

## Research Article

# Visual Tracking Using an Insect Vision Embedded Particle Filter

Wei Guo,<sup>1</sup> Qingjie Zhao,<sup>1</sup> and Dongbing Gu<sup>2</sup>

<sup>1</sup>Beijing Key Laboratory of Intelligence Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China

<sup>2</sup>School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK

Correspondence should be addressed to Wei Guo; gwbit0731@gmail.com

Received 22 September 2014; Accepted 16 January 2015

Academic Editor: Ebrahim Momoniat

Copyright © 2015 Wei Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Particle filtering (PF) based object tracking algorithms have drawn great attention from lots of scholars. The core of PF is to predict the possible location of the target via the state transition model. One commonly adopted approach is resorting to prior motion cues under the smooth motion assumption, which performs well when the target moves with a relatively stable velocity. However, it would possibly fail if the target is undergoing abrupt motion. To address this problem, inspired by insect vision, we propose a simple yet effective visual tracking framework based on PF. Utilizing the neuronal computational model of the insect vision, we estimate the motion of the target in a novel way so as to refine the position state of propagated particles using more accurate transition mode. Furthermore, we design a novel sample optimization framework where local and global search strategies are jointly used. In addition, we propose a new method to monitor long duration severe occlusion and we could recover the target. Experiments on publicly available benchmark video sequences demonstrate that the proposed tracking algorithm outperforms the state-of-the-art methods in challenging scenarios, especially for tracking target which is undergoing abrupt motion or fast movement.

## 1. Introduction

Visual tracking is of great significance in many vision applications such as video surveillance and human-computer interaction. In recent years, numerous algorithms have been proposed in the development of tracking algorithms and much success has been demonstrated under various scenarios. Nevertheless, numerous issues, such as abrupt motion, fast movement, and severe occlusion, remain to be addressed.

Generally, the tracking algorithms fall into two main categories: the generative algorithms [1–5] and the discriminative algorithms [6–10]. Generative tracking algorithms typically learn a target model to represent the target object and then try to search for the best image region most similar to the target model. Discriminative algorithms view the tracking problem as a binary classification task to separate the target object from its local background.

The particle filtering- (PF-) based tracking algorithms are very popular generative algorithms since they can effectively solve the nonlinear and non-Gaussian problems. In

the PF-based tracking process, the state transition model plays a vital role in predicting the possible location of the target. One commonly adopted approach to propagate the sample set is resorting to prior motion cues and amending the prediction according to the Gaussian distribution [5, 11–16]. This approach performs well when target moves with a relatively stable velocity or with a predictable motion pattern [12, 17]. However, in the real world, abrupt motion or fast movement scenarios are frequently available, thereby causing these algorithms to drift away the target objects gradually and even lose the target. Another predictable and reasonable approach is utilizing the motion direction and moving speed of the target to revise the state transition model so as to refine a more accurate position.

To solve those problems, in this paper, we propose a simple yet effective insect vision inspired framework of visual tracking based on PF. In a cluttered moving background, flying insects demonstrate extraordinary capability in locating and detecting visual objects. Insect ommatidium can respond to the motion pattern including the motion direction and

the relative image instant velocity [19, 20]. The insect vision computational model theory [18] is deduced from the neuronal computational model of the way biological ommateum processing information. In this work, unlike conventional state transition models previously mentioned, we take the motion pattern of the target into consideration to construct a more accurate model so as to refine the position state of propagated particles. Furthermore, to seek a more reliable candidate image region, we design a novel sample optimization framework where local and global search strategies are jointly used. The local search method tries to find the best sample using the improved state transition model in a generative subspace. For the global search strategy, unlike the conventional methods which usually search the full frame regions with a sliding window, we try to get the candidate sample in a discriminative subspace and just test the subregions where the response of the insect vision computational model is not zero; that is, the target is likely to appear with a high probability, and then we compare these subregions with the template, thus reducing the computational cost. We always do a local search strategy in every tracking frame while we do a global search strategy for every constant frame. Therefore, such compact associations of discriminative subspace and generative subspace can benefit from each other. In addition, the target may disappear when severe occlusion occurs. To address this issue, we propose a new method to monitor the situation. Once the target is lost, we recover it via the global search method previously mentioned. Besides, we give an incremental-learning-based online template update method to adapt the appearance changes. The main contributions of this paper are summarized as follows:

- (i) an improved transition model that takes into account the motion information obtained by insect vision computational theory to refine the position state of propagated particles in the PF tracking process,
- (ii) an efficient optimization framework where local and global search strategies are jointly used to seek for a more reliable candidate image region,
- (iii) a novel method to judge the tracking failure and a target recovering mechanism,
- (iv) an incremental-learning-based online template update method.

The remainder of this paper is organized as follows. We first review the most relevant work in Section 2. Then, the mathematical concepts of PF and insect vision theory will be introduced briefly in Section 3. The proposed algorithm is detailed in Section 4, including model representation, motion estimation, local and global search strategies, and online template update method. In Section 5, the experimental results are presented with comparisons with state-of-the-art methods on challenging sequences. Section 6 concludes with a discussion and the future work.

## 2. Related Work

Most recently, several studies in the literature have given detailed surveys of tracking [12, 21–24]. Among all the various

algorithms, PF-based tracking algorithms have always drawn the attention from the researchers since Isard and Blake [12, 22] first introduced PF into solving tracking problems. Despite much demonstrated success of these generative tracking approaches, several problems remain to be solved, such as abrupt motion, fast movement, and severe occlusion. In this section, we review the two most relevant topics that motivate this work: the state transition model and the search strategies.

Abrupt motion or fast movement scenarios are frequently available in tracking process. Most of the tracking methods cannot handle these problems because of their smooth motion assumptions. The conventional approach is usually resorting to prior motion cues. Assuming the target region moving with constant velocity and scale change, Nummiaro et al. [11] use a first-order dynamic model to propagate the sample set. These kinds of dynamic propagate modes are very popular now, some with slight modification (e.g., using standard second-order autoregressive process to replace the first-order model) [13, 23–25]. Sullivan and Rittscher [26] propose a deterministic search method to guide the random samples. This method computes the momentum for each particle. A deterministic search and the stochastic diffusion are performed alternatively according to the momentum to generate the offspring of the particles. Zhou et al. [27] propose an adaptive velocity motion model in the prediction scheme. The adaptive velocity is calculated using a first-order linear prediction method based on the appearance difference between two successive frames. Kwon and Lee [3] decompose the motion model into multiple basic motion models in the tracking process. Each basic motion model describes different types of motions made by a Gaussian perturbation with a different variance. This method mainly covers two kinds of motions: the smooth motion with a small variance and the abrupt motion with a large variance. Similarly, Yao et al. [28] utilize a variable variance to balance the smooth motion and fast movement. Kwon and Lee [29] propose a tracking algorithm based on the Wang-Landau Monte Carlo sampling method which efficiently deals with the abrupt motion. This method utilizes the likelihood term and the density of states term to handle the smooth motions and abrupt motions, respectively. To address the abrupt motion problem, Zhou et al. [30] combine a density-gird-based predictive model with the stochastic approximation Monte Carlo sampling to give a proposal adaptation scheme. This method is effective but a large number of particles are needed.

The search strategy is widely used in the tracking process. There are two main reasons. First, a tracker is likely to fail when the distance of the target position is too large between two successive frames (e.g., abrupt motion and fast movement) or severe occlusion occurs. Therefore, target location plays a key role in recovering tracking. Second, the proposal scheme predicting the candidate offspring particles is not entirely convincing which clearly implies the need for a search strategy to give an alternative candidate. The conventional search strategy utilizes motion detection method to locate the target, such as frame difference, background subtraction, and optical flow. However, if the background is cluttered, the frame difference method and the background subtraction

method are likely to fail to detect the target successfully. The optical flow method can deal with the changing background but it is sensitive to the illumination change. Another search strategy utilizes a sliding window to traverse the entire map or the local region surrounding the old object location to locate the target. Zhang et al. [10] deal with the abrupt motion using the brute force search strategy and good results are obtained. Vermaak et al. [31] propose a motion-based object detection method. Based on searching the horizontal and vertical projections of the frame-difference measurements, this method utilizes the motion measurement to determine a rectangular region within which the object is likely to lie. Su et al. [5] detect the target region from salient regions, which are obtained in the saliency map of current frame. This method can also cover the abrupt motion problem. Once the target is lost, the algorithm recovers the target via the saliency map search strategy. Rui et al. [25] identify the target location in the current frame by minimizing the similarity measure function. This method determines a possible direction of the movement of the target center and searches the local optimal position in the neighborhood of the location of the search window in the previous frame. The specific strategies go on until they cover the whole image or the similarity of the candidate regions with the template feature subspaces meets the requirements.

