The Computer Journal Advance Access published October 24, 2012

doi:10.1093/comjnl/bxs141

Efficient Visual Tracking via Hamiltonian Monte Carlo Markov Chain

Fasheng Wang[1,2] and Mingyu Lu[1,*]

[1] *Information Building B-106, School of Information Science and Technology, Dalian Maritime University, No. 1 Linghai Road, Ganjingzi District, Dalian 116026, China*
[2] *Department of Computer Science and Technology, Dalian Neusoft Institute of Information, Dalian 116023, China*
**Corresponding author: lumingyu@dlmu.edu.cn*

Efficient visual tracking is a challenging task in the computer vision community due to its large motion uncertainty induced by occlusion, abrupt motion or appearance changes. In this paper, we propose a Hamiltonian Markov Chain Monte Carlo (MCMC) based tracking scheme for efficient tracking within the Bayesian filtering framework, aiming at handling full or partial occlusions, abrupt motion and appearance changes. In this tracking scheme, no complex models are built for motion uncertainties. The object states are augmented by introducing a momentum item and the Hamiltonian dynamics (HD) is integrated into the traditional MCMC-based tracking method. A new object state is proposed by computing a trajectory according to HD, implemented with the Leapfrog method. The new state can be distant from the current object state but, nevertheless, has a high acceptance probability, which consequently bypasses the slow exploration of the state space suffered by traditional random-walk proposal distribution. In addition, the proposed tracking algorithm can avoid being trapped in local maxima, which is suffered by conventional MCMC-based tracking algorithms. Experimental results reveal that our approach is efficient and effective in dealing with various types of tracking scenarios compared with several alternatives.

Keywords: visual tracking, Monte Carlo Markov Chain, Hamiltonian MCMC

Received 21 April 2012; revised 7 September 2012
Handling editor: Tae-Kyun Kim

1. INTRODUCTION

In recent years, visual target tracking has received tremendous attention in the computer vision community due to its success in a wide range of potential applications [1]. Conventional tracking methods can be divided into two categories: the stochastic approach and the deterministic approach. The deterministic approach usually provides reliable tracking results by using bottom-up information, of which the mean shift [2] algorithm is famous. Particle filter (PF) is a popular stochastic-based approach which has shown efficiency in handling non-Gaussianity and multi-modality [3, 4]. It is a Monte Carlo simulation-based method. Markov Chain Monte Carlo (MCMC) method is also an efficient stochastic approach which has been applied successfully in the tracking of multiple interacting objects [5, 6]. The authors of [5, 6] made strict constraints on the background, which could lead the trackers to fail in tracking with changing background. Lin and Wolf [7] proposed an MCMC-based feature-guided particle tracker for visual tracking with moving cameras, but the proposed tracker can only handle certain kinds of sudden motions. Generally speaking, both of the approaches have been extensively studied. However, both of them are generally based on the smooth motion assumption, that is, the target being tracked is moving with stable motion. In many real-world applications, abrupt motions often appear due to camera switching, low frame rate (LFR) video and uncertain object dynamics, which could cause the conventional tracking approach to fail since they violate the motion smoothness constraint. Additionally, object occlusions and appearance changes in the tracking process make it even harder to effectively track the object.

The easiest way to handle these situations is to guide the tracker searching the whole state space to capture the object. However, it is infeasible for practical applications because searching the whole state space will result in extremely expensive computational cost. Li *et al.* [8] proposed a cascade PF to handle abrupt motion tracking. In this algorithm, recognition techniques are combined into the PF to detect the object. Although it shows efficiency in tracking of multiple faces, it needs a reliable *a priori* object observation model, as well as an off-line learning process. Huang and Essa [9] proposed an approach for tracking varying number of objects through both temporally and spatially significant occlusions. In this method, tracking is performed at both the region level and the object level. Based on sparse representation, Mei *et al.* [10] proposed an efficient *L1* tracker with minimum error bound and occlusion detection which was called bounded particle resampling. The trackers in [9, 10] and some others not mentioned in this paper all need accurate mathematical models to detect and handle occlusions, but it is very difficult to construct these models and they are time consuming in tracking.

