Pattern Recognition 45 (2012) 2584–2597

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

A compact association of particle filtering and kernel based object tracking

Anbang Yao [a,*], Xinggang Lin [b], Guijin Wang [b], Shan Yu [c]

[a] Institute of Automation, Chinese Academy of Science, Beijing 100090, China
[b] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[c] French National Institute for Research in Computer Science and Control, Inria, France

ARTICLE INFO

Article history:
Received 19 June 2011
Received in revised form
20 November 2011
Accepted 17 January 2012
Available online 28 January 2012

Keywords:
Visual tracking
Particle filtering
Kernel based object tracking
Matrix condition number

ABSTRACT

Particle filtering (PF) and kernel based object tracking (KBOT) algorithms have shown their promises in a wide range of visual tracking contexts. This paper mainly addresses the association of PF and KBOT. Unlike other related association approaches which usually directly use KBOT to refine the position states of propagated particles for more accurate mode seeking, we elucidate the problem of what kind of particles is suitable for employing KBOT to refine their position states from a theoretical point of view. In accordance with the theoretical analysis, a two-stage solution is also proposed to resample propagated particles that are suitable for invoking KBOT from a computational perspective. The incremental Bhattacharyya dissimilarity (IBD) based stage is designed to consistently distinguish the particles located in the object region from the others placed in the background, while the matrix condition number based stage is formulated to further eliminate the particles positioned at the ill-posed conditions for running KBOT. Once the appropriate particles are obtained, constrained gradient based mean shift optimization enables us to efficiently refine the particles' position states. Besides, a state transition model embodying object-scale oriented information and prior motion cues is presented to adapt to fast movement scenarios. These ingredients lead to a new tracking algorithm. Experiments demonstrate that the proposed association approach is more robust to handle complex tracking conditions in comparison with related methods. Also, a limited number of particles are used in our association algorithm to maintain multiple hypotheses.

1. Introduction

2-D visual tracking is the problem of automatically estimating the locations of moving objects in the consecutive frames of a video sequence. It is of great pertinence to many applications such as surveillance [1,2], intelligent traffic navigation [3,4], human computer interaction [5–7], content based video indexing and retrieval [8–10]. The challenges in building up a robust visual tracking system arise from the presence of background clutter, occlusion, fast movement, object appearance changes, illumination variations, varying object scales (i.e., object sizes), etc. Up to now, numerous approaches (e.g., [11–22]) have been proposed to overcome these difficulties, the reader is referred to [23] for a comprehensive survey of the literature.

Among the available visual tracking approaches, particle filtering (PF, also known as Condensation) [13] and kernel based object tracking (KBOT, also known as mean shift tracking) [14] algorithms have achieved considerable success over the last decade. As a statistical approach, PF is established for dealing with the multi-modal visual tracking problem of general non-linear and non-Gaussian systems. Its most appealing merit is that it provides a convenient framework for estimating and propagating the posterior probability density function (pdf) of state variable regardless of the underlying distribution [23,24]. However, to precisely approximate the variations of the posterior pdf in state space, the difficulty lies in the fact that PF algorithm usually demands a large number of densely sampled particles. This in turn will incur heavy computational load and further suppress the using of PF in real-time application environments. Unlike PF, KBOT is a non-parametric approach. In KBOT, the target region is represented as a color histogram weighted with an isotropic kernel function, and the gradient based mean shift optimization is used to iteratively seek a target candidate which is the closest mode of the target model in the current frame. Although remarkably lower computational cost and easier implementation are demonstrated by KBOT in comparison with PF, KBOT shows the noticeable deficiency in handling multiple modes (caused by similar object, cluttered background, etc.) and temporary lost (caused by occlusion, quick motion, etc.) [23]. These KBOT's disadvantages can be attributed to the gradient based mean shift optimization used for searching object in the basin of attraction of Bhattacharyya coefficient based similarity function.

* Corresponding author. Tel.: +86 10 82614462; fax: +86 10 62647458.
E-mail address: abyao@nlpr.ia.ac.cn (A. Yao).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2012.01.016

However, it is worth noting that above mentioned strengths and weaknesses suggest that PF and KBOT can complement each other instinctively. In view of this, to the problem of designing a tracking approach whose performance is superior to that of PF/KBOT, a potential solution is the association of PF and KBOT, i.e., to properly integrate PF and KBOT into one optimized tracking framework.

Some important research works have already been done on the issue of the association of filtering approach and KBOT. Comaniciu et al. [14] adopt an integrated framework in which Kalman filtering is used to estimate the state uncertainty of moving object first, and then KBOT is employed to find a more accurate object position. However, Kalman filter assumes that the noise sequences are Gaussian and the time-varying functions are linear, which means that it cannot process the tracking problem of general non-linear and non-Gaussian systems [13,25]. Based on the mode detection algorithm using variable-bandwidth mean shift, Han et al. [24] incorporate incremental kernel density approximation technique into the PF framework for addressing visual tracking. The challenge is that incremental kernel density approximation itself is burdened with expensive computational cost [26], especially that their method still requires a relatively large number of particles to get competitive tracking performance (actually, 400 particles are used in the tracking experiments on real data). Besides, as for algorithm implementation, the number of Gaussian kernels should be properly set in advance, but this is usually not trivial in practice [17,26]. In [27], the authors run PF and KBOT trackers in a parallel way first, and then the object position is determined from comparing the Bhattacharyya coefficient based confidences of the tracking results of two trackers. This kind of balance is somewhat ad hoc, and it pays little attention to the combination of the advantages of PF and KBOT, thus it will not perform well when both trackers are simultaneously unreliable. In addition, as a bin-to-bin similarity measure, Bhattacharyya coefficient is not vey discriminative, especially when it is used for measuring similarities between high dimensional features [17,21,28]. To prevent particles from quickly degenerating into a very limited number of particles which are not positioned at or near to the true mode, Chang and Ansari [29] apply gradient based mean shift optimization to refine the position states of propagated particles for more accurate mode seeking. Although the propagated particles are supposed to move towards high probability modes in state space through iterative mode seeking procedure, it is not reliable enough to directly run KBOT on arbitrary particles. As will be clarified in this paper, this can be traced to the fact that gradient based mean shift iteration usually converges to undesirable positions when particles are placed in the background or are positioned at the ill-posed conditions. With respect to the association structure, the approaches of [30–35] are generally similar to that of [29] regardless of the fact that some of these approaches are also benefited from other cues (e.g., an adaptive transition model is embedded in the association approach of [31]) or are developed for different application contexts (e.g., the association approach of [35] is designed for a hand control wheelchair). To sum up, in spite of the fact that the association approaches of [29–35] have already shown better tracking accuracy than PF and KBOT, the authors have not clearly investigated the core issue which we have addressed in this paper. This core issue is whether KBOT is suitable for refining the position states of arbitrary particles for more accurate mode seeking. If not, how to apply KBOT to refine particle's position state in a reasonable and efficient way? This paper attempts to contribute to a better understanding of the association of PF and KBOT.

Compared with the association approaches described in [29–35], the key differences with respect to our association approach are as follows. (1) we elucidate the problem of what kind of particles is fit for employing KBOT to refine their position states from a theoretical point of view. (2) From a perspective of computation, we present a two-stage solution (i.e., the incremental Bhattacharyya dissimilarity (IBD) based stage and the matrix condition number based stage) to resample propagated particles that are well suited for invoking KBOT. (3) A constrained gradient based mean shift optimization is presented to efficiently move the position states of the appropriate particles towards more accurate modes. (4) A state transition model embodying object-scale oriented information and prior motion cues is proposed to adapt to fast movement scenarios. (5) Based on these components, a new tracking algorithm is also given.

Note that this paper is not focused on object representation in visual tracking paradigm. It is a fact that color histogram based appearance model has been successfully used in many existing PF, KBOT and association based tracking algorithms [14,24,27,29–36]. To fairly compare our approach and related approaches, we also specify color histograms for object representation throughout this paper. Extensive experiments demonstrate that our association approach outperforms the regular PF, KBOT and other two popular association approaches [29,31] in handling challenging tracking tasks. Furthermore, as a general tracking approach, just like the association approaches of [14,24,29–35], different features can also be easily incorporated into our association approach.

The remainder of this paper is organized as follows. We first introduce the mathematical concepts of PF and KBOT in Section 2. Section 3 presents a detailed description of the proposed association approach including the theoretical analysis, the two-stage solution, the constrained gradient based mean shift optimization, the state transition model and the tracking algorithm. Section 4 presents comparative experiments to demonstrate the effectiveness of our association approach. Section 5 concludes with a discussion and makes an outlook of possible future extensions.

2. Preliminary theories

2.1. Particle filtering

PF is a state space approach for implementing recursive Bayesian filter via sequential Monte Carlo (SMC) simulation [13,25]. Let x_t denote the object state at time t, let $Z_t=\{z_1,z_2,\ldots,z_t\}$ denote the observation sequence up to time t, let $p\{z_t|x_t\}$ denote the observation likelihood function and let $p(x_t|x_{t-1})$ denote the state transition model, the visual tracking problem in Bayesian filter is defined to model dynamic system by recursively estimating the posterior pdf

$$p(x_t|Z_t)\propto p(z_t|x_t)\int p(x_t|x_{t-1})p(x_{t-1}|Z_{t-1})dx_{t-1}. \quad (1)$$

Different from other approaches such as Kalman filter and extended Kalman filter which provide the solutions of (1) under their respective conditions [13,14,25], PF is designed to address (1) under more general situations where pdf $p(z_t|x_t)$ and $p(x_t|x_{t-1})$ are usually non-linear and non-Gaussian. The basic idea of PF is to offer a discrete approximation of pdf $p(x_{t-1}|Z_{t-1})$ by randomly sampling a set of K particles with states $\{s_{t-1}^k\}_{k=1}^K$ and importance weights $\{\pi_{t-1}^k\}_{k=1}^K$. By substituting pdf $p(x_{t-1}|Z_{t-1})$ with the sampled particle set $\{s_{t-1}^k,\pi_{t-1}^k\}_{k=1}^K$, (1) can be expressed as

$$p(x_t|Z_t)\propto p(z_t|x_t)\sum_{k=1}^{K}\pi_{t-1}^k p(x_t^k|s_{t-1}^k). \quad (2)$$

Now, according to (2), the state estimation problem can be iteratively solved via prediction and update steps. In practice, to

estimate the object position at time t, regular PF algorithm generally has four steps.