### 3. Preliminary Theories

**3.1. Bayesian Filtering.** The object tracking problem is usually formulated as Bayesian filtering [32], which is a probabilistic approach and provides a reasoned solution framework to estimate states of the dynamic system. Bayesian filtering consists of two phases: state prediction and state update. The state prediction phase utilizes the system model to predict the posterior density. And the state update phase utilizes the latest observation to amend the posterior density. Let  $\mathbf{x}_t$  denote the object state at time  $t$ . Given observation sequences  $\mathbf{z}_{1:t} = \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_t$  up to time  $t$ , Bayesian filter can estimate the target state  $\mathbf{x}_t$ . According to Bayes rule, the state prediction and state update can be formulated recursively as follows:

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}) d\mathbf{x}_{t-1}, \quad (1)$$

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{1:t-1})}{p(\mathbf{z}_t | \mathbf{z}_{1:t-1})}, \quad (2)$$

where  $p(\mathbf{x}_t | \mathbf{z}_{1:t})$  denotes the posterior density at time  $t$ ,  $p(\mathbf{x}_t | \mathbf{x}_{t-1})$  indicates the prior model,  $p(\mathbf{z}_t | \mathbf{x}_t)$  represents the observation model, and  $p(\mathbf{z}_t | \mathbf{z}_{1:t-1})$  is a normalization constant. However, for nonlinear and non-Gaussian systems, integral in (1) is often intractable.

**3.2. Particle Filtering.** Based on Monte Carlo integration method, particle filtering utilizes sequential importance resampling (SIR) method to solve the integral problem in Bayesian filtering. The core idea of SIR particle filtering is to recursively approximate the posterior probability distribution via a finite weighted sample set  $S = \{\mathbf{x}_t^i, \omega_t^i\}_{i=1, \dots, N}$  at each

time step. Each sample represents one hypothetical state of the target, with a corresponding discrete sampling probability  $\omega_t^i$ , which satisfies  $\sum_{i=1}^N \omega_t^i = 1$ . And the posterior density is approximated by

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}) = \sum_{i=1}^N \omega_t^i \delta(\mathbf{x}_t - \mathbf{x}_t^i), \quad (3)$$

$$\omega_t^i \propto \frac{p(\mathbf{z}_t | \mathbf{x}_t^i) p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i, \mathbf{z}_t)},$$

where  $\delta(\cdot)$  is Dirac function. Let the proposal distribution  $q(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i, \mathbf{z}_t) = p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i)$ , and then we get that  $\omega_t^i$  is in proportion to the observation model  $p(\mathbf{z}_t^i | \mathbf{x}_t^i)$ . In practice, the positions of the offspring samples generated via the transition model play a vital role in improving the sampling efficiency.

**3.3. Insect Vision Inspired Motion Detector.** To better simulate the object motion, a more accurate transition model is clearly needed. However, it is often hard to get detailed motion information of the object. Insect vision inspired motion detector is one of the theoretical methods to address this problem. Flying insects, despite their relatively coarse vision and tinny nervous systems, are exquisitely sensitive to motion. Eichner et al. [33] conclude that the insect vision inspired motion detector can respond to the motion pattern of the target including the direction and the relative speed. The theory states that the light distribution of a moving image is the function of space frequency and time frequency. Suppose that the coordinate system of the motion pattern is consistent with the axis of the detector, then the input of the one-dimension detector is  $F = F[x + s(t)]$ , and the output is

$$D(x, t) = F(x + s(t)) \cdot F[x + \Delta x + s(t - \varepsilon)] - F[x + \Delta x + s(t)] \cdot F[x + s(t - \varepsilon)], \quad (4)$$

where  $x$  represents the one-dimensional coordinate,  $s(t)$  is the displacement function,  $\Delta x$  and  $\varepsilon$  specify the distance of the two photoreceptors and the delay time constant of the detector, respectively, and  $D(x, t)$  determines the direction of the moving image. Since  $\Delta x$  and  $\varepsilon$  are small enough, we differentiate (4) as follows:

$$dD(x, t) = -\varepsilon \cdot \frac{ds(t)}{dt} \cdot S(x, t) dx, \quad (5)$$

where  $ds(t)/dt$  gives the image instant velocity and  $S(x, t) = \{\partial F[x + s(t)]/\partial x\}^2 - F[x + s(t)] \cdot \{\partial^2 F[x + s(t)]/\partial x^2\}$  gives the texture information response of the moving target. More detailed mathematical material can refer to [33, 34]. This strong theoretical support motivates us to utilize the relative accurate motion pattern information to establish a more adaptive motion transition model to provide a prediction of possible object movements. In the proposed algorithm, the improved transition model is quite amenable to the changing speeds of moving object, that is, to both smooth motion and abrupt motion or fast movement scenarios.

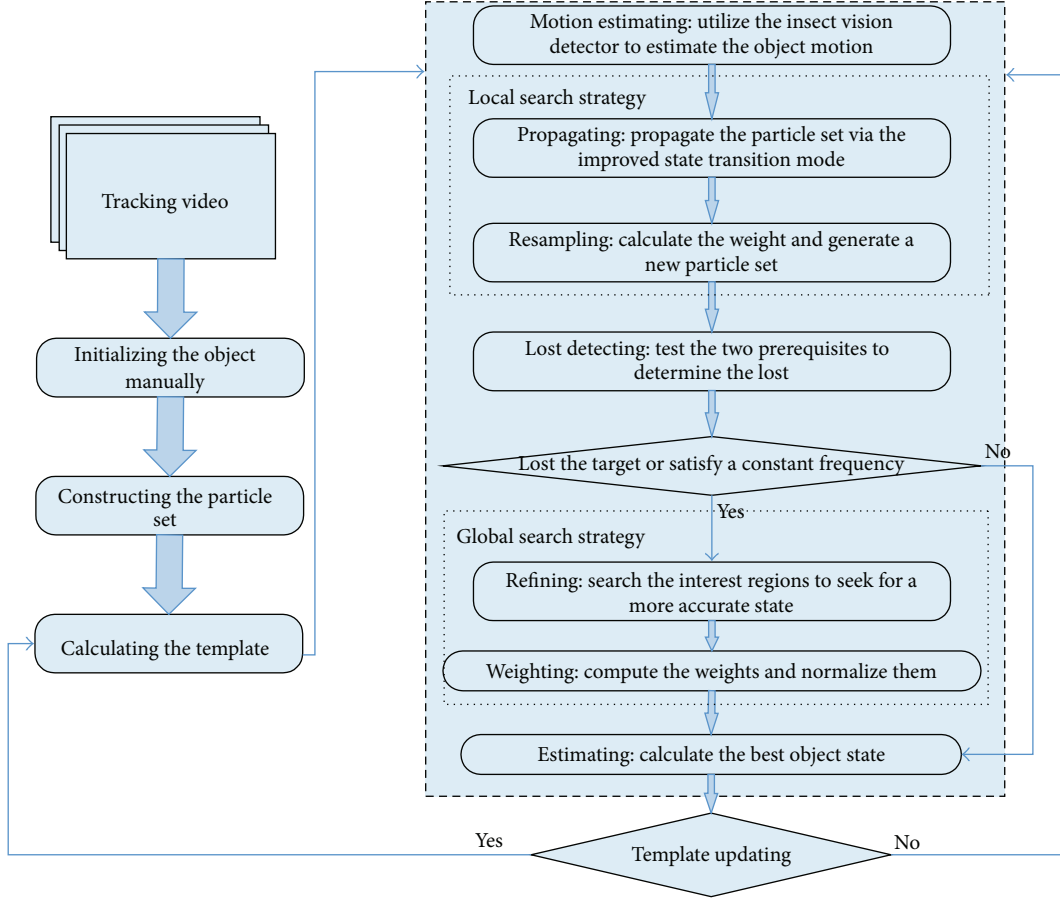


FIGURE 1: Flowchart of the proposed tracking method.

## 4. Proposed Algorithm

In this section, we present the proposed insect vision embedded particle filter tracking algorithm in detail. To address the abrupt motion and fast movement problem, in the particle propagation process, we seek for a more accurate state transition model to refine the position of the offspring particles. Though candidate image regions, predicted by the local search strategy using the improved state transition model, are reliable in most cases, the result is not entirely convincing. Thus, for every  $\eta$  (a constant) frame, a global search strategy is adopted to verify if a more similar image region exists. Besides, a new criterion is investigated to judge whether the target is lost. The global search strategy is immediately utilized to recover the target once the criterion is satisfied. To adapt the changing of the appearance of the target, an online template update method is detailed. The flowchart is shown in Figure 1.

