Multimed Tools Appl (2014) 72:573–589
DOI 10.1007/s11042-013-1379-y

Robust particle tracker via Markov Chain Monte Carlo posterior sampling

Fasheng Wang · Mingyu Lu

Published online: 19 February 2013

Abstract Particle Filter has grown to be a standard framework for visual tracking. This paper proposes a robust particle tracker based on Markov Chain Monte Carlo method, aiming at solving the thorny problems in visual tracking induced by object appearance changes, occlusions, background clutter, and abrupt motions. In this algorithm, we derive the posterior probability density function based on second order Markov assumption. The posterior probability density is the joint density of the previous two states. Additionally, a Markov Chain with certain length is used to approximate the posterior density to avoid the drawbacks of traditional importance sampling based algorithm, which consequently improves the searching ability of the proposed tracker. We compare our approach with several alternative tracking algorithms, and the experimental results demonstrate that our tracker is superior to others in dealing with various types of challenging scenarios.

Keywords Visual tracking · Particle filter · Markov Chain Monte Carlo

1 Introduction

Visual tracking gains special attention in computer vision community due to its success in various real-world applications, such as intelligent visual surveillance, human-computer interaction, traffic monitoring or video indexing [15]. It has recently been addressed in real-world scenarios rather than a lab environment by many researchers, because it is much more challenging in complex real-world settings. Among the large amount of algorithms proposed in the literature, particle filter based tracking method has been applied with great success in solving various kinds of visual tracking problems. The basic idea behind such a method is to approximate the posterior probability density function recursively with a set of weighted particle (or sample) set evolving in the state space. The estimated object states can be as close to the optimal states as possible.

F. Wang · M. Lu (✉)
School of Information Science and Technology, Dalian Maritime University, Information Building B-106, No.1 Linghai Road, Ganjingzi District, Dalian, Liaoning Province, China 116026
e-mail: lumingyu@dlmu.edu.cn

F. Wang
e-mail: fswang@dlmu.edu.cn

Isard M [6] first used the particle filter, which was called CONDENSATION, to solve visual tracking problem. Thereafter, many researchers have been working on this topic. Martínez-del-Rincón J [10] proposed a Rao–Blackwellised particle filter (RBPF) based tracking algorithm, aiming at handling the uncertainties induced by illumination change and short-time occlusions. He introduced a joint image characteristic-space tracking scheme which updates the model simultaneously to the object location, and the RBPF is used to avoid the curse of dimensionality. However, the algorithm can be failed in the face of long time full occlusions, or abrupt change of object position. Khan Z [7] adopted Markov random field (MRF) based motion prior model accompanied with Markov Chain Monte Carlo method within the particle filtering framework, to track multiple interacting ants. But the proposed tracking scheme only works well in lab environment with static background, and tends to fail in real-world scenarios. On the basis of Khan's work, Cong T [3] et al. proposed a new visual tracking algorithm within the particle filtering framework, which combined the color-based observation model with a detection confidence density obtained from the histograms of oriented gradients (HOG) descriptor. The algorithm showed improved robustness in slight object occlusions with static background. However, the authors did not take into account the complex scenarios with severe appearance change and full occlusion, and the algorithm can only yield limited improvement. Choo K [2] proposed a hybrid Monte Carlo (HMC) method for 3D human motion inference, which is much faster than the particle filter in 28D state spaces. But the authors did not use the algorithm in real-world applications, and it tends to fail when several similar objects appeared in the scenario.

Most of the algorithms in the literature are generally based on smooth motion assumption, that is, the target being tracked is moving with stable motion. However, in many real world applications, abrupt motions often appear due to camera switching, low frame rate video and uncertain object dynamics, which could cause the conventional tracking approach to fail since they violate the motion smoothness constraint. Kwon J [8] proposed a Wang-Landau Monte Carlo (WLMC) based tracking algorithm within the Bayesian filtering framework. The algorithm adopted a density of states (DOS) based prior distribution, and the images containing the object are divided into several subspaces with equal size. The tracker is guided to update the object state using the DOS computed online. Experimental results demonstrated that the tracker can handle different kinds of abrupt motions. But there is no rigorous convergence theory to support its convergence, and it could only achieve limited precision in statistical and physical applications [9]. Based on this work, Zhou [16] proposed an adaptive stochastic approximation Monte Carlo (ASAMC) sampling based particle tracking algorithm. The authors constructed explicitly a DOS based trial distribution to replace the traditional filtering distribution as the proposal distribution, and all the samples are drawn from this trial distribution using the Metropolis-Hastings (MH) algorithm. The algorithm showed improved efficiency and accuracy in handling various abrupt motions. However, both the WLMC and the ASAMC based tracking algorithms need firstly divide the sample space (the current video frames) into several equal-size subregions. But the division decreases the algorithms' efficiency when the size of the frame is moderately large. Wang [13] proposed a Hamiltonian Monte Carlo (HMC) estimator for abrupt motion tracking within the Bayesian filtering framework. The HMC is based on Hamiltonian dynamics implemented by Leapfrog iteration which is the same as [2]. The random walk behavior occurred in many MCMC based tracking algorithms can be suppressed. Thus, the HMC algorithm can be avoided being trapped in local maxima during tracking. The main drawback of the HMC is that it is difficult to choose an appropriate step size for Leapfrog iterations. The Gibbs sampling in the proposal step still suffers from random walk behavior.

In addition, the WLMC, ASAMC, and HMC are all sensitive to background clutter and lighting condition changes. Moreover, they failed to track the object when the appearance changes drastically or the object is occluded persistently by obstacles.

As is known to all, most of the particle filter based tracking algorithms in the literature are all based on first-order Markov assumption, that is, the current state at time t only depend on the state at time $t-1$. Although the assumption can simplify the expression of the posterior probability density function and the implementation of visual tracking, it suffers several disadvantages. On the one hand, the first-order assumption cannot accurately model the dynamics of the tracking objects. On the other hand, if the particles at time $t-1$ are lost or delayed, the performance of the algorithm would be severely affected.

In this paper, we address the tracking problems in different complex scenarios and proposed a new tracking algorithm based on Markov Chain Monte Carlo posterior sampling within the particle filtering framework. Firstly, the algorithm is based on second-order Markov assumption. We assume that the object state x_t depends on the states of the previous two time instant, x_{t-1} and x_{t-2}. This assumption can enhance the robustness of the tracker. Secondly, we adopt a Markov Chain with certain length to approximate the posterior probability density function discarding the traditional importance sampling based methods which tends to fail due to sample impoverishment. The Markov Chain is simulated using the MH algorithm; consequently the posterior density function is approximated using a set of unweighted sample sets $\{x_t^i\}_{i=1}^N$. This strategy can avoid the sample impoverishment problem suffered by traditional particle filter based trackers and avoid the local trap problem during visual tracking. We name the proposed algorithm as 2MCMC-PF. The experimental results demonstrated that the proposed tracking algorithm showed improved robustness and accuracy in different types of challenging tracking scenarios.