1. *Re-sampling*: From the particle set $\{s_{t-1}^k,\pi_{t-1}^k\}_{k=1}^K$ at time $t-1$, generate a new particle set $\{\overline{s}_{t-1}^k,\overline{\pi}_{t-1}^k=1/K\}_{k=1}^K$ by removing particles with small weights and concentrating on particles with large weights.
2. *Propagating*: According to the state transition model $p(x_t|x_{t-1}=\overline{s}_{t-1}^k)$, propagate each re-sampled particle state $\overline{s}_{t-1}^k$ to get a new state s_t^k for time t.
3. *Weighting*: Based on the state s_t^k and corresponding observation z_t^k, compute the weight π_t^k of each propagated particle at time t as $p(z_t^k|x_t^k=s_t^k)$ first, and then normalize it by

$$\pi_t^k=\frac{p(z_t^k|x_t^k=s_t^k)}{\sum_{k_1=1}^{K}p(z_t^{k_1}|x_t^{k_1}=s_t^{k_1})}. \tag{3}$$

4. *Estimating*: Calculate the object position at time t as

$$E(x_t)=\sum_{k=1}^{K}\pi_t^k s_t^k. \tag{4}$$

2.2. Kernel based object tracking

Unlike PF, KBOT is a deterministic tracking approach. Its basic purpose is to iteratively seek a target candidate which is the closest mode of the target model in the current frame [14]. Let q and $p(x)$ are two M-bin histogram features (e.g., color histogram) extracted from a hand-drawn target region in the reference frame and a candidate target region centered at 2-D x in the current frame, respectively, $\sum_{u=1}^{M}q_u=1$ and $\sum_{u=1}^{M}p_u(x)=1$, the Bhattacharyya coefficient based similarity function to be maximized in KBOT is defined as

$$\rho(p(x),q)=\sum_{u=1}^{M}\sqrt{p_u(x)q_u}, \tag{5}$$

where

$$p_u(x)=\frac{1}{C}\sum_{i=1}^{N}k\left(\left\|\frac{x-x_i}{h}\right\|^2\right)\delta(b(x_i),u), \tag{6}$$

where

$$C=\sum_{i=1}^{N}k\left(\left\|\frac{x-x_i}{h}\right\|^2\right), \tag{7}$$

$k(x)$ is a non-negative, isotropic and monotonic decreasing kernel profile which weighs over N pixel locations, h is the 2-D bandwidth vector of $k(x)$, δ is the Kronecker delta function, $b(x_i)$ is the feature vector of the pixel at location x_i, and N is the number of pixels located in $k(x)$. To find the new target position in the current frame, the histogram feature $p(x_0)$ of the initial target candidate positioned at 2-D x_0 in the current frame is computed first. Using the Tailor expansion around the value of $p_u(x_0)$, the linear approximation of (5) can then be extended as

$$\rho(p(x),q)\approx\frac{1}{2}\sum_{u=1}^{M}\sqrt{p_u(x_0)q_u}+\frac{1}{2C}\sum_{i=1}^{N}w_i k\left(\left\|\frac{x-x_i}{h}\right\|^2\right), \tag{8}$$

where

$$w_i=\sum_{u=1}^{M}\sqrt{\frac{q_u}{p_u(x_0)}}\delta(b(x_i),u). \tag{9}$$

Note that the first term in (8) is a constant, that is, it is independent of x. Therefore, maximizing (8) only depends on its second term. This can be achieved by using the gradient based mean shift optimization. During the procedure, the object position is iteratively moved from the current position x_0 to the new position $\hat{x}_0$ according to

$$\hat{x}_0=\frac{\sum_{i=1}^{N}x_i w_i g\left(\left\|\frac{x_0-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{N}w_i g\left(\left\|\frac{x_0-x_i}{h}\right\|^2\right)}, \tag{10}$$

where $g(x)=-k'(x)$. As noted in [14], $g(x)$ is a constant if using Epanechnikov profile

$$k(x)=\begin{cases}\frac{1}{2}c_d(d+1)\left(1-\left\|\frac{x_0-x}{h}\right\|\right) & \text{if } \left\|\frac{x_0-x}{h}\right\|\le 1\\ 0 & \text{otherwise}\end{cases}. \tag{11}$$

3. The proposed association approach

According to section 2.1, it can be concluded that the belief of the estimated posterior pdf $p(x_{t-1}|Z_{t-1})$ in PF is only correlated with the sampled particle set $\{s_{t-1}^k,\pi_{t-1}^k\}_{k=1}^K$. To provide a good approximation of the underlying distribution of posterior $p(x_{t-1}|Z_{t-1})$ in state space, PF usually requires a large number of densely sampled particles. However, this in turn will bring about heavy computational burden due in large part to the high dimensionality of state space. Therefore, if using PF in practical applications, how to model the variations of the posterior pdf $p(x_{t-1}|Z_{t-1})$ in state space with a fairly small number of particles is of great importance. A general way to address this efficiency problem is to reallocate the particles to high probability mass areas via a variety of techniques such as unscented particle filter [37] and covariance scaled sampling [38]. Alternatively, according to section 2.2, KBOT also provides a computationally efficient approach to generate a fair particle set mainly due to its simplicity and effectiveness. By employing KBOT to move particles towards mass areas where the dominant modes (i.e., the peaks) of posterior pdf $p(x_{t-1}|Z_{t-1})$ are located, the number of particles needed in PF may be fairly reduced. This avoids having to choose a proposal distribution allowing for efficient allocation of the particles [37,38].

From a theoretical perspective, a central issue should be considered in designing a robust association approach of PF and KBOT is: whether KBOT is suitable for refining the position states of arbitrary particles for more accurate mode seeking. If not, under what conditions particles are fit for invoking KBOT to move their position states to more accurate modes? However, as described in the introduction section, existing association methods [29–35] usually directly use KBOT to refine the position state of each propagated particle. That is, this central issue has not been addressed in the literature. For this reason, the main goal of this section is to sufficiently explore it. We will analyze the theoretical properties of KBOT first, and then present our association approach. The framework of the proposed association approach is summarized in Fig. 1.

3.1. Theoretical analysis

In the following, we first provide and prove the theoretical properties of KBOT for a better understanding of the gradient based mean shift optimization. In accordance with the theoretical analysis, we then derive the theoretical solution for the problem of what kind of particles are well adapted to using KBOT to renew their position states for more accurate mode seeking.

Property 1. *The maximum of possible movement region where the object can be correctly localized by KBOT is the kernel size.*

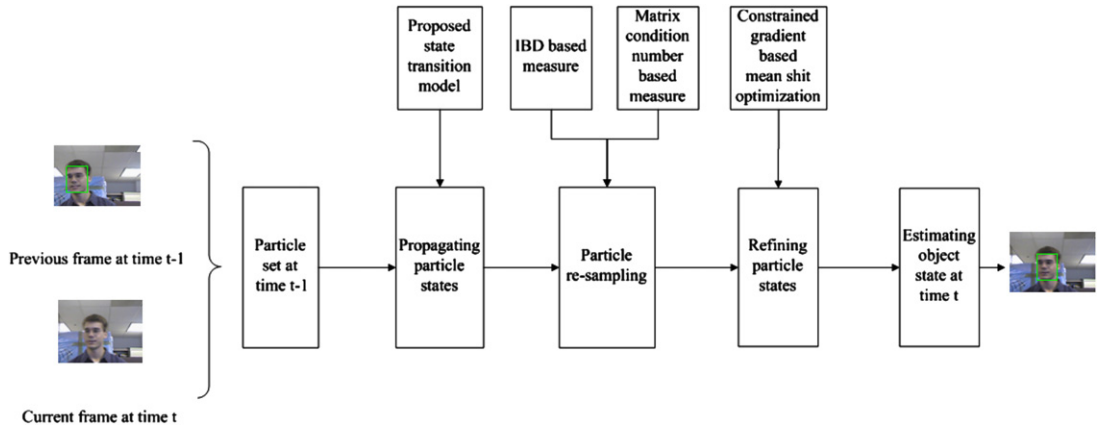

Fig. 1. The framework of the proposed association approach.

Proof. Let $\hat{\omega}_i = \omega_i g\left(\left\| \frac{x_0 - x_i}{h} \right\|^2\right)$, we can derive (12) from (10),

$$\Delta x = \hat{x}_0 - x_0 = \frac{\sum_{i=1}^{N} \hat{\omega}_i x_i}{\sum_{i=1}^{N} \hat{\omega}_i} - x_0 = \frac{\sum_{i=1}^{N} \hat{\omega}_i (x_i - x_0)}{\sum_{i=1}^{N} \hat{\omega}_i}$$
$$= a_1(x_1 - x_0) + , \cdots , + a_N(x_N - x_0), \tag{12}$$

where $a_i = \frac{\hat{\omega}_i}{\sum_{i=1}^{N} \hat{\omega}_i}$, and $\hat{\omega}_i \geq 0$. Thus, $\sum_{i=1}^{N} a_i = 1$. Recall that kernel $k(x)$ is symmetric about the origin x_0, we have

$$\min(|\Delta x|) = \min(|x_1 - x_0|, \cdots, |x_N - x_0|),$$
$$= (0,0)^T \tag{13}$$

$$\max(|\Delta x|) = \max(|x_1 - x_0|, \cdots, |x_N - x_0|) = h. \tag{14}$$

According to (13) and (14), the minimum of iterative step size in KBOT is the zero vector, while the maximum is the kernel size. Thus Property 1 is proved.

Property 2. *The positional displacement vector in KBOT is invariant to the scaling of the weight set by a positive coefficient.*

Proof. Similarly, let $\hat{\omega}_i = a\omega_i$, where $a > 0$, we can readily derive (15) from (10),

$$\Delta x = \frac{\sum_{i=1}^{N} \hat{\omega}_i g\left(\left\| \frac{x_0 - x_i}{h} \right\|^2\right) x_i}{\sum_{i=1}^{N} \hat{\omega}_i g\left(\left\| \frac{x_0 - x_i}{h} \right\|^2\right)} - x_0 = \frac{\sum_{i=1}^{N} \omega_i g\left(\left\| \frac{x_0 - x_i}{h} \right\|^2\right) x_i}{\sum_{i=1}^{N} \omega_i g\left(\left\| \frac{x_0 - x_i}{h} \right\|^2\right)} - x_0. \tag{15}$$

According to (15), it is obvious that the positional displacement vector Δx in KBOT does not change under conditions of $\hat{\omega}_i = a\omega_i$ or ω_i. Thus Property 2 is proved.