**4.1. Model Representation.** Based on estimating the distribution of the target state, it is particularly critical to construct a good model representation to simulate the object motion in our tracking algorithm. Each sample which is a subimage region centered around the object represents a hypothesis state of the target and is defined by  $\mathbf{x} = \{x, y, u, v, s, \alpha\}$

which consist of the 2D image position  $(x, y)$ , the velocity components  $(u, v)$ , the scale of the object  $s$ , and the rotation of the object. And the following constraints should be satisfied:

- (1) the position  $(x, y)$ ,  $0 < x < W$ ,  $0 < y < H$ , where  $W$  and  $H$  are the width and height of the frame, respectively,
- (2) the scale  $s$  which is initialized to 1.0 and varies with the changing of the objects size; that is,

$$s_t = s_{t-1} + q(s), \quad (6)$$

where  $q(s)$  is drawn from zero-mean normal distribution and the variance  $\sigma_s^2$  is set to 0.001,

- (3) the rotation  $\alpha$ ,  $0^\circ \leq \alpha \leq 360^\circ$ . The  $\alpha$  is set to  $0^\circ$  in our approach, for the rotation of the object is not considered in current research.

To propagate the particles, we use the following adaptive state transition model:

$$(x, y)_t = (x, y)_{t-1} + Q(u, v)_{t-1} \cdot \Delta t + q(x, y), \quad (7)$$

where  $q(x, y)$  denotes a zero-mean Gaussian noise and  $\Delta t$  is dependent on the frame-rate of the sequence. And  $Q(u, v)_{t-1}$

is the velocity calculated by our motion estimation method and is discussed below. One should note that we do not assume that the motion is smooth which is commonly adopted by other algorithms. Instead, we directly use the velocity to amend the position. We utilize more accurate motion information to refine a more reliable position in the next frame. The velocity varies in every frame, and hence our model can be easily interpreted as a time-varying state model. The motion velocity is estimated as described below.

**4.2. Motion Estimation.** In order to determine the velocity  $Q(u, v)$ , that is, the direction and the shift of the motion from one frame to the next, we solve the motion estimation problem in the view of insect vision inspired motion detector. We deduce a motion estimation method to get the motion information in a video sequence by applying the computational structure of the biological motion-detection system. For a certain frame of the video, we divide it into orthogonal grids called photoreceptors. The size of the photoreceptors decides the resolution of the detector. To focus on the subimage regions that we are interested in and ignore the unimportant ones, an  $n$ -rank-resolution strategy is adopted. Specifically, the closer to the center of the target, the higher the resolution. In our algorithm, the rank  $n$  is set to 3 and the side length of the photoreceptors denoted by  $l_p$  in the  $p$ th rank is

$$l_p = 2^p + 1 \quad \forall p = 1, \dots, n. \quad (8)$$

The first rank covers an image region which is a square area centered at the location of the target and the side length of the region represented by  $s_1$  is calculated by

$$s_1 = \lambda_1 [(x_t)]^{1/2}, \quad (9)$$

where  $(x_t)$  represents the area size of the target area (a rectangle subimage region). The parameter  $\lambda_1$  is a constant parameter, which determines the size of the first rank region, and is set to 1.5 in all experiments. Thus, the number of the photoreceptors in the first rank is  $N_1 = (s_1/l_1)^2$ .

Similarly, a bigger constant parameter  $\lambda_2 = 2.5$  defines a larger region which is divided into two parts, the central part for the first rank and the surrounding part for the second rank; that is, the second one is purely the surrounding region of the first one but does not embody it. The rest of the entire frame except the first rank and the second rank belongs to the third rank; that is, for a certain pixel at location  $(i, j)$ , we have

$$\varphi(i, j) = \begin{cases} 1 & \text{if } 0 \leq d_v \leq \frac{s_1}{2}, 0 \leq d_h \leq \frac{s_1}{2}, \\ 2 & \text{if } \frac{s_1}{2} < d_v \leq \frac{s_2}{2}, \frac{s_1}{2} < d_h \leq \frac{s_2}{2}, \\ 3 & \text{others,} \end{cases} \quad (10)$$

where  $\varphi(i, j)$  denotes the rank that the pixel belongs to,  $d_v$  and  $d_h$  are the vertical and horizontal distances between the location of the pixel and the location of the target, respectively, and  $s_1$  and  $s_2$  represent the side lengths of the regions defined by  $\lambda_1$  and  $\lambda_2$ , respectively. This  $n$ -rank resolution strategy is very similar to the process of the insect observing a target and can also reduce the computational cost.

Now that a certain frame is divided into finite photoreceptors, many detector units are constructed. A single 2D detector unit is composed of three adjacent photoreceptors, three direct current filters, three low-pass filters, four multiplier units, and two adders. The three photoreceptors collect visual information from central, vertical, and horizontal directions, respectively. First, for a certain photoreceptor at location  $(r, c)$  in the  $t$ th frame, let  $\bar{I}(t, r, c)$  denote the average pixel intensities within the photoreceptor. Next, we apply the direct current filter and the low-pass filter to obtain the direct current components  $\bar{I}_{DC}(t, r, c)$  and the alternating current components  $\bar{I}_{AC}(t, r, c)$ , respectively:

$$\begin{aligned} \bar{I}_{DC}(t, r, c) &= \frac{1}{M} \sum_{l=1}^M \bar{I}(t-l), \\ \bar{I}_{AC}(t, r, c) &= \bar{I}(t, r, c) - \bar{I}_{DC}(t, r, c), \end{aligned} \quad (11)$$

where  $M$  is a time constant in terms of frames and the direct current component is the average of  $M$  frames from the  $(t-M)$ th frame to the  $t$ th frame. Now that every photoreceptor corresponds to its own alternating current component, we can calculate the output of the 2D detector unit in vertical and horizontal directions, respectively. Let  $C_1(t, r, c) = \bar{I}_{AC}(t-\tau, r, c)$ ,  $H_1(t, r, c) = \bar{I}_{AC}(t-\tau, r, c+1)$ ,  $C_2(t, r, c) = \bar{I}_{AC}(t, r, c)$ , and  $H_2(t, r, c) = \bar{I}_{AC}(t, r, c+1)$ , where  $\tau$  is the delay time parameter in terms of frames. Then we have

$$\begin{aligned} R_H(t, r, c) &= C_1(t, r, c) \cdot H_2(t, r, c) \\ &\quad - C_2(t, r, c) \cdot H_1(t, r, c), \end{aligned} \quad (12)$$

where  $R_H(t, r, c)$  denotes the horizontal output of the detector unit and the vertical output can be calculated similarly. All the 2D detector units have no interdependence and can be calculated simultaneously in matrix form.

The outputs of all the  $N$  detector units construct a velocity distribution pool  $G = \{g_i \mid i = 1, \dots, N\}$ , where  $g_i$  denotes the output of the  $i$ th detector unit. We then normalize all the outputs into a canonical range:  $[-\kappa, \kappa]$ . The measure of each output is computed as follows:

$$q^* = \kappa \cdot \text{sgn}(q) \cdot \frac{|q| - \min(\{|g_i\}_{i=1}^N)}{\max(\{|g_i\}_{i=1}^N) - \min(\{|g_i\}_{i=1}^N)} \quad (13)$$

$$\forall q \in G,$$

where  $q^*$  is the canonical value of the output  $q$  and  $\text{sgn}(\cdot)$  is a sign function.  $\kappa$  is a constant parameter and is set between 10 and 30 in our experiments. Thereby, for certain detector units, we obtain the canonical value  $Q(u, v)$ , where  $u$  and  $v$  represent the horizontal and vertical velocities, respectively, and can be computed by (13). We can now substitute  $Q(u, v)$  into (7), yielding  $(x, y)_t$ .

The adaptive improved transition model is simple yet effective. We test our state transition model on a large number of publicly available video sequences and find that it is quite amenable to various motions, such as smooth motion, abrupt motion, and fast movement.

**4.3. The Joint Local and Global Search Strategies for Visual Tracking.** The aforementioned state transition model is used for local search and it works in most cases but not entirely convincing. When severe or long term occlusion occurs, most trackers only based on local search are likely to lose the target. What is worse is that local search tends to lead the tracker falling into local optimum. To address those problems, a global search strategy is adopted. Unlike conventional global search strategy which utilizes a sliding window to search every single pixel of the entire frame, we are only interested in the specific subimage regions where the responses of the insect vision inspired detector are not zero. We calculate the similarities of the regions of interest and the template to determine the likeness of the regions.

In our algorithm, the local search and global search are jointly used to seek for a more accurate candidate image region. First, for every coming frame, we search the object location based on the previous object location by propagating the samples via the improved state transition model. There exist two situations where we apply the global search strategy. For the first situation, the global search is carried out according to a constant frequency, for example, for every  $\eta$  frame (the  $\eta$  is set as 10 in all the experiments). The other is when the tracker loses the target; that is, when the local search cannot find a convincing candidate image region or causes local optimum, the global search is used to recover the target.