In [11], a Wang–Landau Monte Carlo (WLMC) based tracking algorithm has been proposed to handle abrupt motion difficulties. The WLMC tracking method uses the density of states (DOS) as the prior distribution within the Bayesian filtering framework. It uses a random walk method to visit the subspaces of the whole state space and the DOS is adaptively updated during the sampling process. The WLMC tracking scheme could effectively handle the local-trap problem suffered by the conventional MCMC method, which is of great importance in statistical physics. However, there is no rigorous theory so far to support its convergence and it can achieve only a limited statistical accuracy in some applications. Zhou *et al.* [12] proposed an adaptive stochastic approximation Monte Carlo sampling (ASAMC) based tracking algorithm, which is also based on the DOS prior. In these tracking algorithms (WLMC and ASAMC), the frame is divided into M equal-sized subregions, and the DOS of each subregion is estimated to guide the state transition. But the division decreases the efficiency when the size of the frame is large and the state transition based on the density of the subregion is not robust. In general, most MCMC-based algorithms are slow in sampling from target distributions and they are based on random-walk Metropolis proposals, which often need long iterations in order to reach an object state reasonably independent of the current states. Also, the variance of the object states after n iterations of random-walk Metropolis from the start state will grow in proportion to n.

To handle the problems mentioned above, an efficient tracking scheme within the Bayesian filtering framework is proposed, that is, the Hamiltonian MCMC (H-MCMC) based tracking algorithm. H-MCMC was first proposed by Duane *et al.* [13] and was formerly named a hybrid Monte Carlo (HMC) method; it has been used by researchers for solving statistical problems, such as neural network and machine learning [14–16]. Choo and Fleet [17] first used the HMC filter for 3D human motion inference with high-dimensional state space. The HMC method is several thousand times faster than a conventional PF on a 28D human body tracking problem.

H-MCMC is based on Hamiltonian dynamics (HD) which has some excellent properties of reversibility, conservation of the Hamiltonian, volume preservation and symplecticness [18]. The H-MCMC combines the advantages of HD and the Metropolis Monte Carlo approach. It incorporates gradient information in the dynamic trajectories and thus suppresses the random-walk nature in traditional MCMC methods, which ensures rapid mixing, faster convergence and improved efficiency of the Markov Chain [19, 20]. The H-MCMC-based tracking scheme does not need accurate mathematical models to handle the thorny problems mentioned above, but can still give encouraging tracking results.

The remainder of this paper is organized as follows. In Section 2, we introduce the H-MCMC method to facilitate the understanding of the proposed tracker. In Section 3, we introduce the Bayesian object tracking framework, and then, we will integrate the H-MCMC into the Bayesian object tracking framework, and lastly, we will model the object systems under HD. Section 4 shows the experimental results of our tracking scheme in handling tracking difficulties induced by abrupt motions, occlusions and appearance changes. We will compare the H-MCMC-based tracking scheme with the PF, traditional MCMC, adaptive MCMC (AMCMC), WLMC and ASAMC algorithms. Finally, Section 5 draws the conclusion.

2. HAMILTONIAN MCMC

We first introduce a momentum variable $v \in R^{\mathrm{d}}$ to go with the original variable of interest $x \in R^{\mathrm{d}}$, where d denotes the dimension. A Hamiltonian function H is constructed as a potential energy term plus a kinetic energy term, that is:

$$H(x, v) = \rho(x) + K(v), \tag{1}$$

$$\rho(x) = -\log(q(x)), \tag{2}$$

$$K(v) = \frac{v^2}{2}, \tag{3}$$

where $q(x)$ is the target distribution, $\rho(x)$ is the potential energy and $K(v)$ is the kinetic energy. Our goal is to draw random samples from the new probability density function (PDF) that is proportional to $\exp(-H)$.

According to the concept of canonical distribution from statistical mechanics, given the energy function $E(x)$ for the state x of some physical system, the canonical distribution over the states of the system has a PDF:

$$P(x) = \frac{1}{Z}\exp\left(\frac{-E(x)}{T}\right), \tag{4}$$

where T is the system temperature and Z is the normalizing constant. As mentioned above, the Hamiltonian function is an energy function of the augmented state (x, v), and so we define a joint distribution for them as follows:

$$P(x, v) = \frac{1}{Z} \exp\left(\frac{-\rho(x)}{T}\right) \exp\left(\frac{-K(v)}{T}\right). \qquad (5)$$

Each iteration of the H-MCMC algorithm has two steps. The first step changes only the momentum variable, while the second step changes both the state variable and the momentum variable. Both of the two steps leave the canonical joint distribution of (5) invariant. The two steps are summarized as follows:

Step 1: Randomly draw new values of the momentum variable v, which are independent of the current values of the state variable, from a Gaussian distribution. This step may keep the joint distribution of the two variables invariant.