2 Bayesian tracking and particle filter

2.1 Bayesian tracking framework

Suppose the object state x_t at time t is composed of position and scale information, $x_t = \left(x_t^p, x_t^s\right)$, where x_t^p is represented as the center of the object, and x_t^s is represented using the height and width of the object. The object tracking problem can be formulated as the Bayesian filtering problem, that is to say, recursively estimate the hidden state variable x_t, given a series of observations $z_{1:t}=\{z_1, z_2, \ldots, z_t\}$ up to time t. The posterior probability density of state variable x_t is $p(x_{0:t}|z_{1:t})$, but the filtering density $p(x_t|z_{1:t})$ is often adopted which can be estimated by

$$p(x_t|z_{1:t}) = cp(z_t|x_t)\int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1} \tag{1}$$

where $p(x_t|x_{t-1})$ is the system transition model describing the state evolution, $p(z_t|x_t)$ is the observation model, and c is a normalizing constant. After obtaining the posterior probability $p(x|z_{1:t})$, we can compute the Maximum a Posteriori (MAP) estimate over the sample set,

$$x_t^{MAP} = \arg\max p\left(x_t^{(i)}|z_{1:t}\right), i = 1, \ldots, N \tag{2}$$

The estimation x_t^{MAP} calculated by (2) is considered as the best state estimates for the current time instant. But it is usually infeasible to calculate the integral in (1), especially for high dimensional state space.

2.2 Particle filter

The particle filter is based on importance sampling, which uses a set of weighted samples $\{x_t^i, \omega_t^i\}_{i=1}^{N_s}$ to approximate the posterior probability distribution,

$$p(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^{N_s} \omega_k^i \delta\left(x_{0:k} - x_{0:k}^i\right) \tag{3}$$

All the samples are drawn from a so called importance density $q(\bullet)$, and the sample weights under first-order Markov assumption is updated recursively using,

$$\omega_k^i \propto \omega_{k-1}^i \frac{p\left(z_k|x_k^i\right)p\left(x_k^i|x_{k-1}^i\right)}{q\left(x_k^i|x_{k-1}^i, z_k\right)} \tag{4}$$

Details of the derivation of (4) are discussed in [1]. A main drawback of the conventional importance sampling based particle filter is the sample impoverishment problem, which can greatly deteriorate the performance of the tracking algorithm in real-world applications leading to local-trap problem especially in multiple objects tracking. Although many improvement strategies have been proposed in the literature, the applications of these are very limited in visual tracking.

3 Markov Chain Monte Carlo sampling based particle tracker

3.1 Second-order Markov assumption

The second-order Markov assumption assumes that the current state x_t depends on the previous two states x_{t-1}, x_{t-2}, so we can obtain

$$p(x_t|x_{0:t-1}) = p(x_t|x_{t-2:t-1}) \tag{5}$$

Figure 1 shows the states evolution of second-order Markov process.

Based on this assumption, the posterior probability density $p(x_{0:t}|z_{1:t})$ is derived as follows:

$$p(x_{0:t}|z_{1:t}) = \frac{p(z_t, x_{0:t}, z_{1:t-1})}{p(z_t, z_{1:t-1})} \tag{6}$$

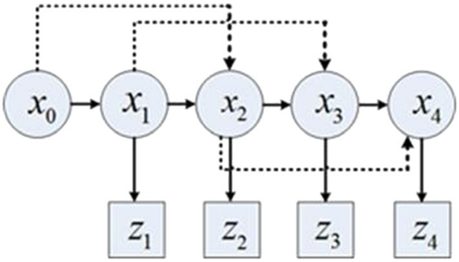


Fig. 1 Second order Markov process. The circle nodes denote the object states, while the square nodes represent the observations correspond to the hidden states. The solid line denotes the first-order Markov model and the dashed line represent the second-order Markov process

$$=\frac{p(z_t|x_{0:t},z_{1:t-1})p(x_{0:t},z_{1:t-1})}{p(z_t,z_{1:t-1})} \tag{7}$$

$$=\frac{p(z_t|x_{0:t},z_{1:t-1})p(x_{0:t}|z_{1:t-1})p(z_{1:t-1})}{p(z_t|z_{1:t-1})p(z_{1:t-1})} \tag{8}$$

$$=\frac{p(z_t|x_{0:t},z_{1:t-1})p(x_{0:t}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \tag{9}$$

$$=\frac{p(z_t|x_{0:t},z_{1:t-1})p(x_t,x_{0:t-1}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \tag{10}$$

$$=\frac{p(z_t|x_{0:t},z_{1:t-1})p(x_t|x_{0:t-1},z_{1:t-1})p(x_{0:t-1}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \tag{11}$$

$$\propto p(z_t|x_{0:t},z_{1:t-1})p(x_t|x_{0:t-1},z_{1:t-1})p(x_{0:t-1}|z_{1:t-1}) \tag{12}$$

$$=p(z_t|x_t)p(x_t|x_{t-2:t-1})p(x_{0:t-1}|z_{1:t-1}) \tag{13}$$

The filtering density $p(x_t|z_{1:t})$ is formulated as:

$$p(x_t|z_{1:t}) \propto p(z_t|x_t)p(x_t|x_{t-2:t-1})p(x_{t-1}|z_{1:t-1}) \tag{14}$$

The following conditional probability density equation is used during the derivation:

$$p(m|n)=\frac{p(m,n)}{p(n)} \tag{15}$$

$$p(m,n|l)=p(m|n,l)p(n|l) \tag{16}$$

Our aim is to estimate the joint posterior probability density (14).

3.2 Markov Chain posterior sampling

The MCMC method constructs a Markov Chain in the state space to approximate the posterior distribution $p(x_t|z_{1:t})$ that converges to the stationary distribution $\pi(x)$. It is typically used to search the state space for the MAP estimates, or introduced into the particle filtering framework to suppress the sample impoverishment problem. Gilks [5] proposed the MCMC based improved particle filtering algorithm, the basic idea of which is to add a MCMC move step after the resampling process, which can guide the samples to move toward the more promising area. In this paper, we use the MCMC method within the particle filtering framework to construct a Markov Chain. Samples are drawn from this chain yielding an unweighted sample set $\{x_t^i\}_{i=1}^{N_s}$ which is used to approximate the filtering posterior probability density $p(x_t|z_{1:t})$, that is,

$$p(x_t|z_{1:t}) = \frac{1}{N_s}\sum_{j=1}^{N_s}\delta\left(x_t - x_t^j\right) \tag{17}$$

In this way, the conventional importance sampling procedure can be avoided, while the MCMC method can enhance the searching ability of the particle filter in the state space.