Now, we give a detailed discussion about the method to judge whether a tracker loses the target. For state  $x_t$  at time  $t$ , the weight responding to each sample is normalized to the range of  $[0, 1]$ . And the sample with a bigger weight is assigned with a higher confidence of belonging to the target. We set a threshold  $\theta_o$  to count the number of the valid samples. In other words, if the weight of a sample is bigger than  $\theta_o$ , the sample is valid, otherwise, invalid. Then we have

$$\Phi = \sum_{i=1}^{N_o} \frac{1}{2} (1 + \text{sgn}(\omega_i - \theta_o)), \quad (14)$$

where  $\omega_i$  is the weight of the  $i$ th sample,  $N_o$  is the total number of the samples, and  $\Phi$  denotes the number of the valid samples. When the following two prerequisites are satisfied, the tracker loses the target:

- (1)  $\Phi < \gamma \cdot N_o$ , where  $\gamma$  denotes the lost factor which is set to 0.15 in this paper; that is, the number of the valid samples is less than  $\gamma \cdot N_o$ ,
- (2) the prerequisite one occurs for  $T$  frames consecutively where  $T = 5$  in this paper.

The global search strategy can definitely reduce the probability of falling into a local optimum by sampling in the interest global state space in a constant frequency. Meanwhile, it can effectively recover our tracker from drifts or heavy occlusions. The lost-judgment prerequisites detect heavy or full occlusions and offer signal to the global search strategy. Therefore, the joint local search and global search strategies seek a more reliable candidate image region making the tracker more convincing.

**4.4. Incremental Learning Based Online Template Update.** Under challenging situations such as drastic luminance change and severe occlusion, the appearance of a target changes over time. Tracking with a fixed template is not likely to capture the appearance variations for a long duration due to the low importance weight of the samples. So, it is important to update the template online to enhance the adaptivity of the tracker. By doing so, we employ an incremental-learning-based online template update method. In the scheme, a sequence of  $H$  max-weighted samples is stored during the tracking process. For every  $U$  frame, we choose the oldest sample from the sequence set and replace it with the latest tracking result of the object. Note that we reserve the sample initialized manually in the first frame. Once the new target location is determined, the template is updated as follows:

$$\begin{aligned} \Omega^{(k)} = & \mu \cdot \Omega^{(0)} + (1 - \mu) \\ & \cdot (\Omega^{(k-1)} - \text{ST}_{\text{oldest}} + \text{ST}_{\text{latest}}), \end{aligned} \quad (15)$$

where  $\Omega^{(k)}$  is the  $k$ th template and  $\Omega^{(0)}$  is the template initialized manually in the first frame,  $\text{ST}_{\text{oldest}}$  and  $\text{ST}_{\text{latest}}$  denote the oldest sample and the latest sample (with the highest weight), respectively, and  $\mu$  is the learning rate. In our experiment, the lengths of sequence  $H$  and the spacing interval  $U$  are set as 5 and 3, respectively, and the learning rate  $\mu$  is set as 0.3.

The incremental-learning-based online template update method ensures that the most recent object appearances are reflected in the template, and the manually initial template is reserved. In this manner, our tracker will not delete all information of the target and keep on learning the appearance changes at the same time, which avoids the drift problem and improves tracking stability.

It is interesting to note that the approach proposed in [34] also uses the latest tracking result to update the template. Our update method is different from [34] in that (1) we store a max-weighted samples sequence for the updating while they store a large number of regions (e.g., 1000 regions) for the updating, (2) we reserve the manually initial sample and utilize a learning rate while they utilize an update rate to decide the number of regions to be updated, and (3) we use the latest tracking result to replace the oldest sample for every  $U$  frame while they set a prerequisite to determine when to update in every frame.

The main steps of our algorithm are summarized in Algorithm 1.

## 5. Experiments

In this section, to demonstrate the performance of the proposed tracking algorithm, extensive experiments are performed on challenging sequences from the existing works. All the sequences are publicly available and the details are listed in Table 1. The challenges include abrupt motion, fast movement, part or full occlusion, illumination changes, and pose variation. In addition, we test other 10 state-of-the-art trackers (the implementations are provided by the authors) on the

**Input:** image sequence, particle set  $\{\mathbf{x}_{t-1}^{(i)}, \omega_{t-1}^{(i)}\}_{i=1}^{N_o}$   
**Output:** particle set  $\{\mathbf{x}_t^{(i)}, \omega_t^{(i)}\}_{i=1}^{N_o}$  and the object state  $\mathbf{x}_t^*$   
**Initialization:**  
(1) Initializing the tracked object manually.  
(2) Constructing the particle set  $\{\mathbf{x}_0^{(i)}, \omega_0^{(i)}\}_{i=1}^{N_o}$ .  
(3) Calculating the template  $\Omega^{(0)}$ .  
**Object tracking:**  
(1) **for**  $k$  from 1 to the last frame **do**  
(2) Estimate the object motion utilizing the proposed insect vision inspired detector and obtain the response  $Q_t$ .  
(3) **Local search strategy:**  
(4) Propagating: According to the proposed state transition model (7), propagate each particle state  $\mathbf{x}_{t-1}^{(i)}$  to get a new state  $\mathbf{x}_t^{(i)}$  for time  $t$ .  
(5) Re-sampling: Based on the state  $\mathbf{x}_t^{(i)}$  and the template, compute the weight  $\omega_t^{(i)}$  of each propagated particle at time  $t$ , and then normalize the particles weight. Generate a new particle set  $\{\mathbf{x}_t^{(i)}, \omega_t^{(i)}\}$ .  
(6) **Lost detection:** (to address severe occlusion and drifts problems)  
(7) Count the number of valid samples and test the Two-Prerequisites. If the Two-Prerequisites are satisfied, let  $LF = 1$ , otherwise,  $LF = 0$ . ( $LF$  is the label to mark the lost.)  
(8) **Global search strategy:**  
(9) **if**  $LF = 1$  (the tracker loses the object) **or** for every  $\eta$  frames **then**  
(10) Refining: Search the interest image region where  $Q_t \neq 0$  to seek for a more accurate state  $\hat{\mathbf{x}}_t^{(i)}$ .  
(11) Weighting: Based on the state  $\hat{\mathbf{x}}_t^{(i)}$  and the template, compute the weight  $\hat{\omega}_t^{(i)}$  of each particle at time  $t$  and normalize it.  
(12) **end if**  
(13) **Estimating:**  
(14) Calculate the object position at time  $t$  and get the best state:  $\mathbf{x}_t^* = \mathbf{x}_t^{i_{\max}}$ , where  $i_{\max} = \operatorname{argmax}_{i \in (0, N_o)}(\omega_t^i)$ .  
(15) **Online template update:**  
(16) Update the template according to (15) for every  $U$  frames.  
(17) **end for**

ALGORITHM 1: Insect vision embedded particle filter tracking.

same sequences for comparison. The 10 evaluated trackers are the Markov Chain Monte Carlo (MCMC) tracker [35], the adaptive MCMC tracker [36], the Wang-Landau Monte Carlo (WLMC) tracker [29], TLD tracker [37], sparsity-based collaborative model (SCM) tracker [38], the fragment tracker (Frag) [39], distribution field (DF) tracker [40], compressive tracker (CT) [10], circulant structure tracker (CST) [41], and adaptive structural local sparse appearance (ASLA) tracker [42]. For fair comparisons, all the evaluated trackers start with the same initial position of the videos and the quantitative results are obtained over 10 runs. For trackers that involve particle filter, we use the same number of samples.

**5.1. Quantitative Evaluation.** Two evaluation criteria are used to evaluate the proposed algorithm in our experiments. The first criterion is the center location error which is defined as follows:

$$C_{\text{error}} = \sqrt{(C_G - C_T)^2}, \quad (16)$$

where  $C_G$  and  $C_T$  denote the center locations of the manually labeled ground truth and the tracked object, respectively. The other criterion is the success rate. Let  $\text{area}(R_G \cap R_T) / \text{area}(R_G \cup R_T)$  denote the overlap ratio, where  $R_G$  is the ground truth

bounding box and  $R_T$  is the tracking bounding box. If the overlap ratio is larger than 0.5 in one frame, the tracking result is considered as a success. The success rate indicates the ratio between the number of successfully tracked frames and the number of total frames.

Table 2 and Figure 2 show the tracking center location errors of all the methods. Table 3 and Figure 3 show the tracking results in terms of success rate. We note that, in the bird sequence, the target disappears for dozens of frames where we cannot determine the ground truth of the target, so we ignore these frames when we compute both evaluation criteria. And the overlap rate plots frame-by-frame shown in Figure 3(b) indicate that when a tracker loses the target, few methods can recover from failure except ours. Besides, the TLD tracker does not report tracking result (or bounding box) when the drift problem occurs and the target object is redetected. Thus, we only report the center location errors for the sequences in which the TLD method does not lose track of target objects. The proposed algorithm achieves the best or the second best performance in most of the sequences against the state-of-the-art algorithms based on both center location error and success rate. The qualitative evaluation of the tracking results of different trackers is detailed below.