Step 2: Perform a Metropolis update step, using HD to propose a new state over L Leapfrog steps using a finite step size ε.

$$v(\tau + \varepsilon/2) = v(\tau) - \frac{\varepsilon}{2} \nabla v(x(\tau)), \qquad (6)$$

$$x(\tau + \varepsilon) = x(\tau) - \varepsilon v\left(\tau + \frac{\varepsilon}{2}\right), \qquad (7)$$

$$v(\tau + \varepsilon) = v\left(\tau + \frac{\varepsilon}{2}\right) - \frac{\varepsilon}{2} \nabla v(x(\tau + \varepsilon)), \qquad (8)$$

where $1 \leq \tau \leq L$ is the step index, ε is the step size and ∇v is the gradient of v. After L steps, the momentum variables at the end of the L-step trajectory are negated, giving a proposed state (x', v'), which is accepted as the next state of the Markov Chain with probability:

$$\alpha = \min(1, \exp(-H(x', v') + H(x, v))). \qquad (9)$$

If this proposed state is rejected, then the current state (x, v) will be used as the next state. The negation of the momentum variables at the end of the trajectory makes the Metropolis proposal symmetrical, which is needed for the acceptance probability above to be valid.

3. H-MCMC-BASED TRACKING METHOD

In this section, we will discuss the tracking algorithm with Hamiltonian Monte Carlo sampling.

3.1. Bayesian object tracking

Let us assume that the state x_t at time t consists of the object position and the scale, that is, $x_t = (x_t^{\mathrm{p}}, x_t^{\mathrm{s}})$. Here, the state x_t corresponds to the variable of interest x in the H-MCMC algorithm described in Section 2.

Object tracking can be formulated as the Bayesian filtering problems, which iteratively estimate the state x_t of the object over the observations $Y_t = \{y_1, y_2, \ldots, y_t\}$ up to time t. The Bayesian filter updates the posterior probability $p(x_t|Y_t)$ with the following recursive rule:

$$p(x_t|Y_t) = cp(y_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|y_{t-1}) dx_{t-1}, \qquad (10)$$

where $p(x_t|x_{t-1})$ denotes the system transition model which describes the time evolution of the object states, $p(y_t|x_t)$ denotes the observation model which measures the similarity between the observation at the estimated state and the given model, and c is the normalizing constant. After computing the posterior probability $p(x_t|Y_t)$ using (10), we obtain the maximum a posteriori (MAP) estimate over the N samples $\{x_t^{(i)}\}_{i=1}^N$ at each time step t:

$$x_t^{\mathrm{MAP}} = \arg\max p(x_t^{(i)}|Y_t), \quad i = 1, \ldots, N, \qquad (11)$$

where x_t^{MAP} represents the best estimated object state which can be used as the current state with the given observation.

However, it is often infeasible to compute the integral in Equation (10) especially for high-dimensional state spaces. To circumvent this problem, the metropolis hastings algorithm is used to sample from a given proposal density Q and the samples $\{x_{i=1:N}^t\}$ are used to approximate the posterior density $p(x_t|Y_t)$. Consequently, the integral will be approximated with the obtained samples, which can avoid computing the complex integrals.

3.2. H-MCMC-based tracking

The conventional MCMC-based methods simulate a Markov Chain that converges to a stationary distribution $p(x_t|Y_t)$, in order to achieve a good approximation to the posterior probability distribution. However, the traditional MCMC sampling methods are easily trapped in local maxima. During the tracking process, the tracker is guided to search the state space in a reasonable way. When it is trapped in local subspace, the tracker will finally lose the object. This situation is common in visual tracking scenarios. So, we designed an H-MCMC-based sampling procedure to avoid being trapped.

In our tracking scheme, the target object is represented as a rectangular region defined by its 2D spatial position and scale factor, $x = (x_0, y_0, s)$. We should carefully design the proposal density to successfully cover the abrupt motion, appearance change and occlusions. Intuitively, the target object with abrupt motion or occlusion can move to any position of the space with smooth scale assumption. The smooth scale assumption is often adopted as is common for visual tracking [11]. Under this assumption, we discompose the proposal density Q into Q_{p} and Q_{s} for the spatial position and scale, respectively.

$$Q = \begin{cases} Q_{\mathrm{p}}, \\ Q_{\mathrm{s}}. \end{cases} \qquad (12)$$

To formulate the tracking process into the H-MCMC framework, we augment the state x as (x, v), where v is the momentum of x.

During the proposal step, we first draw a new value of the momentum variable from a Gaussian distribution $v \sim \mathsf{N}(0, 1)$, independently of the current object state x_t. After that, the Leapfrog method is used to update the momentum variable and state variable. Then, a new sample state (x^*_{t+1}, v^*) is proposed.