The classical algorithm to construct the Markov Chain is the MH algorithm [4]. For the given target posterior density $p(x_t|z_{1:t})$, the algorithm starts from a certain initial sample x^0. The samples are drawn from a so called proposal distribution in order to define the Markov Chain of time*t*. Suppose the state of the Markov Chain is x^k at the k^{th} iteration, the MH algorithm will generate a new sample x^{k+1} for the next iteration. The algorithm is described as in Algorithm 1.

Algorithm 1 Metropolis-Hastings algorithm

Initialization Set the initial sample x_t^1 , begin iteration for k=1:N_s−1

Step 1 Propose a sample $x_t^{'}$ with the proposal distribution $Q\left(x_t^{'};x_t^k\right)$.

Step 2 Compute the acceptance probability:

$$\alpha = \frac{p\left(x_t^{'}|z_{1:t}\right)}{p\left(x_t^k|z_{1:t}\right)}\frac{Q\left(x_t^k;x_t^{'}\right)}{Q\left(x_t^{'};x_t^k\right)} \tag{18}$$

Step 3 Randomly draw a number from a uniform distribution, $\vartheta \sim U$ (0, 1), then decide whether or not accept the proposed sample state.

If $\vartheta < \alpha$, accept the sample, $x_t^{k+1} = x_t^{'}$, else reject the sample, and set $x_t^{k+1} = x_t^k$.

In order to avoid the numerical integration of the predictive density at each MCMC iteration, Pang [11] proposed an improved MCMC algorithm based on the joint posterior probability density $p(x_t, x_{t-1}|z_{1:t})$ of x_t and x_{t-1}. The state variable x_t and x_{t-1} will be updated simultaneously during the MCMC iterations. We will adopt this joint density in our tracking algorithm.

3.3 MCMC posterior sampling based particle tracking

As mentioned above, we consider the joint posterior probability density $p(x_{t-1:t}|z_{1:t})$. According to (14),

$$p(x_{t-1:t}|z_{1:t}) \propto p(z_t|x_t)p(x_t|x_{t-2:t-1})p(x_{t-1}|z_{1:t-1}) \tag{19}$$

By sampling from this target distribution at each iteration, a Markov Chain is constructed. During this procedure, the states x_t and x_{t-1} will be updated simultaneously.

At the proposal step, we propose a new joint state sample $\{x_t^{'}, x_{t-1}^{'}\}$ from the proposal distribution Q(.).

$$\{x_t^{'}, x_{t-1}^{'}\} \sim Q\left(x_t, x_{t-1}|x_t^{k-1}, x_{t-1}^{k-1}\right) \tag{20}$$

Then, compute the acceptance rate of the proposed sample.

$$\psi_1 = \min\left(1, \frac{p\left(x_t', x_{t-1}' | z_{1:t}\right)}{Q\left(x_t', x_{t-1}' | x_t^{k-1}, x_{t-1}^{k-1}\right)} \frac{Q\left(x_t^{k-1}, x_{t-1}^{k-1} | x_t', x_{t-1}'\right)}{p\left(x_t^{k-1}, x_{t-1}^{k-1} | z_{1:t}\right)}\right) \tag{21}$$

Using ψ_1, the sample is decided whether or not accepted at the acceptance step. If accepted, set $\{x_t^k, x_{t-1}^k\} = \{x_t', x_{t-1}'\}$; otherwise, set $\left\{x_t^k, x_{t-1}^k\right\} = \left\{x_t^{k-1}, x_{t-1}^{k-1}\right\}$.

In the following process, individual refinement steps are taken to sample the individual component of the joint state. Firstly, sample a state sample of x_{t-1} from the proposal distribution.

$$\{x_{t-1}'\} \sim Q(x_{t-1} | x_t^k, x_{t-1}^k) \tag{22}$$

Calculate the acceptance probability using

$$\psi_2 = \min\left(1, \frac{p\left(x_{t-1}' | x_t^k, z_{1:t}\right)}{Q\left(x_{t-1}' | x_t^k, x_{t-1}^k\right)} \frac{Q\left(x_{t-1}^k | x_t^k, x_{t-1}'\right)}{p\left(x_{t-1}^k | x_t^k, z_{1:t}\right)}\right) \tag{23}$$

Decide whether or not accept the proposed sample x_{t-1}' . If accepted, set $\left\{x_{t-1}^k\right\} = \left\{x_{t-1}'\right\}$, otherwise, set $\left\{x_{t-1}^k\right\} = \left\{x_{t-1}^{k-1}\right\}$.

Secondly, sample a state sample of x_t.

$$\left\{x_t'\right\} \sim Q\left(x_t | x_t^k, x_{t-1}^k\right) \tag{24}$$

The acceptance probability is computed using

$$\psi_3 = \min\left(1, \frac{p\left(x_t' | x_{t-1}^k, z_{1:t}\right)}{Q\left(x_t' | x_t^k, x_{t-1}^k\right)} \frac{Q\left(x_t^k | x_t^k, x_{t-1}^k\right)}{p\left(x_t^k | x_{t-1}^k, z_{1:t}\right)}\right) \tag{25}$$

Decide whether or not accept the proposed sample x_t' . If accepted, $\left\{x_t^k\right\} = \left\{x_t'\right\}$; otherwise $\left\{x_t^k\right\} = \left\{x_t^{k-1}\right\}$.

We can finally obtain the sample set $\{x_t^k\}_{k=1}^{N_s}$ which is used to compute the posterior probability density. The algorithm can be summarized as follow.

Algorithm 2 MCMC posterior sampling based particle tracking

```
Input Sample set of time $t-1$ $\{x_{t-1}^k, x_{t-2}^k\}_{k=1}^{N_s}$

Output Sample set of timet $\left\{x_t^k\right\}_{k=1}^{N_s}$ , and state estimate $\widehat{x}_t$ .
    For $k$=1, 2, …, $N_s$

Step 1  Proposal step- Propose $\left\{x_t', x_{t-1}'\right\}$ using (20).
Step 2  Acceptance step- Compute the acceptance probability $\psi_1$using (21), and then accept
        $\left\{x_t', x_{t-1}'\right\}$ with probability $\psi_1$, $\left\{x_t^k, x_{t-1}^k\right\} = \left\{x_t', x_{t-1}'\right\}$ . If the proposed state is
        rejected, $\left\{x_t^k, x_{t-1}^k\right\} = \left\{x_t^{k-1}, x_{t-1}^{k-1}\right\}$ .
Step 3  Refine $x_{t-1}$- Propose $x_{t-1}'$ using (22).
Step 4  Compute the MH acceptance probability$\psi_2$, then accept $x_{t-1}'$ with probability $\psi_2$
        (23), $\left\{x_{t-1}^k\right\} = \left\{x_{t-1}'\right\}$ ; If the proposed state is rejected, $\left\{x_{t-1}^k\right\} = \left\{x_{t-1}^{k-1}\right\}$ .
Step 5  Refine$x_t$- Propose $x_t'$ using (24)
```

Step 6 Compute the MH acceptance probability ψ_3 and accept $x_t^{'}$ with probability ψ_3(25), $\{x_t^k\}=\{x_t^{'}\}$; If the proposed state is rejected, $\{x_t^k\}=\{x_t^{k-1}\}$.

