Occlusion, non-rigid object deformation, Out-of-Plane Rotation
Skating2	Scale Variation, Occlusion, non-rigid object deformation, Fast Motion, Out-of-Plane Rotation

In this paper, we adopt$q(\mathrm{x}_t|\,\mathrm{x}_{1:t-1},\mathrm{y}_{1:t})=p(\mathrm{x}_t|\,\mathrm{x}_{t-1})$, which is assumed as a Gaussian distribution similar to [37]. In detail, six parameters of the affine transform are used (i.e., $\mathrm{x}_t=\{x_t,y_t,\theta_t,s_t,\alpha_t,\phi_t\}$, where$x_t,y_t,\theta_t,s_t,\alpha_t,\phi_t$denote$x,y$translations, rotation angle, scale, aspect ratio, and skew, respectively). The state transition is formulated by random walk, i.e., $p(\mathrm{x}_t|\,\mathrm{x}_{t-1})=\mathbb{N}(\mathrm{x}_t;\mathrm{x}_{t-1},\psi)$, where$\psi$ is a diagonal covariance matrix. Finally, the state$\mathrm{x}_t$is estimated as$\mathrm{x}_t=\sum_{i=1}^{N}w_t^i\mathrm{x}_t^i$. We note that the key of designing a practical tracking algorithm is to develop an effective and efficient observation likelihood$p(\mathrm{y}_t|\,\mathrm{x}_t)$.

4 Experiment and results

4.1 Datasets and baselines

The VOT2013 dataset [42] includes various real-life visual phenomena, while containing a small number of sequences to keep the time for performing the experiments reasonably low. The VOT2013 challenge consists of 16 color image sequences with 172 to 770 frames: bicycle, bolt, car, cup, david, diving, face, gymnastics, hand, iceskater, juice, jump, singer, sunshade, torus, and woman. The sequences have been selected to make the tracking a challenging task: objects change aspect or are articulated, the scale and orientation vary, illumination changes and occlusions occur.

Some example frames are shown in Fig.9. The attributes of testing sequences used in our paper are shown in Table 1.

In addition, given the tracking result $R_T$and the ground truthR_G, we use the detection criterion in the PASCAL VOC [19] challenge, i.e., Eq. (25) to evaluate the success rate.

$$\text{score}=\frac{\text{area}(R_T\cap R_G)}{\text{area}(R_T\cup R_G)} \tag{25}$$

4.2 Implementation details

The proposed method has been implemented in Matlab and tested on a 1.73 GHz PC with 2 GB memory. The number of particle samples processed in the experiments is 100. During

Table 2 Time Complexity of Compared Methods (fps)

Method	OAB	L1	IVT	MIL	ASLA	FCNT	ASLA_DW
fps	6.042	13.904	16.818	0.176	5.945	3	1.108

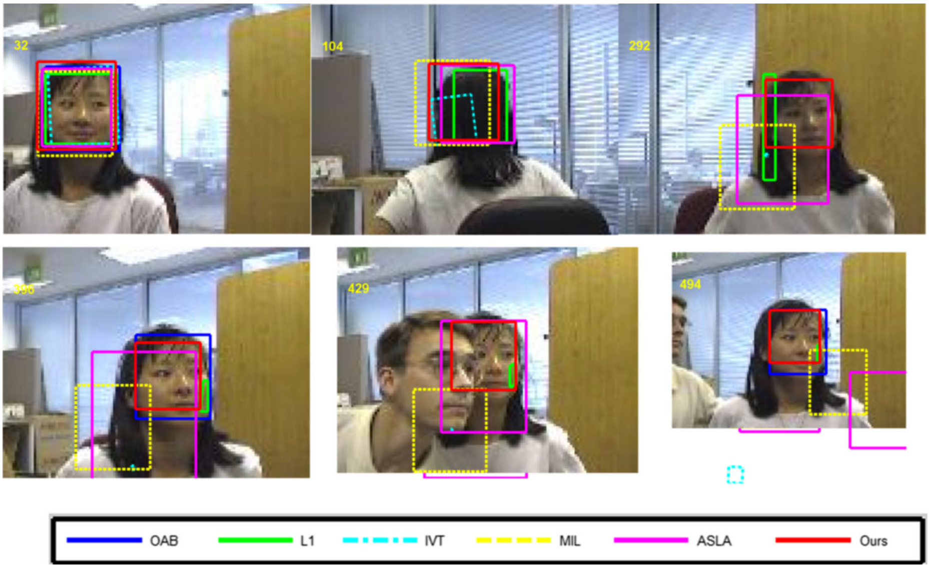


Fig. 10 The tracking results of Gril sequence

tracking, the pixel's values of each frame are normalized into [0, 1]. Each particle is associated with an image patch. After image scaling, the image patch is normalized to$N_1 \times N_2$ pixels. In the experiments, the parameters (N1, N2) are chosen as (32, 32). Due to the multiple sub process used in our model, the time complexity of our proposed method is about 1 fps. The time complexity of different methods is shown in Table 2. In our experiments, we have compared the tracking results of our proposed method with those of state-of-the-art methods, such as OAB [20], L1 [29], MIL [4], IVT [37], ASLA [27], FCNT [41]. We implemented these trackers using publicly available source codes or binaries provided by the authors. They were initialized using their default parameters.