In the acceptance step, we determine whether the proposed sample is accepted or not. The acceptance probability α^* is defined by $\alpha^* = \min(1, \alpha_d)$, where

$$\alpha_d = \exp(-H(x'_{t+1}, v') + H(x_t, v)). \tag{13}$$

Algorithm 1 H-MCMC-based tracking method.

```
1). Initialize object state $x_0$ and momentum variable $v_0$;
2). Set $\varepsilon = 0.02$ and $L = 30$    // Initialize the parameters
3). For $i = 1$ to $N$ do
    - Draw the value of momentum variable $v \sim \mathsf{N}(0, 1)$
    - Set $(x^{(0)}, v^{(0)}) = (x_{i-1}, v)$
    - For $j = 1$ to $L$ do    // Do L leapfrog iterations
        -- $v^{(j-1/2)} = v^{(j-1)} - \frac{\varepsilon}{2}\nabla v(x^{(j-1)})$
        -- $x^{(j)} = x^{(j-1)} + \varepsilon v^{(j-1/2)}$
        -- $v^{(j)} = v^{(j-1/2)} - \frac{\varepsilon}{2}\nabla v(x^{(j)})$
    - EndFor
    - Propose a new state according to (12) :
        -- $(x^*, v^*) = (x^{(L)}, v^{(L)})$
    - Draw $\alpha \sim U(0, 1)$
    - Compute the increments of the energy function $H$:
        -- $\delta H = H(x^*, v^*) - H(x^{(0)}, v^{(0)})$
    - If $\alpha < \min\{1, \exp(-\delta H)\}$
        -- $(x_i, v_i) = (x^*, v^*)$    // accept the proposed state as the next state
    - Else
        -- $(x_i, v_i) = (x_{i-1}, v_{i-1})$    // reject the proposed state, use the current state as the next state.
4). EndFor
5). Return $\{x_i, v_i\}_{i=1}^N$
6). Computer MAP estimates based on (11)
```

The H-MCMC algorithm is typically ergodic, hence it will not be trapped in some subsets of the state space. Thus, the local-trap problem suffered by the traditional MCMC algorithm should be essentially avoided.

3.3. Leapfrog parameters for H-MCMC-based tracking

Since the HD is time reversible and preserves the volume and the total energy, the movement along the object trajectories of constant energy H will leave the joint distribution (5) invariant if the dynamics is simulated exactly. However, in practice, the dynamic is simulated with a finite number of steps using the leapfrog scheme and this leads to errors in the simulation. Thus, when using the HMC algorithm in practice, we have to address two main issues to get good performance: controlling the step size ε and choosing the simulation length L.

Generally speaking, small ε results in good exploration of the distribution space, while too large a step size will result in very low acceptance rates for the states proposed by the simulating trajectory. Thus, users can choose a reasonable value of ε according to their practical requirement. We will show how this parameter influences the successful rate (SR) of the tracker in Section 4.

For trajectory length L, setting it by trial and error seems necessary. As is discussed in [18], for a problem thought to be fairly difficult, $L = 100$ might be a suitable point.

4. EXPERIMENTAL RESULTS

In this paper, we have used a standard second-order constant acceleration model as the dynamic model for state evolution. So we have used the color-based tracking model as described in [21]. The likelihood function for the filtering distribution is based on the HSV color histogram similarity that is defined by

TABLE 1. Test sequences used in our experiments.

Seq. no.	Length/frames	Description
1	813	‘ChoiHongMan’, Abrupt motion induced by camera switching
2	218	‘SonYeJin’ Abrupt motion and occlusions
3	51	‘Tennis’ Abrupt motion induced by LFR video
4	770	‘YoungKi’ Abrupt motion induced by camera switching, and an illumination change
5	150	‘PingPong’ Fast and random moving motion of a yellow ping-pong ball
6	52	‘Ball’ Abrupt motion induced by sudden dynamic changes, a white ping-pong ball
7	139	‘Sudden’ Abrupt motion induced by sudden dynamic changes, a yellow ping-pong ball
8	31	‘Face1’ Fast-moving face
9	501	‘Face2’ Appearance changes, occlusion and clutter of a moving girl’s face

Some of the sequences are public data in many research works, and some were taken by us.