EndFor

Step 7 Obtain the sample set $\{x_t^k, x_{t-1}^k\}_{k=1}^{N_s}$, and compute the state estimate $\widehat{x}_t$.

4 Implementation and experiments

Firstly, we use rectangular area to represent the tracking object, which is defined by its spatial position center and the object scale, that is x=(x_0, y_0, s). As to the proposal distribution, we adopt the following as in [8].

$$Q\left(x_t^{'};x_t\right)=\begin{cases} Q_{AR}\left(x_t^{s'};x_t^s\right) & scale \\ Q_U\left(x_t^{p'}\right) & position \end{cases} \tag{26}$$

where Q_{AR} is used to propose a new scale state $x_t^{s'}$ based on x_t^s with a second-order autoregressive process. We assume that the object scale is changed smoothly. Q_U is used to propose a new position state $x_t^{p'}$. The new position state is proposed uniformly from a uniform distribution on the 2D spatial space.

We use a HSV color histogram based appearance model for the sake of handling illumination change. The likelihood function for the filtering distribution based on the HSV color histogram similarity is defined by the following equation.

$$p(z_t|x_t)=e^{-\xi d^2(H',H(x_t))} \tag{27}$$

where the H' is the target reference model, $H(x_t)$ is the candidate model, ξ is a scaling parameter, and d is the Bhattacharyya distance over the HSV histogram which is defined by

$$d=\sqrt{1-\zeta(H',\mathrm{H}(\mathrm{x_t}))} \tag{28}$$

where ζ is the Bhattacharyya similarity coefficient.

As for the state transition dynamic models, we adopt a standard second order constant acceleration model (29) under second-order Markov assumption for the proposed algorithm, while a first order model (30) for the other tracking algorithms.

$$x_t=A_0x_{t-1}+B_0x_{t-2}+\Upsilon\upsilon_t \tag{29}$$

$$x_t=A_1x_{t-1}+B_1\upsilon_t \tag{30}$$

where A_0, B_0, A_1, B_1, and Υ are predefined coefficient matrix.

For the sake of evaluating the performance of our proposed tracking algorithm, we use nine video sequences to test its performance and make comparison to the other alternatives: ASAMC [16], adaptive MCMC [12], WLMC [8], and HMC [2, 13]. The video sequences are listed in Table 1.

Table 1 Video sequences used in our experiments

Seq. No.	Length (frames)	Details
1	813	"ChoiHongMan" Abrupt motion induced by camera switch.
2	195	"Baby" Full and partial occlusions, illumination change.
3	539	"Hockey" Background clutter, object interactions, similar moving object, and partial occlusion.
4	101	"Hockey2" Background clutter, object interactions, similar moving object, and partial occlusion.
5	31	"Face1" Fast moving human face.
6	500	"Face2" Partial and full occlusions, background clutter, object appearance and pose change.
7	113	"Soccer" Background clutter, object interactions, similar moving objects, and object disappearing and re-entering..
8	50	"SeqMS" Frequently occlude human face.
9	220	"ChenNa" Illumination change.

4.1 Qualitative results

For qualitative comparison purpose, we test our proposed tracker over different tracking scenarios and compare the tracking results with other algorithms. 300 samples are used by default.

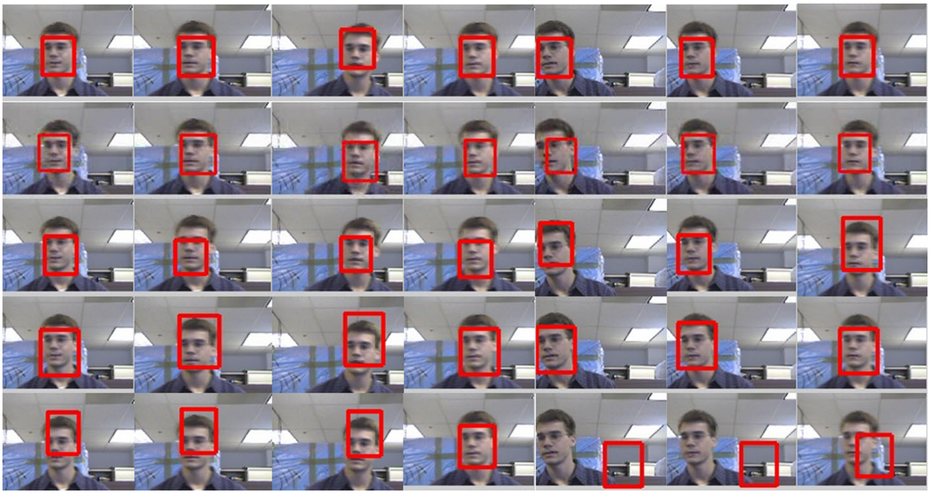


Fig. 2 Tracking results over the *face1* sequence. From top to bottom: 2MCMC-PF, ASAMC, WLMC, HMC, and AMCMC, the sample frames are #2, #18, #19, #23, #25, #27, #28 (from left to right)

Abrupt motion-fast moving object Sequence 5 is used in this experiment to test the ability of the trackers of tracking a fast moving object with abrupt dynamic changes. The 2MCMC-PF tracker uses 300 samples while the others use 600. Figure 2 displays the tracking results. The results show that the proposed tracker shows better tracking accuracy than the other four trackers, although some trackers give acceptable results at certain frames. AMCMC shows the worst results, losing the object at frame #25, #27 and #28.

Object appearance change The goal of the experiment is to test the ability of handling object appearance change of different trackers. Sequence 6 is used in which a girl face is moving and rotating with appearance and pose changes in front of the camera. The tracking results are shown in Fig. 3. It is clearly shown that the 2MCMC-PF tracker can accurately capture the object throughout the sequence. The first object rotation occurred between frame #86 and frame #100, and our tracker accurately tracked the object while other tracker failed. The second object rotation occurred between frame #176 and #246, where all the trackers can capture the object at frame #176, but only the 2MCMC-PF tracker can accurately track the object throughout the rotation process, and the other four frequently lose the object.