4.3 Qualitative comparisons

The first test sequence is "Girl". We show some representative frames of the tracking results of five different trackers in Fig.10. The six representative frame indices are 32, 104, 292, 396, 429, and 494. The man's face is passing in front of the woman's face. From the tracking results, we can see that MIL tracker loses the target from frame 292, and finally causes a big

Table 3 Center Location Errors of Testing Sequences

Sequence	OAB	L1	IVT	MIL	ASLA	FCNT	ASLA_DW
Girl	57.5996	11.3976	27.3029	20.9609	18.2574	7.0670	5.1502
FaceOcc1	32.1380	15.0316	17.5749	43.2920	34.3677	21.8209	12.3887
Seq_sb	75.6947	51.1431	16.7259	18.5635	68.6419	15.5634	12.5541
Deer	52.1110	221.1957	226.5850	28.8649	120.6866	7.8316	4.7234
Jogging	15.1373	95.5248	83.2741	113.6823	102.8373	5.8445	4.8853
Skating2	352.5803	169.2047	152.2267	79.6480	49.7983	32.0768	25.0924

Table 4 Overlap Metric of Testing Sequences

Sequence	OAB	L1	IVT	MIL	ASLA	FCNT	ASLA_DW
Girl	0.2032	0.2839	0.1606	0.3009	0.3954	0.5416	0.5557
FaceOcc1	0.7132	0.7512	0.7462	0.4681	0.5869	0.6490	0.7946
Seq_sb	0.0611	0.0394	0.2462	0.3918	0.0480	0.2365	0.6000
Deer	0.5426	0.0442	0.0314	0.5100	0.0449	0.7052	0.7631
Jogging	0.7067	0.1671	0.1490	0.1668	0.1661	0.7175	0.7538
Skating2	0.0449	0.0297	0.0454	0.2326	0.2866	0.2836	0.4993

drift, the same as IVT tracker. The OAB tracker fails to track the target in frames 104, 429 due to large pose changes and occlusion of another man. For the l1 tracker, it can not track the target in frames 292, 396, 429, and 494. Compared with other trackers, although ASLA tracker can track the target, however, the location accuracy is lower than our proposed tracker (ASLA_DW) which can obtain good tracking results and improve location accuracy. The location errors and the overlap metric of six comparison methods on video sequence “Girl” are respectively shown in Table 3 and Table 4. Figure 16 is the location error curve of tracking results on Girl sequence for the compared methods, while Fig.17 is the overlap metric results.

The second test sequence is “FaceOcc1”. We show some representative frames of the tracking results of five different trackers in Fig.11. The six representative frame indices are 60, 196, 356, 418, 558, and 774. The woman’s face is partially occluded by a book from left part of face to the right part of the face. From the tracking results, we can see that the tracking accuracy of MIL tracker is lower than other trackers. For the OAB、L1、 IVT and our proposed tracker, the target can be well tracked, although the location accuracy of l1 tracker is slightly lower. Compared with other trackers, ASLA tracker can not obtain good tracking