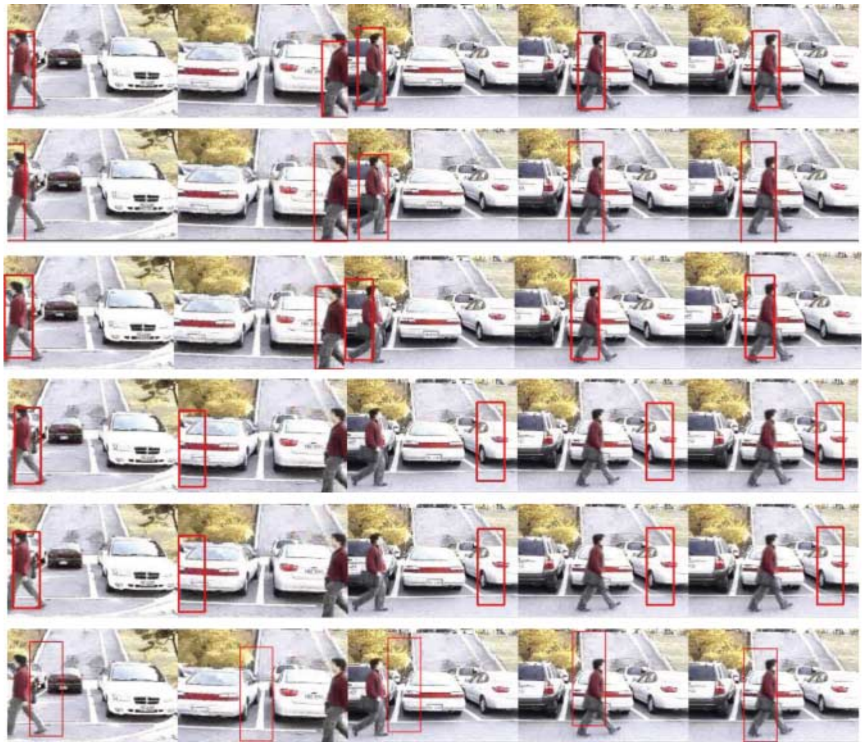

FIGURE 1. Tracking results of different trackers over the *YoungKi* sequence. From top to bottom are the results of our proposed tracker, ASAMC, WLMC, AMCMC, MCMC and PF, respectively. The sample frames from left to right are #38, #39, #403, #432 and #433, respectively. Abrupt motion appeared between frame #38 and #39. The proposed tracker can accurately capture the walking man after the camera switch, and the performances of the ASAMC and the WLMC are slightly worse than ours, while the other three lose the object. Our tracker can also obtain much more acceptable results in smooth motion than the other algorithms.

the following equation:

$$p(y_t|x_t) = e^{-\lambda B(\mathrm{HS_r}, \mathrm{HS}(x_t))}, \quad (14)$$

where $\mathrm{HS_r}$ is the reference target model, $\mathrm{HS}(x_t)$ is the candidate object model, λ is a predefined parameter and $B(.)$ is the Bhattacharyya distance over the HSV histogram. Since our tracking scheme is more about the abrupt motion and occlusions, we only designed a simple appearance model.

To evaluate the performance of the proposed tracking method, we used nine video sequences to test it and compared its performance with several other alternatives: PF is based on [4], MCMC is based on [7], AMCMC is based on [11, 22], WLMC is based on [11] and ASAMC is based on [12]. The details of the video sequences are listed in Table 1.

To evaluate the performance of the trackers at frame k, we have used the *f-measure* that was often used in information retrieval literatures. The recall ϕ and the precision ζ are used to measure the performance of the trackers at frame k:

$$\phi = \frac{U_t^{\mathrm{E}} \cap U_t^{\mathrm{G}}}{U_t^{\mathrm{G}}}, \quad (15)$$

$$\zeta = \frac{U_t^{\mathrm{E}} \cap U_t^{\mathrm{G}}}{U_t^{\mathrm{E}}}, \quad (16)$$

where U_t^{E} is the estimated object area and U_t^{G} is the ground truth area of the target at time t. The ground truth areas of the target are manually calibrated frame by frame.

The *f-measure* is defined as

$$F = \frac{2\phi\zeta}{\phi + \zeta}. \quad (17)$$

When the ground truth area and the estimated object area perfectly overlap, *f-measure* is 1.0. The larger the value of *f-measure*, the better are the tracking results.

4.1. Qualitative results

For qualitatively evaluating the performance of the proposed tracker, we test the algorithms in various scenarios: camera switching, sudden dynamic changes, LFR videos, full and partial occlusions, frequent appearance changes and background clutter. All the algorithms use 300 samples and 600 iterations are executed at each time step by default.

4.1.1. Camera switching

For tracking with frequent camera switching, we used sequence 4 from [11] to test the proposed tracking scheme. The