Background clutter, object interaction and similar moving object We use sequence 3, 4, and 7 for this experiment. In these video sequences, the players wear the same sports suites moving with frequent occlusions and interactions, which make tracking a player more difficult. Sample frames of tracking results are shown in Figs. 4, 5 and 6, which demonstrated that the 2MCMC-PF tracker could consistently track the object while other tracking algorithms failed frequently. In sequence 7, the player disappeared from the scenario and re-appear after several frames, and the 2MCMC-PF tracker can efficiently capture the player exactly after re-appearing.

Illumination change, full and partial occlusions This experiment aims to test the algorithms' ability of handling illumination change and occlusions. We use the "Baby"

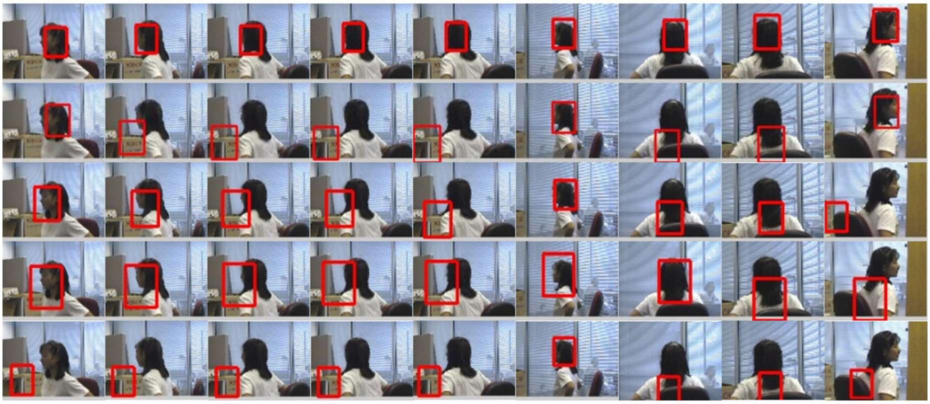


Fig. 3 Tracking results over the *face2* sequence. From top to bottom: 2MCMC-PF, ASAMC, WLMC, HMC, and AMCMC, the sample frames are #86, #94, #98, #99, #100, #176, #188, #215, #246 (from left to right)

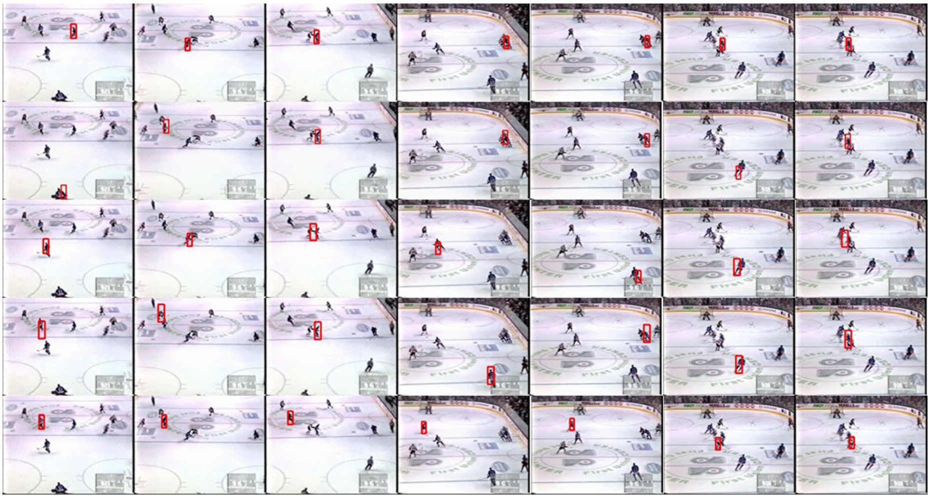


Fig. 4 Tracking results over the *Hockey1* sequence. From top to bottom: 2MCMC-PF, ASAMC, WLMC, HMC, and AMCMC, the sample frames are #69, #303, #320, #458, #470, #510, #512 (from left to right)

and “ChenNa” sequences for this tracking experiment. 100 samples are used for the 2MCMC-PF tracker, while 300 for the others. For the “Baby” sequence, the illumination change occurred after the baby entering the shadow, and it was full occluded by the adult for a long time interval. Sample tracking frames are shown in Fig. 7. The

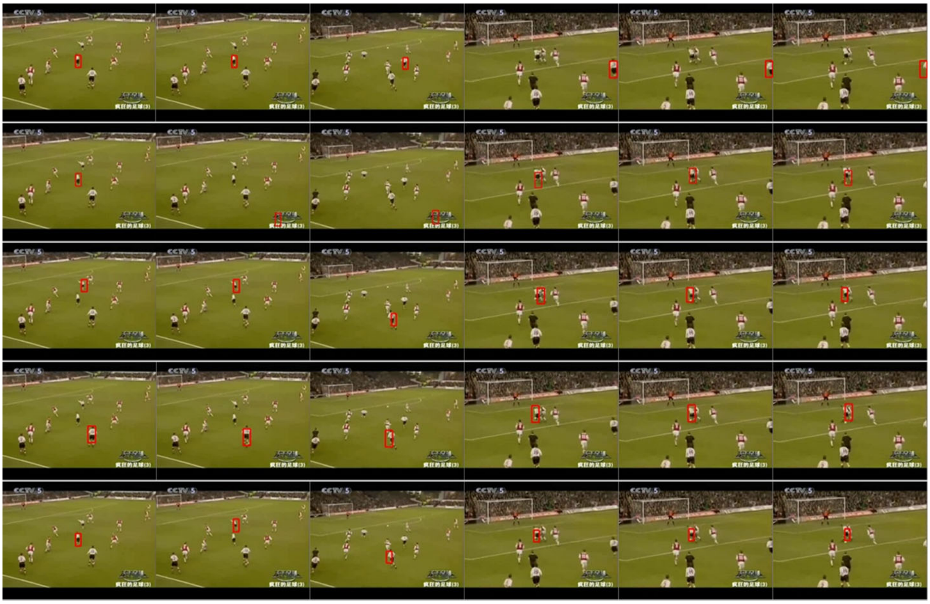


Fig. 5 Tracking results over the *Soccer* sequence. From top to bottom: 2MCMC-PF, ASAMC, WLMC, HMC, and AMCMC, the sample frames are #12, #13, #29, #58, #59, #60 (from left to right)