TABLE 1: The details of video sequences.

Sequence	Frames	Initial position				Challenges
Biker [10, 35]	1-180	250	110	45	75	Abrupt motion
Bird [9]	1-408	443	94	37	27	Abrupt motion, disappear, and reappear
Bolt [9, 10, 35]	1-350	336	165	25	60	Fast movement
Cartoon	1-200	83	134	56	56	Significant pose variation
Coke [8]	1-292	148	82	20	35	Drastic illumination variation and occlusion
Dragon baby [35]	1-112	167	74	40	70	Abrupt motion, pose variation
Girl move [5, 9]	1-1500	303	161	28	113	Significant occlusion
Indian dancer	1-150	157	61	20	23	Pose variation and abrupt motion
Lemming [9]	1-681	40	200	54	92	Abrupt motion and significant occlusion
Liquor [9]	1-1553	257	226	67	125	Occlusion and fast movement
Panda [10, 37]	1-241	257	171	45	55	Fast movement, occlusion, and pose variation
Stennis [5]	1-55	133	39	23	23	Abrupt motion and fast movement
Surfer [35]	1-400	282	165	35	45	Significant and long duration occlusion
Sylvester [2, 10]	1-1344	125	56	43	50	Drastic illumination and pose variations
Woman [38, 41]	1-553	208	125	27	75	Occlusion and significant appearance changes

TABLE 2: Center location error (CLE) (in pixels).

Sequence	Ours	SCM	TLD	MCMC	AMCMC	AWLMC	Frag	DF	CT	CST	ASLA
Biker	<b>7</b>	41	—	<b>18</b>	20	20	88	99	27	<b>12</b>	65
Bird	<b>11</b>	<b>38</b>	—	40	122	222	156	146	<b>33</b>	368	215
Bolt	<b>4</b>	512	—	<b>16</b>	<b>16</b>	77	129	207	327	409	158
Cartoon	<b>20</b>	68	—	21	<b>15</b>	<b>14</b>	71	126	73	96	68
Coke	<b>9</b>	56	<b>7</b>	55	55	77	42	32	<b>16</b>	60	38
Dragon baby	<b>17</b>	52	—	65	<b>39</b>	<b>18</b>	43	67	63	170	45
Girl move	<b>7</b>	282	—	49	<b>31</b>	<b>16</b>	309	242	154	884	158
Indian dancer	<b>4</b>	14	<b>6</b>	<b>5</b>	9	25	15	31	17	12	14
Lemming	<b>9</b>	44	12	7	7	15	168	99	24	30	114
Liquor	<b>10</b>	<b>31</b>	<b>16</b>	143	93	32	138	229	184	165	122
Panda	<b>6</b>	<b>18</b>	—	131	136	20	90	96	88	119	<b>13</b>
Stennis	<b>3</b>	13	<b>3</b>	15	88	21	28	45	<b>7</b>	97	26
Surfer	<b>8</b>	<b>13</b>	<b>11</b>	19	19	43	27	94	43	290	85
Sylvester	11	18	13	<b>9</b>	<b>9</b>	11	14	53	22	<b>10</b>	11
Woman	<b>3</b>	10	37	<b>5</b>	<b>5</b>	39	116	99	124	8	7
Average CLE	<b>8</b>	80	<b>13</b>	<b>39</b>	44	43	95	111	80	182	75

Bold fonts indicate the top three best performances. The total number of evaluated frames is 8019.

## 5.2. Qualitative Evaluation

**5.2.1. Abrupt Motion and Fast Movement.** For the biker sequence shown in Figure 4(a), the target object undergoes abrupt movements with 180-degree out-of-plane rotation when the biker jumps from one side of the railway to the other. The AMCMC, MCMC, AWLMC, and CST methods and our method perform well on this sequence. The MCMC method replaces the traditional importance sampling step in the particle filter with a Markov chain Monte Carlo (MCMC) sampling to obtain a more efficient filter and the AMCMC method automatically tunes the Markov chain parameters during a run. The WLMC method utilizes the density of states (DoS) which is estimated by the Wang-Landau algorithm to capture abrupt motions, so these three methods work

well during tracking. The CST method performs well due to the use of circulant structure of tracking-by-detection with kernels mechanism. In the bird 1 sequence shown in Figure 4(b), the target moves flexibly. Worse yet, it disappears from frame 130 and reappears in frame 190. Our method can quickly locate the target and recover tracking when the target appears again due to the use of the global search mechanism for target detection, while most other methods work well before the long duration occlusion occurs but lost the target and cannot relocate after occlusion because they have no mechanism to recover tracking. Some methods are disturbed by the similar bird flying aside. In Figure 4(c), the target object moves fast. Only MCMC and AMCMC methods and our method perform well on this sequence. The WLMC method cannot handle the sequence since it



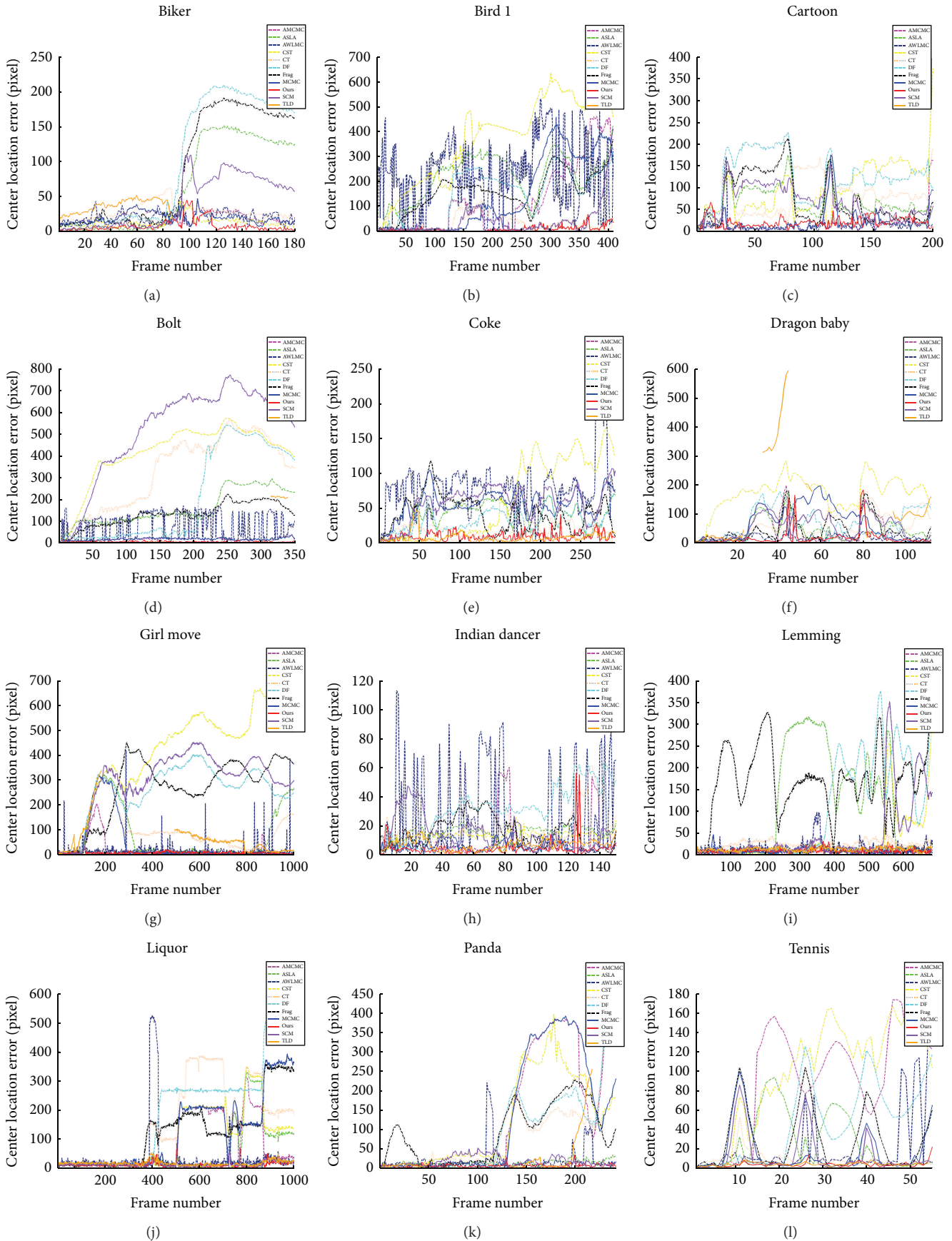


FIGURE 2: Continued.