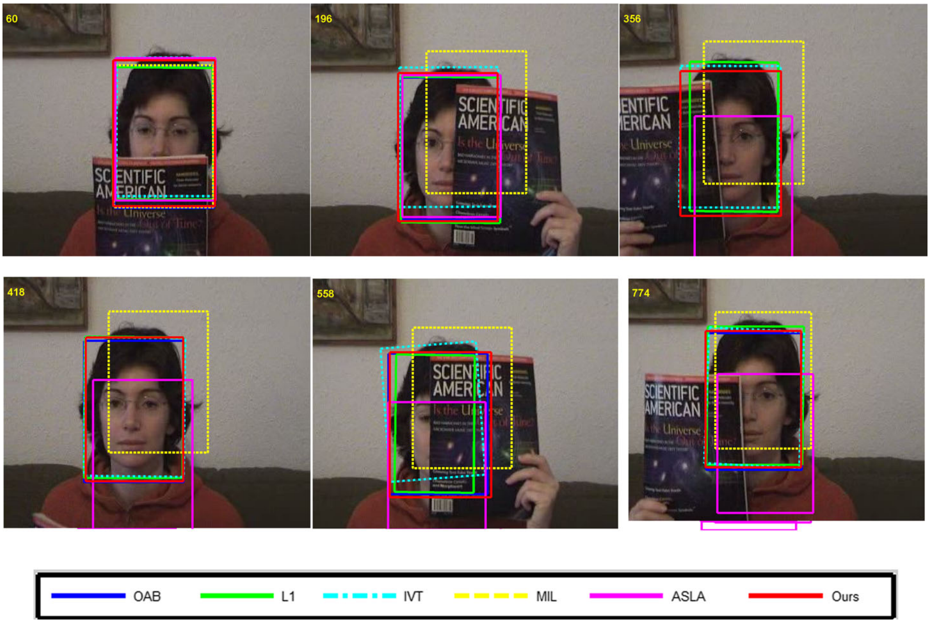


Fig. 11 The tracking results of FaceOcc1 sequence

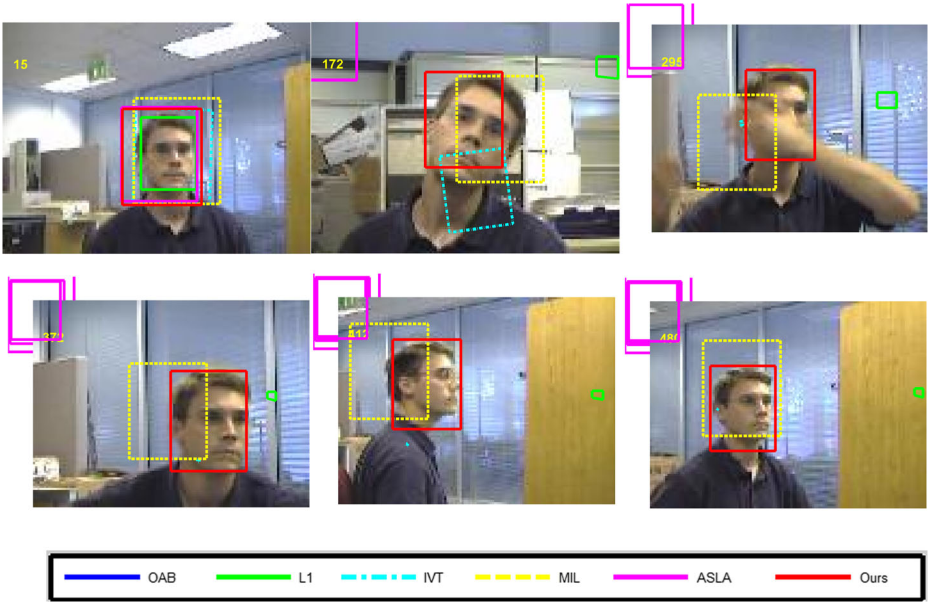


Fig. 12 The tracking results of seq_sb sequence

results in frames 356,418,558,774, whereas our proposed tracker can track the woman' face robustly and improve the tracking location accuracy better. The location errors and the overlap metric of six comparison methods on video sequence "FaceOcc1" are respectively shown in Table 3 and Table 4. Figure 16 is the location error curve of tracking results on FaceOcc1 sequence for the compared methods, while Fig.17 is the overlap metric results.

The third test sequence is "seq_sb". We show some representative frames of the tracking results of five different trackers in Fig.12. The six representative frame indices are 15, 172, 295, 372, 412, and 480. The man's face is totally occluded by a book with out of plane rotation. From the tracking results, we can see that MIL tracker loses the target from frame 295, and finally causes a big drift, the same as IVT,OAB,L1, ASLA tracker. Compared with other trackers, our

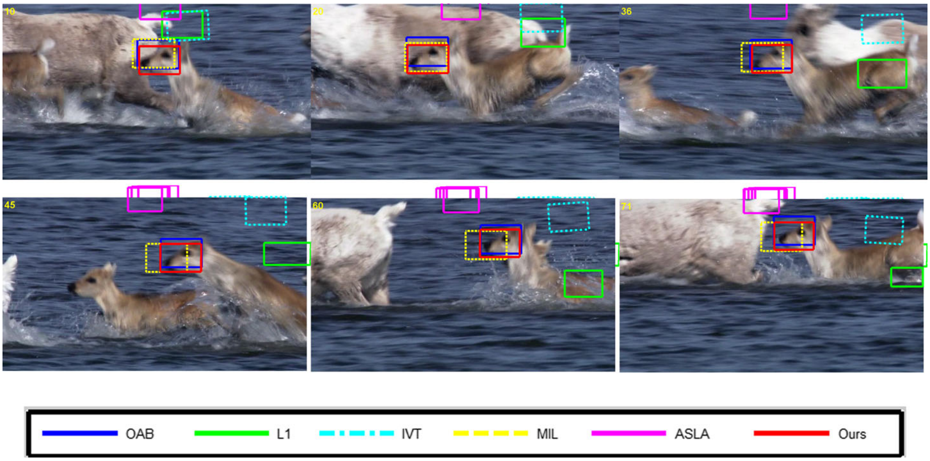


Fig. 13 The tracking results of Deer sequence

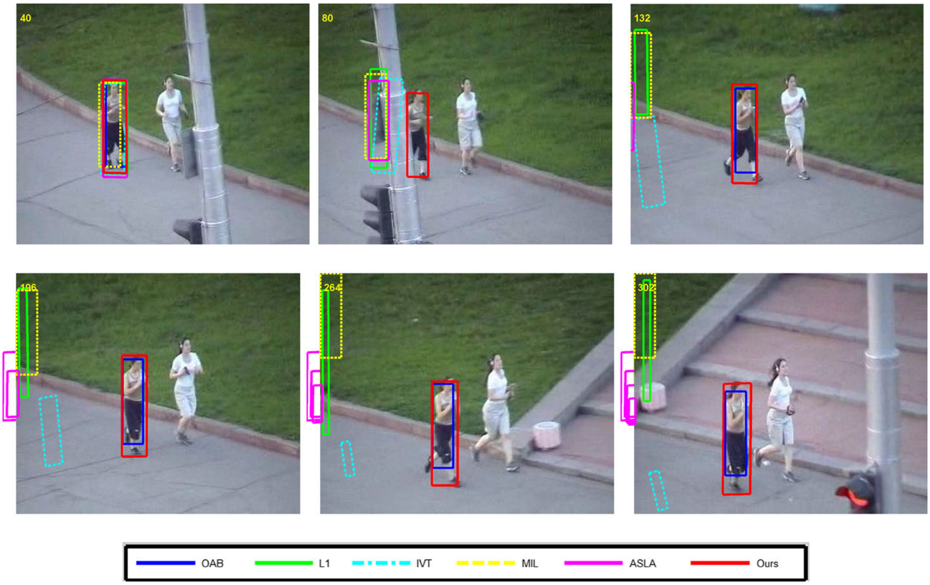


Fig. 14 The tracking results of Jogging sequence

proposed tracker can still obtain good tracking results under such big challenging video. The location errors and the overlap metric of six comparison methods on video sequence “seq_sb” are respectively shown in Table 3 and Table 4. Figure 16 is the location error curve of tracking results on seq_sb sequence for the compared methods, while Fig.17 is the overlap metric results.