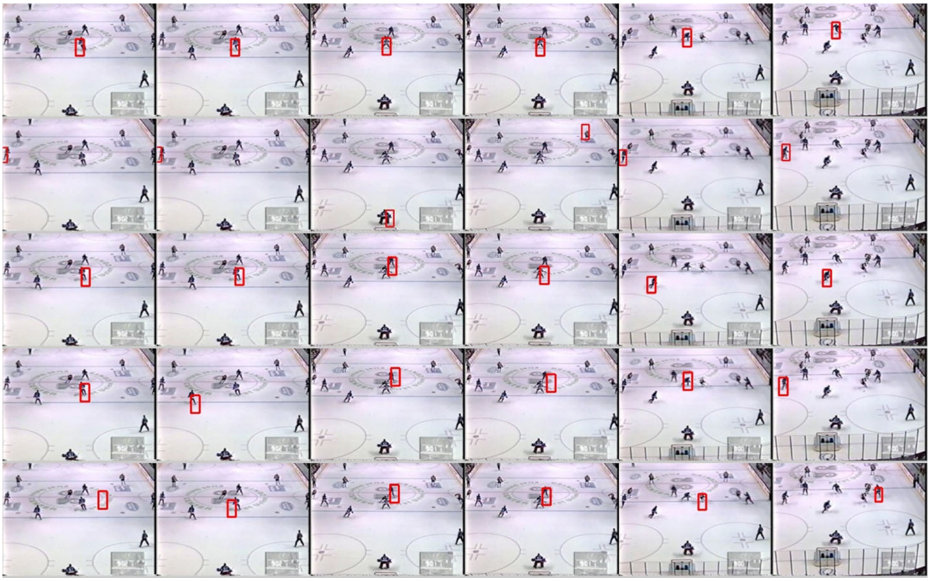


Fig. 6 Tracking results over the *Hockey2* sequence. From top to bottom: 2MCMC-PF, ASAMC, WLMC, HMC, and AMCMC, the sample frames are #23, #24, #39, #40, #61, #82 (from left to right)

2MCMC-PF shows the best tracking accuracy compared to the other four algorithms. The AMCMC and HMC shows worse tracking results than our tracker, while better than the ASAMC and the WLMC. The latter two frequently lost the object.

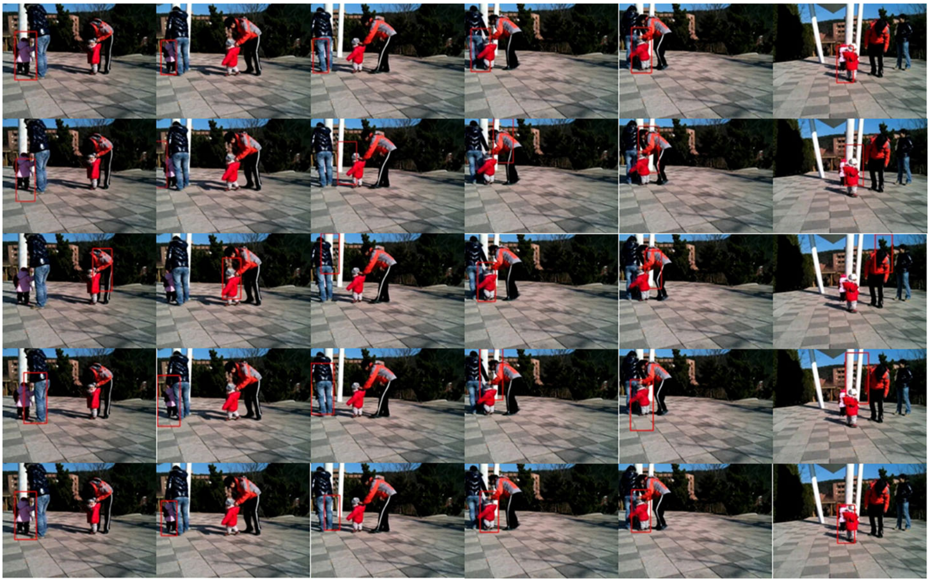


Fig. 7 Tracking results over the *Baby* sequence. From top to bottom: 2MCMC-PF, ASAMC, WLMC, HMC, and AMCMC, the sample frames are #35, #52, #66, #82, #87, #138 (from left to right)