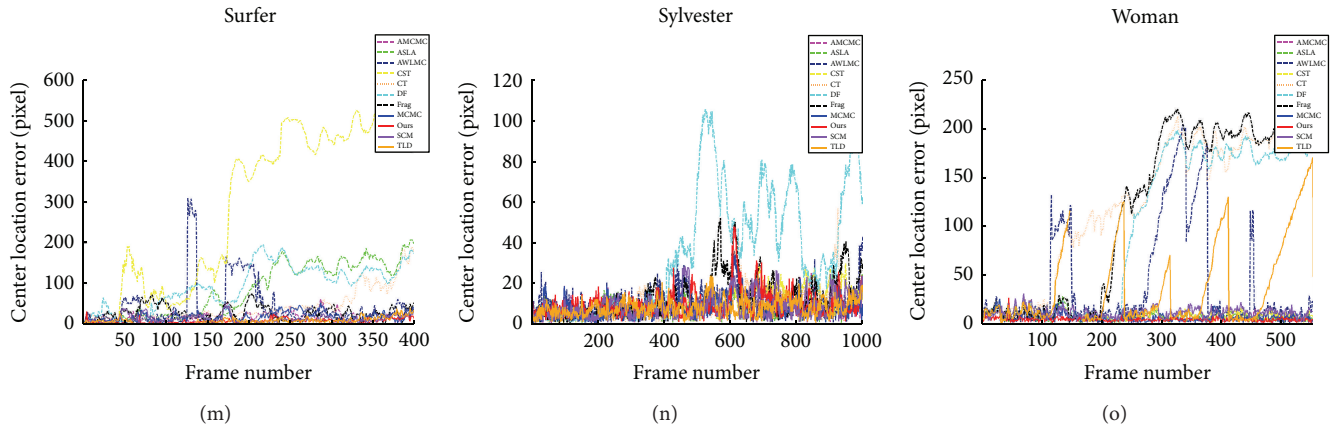


FIGURE 2: Center location error plots for all test sequences (the red curve denotes the proposed method). For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.

TABLE 3: Success rate (SR) (%).

Sequence	Ours	SCM	TLD	MCMC	AMCMC	AWLMC	Frag	DF	CT	CST	ASLA
Biker	<b>90.0</b>	49.4	8.9	<b>58.9</b>	41.7	32.8	29.4	27.2	9.4	<b>55.0</b>	47.8
Bird	<b>46.8</b>	<b>17.4</b>	—	15.7	<b>19.8</b>	3.9	3.9	15.4	10.8	2.2	2.5
Bolt	<b>84.6</b>	3.7	1.1	<b>32</b>	<b>38.3</b>	7.4	2.6	2.6	0.2	0.9	1.1
Cartoon	44	3.0	0.5	<b>51</b>	<b>62.0</b>	<b>73.5</b>	5.5	3.5	6.0	2.0	2.5
Coke	<b>48.3</b>	3.4	<b>56.5</b>	4.1	4.1	1.7	2.1	5.5	<b>34.9</b>	21.6	4.1
Dragon baby	<b>76.8</b>	27.7	13.4	<b>35.7</b>	<b>47.3</b>	18.8	28.6	17.8	9.8	4.5	30.3
Girl move	<b>80.3</b>	6.2	16.6	<b>58.3</b>	<b>59.5</b>	47.1	5.3	5.7	10.9	6.9	16.1
Indian dancer	<b>84.0</b>	3.3	39.3	<b>44.7</b>	<b>48.7</b>	32.7	28.0	12.0	11.3	7.3	8.7
Lemming	<b>89.0</b>	56.0	73.7	<b>93.4</b>	<b>93.5</b>	71.7	5.6	48.5	47.7	72.7	45.4
Liquor	<b>91.0</b>	68.8	<b>88.8</b>	30.7	42.9	<b>71.4</b>	23.3	26.0	23.4	30.1	43.8
Panda	<b>92.5</b>	<b>44.4</b>	41.5	24.9	25.3	47.3	19.5	44.4	25.3	19.9	<b>48.5</b>
Tennis	<b>87.3</b>	<b>67.3</b>	60	<b>63.6</b>	21.8	40.0	36.4	32.7	45.5	16.4	43.6
Surfer	<b>84.5</b>	<b>41.0</b>	<b>45.3</b>	24.3	18.8	14.5	13.0	8.5	10.0	11.0	14.0
Sylvester	63.2	49.1	60.7	<b>68.5</b>	<b>68.5</b>	60.2	55.4	27.7	56.9	60.0	<b>69.4</b>
Woman	64.7	51.7	32.2	<b>91.5</b>	<b>92.7</b>	39.2	24.6	29.5	11.9	<b>74.1</b>	5.2
Average SR	<b>75.1</b>	32.8	38.4	<b>46.4</b>	<b>45.6</b>	37.5	18.8	20.5	20.9	25.6	25.5

Bold fonts indicate the top three best performances. The total number of evaluated frames is 8019.

assumes that the object can go anywhere in a scene even at one proposal step. This mechanism avoids the local-trap problem and is effective to abrupt motion, but it leads the method to be susceptible to interference, for example, the athlete wearing the similar cloth with Bolt. In the dragon baby sequence (Figure 4(d)), the target moves abruptly in action scene. Despite all the abrupt movements and 360-degree out-of-plane rotation, our tracker and the AWLMC method are able to track the baby well. For the tennis sequence shown in Figure 4(e), the ping pong moves fast and changes its motion direction suddenly when it hits the racket. The feature of the target is not significant since it is small. Only the WLMC, SCM, and TLD methods and our method perform well on this sequence. The TLD approach works well because it maintains a detector which uses Haar-like features during tracking. The proposed algorithm handles abrupt motion and fast movement well as the adopted state transition model and

the joint local and global searches mechanism well describe the characteristic of the motion in a real scene.

**5.2.2. Occlusion and Drift.** The target object in the girl move sequence in Figure 5(a) undergoes heavy occlusion (from frame 108 to frame 123 and from frame 1384 to frame 1400). Only the AWLMC method and our method perform well after the severe occlusion. In the lemming sequence (Figure 5(b)), the target undergoes heavy occlusion. Overall, the AMCMC, MCMC, AWLMC, CST, and TLD methods and our method perform well on this sequence. The SCM algorithm uses a sparsity-based discriminative classifier (SDC) and a sparsity-based generative model (SGM), so it works well in most frames. But the heavy occlusion can damage the SDC and SGM mechanisms seriously and hence this method is less effective in handling severe occlusion. Due to the same reasons, the SCM method fails to track the target object