The fourth test sequence is “Deer”. We show some representative frames of the tracking results of five different trackers in Fig.13. The six representative frame indices are 10, 20, 36, 45, 60, and 71. The challenging factors are motion blur, fast motion, in-plane rotation, background clutters, low-resolution. From the tracking results, we can see that MIL tracker can track the target in frames 20, 36, 45, 60, and 71, while the location accuracy is lower than OAB Tracker. For the l1, IVT, ASLA tracker, they totally lose the target in the whole video sequence. Compared with other

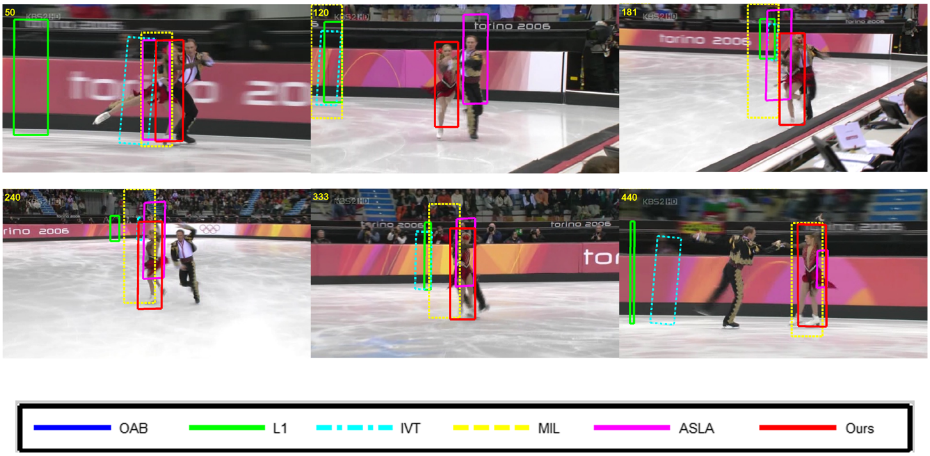
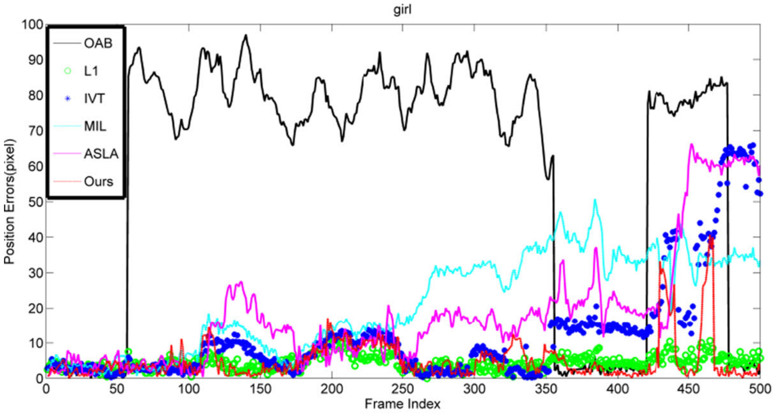
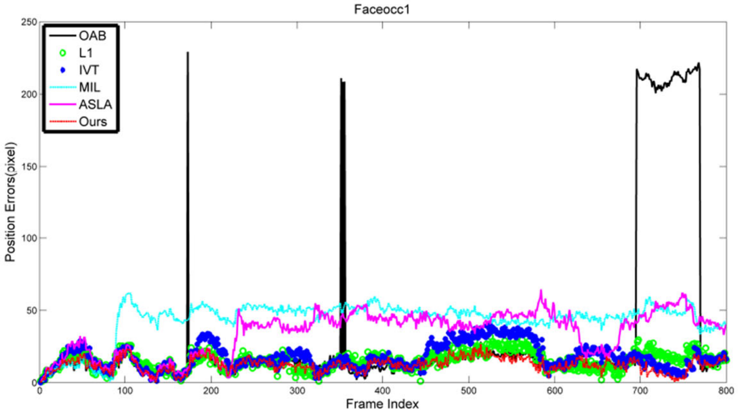


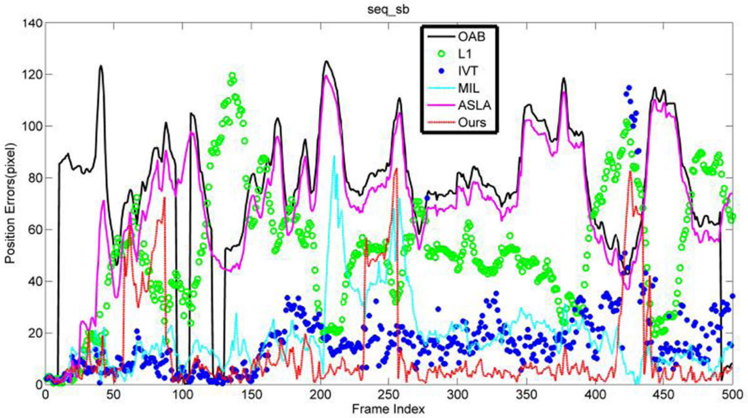
Fig. 15 The tracking results of Skating2 sequence