Property 3. *The positional displacement vector in KBOT is not invariant to a positive constant offset from the initial weight set to a new weight set.*

Proof. Recall that $g(\left\| \frac{x_0 - x_i}{h} \right\|^2) = g(\left\| \frac{-(x_0 - x_i)}{h} \right\|^2)$ and $\sum_{i=1}^{N} g(\left\| \frac{x_0 - x_i}{h} \right\|^2)(x_i - x_0) = 0$, let $\hat{\omega}_i = \omega_i + a$, where $a > 0$, we can derive (16) from (10)

$$\Delta x = \frac{\sum_{i=1}^{N} \hat{\omega}_i g\left(\left\| \frac{x_0 - x_i}{h} \right\|^2\right) x_i}{\sum_{i=1}^{N} \hat{\omega}_i g\left(\left\| \frac{x_0 - x_i}{h} \right\|^2\right)} - x_0$$
$$= \frac{\sum_{i=1}^{N} \omega_i g\left(\left\| \frac{x_0 - x_i}{h} \right\|^2\right)(x_i - x_0) + a \sum_{i=1}^{N} g\left(\left\| \frac{x_0 - x_i}{h} \right\|^2\right)(x_i - x_0)}{\sum_{i=1}^{N} (\omega_i + a) g\left(\left\| \frac{x_0 - x_i}{h} \right\|^2\right)}$$
$$= \frac{\sum_{i=1}^{N} \omega_i g\left(\left\| \frac{x_0 - x_i}{h} \right\|^2\right)(x_i - x_0)}{\sum_{i=1}^{N} \omega_i g\left(\left\| \frac{x_0 - x_i}{h} \right\|^2\right) + b}, \tag{16}$$

where

$$b = a \sum_{i=1}^{N} g\left(\left\| \frac{x_0 - x_i}{h} \right\|^2\right). \tag{17}$$

According to (16), it is clear that the direction of the positional displacement vector Δx in KBOT is invariant under conditions of $\hat{\omega}_i = \omega_i + a$ or ω_i, only the iterative step size changes. Thus Property 3 is proved.

Property 4. *Maximizing Bhattacharyya coefficient based similarity function (5) is equivalent to finding the solution of a system of linear equations.*

Proof. For KBOT, it has already been stated in [39] that maximizing (5) defined similarity function can be replaced by the minimization of

$$O(x) = \left\| \sqrt{q} - \sqrt{p(x_0 + \Delta x)} \right\|, \tag{18}$$

where $x = x_0 + \Delta x$, Δx is the positional displacement vector. On the other side, Eq. (6) can be rewritten in a more concise form

$$p(x) = U^T K(x), \tag{19}$$

where

$$U = [u_1, u_2, \ldots, u_M]$$
$$K(x) = [k(x_1, x), k(x_2, x), \ldots, k(x_N, x)]^T, \tag{20}$$

where

$$u_j = [\delta(b(x_1), u_j), \delta(b(x_2), u_j), \ldots, \delta(b(x_N), u_j)]^T$$
$$j = 1, \ldots, M. \tag{21}$$

Now, by expanding (18) in a Taylor series with respect to Δx and dropping the higher order terms, we have

$$A\Delta x = \sqrt{q} - \sqrt{p(x)}, \tag{22}$$

where,

$$A=\frac{1}{2}diag\{p_1(x),p_2(x),\ldots,p_M(x)\}^{-\frac{1}{2}}U^TJ(x), \tag{23}$$

where

$$J(x)=\left[\frac{\partial K}{\partial x_x},\frac{\partial K}{\partial x_y}\right]=\begin{bmatrix}\frac{\partial k(x_1,x)}{\partial x_x},\frac{\partial k(x_1,x)}{\partial x_y}\\ \frac{\partial k(x_2,x)}{\partial x_x},\frac{\partial k(x_2,x)}{\partial x_y}\\ ,\ldots,\\ \frac{\partial k(x_N,x)}{\partial x_x},\frac{\partial k(x_N,x)}{\partial x_y}\end{bmatrix}. \tag{24}$$

Here $A\in\Re^{M\times 2}$, $\Delta x\in\Re^2$, $q\in\Re^M$ and $p(x)\in\Re^M$, hence, it is evident that minimizing (18) is equivalent to solving a system of linear Eq. (22). Thus Property 4 is proved.

Following the theoretical properties of KBOT proved so far, we can easily derive that KBOT is not suitable for refining the position states of arbitrary particles. Further, the theoretical results for the problem of what kind of particles are adapted to using KBOT to refine their position states for more accurate mode seeking are concluded as follows.

(1) Since kernel size reflects object scale, Property 1 implies that KBOT cannot deal with fast movement scenarios where there is usually no or little overlap between the object regions in two consecutive frames. That is, KBOT can only be used for refining the position states of particles which are completely or at least partially located in the object region. This is why KBOT is easy to be trapped into a local maximum [14,23,39].

(2) For any two particles, Property 2 and Property 3 demonstrate that if there exists a linear relation between their respective weight sets of sample pixels, the particle which is closer to the true mode is easier to converge to a more accurate solution of (22) in comparison with the other one.

(3) From the perspective of getting a numerically stable solution of (22), Property 4 shows that only the particles whose positions are placed at well-posed conditions are adapted to invoking KBOT for more accurate mode seeking. That is, KBOT cannot be used for refining the position states of particles that are positioned at ill-posed conditions.

3.2. Two-stage solution

Up to now, we have clarified the problem of what kind of particles are well suited for invoking KBOT to seek more accurate modes from a theoretical point of view. However, in practice, it still remains unclear on the problem of how to efficiently measure the quality of particles in terms of the theoretical results. To this end, we propose a computationally tractable solution to select appropriate particles from the propagated particle set while meeting theoretical results. In the light of theoretical results (1) and (2), the particles located in the background should be eliminated, while the particles positioned at ill-posed conditions should be discarded according to (3). Therefore, our solution is composed of two stages which well address above considerations.

The first stage of our solution is devoted to distinguishing the particles located in the object region from the others placed in the background. In the context of using histogram features as observations to represent particles, this can be done by comparing particle weights which are usually computed from available measures. However, the widely used measures such as Bhattacharyya coefficient and Kullback–Leibler divergence [40] are not discriminative enough, especially for measuring the similarities between high dimensional histogram features [17,21,28]. Here, we employ our previously proposed IBD [41] to discriminate the particles located in the object region from the others placed in the background. Our IBD has two attractive properties. (1) An incremental similarity matrix (ISM) is embedded in the traditional Bhattacharyya coefficient based dissimilarity for better evaluating the difference between a target histogram and a particle histogram. Such an ISM works as a bin-mixing matrix and enables a cross-bin interaction. (2) With the support of ISM computed in joint spatial temporal space, the discriminative capability of IBD is superior to the state of the art measures.

Given a target histogram q and a particle histogram p that have the same dimensions to those of the inputs of (5), IBD is defined as

$$d=\sqrt{1-\left(\sqrt{q}^TW\sqrt{p}\right)}, \tag{25}$$

where W is the ISM, $W=[w_{ij}]\in\Re^{M\times M}$, $\forall i,j,\quad 0\le w_{ij}\le 1$, $w_{ij}=w_{ji}$, and $\sum_j w_{ij}=1$. The entry w_{ij} denotes the incremental similarity belief between the matched bins q_i and p_j. Usually, a large w_{ij} set is easier to obtain a small dissimilarity value in comparison with a small w_{ij} set. Analogous to (5), (25) is bounded by zero and one, thus such an ISM also has the advantage of being not needed to renormalize the dissimilarity in the range of zero to one.

In the computation process, we first construct each ISM in spatial space and in temporal space, respectively. Then, the final ISM in joint spatial temporal space is obtained through a filtering approach. The detailed computation process of IBD and more analysis are referred to [41]. Here, Fig. 2 shows some example results to compare the discriminative capability of Bhattacharyya coefficient, Kullback–Leibler divergence and our incremental Bhattacharyya similarity ($\rho=\sqrt{q}^TW\sqrt{p}$). The results are obtained by calculating the similarities between the histogram pairs of each particle region (indicated with red rectangle) and the target region (indicated with green rectangle). Each particle region has a same size to the target region, and its center is scanned all over the target region. For color histograms, they are computed at a $8\times 8\times 8$-*bin* dimension in RGB space. It can be observed that Bhattacharyya coefficient and Kullback–Leibler divergence obtain very similar scores for many particle regions, while our incremental Bhattacharyya similarity can better discriminate the particles that are close to target center from the others that are relatively far from it.