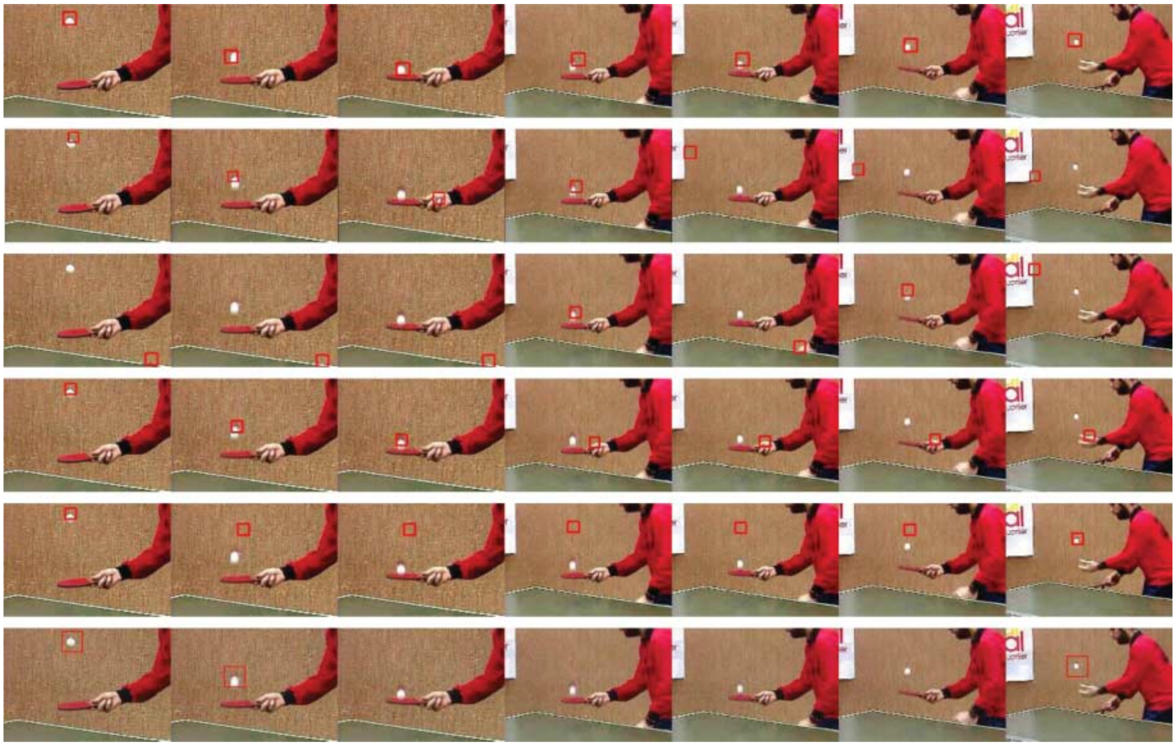

FIGURE 2. Tracking results of the trackers over the *Ball* sequence. From top to bottom are the results of our proposed tracker, ASAMC, WLMC, AMCMC, MCMC and PF, respectively. The sample frames from left to right are #5, #10, #11, #40, #41, #43 and #51, respectively. The proposed tracker can effectively capture the ball after it bounced against the racket between frame #10 and #11, and the performances of the ASAMC and the AMCMC are worse than ours. The WLMC and MCMC failed to track the ball, while the PF tracker was lost at frame #11, #41 and #43. The last several frames of the sequence have background clutter for the white ball, but our tracker can still capture it accurately.

experimental results show that our proposed tracking method, the ASAMC and the WLMC can successfully capture the abrupt motion induced by camera switching, while the other three failed. Our tracking method can also track the smooth motion and outperforms the other algorithms. Sample tracking frames are shown in Fig. 1.

4.1.2. *Sudden dynamic change*

The scenario in sequence 6 is a white ball bounced from a racket with sudden dynamic changes. When the ping-pong ball hits the racket, the dynamics of the ball change suddenly. There is also camera switching at the end of the scenario, which makes it even harder for tracking the ping-pong ball accurately. Our experimental results demonstrated that the proposed tracking method can effectively handle the difficulty. The ball is successfully tracked throughout the sequence. Sample frames are shown in Fig. 2. Tracking results of the other five algorithms are also shown in Fig. 2. As can be seen from the figure, the other five algorithms perform worse than ours.

4.1.3. *LFR video*

This experiment is to track a tennis player in an LFR video which is manually down-sampled from a video by keeping one frame in every 35 frames, and the sequence 3 is adopted. We used 1000 samples (particles) for the PF, MCMC and AMCMC, and 300 for the other three. Six methods were tested over this video sequence. Sample frames of the tracking results are depicted in Fig. 3. As is shown in this figure, even with 1000 samples, the PF, MCMC and AMCMC failed to track the player accurately due to the abrupt motion caused by severe frame dropping. But our method could effectively deal with the difficulty even with only 300 samples, while the ASAMC and the WLMC could not effectively capture the abrupt motion and shows inferior performance to ours.

4.1.4. *Full and partial occlusions*

We used sequence 2 to test the algorithms' ability of dealing with full and partial occlusions accompanied with camera switching. Figure 4 shows the comparison of the tracking results. The running girl is occluded from frame #20 to #26. The proposed tracker captured the object accurately throughout the occluding process, and its performance is much better than the ASAMC, and the WLMC, while the other three failed to capture the object after occlusion.