(a) girl

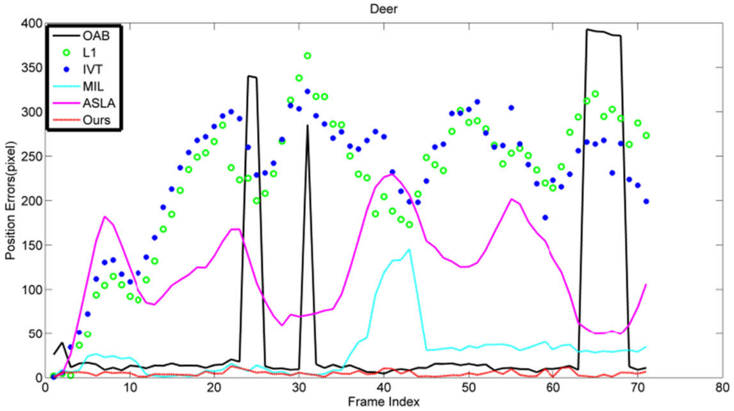


(b) Faceocc1

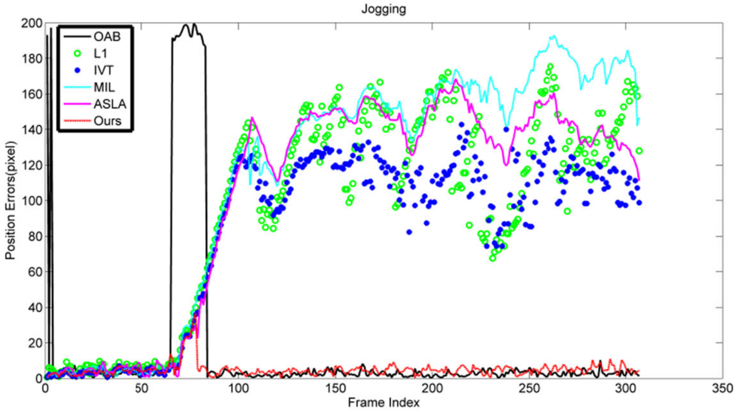


(c) seq_sb

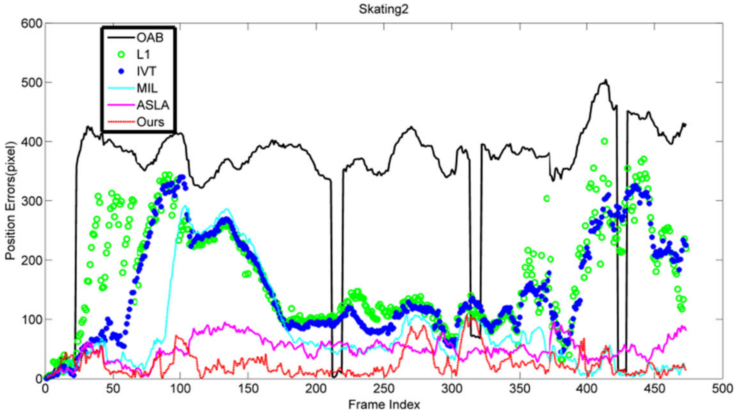
Fig. 16 The tracking error for each test sequence. The error is measured the same as in Table 3



(d) Deer

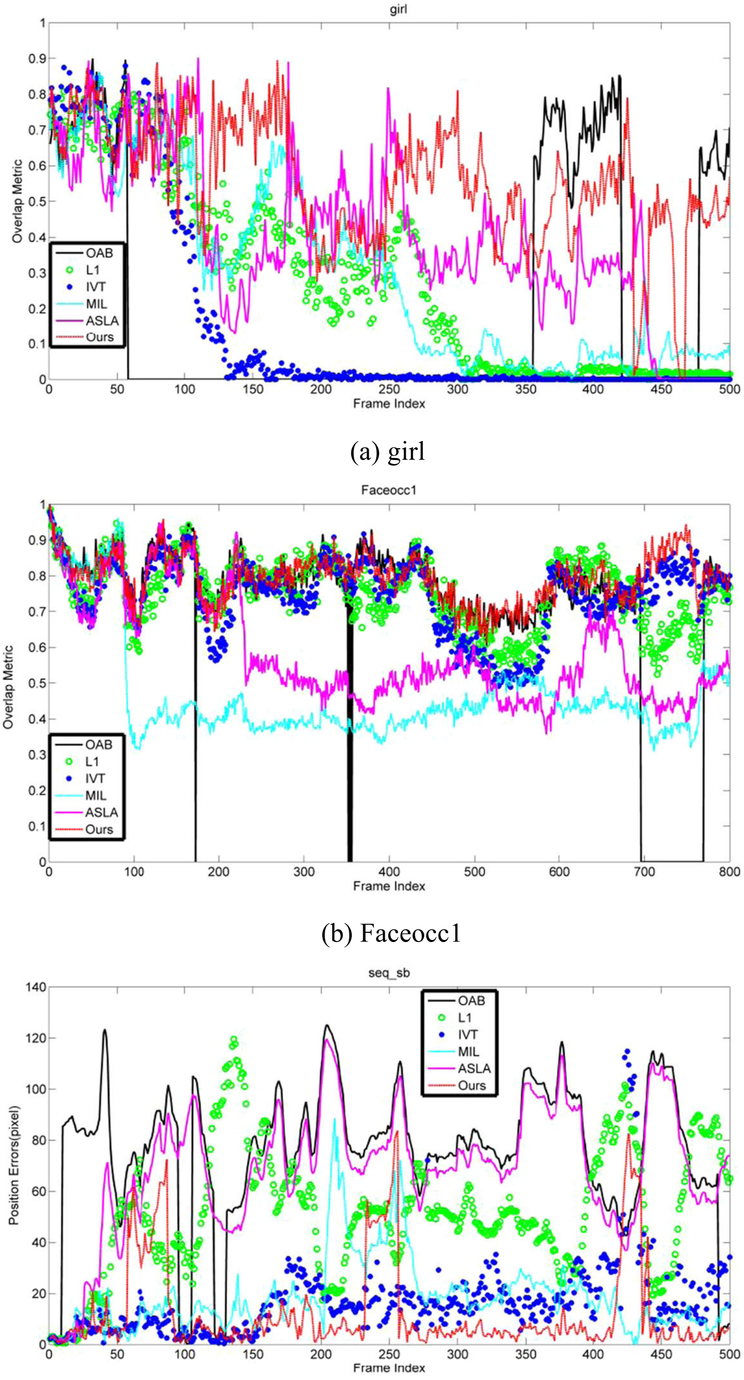


(e) Jogging



(f) Skating2

Fig. 16 (continued)