Once the particles located in the object region have been consistently distinguished from the others placed in the background. We carry out the second stage of our two-stage solution. The second stage is dedicated to eliminating the particles positioned at the ill-posed conditions for invoking KBOT. Knowing from the theories of linear algebra, the system of linear Eq. (22) will have a unique solution if and only if A is a full rank matrix. However, since $A\in\Re^{M\times 2}$, where M is the dimension of histogram feature (generally, $M\ge 8$), (22) is usually an over-determined system of linear equations. Hager et al. [39] present a modified iteration procedure to compensate un-converged problem of (22). A more general way is to analyze the numerical stability of the system of linear Eq. (22) by calculating the condition number of matrix A^TA [42]. Here, we use norm-2 based matrix condition number to evaluate whether a particle is positioned at ill-posed conditions or not,

$$cond(A^TA)=\left(\frac{\lambda_{\max}((A^TA)^TA^TA)}{\lambda_{\min}((A^TA)^TA^TA)}\right)^{1/2}, \tag{26}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximal and minimal eigenvalues of the symmetric real matrix $(A^TA)^TA^TA$, respectively. As for KBOT, the particle with a small matrix condition number is easier to converge to a numerically stable solution of (22) in comparison with the others with larger matrix condition numbers. That is,

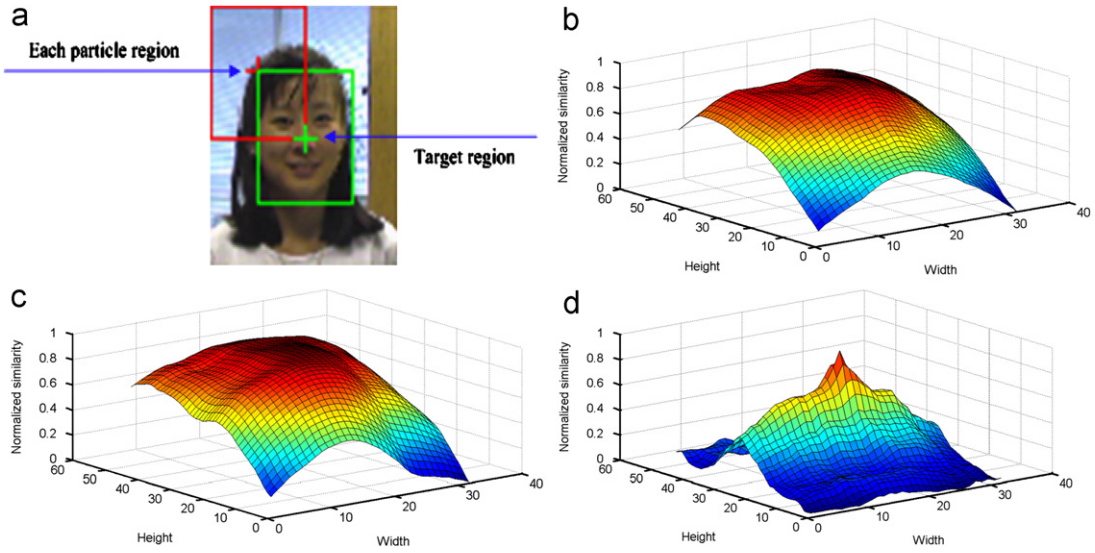

Fig. 2. Comparison of three different similarity measures. (a) The sample image. (b) Results of Bhattacharyya coefficient. (c) Results of Kullback-Leibler divergence. (d) Results of incremental Bhattacharyya similarity.

more large condition number indicates that the particle is more susceptible to trapping into a problematic solution.

From the above description, we can see that the IBD and matrix condition number based stages of the proposed approach are run in a cascaded manner. In practice, when most of the propagated particles are located in the object region, the matrix condition number based stage usually contributes more to the improved tracking performance compared with the IBD based stage. Opposite to the above case, when a large portion of the propagated particles are positioned in the background, the improved tracking performance is usually attributed in a large part to the IBD based stage. In other successful scenarios, the tracking performance is more likely to benefit from the joint contribution of two stages. More precisely speaking, the matrix condition number based stage can be viewed as a boosting step following the IBD based stage to seek more reliable particles which are well suited for invoking KBOT to refine their position states. Thus, in general, neither the contribution of IBD base stage nor the contribution of matrix condition number based stage can be neglected.

Here, we provide an example to show the effectiveness of our two-stage solution. Fig. 3a and b are two consecutive frames in a video sequence. Based on the estimated object region in the previous frame, we first uniformly sample 25 particles (whose centers are indicated with red crosses) in the current frame. Next, we run KBOT on each particle for more accurate mode seeking. In the experiments, the iteration upper bound of KBOT is set equal to 20. Fig. 3c and d show the results. In Fig.3d, red arrows represent the top 10 particles of 20 candidates sorted in an ascending order of IBD values, and green arrows point to the best 5 particles of 10 candidates (selected with IBD from 20 particles) sorted in an ascending order of the matrix condition numbers. It is obvious that our two-stage solution can well discriminate the particles located in the object from the particles placed in the background. In addition, for KBOT, the particles further singled out with respect to matrix condition number have the property of quickly converging to accurate positions as shown in Fig. 3c.

As for the association of PF and KBOT, our two-stage solution can be directly inserted between the weighting and estimating steps of PF. After selecting more appropriate particles from the propagated particle set, KBOT may be used to move them to more accurate modes in state space. However, in this way, the resulting particle set is smaller than the original set in size, which means the main concept of filtering in state space will be lost. That is not what we want to see. Note that the main goal of our two-stage solution is essentially consistent with that of particle re-sampling from the perspective of using KBOT to refine the position states of particles. In our association approach (as shown in Fig. 1), the two-stage solution is naturally incorporated into the particle re-sampling step. Its main advantage is that the measures of IBD and matrix condition number are not only used for suppressing the effects of the degeneracy problem but also for guaranteeing the fitness of re-sampled particles for running KBOT. Apart from that, the steps of particle re-sampling and state propagation are exchanged for compactness [13,24,25]. Consequently, similar to the association approaches of [24,29–35], the proposed association approach remains in the PF paradigm.

3.3. Constrained gradient based mean shift optimization

Once appropriate particles are re-sampled from the propagated particle set, KBOT can be directly used to move their position states to more accurate modes along the gradient descent direction of the likelihood surface. Different from this general way, we use the constrained gradient based mean shift optimization in which we add a relatively small upper bound (in this paper, we set it equal to 8 during experiments) as a restriction on the iteration procedure mainly due to two facts. One is that the re-sampled particles from our two-stage solution have the property of fast convergence. With respect to the example shown in Fig. 3, the average number of iterations for the top 5 particles selected from our two-stage solution is 4. The other is that the later possible stages of iteration procedure usually move very little displacements [14,39]. Therefore, running KBOT on each re-sampled particle in the modified way not only reasonably compensates particle set for the un-densely sampling of the posterior pdf but also rationally saves some unnecessary computational cost.

3.4. State transition model

One important goal of the association of PF and KBOT is to make a tradeoff between particle number and particle diversity so that well suited particles can be consistently re-sampled from the propagated particle set, further KBOT can be used on them to seek more accurate modes. This goal can easily be achieved when the target moves with a relatively stable velocity or with a predictable

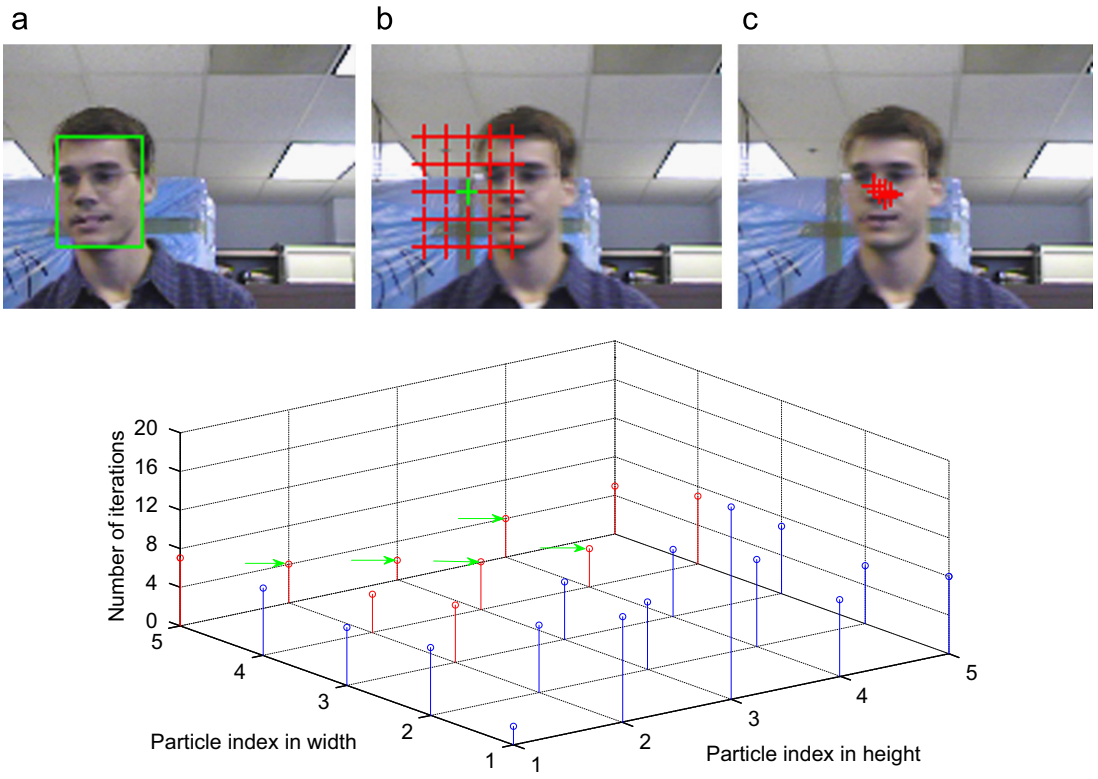

Fig. 3. The efficacy of our two-stage solution. (a) The previous frame, in which the green rectangle denotes the object region. (b) The current frame, in which the red crosses represent the positions of sampled particles and the green cross denotes the center of initial object candidate. (c) The refined position states of the top 5 particles selected from our two-stage solution. (d) The number of iterations needed if invoking KBOT on each particle position. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

motion pattern [13,29]. However, for the association algorithms of PF and KBOT, when fast movement scenarios are frequently available, achieving the goal will not easy any more. In this case, making a tradeoff between particle number and particle diversity demands that at least some of the particles in the propagated particle set should be allocated in the target area, which implies that the prediction of future positions of moving object should be properly harnessed. In this subsection, we advance a new state transition model to make an attempt to account for searching region where the target is covered. Instinctively, under the same fast movement scenarios, a small size object is easier to be lost in comparison with a big size object owing to the less object overlap between two continuous frames. Consequently, unlike conventional state transition models which usually resort to prior motion cues [13] or sophisticated learning techniques [31], our dynamic model jointly utilizes object-scale oriented information and prior motion cues to provide a prediction of possible object movements. Specifically, the proposed state transition model is defined as

$$x_t = x_{t-1} + \frac{1}{2}(x_{t-1} - x_{t-3}) + s_t. \tag{27}$$

Here, the second term represents the average speed of a moving object in the previous two frames. Term s_t is generated from a Gaussian distribution $N_t(0,\sigma)$, where standard variation vector σ reflects the object scale at time $t-1$. Note that s_t acts as a balance term to decrease the possibility of false prediction under scenarios of unpredictable fast movement. In the algorithm, we set σ according to

$$\sigma(h) = \alpha h, \tag{28}$$

where h is the kernel size, α is a positive coefficient. In general, the larger the kernel size h is, the smaller will be the coefficient α. We recommend to choose α in the range of 0.25–2 (its default value is 0.5). In the experiments, we test our state transition model on a large number of publicly available video sequences [44,45] and find it is quite amenable to the changing speeds of moving object.