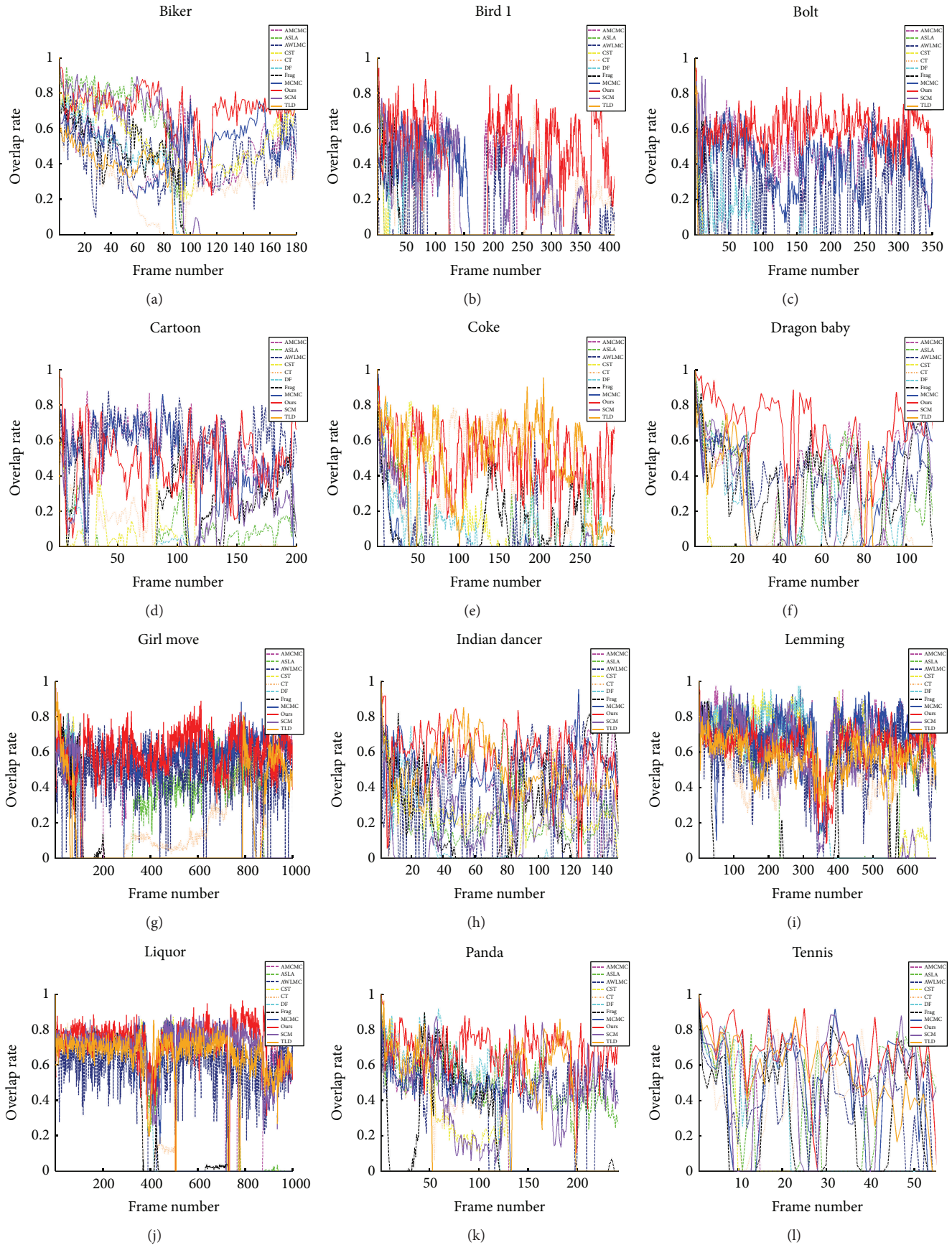


FIGURE 3: Continued.

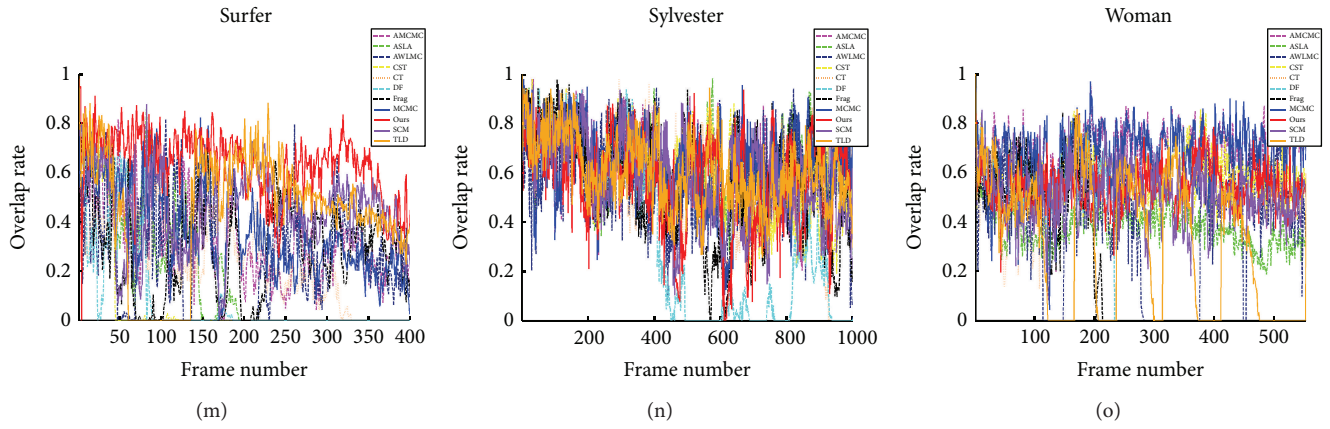
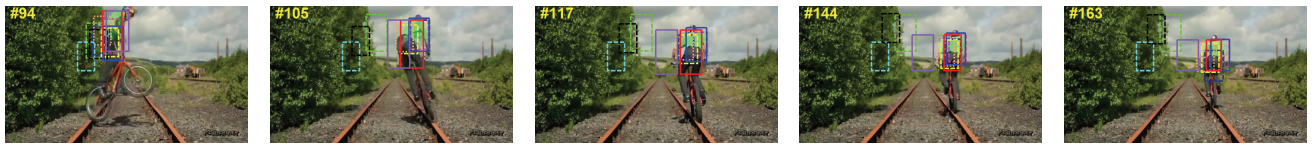


FIGURE 3: Overlap rate plots for all test sequences (the red curve denotes the proposed method). For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.



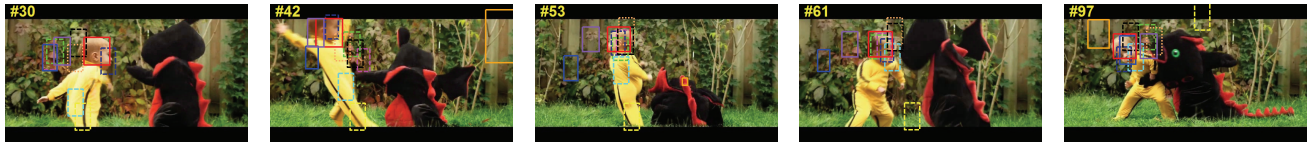
(a) Tracking results of the biker sequence



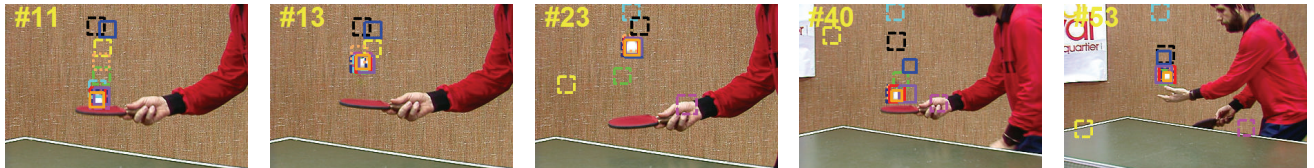
(b) Tracking results of the bird 1 sequence



(c) Tracking results of the Bolt sequence



(d) Tracking results of the dragon baby sequence



(e) Tracking results of the tennis sequence

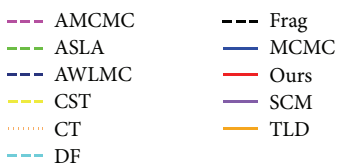


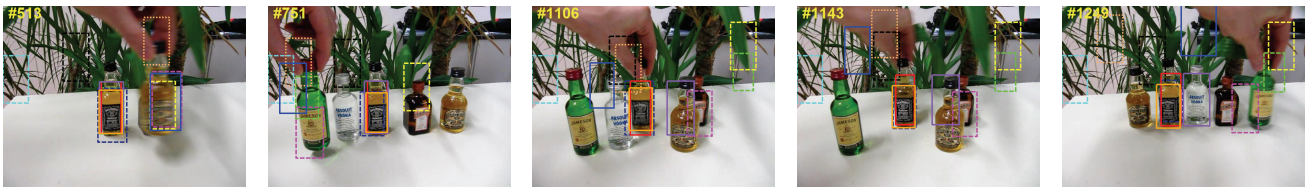
FIGURE 4: Screenshots of some sample tracking results when there are abrupt motion and fast movement.



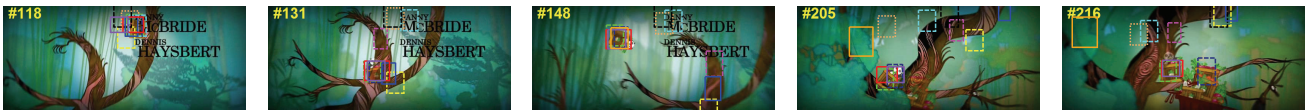
(a) Tracking results of the girl sequence



(b) Tracking results of the lemming sequence



(c) Tracking results of the liquor sequence



(d) Tracking results of the panda sequence



- |           |          |
|-----------|----------|
| --- AMCMC | --- Frag |
| --- ASLA  | --- MCMC |
| --- AWLMC | --- Ours |
| --- CST   | --- SCM  |
| --- CT    | --- TLD  |
| --- DF    |          |

(e) Tracking results of the surfer sequence

FIGURE 5: Screenshots of some sample tracking results when there are severe occlusion and drift.

stably in the liquor sequence (Figure 5(c)). In the liquor sequence, the bottle undergoes several heavy occlusions (e.g., frames 508, 727, 772, 1183, and 1236). Only AWLMC and TLD methods and our method track the object stably. In the panda variation (Figure 5(d)), the target undergoes in-plane pose variation, occlusion, and fast motion. Tables 2 and 3 show that only the proposed method outperforms the other methods on this sequence in terms of success rate and center location error. In the surfer sequence shown in Figure 5(e), the target object undergoes heavy occlusion and the appearance changes dramatically. In addition, the resolution of the video

sequence is low, which make it a challenge to track the target. It is clear to see that, in terms of both success rate and center location error, our method performs better than the other methods on this sequence. The SCM and TLD methods work well on this sequence as both of them select adaptive appearance model to account for appearance change.