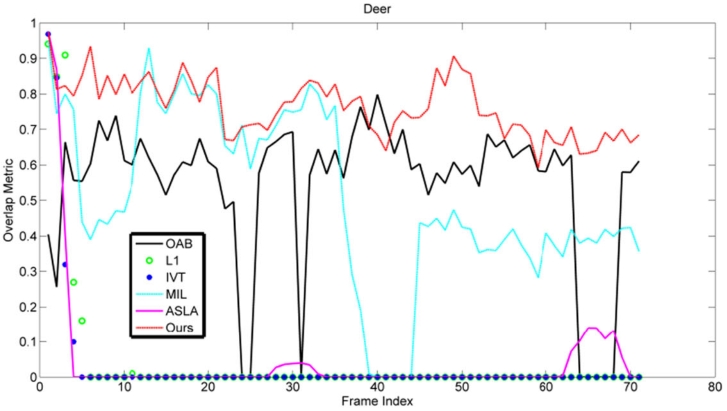


(a) girl

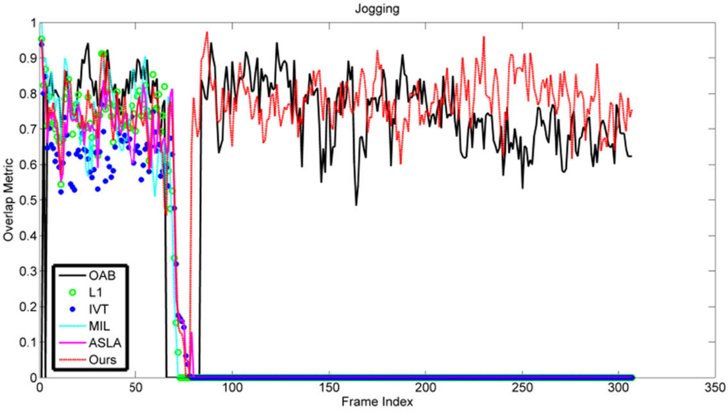
(b) Faceocc1

(c) seq_sb

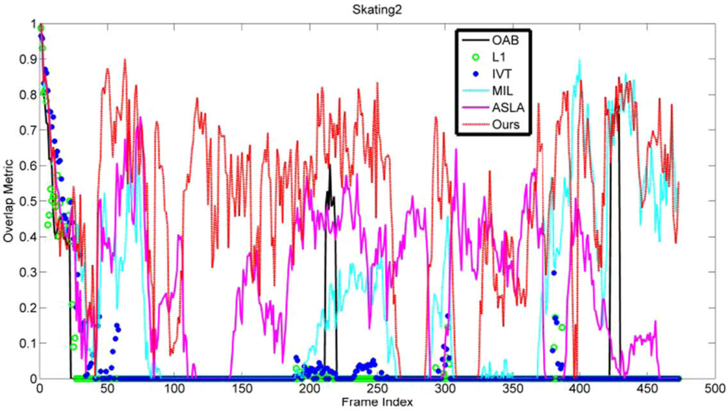
Fig. 17 The Overlap metric for each test sequence. The overlap metric is measured the same as in Table 4



(d) Deer



(e) Jogging



(f) Skating2

Fig. 17 (continued)

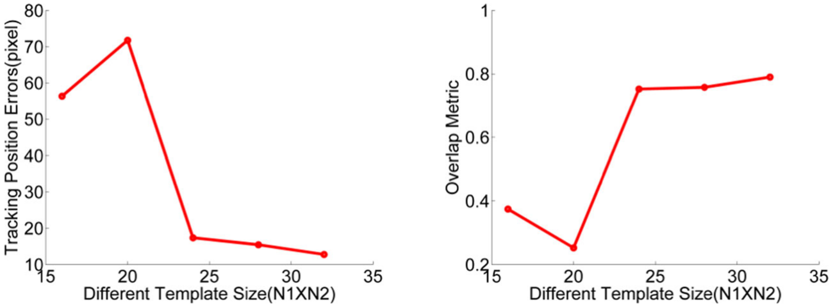
Table 5 Overall performance for different methods

Method	OAB	L1	IVT	MIL	ASLA	FCNT	ASLA_DW
Ranking	7	6	4	5	3	2	1

trackers, OAB tracker and our proposed tracker can obtain good tracking results. The location errors and the overlap metric of six comparison methods on video sequence "Deer" are respectively shown in Table 3 and Table 4. Figure 16 is the location error curve of tracking results on Deer sequence for the compared methods, while Fig.17 is the overlap metric results.

The fifth test sequence is "Jogging". We show some representative frames of the tracking results of five different trackers in Fig.14. The six representative frame indices are 40, 80, 132, 196, 264, and 302. The jogging woman is occluded by a tree. From the tracking results, we can see that MIL, L1, IVT, ASLA tracker lose the target in frames 80, 132, 196, 264, and 302 due to the jogging woman totally occluded by the tree. For the OAB tracker, it loses the target in frame 80, whereas it can track the target in the remaining frames. Compared with OAB Tracker, our proposed tracker can obtain good tracking results in spite of the target totally occluded by the tree. The location errors and the overlap metric of six comparison methods on video sequence "Jogging" are respectively shown in Table 3 and Table 4. Figure 16 is the location error curve of tracking results on Jogging sequence for the compared methods, while Fig.17 is the overlap metric results (Table 4).

The six test sequence is "Skating2". We show some representative frames of the tracking results of five different trackers in Fig.15. The six representative frame indices are 50, 120, 181, 240, 333, and 440. This video sequence undergoes fast motion, occlusion and deformation. From the tracking results, we can see that MIL tracker loses the target in frame 120, and finally causes a big drift in frames 181,240,333. However, it tracks the target rightly in frame 440. The L1 tracker loses the target in the representative frames, the same as OAB, IVT tracker. The IVT tracker can not track the target because of its incapable of dealing with occlusion. The tracking accuracy of ASLA tracker is not good than our proposed tracker. The location errors and the overlap metric of six comparison methods on video sequence "Skating2" are respectively shown in Table 3 and Table 4. Figure 16 is the location error curve of tracking results on Skating2 sequence for the compared methods, while Fig.17 is the overlap metric results. From Tables 3 and 4, we can see that the overall performance for different methods of is shown in Table 5.