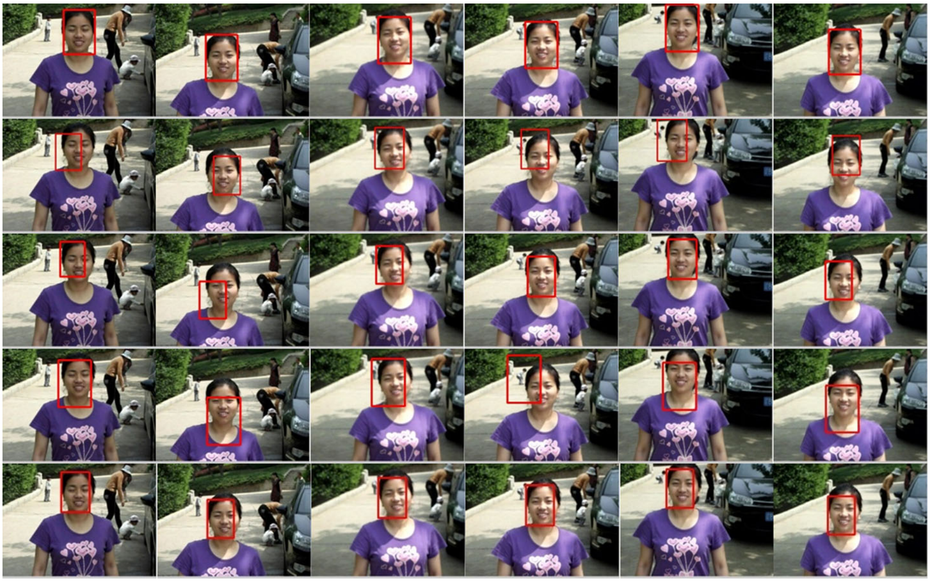
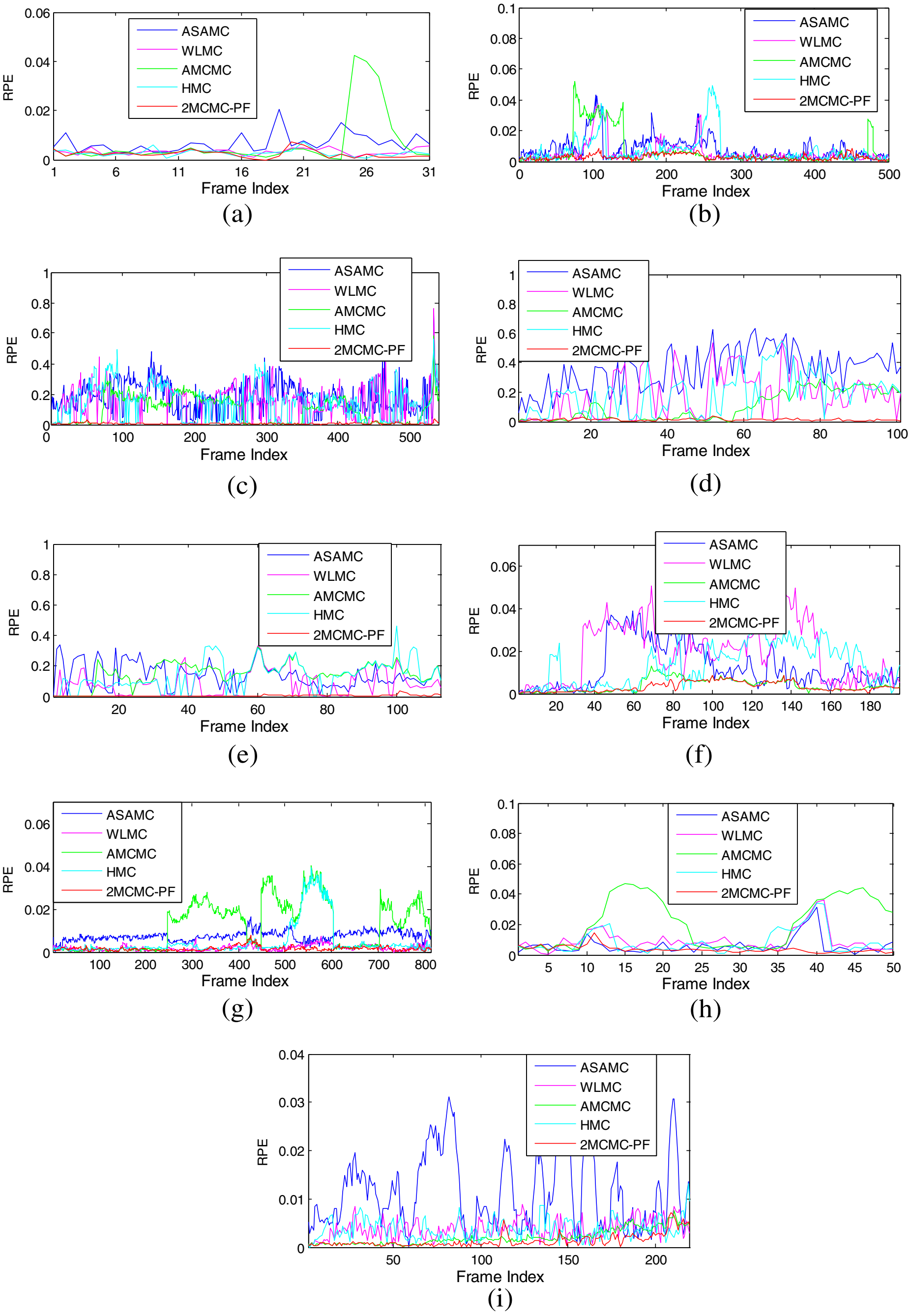


Fig. 8 Tracking results over the *ChenNa* sequence. From top to bottom: 2MCMC-PF, ASAMC, WLMC, HMC, and AMCMC, the sample frames are #72, #108, #132, #136, #149, #157 (from left to right)

As for the "ChenNa" sequence, we aim to track the girl's face in a scenario with illumination change. Tracking results are shown in Fig. 8. When illumination change occurred between frame #108 and #157, the 2MCMC-PF tracker can accurately track

Table 2 RPEs of the algorithms over the 9 video sequences

Tracker Seq.		2MCMC-PF	ASAMC	WLMC	HMC	AMCMC
Face1	RPE mean	0.024	0.067	0.033	0.028	0.065
	RPE std	0.0016	0.0040	0.0013	0.0014	0.011
Face2	RPE mean	0.0028	0.0082	0.0055	0.0068	0.0070
	RPE std	0.0018	0.0073	0.0068	0.0087	0.0107
Hockey1	RPE mean	0.0070	0.1821	0.1450	0.1491	0.0948
	RPE std	0.0046	0.0984	0.1037	0.1120	0.0862
Hockey2	RPE mean	0.0127	0.3511	0.1840	0.1871	0.1010
	RPE std	0.0076	0.1423	0.1537	0.1424	0.0977
Soccer	RPE mean	0.0057	0.1408	0.0901	0.1386	0.1519
	RPE std	0.0053	0.0819	0.0815	0.0968	0.0725
Baby	RPE mean	0.0031	0.0125	0.0189	0.0132	0.0037
	RPE std	0.0022	0.0106	0.0135	0.0096	0.0030
ChoiHongMan	RPE mean	0.0016	0.0074	0.0019	0.0049	0.0124
	RPE std	0.0010	0.0022	0.0013	0.0079	0.0111
SeqMS	RPE mean	0.0036	0.0059	0.0093	0.0083	0.0215
	RPE std	0.0023	0.0056	0.0074	0.0080	0.0162
ChenNa	RPE mean	0.0014	0.0116	0.0038	0.0038	0.0021
	RPE std	0.0014	0.0080	0.0019	0.0022	0.0015



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Fig. 9 RPEs of the algorithms over the 9 video sequences. **a** Face1, **b** Face2, **c** Hockey1, **d** Hockey2, **e** Soccer, **f** Baby, **g** ChoiHongMan, **h** SeqMS **i** ChenNa

the object. AMCMC algorithm shows acceptable results, but is worse than our tracker, while better than the other three algorithms.

Table 3 SRs of the algorithms over the 9 video sequences

Tracker Seq.	2MCMC-PF	ASAMC	WLMC	HMC	AMCMC
Face1	100 %	83.87 %	100 %	100 %	87.10 %
Face2	100 %	68.80 %	87 %	83.40 %	84.80 %
Hockey1	95.19 %	0.74 %	18.52 %	19.26 %	39.44 %
Hockey2	82.18 %	0 %	22.77 %	18.81 %	37.62 %
Soccer	80.53 %	1.77 %	30.97 %	16.81 %	11.50 %
Baby	69.74 %	23.08 %	14.36 %	21.54 %	60.51 %
ChoiHongMan	94.10 %	3.20 %	88.19 %	75.52 %	42.31 %
SeqMS	98 %	94 %	84.00 %	78.00 %	44.00 %
ChenNa	97.73 %	32.27 %	82.73 %	81.82 %	97.27 %
Overall SR	93.05 %	21.90 %	62.96 %	57.18 %	55.15 %

4.2 Quantitative results

4.2.1 Relative position error

In the quantitative comparison experiment, we compare the relative position error (RPE) of different trackers which is defined as:

$$\Delta p = \left\|(x, y) - (x_g, y_g)\right\| / s_g \tag{31}$$

where (x_g, y_g, s_g) is the ground truth state calibrated manually. We adopt this measurement for evaluating the tracking accuracy because it can facilitate comparing the tracking accuracy for the objects with different sizes [14]. As shown in Table 2, both the mean and standard deviation of the 2MCMC-PF tracker over the nine sequences are almost consistently smaller than those of other trackers. Figure 9 shows the curves of the RPEs of different trackers over the nine sequences, which indicate that the 2MCMC-PF tracker is more accurate and stable than the alternatives.