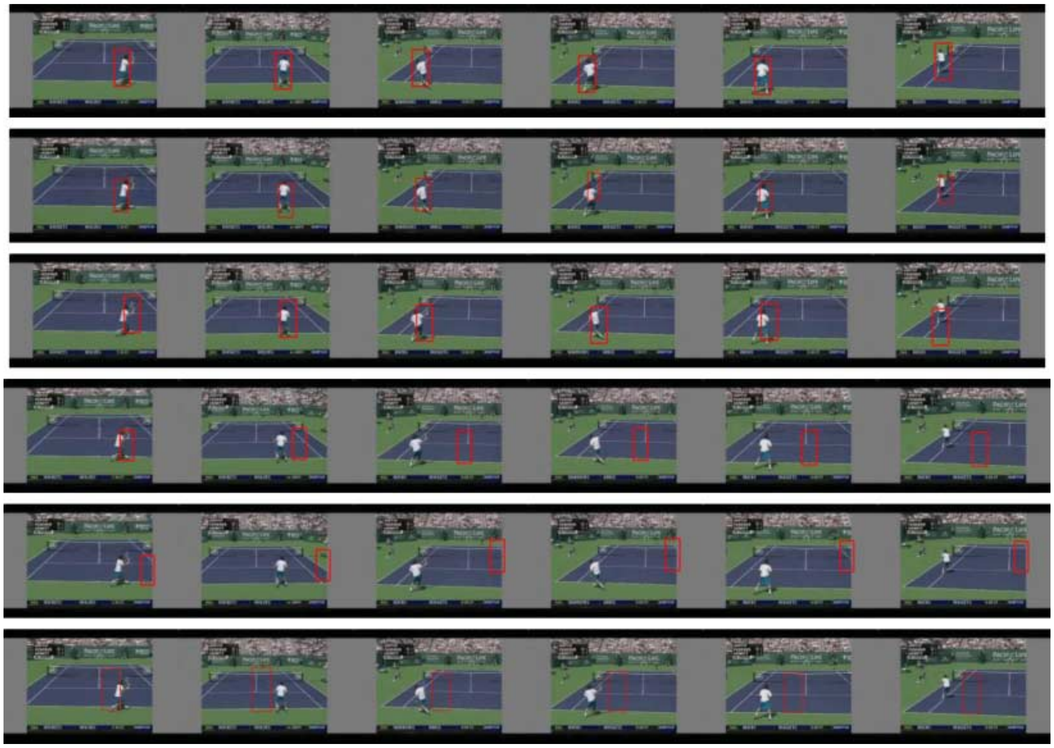

FIGURE 3. Tracking results of different methods over the *Tennis* sequence (1000 samples for PF, MCMC and AMCMC; 300 samples for ASAMC, WLMC and our method). From top to bottom are the results of our proposed tracker, ASAMC, WLMC, AMCMC, MCMC and PF, respectively. The sample frames from left to right are #16, #19, #26, #27, #28 and #31, respectively. Our method can accurately capture the player, and shows much better performance than the ASAMC and the WLMC, but the AMCMC, MCMC and PF failed to track the player. This experiment shows that our proposed tracker can efficiently deal with abrupt motion caused by severe frame dropping.

4.1.5. *Fast-moving object*

Sequence 8 is used to test the ability of the algorithms to capture a fast-moving object. We used 50 samples for the proposed tracker, the ASAMC and the WLMC, 100 samples for the MCMC and the AMCMC, and 200 for the PF. The tracking results demonstrated that the proposed algorithm can accurately track the fast-moving face throughout the sequence. The WLMC deteriorates the performance, while the MCMC and the AMCMC failed with even more samples, and the ASAMC and PF show better performance than the MCMC and AMCMC, but all are inferior to ours. Experimental results showed that when the number of samples grew up to ∼300, the MCMC and the AMCMC could yield moderately acceptable tracking results. Sample frames of the tracking results are shown in Fig. 5.

4.1.6. *Appearance change and clutter*

For the sake of testing the trackers' ability to handle object appearance changes and scenario clutter, we used the popular face sequence 9 for this experiment. In this sequence, the girl is moving with rotation of the head, change of the pose and a noisy human face. All the algorithms use 500 samples and 300 iterations. Sample figures of tracking results are shown in Fig. 6. The proposed tracker can effectively handle the appearance change of the object which was induced by pose change and head rotation. Our tracker can also handle the human face clutter (frames #440, #441). The other five trackers may give acceptable results at specific frames, but the overall performance is inferior to that of the proposed tracker.