(a) Different Template size vs Tracking errors (b) Different Template size vs Tracking Overlap metric

Fig. 18 The sensitivity of the two normal sizes to the overall results

4.4 Parameters selection

Since there are many parameters are involved in the formulations, in fact, each parameter may influence the overall results. The difference is that the importance of each parameter. We only show the primary parameters. We settle the optimal parameters based on our experiments. We first set one parameter change, the rest of parameters are not changing, through this we get the optimal one parameter; then get the second optimal parameters one by one. For the normal size$N_1 \times N_2$of image patch, let$N_1 = N_2$, and range of $N_1 = [16\ 20\ 24\ 28\ 32]$, we verify the sensitivity of the two parameters to the overall results. The results are shown in Fig. 18.

4.5 Analysis of the proposed method

Since for the problem of visual tracking, it is in fact a combination of approaches. Each method can have some effect on the result of visual tracking approach, such as for different motion models and different appearance models. Our method is designed to improve the appearance and model models, so we have integrated some methods. Experimental results on some publicly available benchmarks of video sequences demonstrate the accuracy and effectiveness of our tracker.

5 Conclusion

In this paper, we propose a robust tracking algorithm by integrating the generative and discriminative model. The object appearance model is composed of generative target model and a discriminative classifier. For the generative target model, we adopt the weighted structural local sparse appearance model combining patch based gray value and Histogram of Oriented Gradients feature as the patch dictionary. By sampling positives and negatives, alignment-pooling features are obtained based on the patch dictionary through local sparse coding, then use a support vector machine to train the discriminative classifier. A robust inter-frame matching based on optical flow and Delaunay triangulation accompanied with template matching is adopted to improve the proposal distribution of particle filter to enhance the performance of tracking. Our approach is shown to effectively improve the tracking performance on challenging scenarios.

Acknowledgments This work was supported by the by Shanghai University Outstanding Teachers Cultivation Fund Program A30DB1524011-21 and 2015 School Fund Project A01GY15GX48.

References

1. Adam A, Rivlin E, and Shimshoni I (2006) Robust fragments-based tracking using the integral histogram. In: 2006 I.E. conference on computer vision and pattern recognition (CVPR). IEEE, pp 798–805
2. Attali D, Boissonnat J-D , and Lieutier A (2003) Complexity of the delaunay triangulation of points on surfaces the smooth case. In: 2003 Proceedings of the nineteenth annual symposium on Computational geometry ACM, pp 201–210
3. Avidan S (2004) Support vector tracking. IEEE Trans Pattern Anal Mach Intell 26(8):1064–1072
4. Babenko B, Yang M-H, and Belongie S (2009) Visual tracking with online multiple instance learning. In: 2009 I.E. conference on computer vision and pattern recognition (CVPR) IEEE, pp 983–990

5. Babenko B, Yang MH, Belongie S (2011) Robust object tracking with online multiple instance learning. IEEE Trans Pattern Anal Mach Intell 33(8):1619–1632
6. Barron JL, Fleet DJ, Beauchemin SS (1994) Performance of optical flow techniques. Int J Comput Vis 12(1): 43–77
7. Bay, H., Tuytelaars, T., and Van Gool, L (2006) Surf: Speeded up robust features. In: 2006 European conference on computer vision (ECCV) Springer Berlin Heidelberg, pp 404–417
8. Black MJ, Jepson AD (1998) Eigentracking: robust matching and tracking of articulated objects using a view-based representation. Int J Comput Vis 26(1):63–84
9. Boissonnat J-D and Yvinec M, (1998) Algorithmic Geometry, chapter Voronoi diagrams: Euclidian metric, Delaunay complexes(Cambridge University Press).
10. Brox T, Bruhn A, Papenberg N, and Weickert J (2004) High accuracy optical flow estimation based on a theory for warping. In: 2004 European Conference on Computer Vision(ECCV) Springer Berlin Heidelberg, pp 25–36
11. Burgess C (1998) A tutorial on support vector machines for pattern recognition. Data Mining Knowl. Discovery 2(2):121–167
12. Candes EJ, Romberg JK, Tao T (2006) Stable signal recovery from incomplete `and inaccurate measurements. Commun Pure Appl Math 59(8):1207–1223
13. Collins RT, Liu Y, Leordeanu M (2005) Online selection of discriminative tracking features. IEEE Trans Pattern Anal Mach Intell 27(10):1631–1643
14. Comaniciu D, Ramesh V, Meer P (2003) Kernel-based object tracking. IEEE Trans Pattern Anal Mach Intell 25(5):564–575
15. Cortes C, Vapnik VN (1995) Support-vector networks. J Mach Learn 20(3):273–297
16. Dalal, N., Triggs, B (2005) Histograms of Oriented Gradients for Human Detection. In: 2005 I.E. conference on computer vision and pattern recognition (CVPR) IEEE, pp 886–893
17. Dou J, Li J (2012) Robust image matching based on SIFT and delaunay triangulation. Chin Opt Lett 10(s1): 11001
18. Dou J, Li J (2014) Image matching based local Delaunay triangulation and affine invariant geometric constraint. Optik-International Journal for Light and Electron Optics 125(1):526–531
19. Everingham M, Van Gool L, Williams CKI, et al. (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338
20. Grabner H and Bischof H (2006) On-line boosting and vision. In: 2006 I.E. conference on computer vision and pattern recognition (CVPR) IEEE, pp 260–267
21. Grabner H, Leistner C, and Bischof H (2008) Semi-supervised on-line boosting for robust tracking. In: 2008 European Conference on Computer Vision (ECCV) Springer Berlin Heidelberg, pp 234–247
22. Hare, S., Saffari, A., and Torr, P. H (2011) Struck: Structured output tracking with kernels. In: 2011 International Conference on Computer Vision (ICCV) IEEE, pp. 263–270
23. Henriques JF, Caseiro R, Martins P, and Batista J (2012) Exploiting the circulant structure of tracking-by-detection with kernels. In: 2012 European Conference on Computer Vision(ECCV). Springer Berlin Heidelberg, pp 702–715
24. Horn BKP, Schunck BG (1981) Determining optical flow. Artif Intell 17(1–3):185–203
25. Isard M, Blake A (1998) CONDENSATION: Conditional density propagation for visual tracking. Int J Comput Vis 29(1):5–28
26. Jang D, Choi H, and Kim G (1996) Real-time tracking with Kalman filter. In: 1996 Proceeding of IAPR Workshop on Machine Vision Applications (MVA). pp 10–13
27. Jia X, Lu H, Yang M-H (2012) Visual tracking via adaptive structural local sparse appearance model. In: 2012 I.E. conference on computer vision and pattern recognition (CVPR) IEEE, pp 1822–1829
28. Kalal Z, Mikolajczyk K, Matas J (2012) Tracking-learning-detection. IEEE Trans Pattern Anal Mach Intell 34(7):1409–1422
29. Li Xi , et al (2007) Robust visual tracking based on incremental tensor subspace learning. In: 2007 I.E. International Conference on Computer Vision (ICCV). IEEE, pp 1–8
30. Li X, Hu W, Shen C, Zhang Z, Dick A, Van Den Hengel A (2013) A survey of appearance models in visual object tracking. ACM Trans. Intell. Syst. Technol 4(4):58
31. Lowe DG (2004) Distinctive image features form scale-Invariant keypoints. Int J Comput Vis 60(2):91–110
32. Mei, X., and Ling, H. (2009) Robust visual tracking using l1 minimization. In: 2009 I.E. International Conference on Computer Vision (ICCV) IEEE, pp 1436–1443
33. Mei X, Ling H (2011) Robust Visual Tracking and Vehicle Classification via Sparse Representation. IEEE Trans Pattern Anal Mach Intell 33(11):2259–2272