The proposed algorithm handles occlusion and drift well as we utilize a joint local and global search mechanism, where the two strategies complement each other to get a more reliable location of the target. Furthermore, the proposed algorithm performs well even when the target undergoes



FIGURE 6: Screenshots of some sample tracking results when there are severe occlusion and drift.

long duration or full occlusion as we can detect the severe occlusion. Once the algorithm determines that the target is lost, the global search strategy is used to relocate the target and recover the tracking immediately.

**5.2.3. Pose and Illumination Change.** For the coke sequence shown in Figure 6(a), the appearance changes dramatically due to illumination and pose variation when the target moves here and there under a table lamp. The target object

shown in Figure 6(d) undergoes the similar problems. The CT and TLD methods and our method achieve favorable performance on these two sequences. In the cartoon sequence shown in Figure 6(b), the target object has a large pose variation and shape deformation. In addition, the target object undergoes abrupt movements. The AMCMC, MCMC, and AWLMC methods and our method work well on this sequence. In Figure 6(c), the appearance of the target object changes when the dancer moves up and down, left and right.

Most methods cannot track the target stably and drift away from the target object. When the illumination variations and partial occlusion both occur in the woman sequence (Figure 6(e)), most algorithms fail to track the target object well. Our method utilizes an effective incremental-learning-based online template update mechanism to adapt to the appearance changes. Thereby, we can handle the pose and illumination changes in the test sequences. Furthermore, the proposed algorithms can predict a more accurate position via the adaptive state transition model, which makes our method more flexible.

## 6. Conclusion

In this paper, we propose a novel insect vision embedded particle filter tracking algorithm. An adaptive state transition model utilizing the insect vision detector is adopted to cope with abrupt motion and fast movement tracking problems. We use the joint local and global search strategies to refine a more accurate candidate state. Furthermore, a two-prerequisite method is discussed to detect the lost target caused by heavy occlusion and drift problems. Once the target is lost, the interesting regions where the responses of the insect vision detector are not zero will be detected via the global search strategy to determine the most similar region. Moreover, to address the dramatic appearance changes problem, an incremental-learning-based online template update method is adopted. Numerous experiments on challenging sequences demonstrate that the proposed tracking algorithm performs well against state-of-the-art algorithms. In our future work, we will consider a mixture appearance model to strengthen the prior knowledge of the target.

## Conflict of Interests

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

## Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under grants no. 61175096 and no. 61300082. The authors would like to thank the associate editor and all anonymous reviewers for their insightful comments. They also thank the authors in [10, 29, 35–42] for providing the compare codes.

## References

- [1] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.
- [2] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 125–141, 2008.
- [3] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1269–1276, June 2010.
- [4] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1830–1837, June 2012.
- [5] Y. Su, Q. Zhao, L. Zhao, and D. Gu, "Abrupt motion tracking using a visual saliency embedded particle filter," *Pattern Recognition*, vol. 47, no. 5, pp. 1826–1834, 2014.
- [6] S. Avidan, "Support vector tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004.
- [7] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proceedings of the British Machine Vision Conference (BMVC '06)*, vol. 1, pp. 6.1–6.10, BMVA Press, Edinburgh, Scotland, September 2006.
- [8] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.
- [9] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 1323–1330, IEEE, Barcelona, Spain, November 2011.
- [10] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Computer Vision—ECCV 2012*, pp. 864–877, Springer, Berlin, Germany, 2012.
- [11] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "Object tracking with an adaptive color-based particle filter," in *Pattern Recognition*, vol. 2449, pp. 353–360, Springer, Berlin, Germany, 2002.
- [12] M. Isard and A. Blake, "CONDENSATION—conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [13] K. Nummiaro, E. Koller-Meier, and L. van Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, no. 1, pp. 99–110, 2003.
- [14] B. Han, Y. Zhu, D. Comaniciu, and L. Davis, "Kernel-based Bayesian filtering for object tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 227–234, IEEE, June 2005.
- [15] S. Stalder, H. Grabner, and L. van Gool, "Cascaded confidence filtering for improved tracking-by-detection," in *Computer Vision—ECCV 2010*, vol. 6311 of *Lecture Notes in Computer Science*, pp. 369–382, Springer, Berlin, Germany, 2010.
- [16] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 1515–1522, IEEE, Kyoto, Japan, September–October 2009.
- [17] C. Chang and R. Ansari, "Kernel particle filter for visual tracking," *IEEE Signal Processing Letters*, vol. 12, no. 3, pp. 242–245, 2005.
- [18] M. Egelhaaf, A. Borst, and W. Reichardt, "Computational structure of a biological motion-detection system as revealed by local detector analysis in the fly's nervous system," *Journal of the Optical Society of America A, Optics and Image Science*, vol. 6, no. 7, pp. 1070–1087, 1989.
- [19] J. Yin, D.-G. Li, H.-M. Fang, and Z.-F. Liu, "Moving objects detection by imitating biologic vision based on fly's eyes," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '04)*, pp. 763–766, August 2004.

- [20] G. Yu, Q. Zhao, and Y. Xu, "Insect vision inspired object detection in the video with a moving cluttered background," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '11)*, pp. 2931–2936, December 2011.
- [21] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, article 58, 2013.
- [22] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Computer Vision—ECCV '96*, vol. 1064 of *Lecture Notes in Computer Science*, pp. 343–356, Springer, Berlin, Germany, 1996.
- [23] Z. Wang, X. Yang, Y. Xu, and S. Yu, "CamShift guided particle filter for visual tracking," *Pattern Recognition Letters*, vol. 30, no. 4, pp. 407–413, 2009.
- [24] H. Liu and F. Sun, "Efficient visual tracking using particle filter with incremental likelihood calculation," *Information Sciences*, vol. 195, pp. 141–153, 2012.
- [25] T. Rui, Q. Zhang, Y. Zhou, and J. Xing, "Object tracking using particle filter in the wavelet subspace," *Neurocomputing*, vol. 119, pp. 125–130, 2013.
- [26] J. Sullivan and J. Rittscher, "Guiding random particles by deterministic search," in *Proceedings of the 8th International Conference on Computer Vision (ICCV '01)*, vol. 1, pp. 323–330, July 2001.
- [27] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1491–1506, 2004.
- [28] A. Yao, X. Lin, G. Wang, and S. Yu, "A compact association of particle filtering and kernel based object tracking," *Pattern Recognition*, vol. 45, no. 7, pp. 2584–2597, 2012.
- [29] J. Kwon and K. M. Lee, "Tracking of abrupt motion using Wang-Landau Monte Carlo estimation," in *Computer Vision—ECCV 2008*, pp. 387–400, Springer, Berlin, Germany, 2008.
- [30] X. Zhou, Y. Lu, J. Lu, and J. Zhou, "Abrupt motion tracking via intensively adaptive markov-chain monte carlo sampling," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 789–801, 2012.
- [31] J. Vermaak, P. Pérez, M. Gangnet, and A. Blake, "Towards improved observation models for visual tracking: selective adaptation," in *Computer Vision—ECCV 2002*, vol. 2350 of *Lecture Notes in Computer Science*, pp. 645–660, Springer, Berlin, Germany, 2002.
- [32] D. Fox, D. Schulz, G. Borriello, J. Hightower, and L. Liao, "Bayesian filtering for location estimation," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 24–33, 2003.
- [33] H. Eichner, M. Joesch, B. Schnell, D. F. Reiff, and A. Borst, "Internal structure of the fly elementary motion detector," *Neuron*, vol. 70, no. 6, pp. 1155–1164, 2011.
- [34] S. He, Q. Yang, R. W. H. Lau, J. Wang, and M.-H. Yang, "Visual tracking via locality sensitive histograms," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2427–2434, June 2013.
- [35] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805–1819, 2005.
- [36] G. O. Roberts and J. S. Rosenthal, "Examples of adaptive mcmc," *Journal of Computational and Graphical Statistics*, vol. 18, no. 2, pp. 349–367, 2009.
- [37] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: bootstrapping binary classifiers by structural constraints," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 49–56, June 2010.
- [38] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1838–1845, IEEE, Providence, RI, USA, June 2012.
- [39] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 1, pp. 798–805, IEEE, June 2006.
- [40] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1910–1917, IEEE, June 2012.
- [41] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision—ECCV 2012*, vol. 7575 of *Lecture Notes in Computer Science*, pp. 702–715, Springer, Berlin, Germany, 2012.
- [42] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1822–1829, IEEE, June 2012.





# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