3.5. The tracking algorithm

Given the target model q containing M bins and the particle set $\{s_{t-1}^k, \pi_{t-1}^k\}_{k=1}^K$ at time $t-1$, the implementation of our tracking algorithm is described as follows:

1. *Propagating*: According to the proposed state transition model (27), propagate each particle state s_{t-1}^k to get a new state s_t^k for time t.
2. *Re-sampling*: Based on the state s_t^k and respective histogram feature p_t^k, compute the weight π_t^k of each propagated particle at time t according to $p(p_t^k | x_t^k = s_t^k)$ and (25), and calculate matrix condition number c_t^k according to (26), then generate a new particle set $\{\bar{s}_t^k, \bar{\pi}_t^k = 1/K\}_{k=1}^K$ by concentrating on particles with large weights and small matrix condition numbers (the way is similar to that provided in [25]).
3. *Refining*: Run constrained KBOT for each re-sampled particle to move its state $\bar{s}_t^k$ to a more accurate state $\hat{s}_t^k$.
4. *Weighting*: Based on the state $\hat{s}_t^k$ and respective histogram feature $\hat{p}_t^k$, compute the weight $\hat{\pi}_t^k$ of each refined particle at time t according to $p(\hat{p}_t^k | x_t^k = \hat{s}_t^k)$ and (25), and normalize it according to (3).
5. *Estimating*: Calculate the object position at time t as $E(x_t) = \sum_{k=1}^K \hat{\pi}_t^k \hat{s}_t^k$.

We shall now proceed to provide a complexity analysis. To make the analysis clear and concise, we focus on the basic

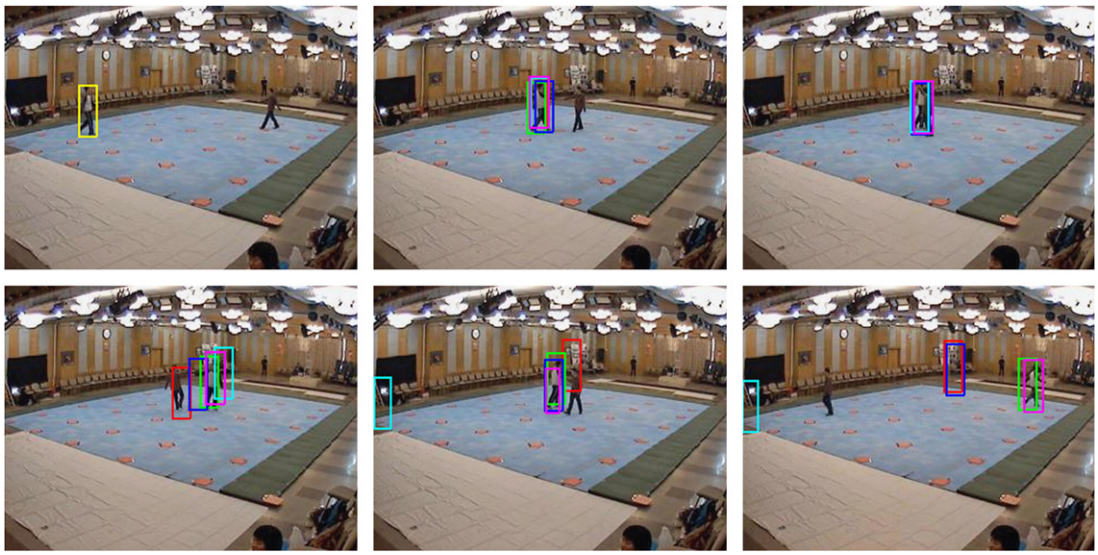

Fig. 4. Pedestrian sequence: the frames #1, #46, #77, #97, #140 and #198 are shown. The reference object in the first frame is represented in yellow rectangle, and the results of our tracker, PF tracker, BKOT tracker, the trackers of [29] and [31] are indicated in magenta, green, red, blue and cyan rectangles, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

operations needed for performing each step of the proposed tracking algorithm. Since histogram feature can be generated in a constant time via a pre-computed integral histogram [43], we omit its cost analysis here. Among the five steps listed above, we can easily see that the operations required to perform steps 1 and 5 are linearly proportional to the particle set size K. According to the definitions of (25) and (26), the operations required to calculate the IBD based weight and matrix condition number of every particle are $O(M^2)$ and $O(2\times M\times 2)$, respectively. Thus step 2 has a running time of $O(KM^2)$ operations. Note that step 4 mainly repeats the process of computing IBD based particle weight K times, it also takes $O(KM^2)$ operations. Denote a_t^k as the number of iterations performed when using constrained KBOT on the k^{th} re-sampled particle to seek a more accurate position state, the total iterations executed in step 3 is $\sum_{k=1}^{K} a_t^k$. Recall that we set the iteration upper bound in KBOT equal to 8, thus step 3 at most executes $8K$ iterations. From the above description, it can be concluded that the relatively expensive parts in the proposed tracking algorithm are the computation of IBD based particle weight and the running of iterative procedure in KBOT. By using the skills we previously proposed in [41], running IBD on each particle will asymptotically take $O(M)$ operations. Section 4.3 provides a processing speed comparison of our tracking algorithm and other related algorithms on real data.

4. Experimental results

Performance comparison of our association approach and related approaches on real data is given in this section.

4.1. Implementation

Our reference approaches include the regular PF, KBOT and two popular association based approaches [29,31]. Compared with the approach of [29], an adaptive state transition model is embedded in the algorithm of [31]. As we have already clarified in the previous section that the association structures of the approaches of [30–35] are basically similar to that of [29] irrespective of the other factors such as features and application environments. Thus under the same experimental settings, it can be supposed that the performance of tracking algorithms of [30–35] in a certain sense is comparative to that of [29]. The source code of [29] is provided by Dr. Chang, and the MATLAB codes of the other three reference algorithms are strictly implemented according to the pseudo-codes provided in [13,14] and [31], respectively, no computational optimization is considered. In PF tracker, the particle number is 500, and a widely used second order autoregressive model

$$x_t=\beta x_{t-1}+\gamma x_{t-2}+v_t \quad \text{where } v_t\sim N(0,1),\quad \beta=2,\quad \gamma=-1 \tag{29}$$

is used as the state transition model. In KBOT tracker, when the difference between the results of two continuous iteration steps is equal to or less than 0.8, the mean shift procedure is stopped. As for the two association based trackers, we completely follow the default parameter settings reported in [29] and [31], respectively. Specifically, we set $N=50$, $K(x)=\exp^{\|x\|^2}$, $I=3$, $f(\lambda_0,i)=2^{-i}\lambda_0$ and $n_t\sim N(0,3)$ for the tracker of [29], while we set $\sigma_x^0=\sigma_y^0=14$, $\sigma_h^0=0.13$, $k_p=10$, $k_s=5$ and $N_s=30$ for the tracker of [31]. Our tracking algorithm is also implemented in MATLAB, and 50 particles are used to maintain multiple hypotheses. Note that color histograms have already been used for representing object in four reference approaches, thus to have a fair comparison, we also specify color histograms for object representation in the experiments, and color histograms are generated in RGB color space with $8\times 8\times 8$ bins. Considering that our algorithm, PF algorithm, the algorithms of [29] and [31] all belong to the statistical algorithms, we run them in 20 times and take the averaged results as their final outputs.

On a desktop PC (2.5 GHz Intel Pentium 4 processor, 512 MB RAM, 120 GB hard disk), we run these five trackers on a number of real video sequences most of which are publicly available from [44,45]. For each test video sequence, the target in the first frame is initialized with a hand-drawn rectangle region and tracking results are also indicated with rectangle regions. In the following subsections, we just present representative results on several real video sequences containing different objects and challenges to show the efficacy of the proposed association approach.

4.2. Comparison of tracking performance

The first set of experiments is done on a video sequence containing 235 320×240-pixel color images. In this video, the

target is a pedestrian which moves across a dancing room with a relatively stable velocity while mainly experiencing background clutter (e.g., frames #1, #97, #140, #198) and partial occlusion (e.g., from frames #70 to #85). Fig. 4 shows some frames of this video sequence. The root square error (RSE) between the estimated object center and the manually labeled ground truth is plotted for each frame in Fig. 8a, and the average RSEs of five trackers over all frames of each test video sequence are presented in Table 1. In Fig. 4, the yellow rectangle in the first frame shows the reference object, and the magenta, green, red, blue and cyan rectangles in the other frames show the results of our tracker, PF tracker, KBOT tracker, the trackers of [29] and [31], respectively. Note that KBOT tracker fails when the color values of the surrounding region become very similar to those of the pedestrian at frame #172, and it cannot recover from the lost from then on. Compared with KBOT tracker, PF tracker shows much better capacity to deal with temporary failure. However, it temporarily loses the pedestrian several times when partial occlusion is occurred together with serious background clutter (e.g. from frames #101 to #106). The tracker of [29] sometimes exhibits a bit worse performance than PF tracker in the cases of background clutter and partial occlusion, and it completely loses the target at frame #185. Due to the fact that the tracker of [31] entirely loses the pedestrian from frame #127 to the last, its average RSE on this video sequence is largely magnified as shown in Table 1. As a result, the tracker of [31] generally shows the worst performance among five trackers. The unsatisfactory performance of the trackers of [29] and [31] is mainly due to the fact that KBOT is not suitable for refining the position states of arbitrary particles for more accurate mode seeking. By invoking KBOT on the particles located in the cluttered background or positioned at the ill-posed conditions, these particles will easy to be moved toward false modes that are not close to the object position. In this case, as the tracking error is accumulated over time, tracking lost is likely to happen. However, this issue is not well addressed in the association approaches of [29] and [31]. The fact of the higher average RSEs of the trackers of [29] and [31] to that of PF tracker serves as a concrete example. As for our tracker, this issue is sufficiently considered. By encoding the spatial-temporal attributes of the target, the IBD based measure can reliably distinguish the foreground particles from the background particles in the scenarios of cluttered background and partial occlusion. Further, the matrix condition number based measure discards the particles located at ill-posed conditions for invoking KBOT. Consequently, the selected particles have higher probability to move towards more accurate modes in comparison with the neglected particles. According to Figs. 4, 8a and Table 1, our association approach shows much better capability to suppress the influences of cluttered background and partial occlusion in comparison with the association algorithms of [29] and [31], and the best tracking performance on this video sequence has also been obtained by our association approach.