4.2.2 Success rate

We define tracking to be lost when the distance between ground truth center and the estimated object center is larger than the calibrated radius of the object. The calibrated radius is defined as the smaller one of the half the width and half the height. This definition is different from that of [16], which defined tracking to be lost when the estimated center was not in the calibrated object area. The success rate (SR) is defined as the ratio between the successfully tracked frames and the total frames of the sequences. In this experiment, we calculate the overall success rate of different trackers over the nine sequences. We also compare the success rate of the trackers over one single sequence. Results are shown in Table 3. The results indicate that all the SRs of our proposed tracker over each of the sequences are higher than that of the other algorithms. The SRs of the other four algorithms fluctuate severely over different sequences which indicate that the robustness of the four trackers is worse than the 2MCMC-PF. We should note that our tracking lost calculation is much stricter than that of [16]. If we use the method mentioned in [16], the SR would be greatly different from that of calculated using our method. Take the ASAMC algorithm over the "ChoiHongMan" sequence as example, the SR calculated using our method is about 3.2 %, but the 2MCMC-PF tracker is 94.10 %, while the SR calculated using method of [16] yields 88.68 % and 100 % respectively.

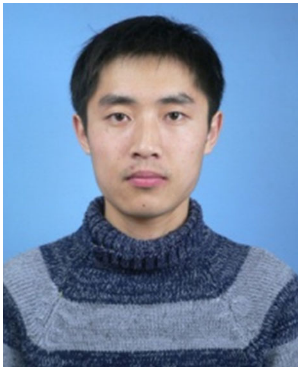
5 Conclusions

We have proposed a robust tracking algorithm within the particle filtering framework using the Markov Chain Monte Carlo posterior sampling and second-order Markov assumption. In our tracking algorithm, we use a Markov Chain with certain length to simulate the approximated posterior probability density, which avoid the drawbacks of the traditional importance sampling based algorithm. The second-order Markov assumption can make better use of the history information and enhance the searching ability of our tracker. Experimental results have demonstrated that our proposed tracker can give stable and accurate tracking results in various tracking scenarios. We have to mention that the proposed algorithm can only handle abrupt motions with camera switching, as for abrupt motions induced by sudden dynamic changes, low frame rate videos and rapid motion, it tends to fail. Our further study will focus on background modeling and object appearance adaptation in much more complex tracking scenarios.

Acknowledgments This work was supported by the National Natural Science Foundation of China (No. 60772063, 61073133, 61175053, 60973067, 61175096, 61272369), the Innovation Group Project of China Education Ministry (No. 2011ZD010, Fundamental research fund of DMU (NO. 2012QN030), Foundation of Scientific Planning Project of Dalian City (No. 2011E15SF100), the Natural Science Foundation of Liaoning Education Ministry (No. L2011241, L2010043).

References

1. Arulampalam S, Maskell S, Gordon N, Clapp T (2002) A tutorial on particle filters for on-line Non-linear/Non-Gaussian Bayesian tracking. IEEE T Signal Process 50(2):174–188
2. Choo K, Fleet DJ (2001) People Tracking Using Hybrid Monte Carlo Filtering. Proceedings of International Conference on Computer Vision, pp. 321–328
3. Cong TD, F Septier, Garnier C et al. (2012) Robust Visual Tracking via MCMC-based Particle Filtering. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Apr. 2012, Kyoto, Japan
4. Geyer C (2011) Introduction to Markov Chain Monte Carlo. In: Steve B et al. (ed) Handbook of Markov Chain Monte Carlo. Chapman and Hall/CRC press, pp.1–46
5. Gilks W, Berzuini C (2001) Following a moving target: Monte Carlo inference for dynamic Bayesian models. J Royal Stat Soc B 63(1):127–146
6. Isard M, Blake A (1998) CONDENSATION-conditional density propagation for visual tacking. Int J Comput Vision 29(1):5–28
7. Khan Z, Balch T, Dellaert F (2005) MCMC-based particle filter for tracking a variable number of interacting targets. IEEE T Pattern Anal 27(11):1805–1819
8. Kwon J, Lee K (2008) Tracking of Abrupt Motion Using Wang- Landau Monte Carlo Estimation. Proceedings of European Conference on Computer Vision, pp. 387–400
9. Liang F, Liu C et al (2007) Stochastic approximation in Monte Carlo computation. J Am Stat Assoc 102:305–320
10. Martinez-del-Rincon J, Orrite C, Medrano C (2011) Rao-blackwellised particle filter for color-based tracking. Pattern Recogn Lett 32:210–220
11. Pang SK, Li J, Godsill G (2008) Models and Algorithms for Detection and Tracking of Coordinated Groups," Proceedings of IEEE Aerospace Conference, Big Sky, MT, 2008, pp. 1–17
12. Roberts O, Rosenthal S (2009) Examples of adaptive MCMC. J Comput Graph Stat 18(2):349–367
13. Wang F, Lu M (2012) Hamiltonian Monte Carlo Estimator for Abrupt Motion Tracking. Proceedings of International Conference on Pattern Recognition, pp. 3066–3069
14. Yang M, Fan Z, Fan J, Wu Y (2009) Tracking nonstationary visual appearances by data-driven adaptation. IEEE T Image Process 18(7):1633–1644
15. Yilmaz A, Javed O, Mubarak S (2006) Object tracking: a survey. ACM Comput Surv 38:1–45
16. Zhou X, Lu Y, Lu J, Zhou J (2012) Abrupt motion tracking via intensively adaptive Markov Chain Monte Carlo sampling. IEEE T Image Process 21:789–801



Fasheng Wang born in 1983, received his bachelor's degree in 2005 from Qufu Normal University, and master's degree from Beijing Institute of Technology. He is now pursuing his Doctor's degree in Dalian Maritime University. His research interests mainly focus on Computer Vision and Pattern Recognition.

Mingyu Lu born in 1963, received his doctor's degree from Tsinghua University in 2003, and is now a Profession and doctoral supervisor in Dalian Maritime University. His research interests include data mining, pattern recognition, machine learning and computer vision.