34. Nie L, Wang M, Zha ZJ, Chua TS (2012a) Oracle in image search: a content-based approach to performance prediction. ACM Transactions on Information Systems (TOIS) 30(2):13
35. Nie, L., Yan, S., Wang, M., Hong, R., & Chua, T. S. (2012b) Harvesting visual concepts for image search with complex queries. In: 2012 Proceedings of the 20th ACM international conference on Multimedia. ACM, pp. 59–68
36. Ozuysal M, Calonder M, Lepetit V, et al. (2010) Fast keypoint recognition using random ferns. IEEE Trans Pattern Anal Mach Intell 32(3):448–461
37. Ross DA, Lim J, Lin RS, et al. (2008) Incremental learning for robust visual tracking. Int J Comput Vis 77(1–3):125–141
38. Saffari, A., Leistner, C., Santner, J., Godec, M., & Bischof, H (2009) On-line random forests. In: 2009 I.E. International Conference on International Conference on Computer Vision Workshops (ICCV Workshops). IEEE, pp 1393–1400
39. Tang F, Brennan S, Zhao Q, and Tao H (2007) Co-tracking using semi-supervised support vector machines. In: 2007 I.E. International Conference on Computer Vision (ICCV) IEEE, pp 1–8
40. Wang Y, Tang X, Cui Q (2012) Dynamic appearance model for particle filter based visual tracking. Pattern Recogn 45(12):4510–4523
41. Wang L, Ouyang W, Wang X, et al (2015) Visual tracking with fully convolutional networks. In 2015 I.E. International Conference on Computer Vision (CVPR) IEEE, pp 3119–3127.
42. Wu, Y., Lim, J., and Yang, M. H (2013) Online object tracking: A benchmark. In: 2013 I.E. conference on computer vision and pattern recognition (CVPR) IEEE, pp 2411–2418
43. Yang H, Shao L, Zheng F, Wang L, Song Z (2011) Recent advances and trends in visual tracking: a review. Neurocomputing 74(18):3823–3831
44. Yilmaz A, Javed O, Shah M (2006) Object Tracking: A Survey. ACM Comput Surv 38(4):1–45
45. Zhang K, Zhang L, and Yang M-H (2012) Real-time compressive tracking. In: 2012 European Conference on Computer Vision(ECCV) Springer Berlin Heidelberg, pp 864–877
46. Zhong B et al. (2014) Visual tracking via weakly supervised learning from multiple imperfect oracles. Pattern Recogn 47(3):1395–1410
47. Zhou, T., Lu, Y., and Qiu, M. (2015) Online visual tracking using multiple instance learning with instance significance estimation. CoRR

Jian-fang DOU DingZhou City, Hebei Province, China. He is currently a teather with the School of Intelligent Manufacturing and Control Engineering, Shanghai Second Polytechnic University, Shanghai, China. In 2006, obtained the Bachelor Degree, Geography Information System, College of Traffic and Transport, Hebei Polytechnic University, Hebei Province, China. In 2009, obtained the Master Degree, Photogrammetry and Remote Sensing, School of Civil Engineering, Tongji University, Shanghai, China. In 2014, received the PhD Degree from Shanghai Jiaotong University, Shanghai , China. His major interests are object detection and tracking, machine learing, Intelligent Measurement and Control.

Qin Qin She is currently a Associate Professor with the School of Intelligent Manufacturing and Control Engineering, Shanghai Second Polytechnic University, Shanghai, China. In 2006, received the PhD Degree from The Shanghai Institute of Technical Physics of the Chinese Academy of Sciences, Shanghai, China. Her major interests are Intelligent Detection, Image processing and pattern recognition.

Zi-mei Tu Zhejiang Province, China. She is currently a teather with the School of Intelligent Manufacturing and Control Engineering, Shanghai Second Polytechnic University, Shanghai, China. In 2012, obtained the Bachelor Degree, Information display and photoelectric technology, Shanghai Second Polytechnic University, Shanghai, China. In 2014, obtained the Master Degree, Shanghai Normal University, China. Her major interests are image segmentation and three dimension reconstruction.