Table 1
Comparison of the average RSEs for five trackers on four video sequences (in pixels, all frames are considered).

Video sequence	PF	KBOT	[29]	[31]	Our tracker
Pedestrian	7.88	28.68	23.69	102.29	5.65
Man's face	26.63	32.68	8.73	11.64	5.48
Ping-pong ball	39.00	37.18	18.73	12.41	5.00
Girl's face	100.95	12.36	29.18	28.48	5.22

Next, in the second set of experiments, we mainly consider visual tracking under fast movement conditions. The experiments are implemented on a challenging video sequence studied and provided by Birchfield [44]. The video sequence is composed of 128×96-pixel color images recorded under laboratory conditions, in which a man's face undergoes back-and-forth movement with a fast speed (e.g., from frames #1 to #4, from frames #13 to #16). There also exist unpredictable camera motion and transient variations in face pose. Fig. 5 shows some frames of this video sequence, and the RSE curves are shown in Fig. 8b. Note that KBOT tracker quickly drifts away from the target when there is little or no overlap between the face regions in the two consecutive frames. This can be attributed to the limitation of maximum iteration step size of gradient based mean shift optimization. Although PF tracker shows better performance than BKOT tracker, its tracking accuracy is not high because it is not well adapted to handling the scenario of fast movements where most of the particles are usually not located in the target region. Comparatively speaking, since no serious background clutter or partial occlusion is available in this video sequence, the tracker of [31] does not reach the man's face only at frame #21, while the tracker of [29] exhibits even more accurate results. Specifically, on this video sequence, the average RSEs of the trackers of [29] and [31] are 8.73 and 11.64 pixels per frame, respectively, which

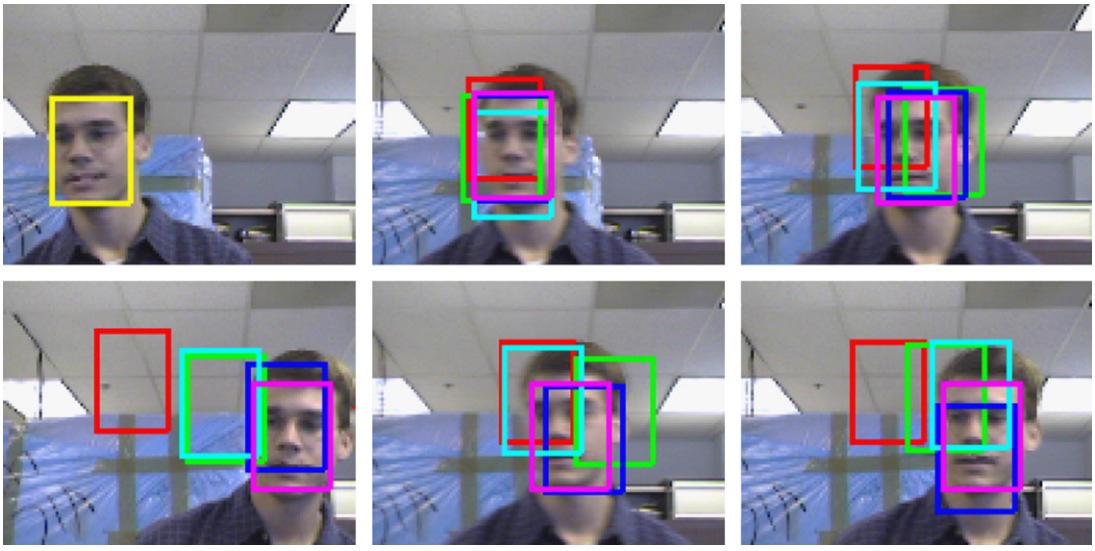

Fig. 5. Man's face sequence: the frames #1, #2, #13, #14, #21, and #23 are shown.

are much smaller than those of KBOT tracker and PF tracker as shown in Table 1. Recall that our tracker introduces a new state transition model which embodies object-scale oriented information and prior motion cues to handle particle diversity under fast movement scenarios. Once the particle diversity is reasonably guaranteed, the particles which are suitable for invoking KBOT for more accurate mode seeking can be picked out via our two-stage solution. Therefore, as can be seen from Fig. 8b and Table 1, our association approach accurately tracks the man's faces throughout the video sequence. Further, our approach provides comparable performance on this video sequence to the trackers of [17,29,44,46].

Subsequently, the third set of experiments is implemented on a popularly used ping-pang ball video sequence [14] containing 88 frames with resolution of 352×240 pixels. In the former frames of this video sequence, the ping-pong ball moves up-and-down with a fast speed (e.g., frames #9, #10, #12). In the later frames, partial occlusion (e.g., frames #63, #64, #65) and background clutter (e.g., frames #73, #75) are the main difficulties. Apart from that, there exist transient variations in ping-pong ball size over the video sequence. For these trackers, we use the linear filtering technique described in [14] to adapt object scale changes over time. With respect to the frames shown in Fig. 6, the RSE curves plotted in Fig. 8c and the average RSEs given in Table 1, it can be noticed that both PF and KBOT trackers temporarily fail when the ping-pong ball falls down with a fast speed in the former frames. Although the average RSE of KBOT tracker is a bit smaller than that of PF tracker, PF tracker shows better performance than KBOT tracker in the later frames where the target mainly undergoes partial occlusion and background clutter. By contrast, the trackers of [29] and [31] prove to be more robust, similar results and conclusions have also been reported in [31]. Here, we want to point out that [31] only reports the results on the former frames from #1 to #60. That is, the latter 28 frames of the ping-pong ball video sequence are not considered in [31]. For our tracker, with the auxiliary support of the proposed state transition model, the information of prior object size and motion cues is captured. On the other hand, our association approach uses IBD and matrix condition number to jointly measure the confidences of the propagated particles from the perspective of employing KBOT to seek more accurate modes. Benefited from these ingredients, our algorithm precisely tracks the ping-pong ball almost throughout the video sequence. Only when the ping-pong ball shrinks into a small size (less than 8×8 pixels) at frame #84, our tracker drifts off the target. Specifically, the average RSE of our tracker on this video sequence is 5 pixels per frame. Further, the results of our approach on the frames from #1 to #60 are also comparative to the trackers of [31], and the integrated approach of Kalman filter and KBOT [14] (in which the latter 28 frames are also not considered).

In the fourth set of experiments, we further evaluate the robustness of five trackers to handle hybrid difficulties including object appearance change (e.g., frames #1, #81, #117, #172), partial occlusion (e.g., from frames #440 to #464), cluttered background (e.g., frames #8, #331, #440) and slight object scale variation (e.g., frames #1, #117, #331). In the experiments, a widely used girl's face video sequence [44] consists of 501 frames with resolution of 128×96 pixels is used. Some frames of this video sequence are illustrated in Fig. 7, and Fig. 8d shows the RSE curves. It can be observed that PF tracker fails at frame #88 where the girl's face deforms non-rigidly in pose under cluttered background conditions, and it cannot recover the lost from then on. As a result, PF tracker comparatively produces the worst average RSE than reference trackers as shown in the last row of Table 1. The main reason is that Bhattacharyya coefficient is not discriminative enough to measure particles located in the cluttered environments. As for KBOT algorithm, the tracked face center is placed within the target region for almost all frames of the video, but its accuracy is not consistently high. When the man's face gets close to and further occludes the girl's face, KBOT track quickly drifts to the man's face. Comparatively speaking, the trackers of [29] and [31] are proved to be more robust than PF tracker. However, these

Fig. 6. Ping-pong ball sequence: the frames #1, #9, #10, #12, #64, #73, #75, #82, #83 and #84 are shown.

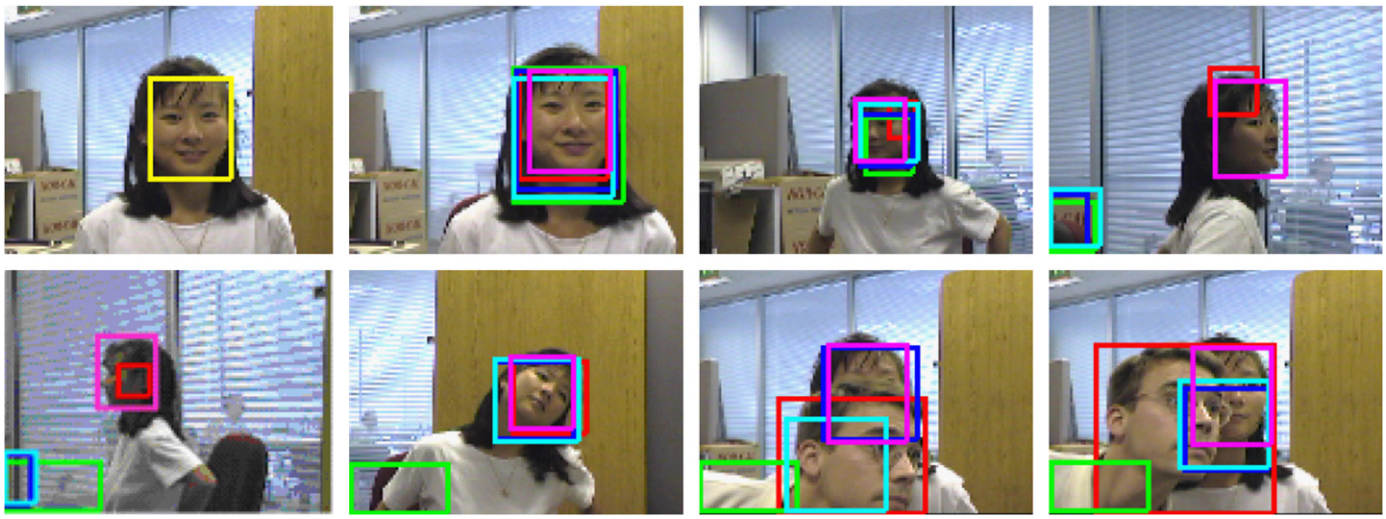

Fig. 7. Girl's face sequence: the frames #1, #8, #81, #117, #172, #331, #440 and #464 are shown.

two association based trackers also temporarily lose the object in the serious background clutter scenarios (e.g., from frames #100 to #150). As we described beforehand, this can be traced to the fact that direct implementing KBOT on arbitrary particles may incur unexpected tracking errors. Since this issue is properly addressed in the proposed approach, our tracker again achieves more accurate results on this video sequence compared with the other four reference trackers.

4.3. More analysis

In this subsection, we proceed with more analysis. First, we compare the tracking precision of five different algorithms over respective successful tracks on each test video sequence. Unlike the way of calculating the average RSEs shown in Table 1, in the statistical process, when an estimated target is sufficiently far from the ground truth (e.g., there is no overlap between the estimate target and the ground truth), it is considered as a "lost track" and the error is not accumulated in the accuracy measure. That is, the tracking accuracy has a value of interest in the comparison only when all trackers under analysis are close to the real target. Following this protocol, the average RSE on each video sequence is computed for every tracker. Table 2 presents the detailed results. Note that the average RSE of our tracker is 5.65, 5.48, 4.00 and 5.22 pixels per frame for pedestrian, man's face, ping-pong ball and girl's face video sequences, respectively, which is more accurate than those of the other four trackers. Further, two association based trackers [29,31] usually shows more accurate tracking results than PT tracker and KBOT tracker, which implies that if association approach is effective, better tracking performance can be achieved. Up to now, we can conclude that our tracker is more robust than the other four reference trackers to handle complex tracking tasks. The favorable performance of our algorithm can be attributed to several factors. The most important one is our two-stage solution which presents a reliable way to suppress the influence of particles located in the background or placed at the ill-conditioned positions. Besides that, the proposed state transition model is really adapted to keeping particle diversity in the fast movement scenarios.

Second, empirical results show that a tracking algorithm with scale adaptation usually exhibits better performance in contrast with the one having no scale adaptation [14,23,47]. Here, we further implement another set of experiments on the ping-pong ball sequence to exploit the influence of scale changes. In the experiments, our tracker is run in two different ways. In the first way, scale adaption is not considered, while in the other way it is considered. Fig. 9 depicts the RSE curves of two trackers. Note that without scale adaptation, the tracking errors of most frames are rather small. However, the tracker drifts off the target at frame #72 when the ping-pong ball size largely shrinks. In addition, the average RSE over successful tracks increases from 4.00 to 6.28 pixels. Therefore, with scale adaptation, our approach performs more accurate and robust. These results clearly demonstrate that scale adaptation acts as a boosted ingredient to tracking performance in the scenario of object scale changes, especially for drastic scale changes. This topic has been explored in [47,48] and our previous work [49].

Finally, the computational cost of a tracking algorithm usually affects its application in real-time environments. Although we have already stated the complexity of the proposed tracking algorithm, it is still necessary to compare its actual processing speed with those of the other reference algorithms. To this purpose, the overall running time of each tracking algorithm has been recorded for each video sequence during the experiments. Recall that all these tracking algorithms are implemented in MATLAB, we compute the average processing speed of each tracker so that to have a simple comparison of computation cost. For KBOT tracker, the averaged processing speed is about 9–14 frames per second (fps). For PF tracker, it is only about 0.3–0.6 fps. Our approach achieves an average processing speed of 1–3 fps. Since the number of particles is same for [29] and our tracker, the average processing speed of [29] is similar to that of our approach. As for the tracker of [31], it exhibits a slower speed because it applies mean shift procedure on each particle for two separate times.

5. Conclusion

Visual tracking is a hot research topic in computer vision. In this paper, we address visual tracking via considering the association of PF and KBOT. The main purpose of PF and KBOT association is to build up a more robust visual tracking approach via combining their strengths and alleviating their weaknesses. To achieve this purpose, the following items have been explored. (1) For the problem of what kind of particles are fit for applying KBOT to refine their position states for more accurate mode seeking, it has been theoretically analyzed. (2) In accordance

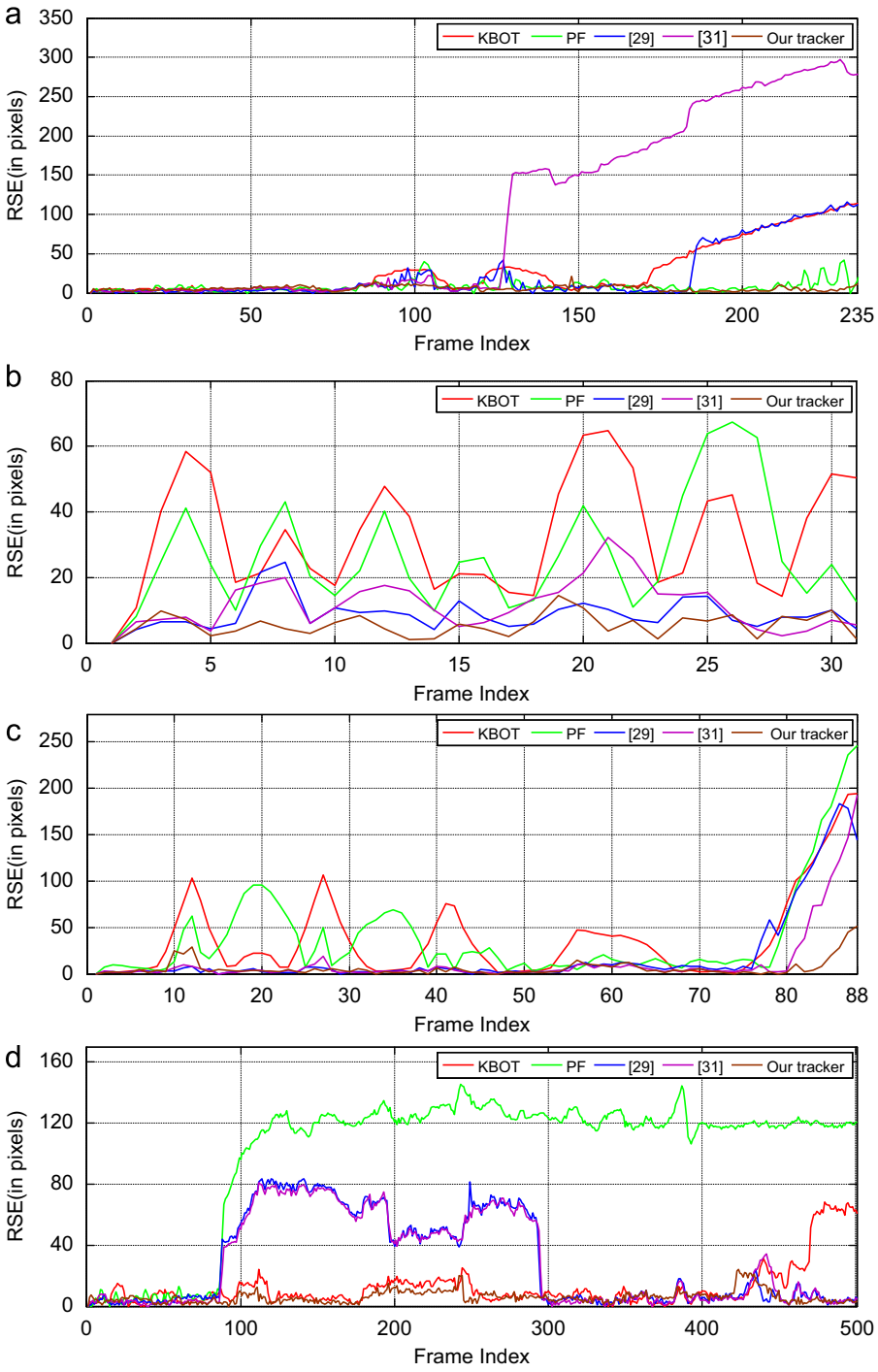

Fig. 8. Comparison of the RSE between our tracker and the other four trackers on four video sequences. (a) Pedestrian. (b) Man's face. (c) Ping-pong ball. (d) Girl's face.

Table 2
Comparison of the average RSEs for five trackers on four video sequences (in pixels, lost tracks are discarded).

Video sequence	PF	KBOT	[29]	[31]	Our tracker
Pedestrian	7.73	13.89	6.09	6.00	5.65
Man's face	18.30	20.90	8.73	11.64	5.48
Ping-pong ball	13.06	12.41	4.79	4.62	4.00
Girl's face	5.81	9.02	5.74	5.93	5.22

with the theoretical analysis, the problem of how to design a computationally tractable way to pick out propagated particles that are well suited for invoking KBOT is also addressed. As a result, a two-stage solution is proposed. In our solution, the IBD based stage is devoted to distinguishing the particles located in the object region from those placed in the background, and the matrix condition number based stage is dedicated to eliminating the particles positioned at the ill-posed conditions for invoking KBOT. (3) To deal with the difficulties caused by fast

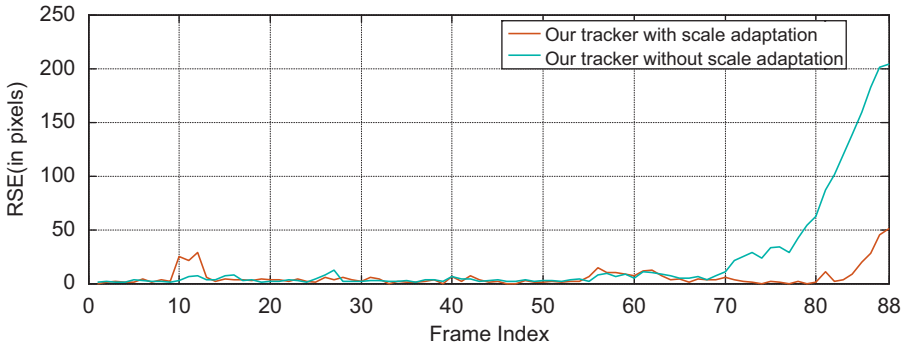

Fig. 9. A comparison of our approach with and without scale adaptation on ping-pong ball sequence.

movement scenarios, a state transition model embodying object-scale oriented information and prior motion cues is proposed. Also, the conventional gradient based mean shift optimization is modified by discarding the later unnecessary iterations. The efficacy of the proposed association approach has been fully demonstrated by comparative experiments.

Although we have presented the theoretical and computational solutions to the key problem in the association of PF and KBOT, there are still some aspects that deserved further study. Color histograms are used as the object descriptors for performing fair comparison of our association approach and related methods in the experiments. However, as we mentioned in the introduction section, features play an important role in practical tracking tasks. As for color histograms, they are not well adapted to handling varying light conditions, etc [17,19]. Therefore, in our future work, we first plan to combine heterogeneous features such as color, texture and shape into the proposed association algorithm for achieving more robust tracking performance. We do not directly address the long-term object appearance changes in the current work. However, this problem can be effectively resolved by appropriately updating the target model [13,36,41]. Particle sampling is an important component of PF. When the particles are sparsely sampled and the object is not covered in search region, PF algorithms will not converge to the true mode of the target any more. This is an open issue in PF [25].

Acknowledgments

Part of the work was done when author Anbang Yao was studying at the Department of Electronic Engineering, Tsinghua University, Beijing, China. The authors would like to thank Dr. Chang for providing the source code of hybrid particle filter for comparisons.

References

[1] W. Hu, T. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object motion and behaviors, IEEE Transactions on Systems, Man, and Cybernetics C 34 (3) (2004) 334–352.
[2] H. Wang, D. Suter, A consensus-based method for tracking: modelling background scenario and foreground appearance, Pattern Recognition 40 (3) (2007) 1091–1105.
[3] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P.J. Nordlund, Particle filters for positioning, navigation, and tracking, IEEE Transactions on Signal Processing 50 (2) (2002) 425–437.
[4] B. Coifmana, D. Beymer, P. McLauchlan, J. Malik, A real-time computer vision system for vehicle tracking and traffic surveillance, Transportation Research Part C: Emerging Technologies 6 (4) (1998) 271–288.
[5] J.M. Rehg, T. Kanade, DigitEyes: vision-based hand tracking for human–computer interaction, in: Proceedings of the IEEE Workshop on Motion of Non-rigid and Articulated Objects, 1994, pp. 16–22.
[6] B. Stenger, A. Thayananthan, P.H.S. Torr, R. Cipolla, Model-based hand tracking using a hierarchical Bayesian filter, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (9) (2006) 1372–1384.
[7] S. Choi, D. Kim, Robust head tracking using 3D ellipsoidal head model in particle filter, Pattern Recognition 41 (9) (2008) 2901–2915.
[8] H. Li, D. Doermann, O. Kia, Automatic text detection and tracking in digital video, IEEE Transactions on Image Processing 9 (1) (2000) 147–156.
[9] Z. Yin, R. Collins, On-the-fly object modeling while tracking, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
[10] C. Kim, J. Hwuang, Fast and automatic video object segmentation and tracking for content-based applications, IEEE Transactions on Circuits and Systems for Video Technology 12 (2) (2002) 122–129.
[11] K. Hotta, Adaptive weighting of local classifiers by particle filters for robust tracking, Pattern Recognition 42 (5) (2009) 619–628.
[12] C.R. Wren, A. Azarbayejani, T. Darrell, A.P. Pentland, Pfinder: real-time tracking of the human body, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 780–785.
[13] M. Isard, A. Blake, Condensation—conditional density propagation for visual tracking, International Journal of Computer Vision 29 (1) (1998) 5–28.
[14] D. Comaniciu, V. Ramesh, P. Meer, Kernel based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 564–577.
[15] S. Avidan, Ensemble tracking, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2005, pp. 494–501.
[16] R.T. Collins, Online selection of discriminative tracking features, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (10) (2005) 1631–1643.
[17] H. Wang, D. Suter, K. Schindler, C. Shen, Adaptive object tracking based on an effective appearance filter, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (9) (2007) 1661–1667.
[18] D.A. Ross, J. Lim, R.S. Lin, M.H. Yang, Incremental learning for robust visual tracking, International Journal of Computer Vision 77 (1–3) (2008) 125–141.
[19] B. Babenko, M. Yang, S. Belongie, Visual tracking with online multiple instance learning, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2009, pp. 983–990.
[20] X. Mei, H. Ling, Robust visual tracking using l1 minimization, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2009.
[21] S.T. Birchfield, S. Rangarajan, Spatiograms versus histograms for region-based tracking, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2005, pp. 1158–1163.
[22] E. Ozyildiz, N. Krahnstover, R. Sharma, Adaptive texture and color segmentation for tracking moving objects, Pattern Recognition 35 (10) (2002) 2013–2029.
[23] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Computing Surveys 38 (4) (2006) 1–45.
[24] B. Han, D. Comaniciu, Y. Zhu, L. Davis, Incremental density approximation and kernel-based Bayesian filtering for object tracking, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2004, pp. 638–644.
[25] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear /non-Gaussian Bayesian tracking, IEEE Transactions on Signal Processing 50 (2) (2002) 174–188.
[26] A. Elgammal, R. Duraiswami, D. Harwood, L.S. Davis, Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, Proceedings of the IEEE 90 (7) (2002) 1151–1163.
[27] J. Jia, Q. Wang, Y. Chai, R. Zhao, Object tracking by multi-degrees of freedom mean shift procedure combined with the Kalman particle filter algorithm, in: Proceedings of the IEEE International Conference on Machine Learning and Cybernetics (ICMLC), 2006, pp. 3793–3797.
[28] C. Yang, R. Duraiswami, L. Davis, Efficient mean-shift tracking via a new similarity measure, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2005, pp. 176–183.
[29] C. Chang, R. Ansari, Kernel particle filter for visual tracking, IEEE Signal Processing Letters 12 (3) (2005) 242–245.
[30] K. Deguchi, O. Kawanaka, T. Okatani, Object tracking by the mean-shift of regional color distribution combined with the particle-filter algorithm, in: Proceedings of the IEEE International Conference on Pattern Recognition (ICPR), 2004, pp. 506–509.

[31] E. Maggio, A. Cavallaro, Hybrid particle filter and mean shift tracker with adaptive transition model, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2005, pp. 221–224.
[32] C. Shan, Y. Wei, T. Tan, F. Ojardias, Real time hand tracking by combining particle filtering and mean shift, in: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FGR), 2004, pp. 669–674.
[33] C. Chang, R. Ansari, Kernel particle filter: iterative sampling for efficient visual tracking, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2003, pp. 977–980.
[34] C. Chang, R. Ansari, A. Khokhar, Multiple object tracking with kernel particle filter, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2005, pp. 566–573.
[35] C. Shan, T. Tan, Y. Wei, Real-time hand tracking using a mean shift embedded particle filter, Pattern Recognition 40 (7) (2007) 1958–1970.
[36] K. Nummiaro, E. Koller-Meier, L. van Gool, An adaptive color-based particle filter, Image and Vision Computing 21 (1) (2003) 99–110.
[37] Y. Rui, Y. Chen, Better proposal distributions: object tracking using the unscented particle filter, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2001, pp. 786–793.
[38] C. Sminchisescu, B. Triggs, Covariance scaled sampling for monocular 3D body tracking, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2001, pp. 447–454.
[39] G.D. Hager, M. Dewan, C.V. Stewart, Multiple kernel tracking with SSD, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2004, pp. 790–797.
[40] J. Lin, Divergence measures based on the Shannon entropyInform, IEEE Transactions on Information Theory 37 (1) (1991) 145–151.
[41] A. Yao, G. Wang., X. Lin, X. Chai, An incremental Bhattacharyya dissimilarity measure for particle filteirng, Pattern Recognition 43 (4) (2010) 1244–1256.
[42] Z. Fan, M. Yang, Y. Wu, G. Hua, T. Yu, Efficient optimal kernel placement for reliable visual tracking, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2006, pp. 658–665.
[43] F. Porikli, Integral histogram: a fast way to extract histograms in Cartesian spaces, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2005, pp. 829–836.
[44] 〈http://vision.stanford.edu/~birch/headtracker/seq/〉.
[45] 〈http://media.xiph.org/video/derf/〉.
[46] F. Dornaika, F. Davoine, On appearance based face and facial action tracking, IEEE Transactions on Circuits and Systems for Video Technology 16 (9) (2006) 1107–1124.
[47] R.T. Collins, Mean-shift blob tracking through scale space, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2003, pp. 234–240.
[48] Yilmaz A., Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–6.
[49] A. Yao, G. Wang, X. Ling, H. Wang, Kernel based articulated object tracking with scale adaptation and model update, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2008, pp. 945–948.

Anbang Yao received his B.S. degree from Nanjing University of Science and Technology, China in 2002, and received his M.S. degree and Ph.D. degree from Tsinghua University, China in 2005 and 2010, respectively. Currently, he is a postdoctoral research fellow at National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science. His research interests include visual tracking, object detection, face and gesture analysis, human computer interaction and optimization methods, etc.

Xinggang Lin received the Ph.D. degree and M.S. degree both in information science from Kyoto University, Japan in 1986 and 1982, respectively, and received the B.S. degree in electronic engineering from Tsinghua University, China in 1970. He joined the Department of Electronic Engineering, Tsinghua University, Beijing, China in 1986, where he has been a full professor since 1990.

Guijin Wang received the B.S. degree and Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, China in 1998 and 2003, respectively. In 2003, he joined the Information Technologies Laboratories, Sony Corporation, Japan as a researcher. Since 2006, he has been with the Department of Electronic Engineering, Tsinghua University, China as an associate professor.

Shan Yu received her B.S. degree in biomedical instrumentation from Shanghai Jiao Tong University, China in 1985, and received her Ph.D. degree in science of engineering from University of Nice-Sophia Antipolis, France in 1992. She joined the French National Institute for Research in Computer Science and Control, INRIA as a research scientist in 1993. From 2000 to 2009, she took leave from INRIA and joined the Medical Center of Columbia University and New York State Psychiatric Institute. Now, she has come back to INRIA. Her research interests include image processing, patter recognition and the understanding of neural basis of psychiatric disorders through the use of fMRI and PET.