4.2. Quantitative results

For the purpose of quantitative evaluation, we compared our method with the other five tracking methods over all the nine video sequences. A total of 600 samples are used for the proposed tracker, the ASAMC and the WLMC, and 1000 for the other three.

4.2.1. *Successful rate*

We define SR as the value of *f-measure*. That is, an object is considered as correctly tracked only if the recall and precision are both higher than 50%. It indicates the ratio between the number of correctly tracked frames and the total number of frames.

The overall performance of the six different tracking methods is shown in Table 2. It is observed that our method (600 samples),

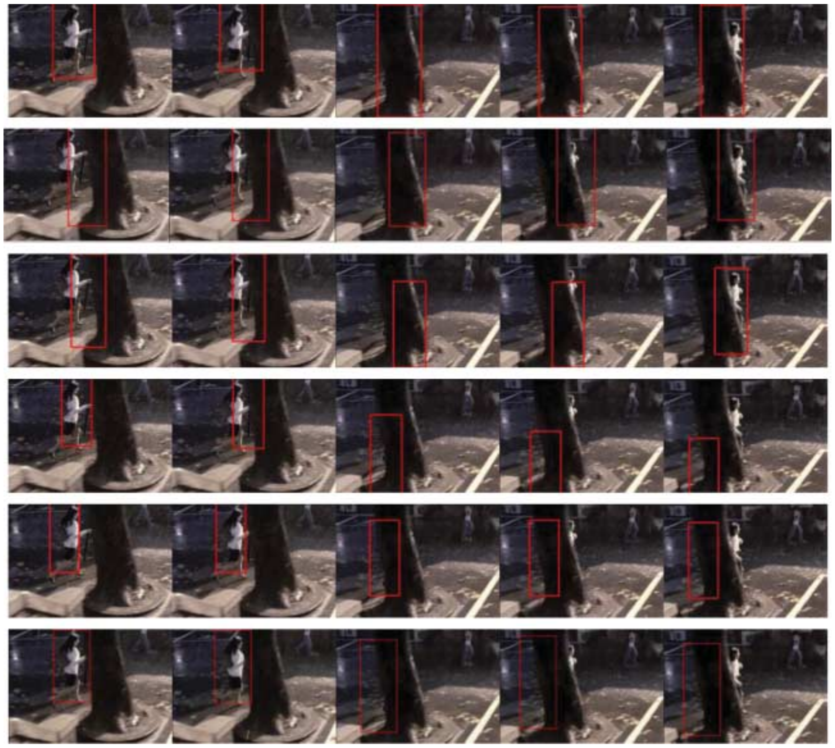

FIGURE 4. Tracking results of different methods over the *SonYeJin* sequence. From top to bottom are the results of our proposed tracker, ASAMC, WLMC, AMCMC, MCMC and PF, respectively. The sample frames from left to right are #17, #18, #24, #25 and #26, respectively. The object is occluded between frames #24 and #26. Our tracker can track the object accurately before and after occlusion. The ASAMC can track the person after occlusion but shows worse performance before occlusion. The WLMC tracker can also capture the object, but is worse than ours. After occlusion, the AMCMC, MCMC and PF lose the object.

the ASAMC (600 samples) and the WLMC (600 samples) outperform the other three, while our method is superior to the ASAMC and the WLMC method.

4.2.2. Time consumption

We compare the time consumption of different trackers over the *SonYeJin* sequence. For fair comparison, 600 samples are used for all of the trackers. The image size is 800 × 332 pixels and all the algorithms are run on a personal computer (CPU: Intel Core i3 2.4 GHz, Memory: 2 GB, OS: Windows 7). The result is shown in Table 3. It is clear that the running times of the ASAMC and the WLMC are longer than that of the PF, MCMC and AMCMC, while our proposed tracker needs the least computing time.

4.2.3. Sample interval influence

In LFR video tracking, the down-sampling interval can affect the tracking performance of different trackers, and so we tested these six trackers over four LFR video sequences, which were down-sampled from an original sequence with a sampling interval from 10 to 35 frames. We use 300 samples for the H-MCMC, and 600 for the other 5 trackers. Figure 7 illustrates the successful tracking rates of the six methods over the down-sampling interval. We can observe that our method (H-MCMC) is markedly better than others with fewer samples. Also, the tracking performance of our method, ASAMC and WLMC are less affected by the sampling interval. This result revealed the robustness of our tracker in tracking the target even with severe abrupt motions.

4.3. Parameters for H-MCMC tracker

As discussed in Section 3.3, the value of step size ε can greatly affect the performance of the proposed tracker. To evaluate the performance of the tracker over different step sizes, we used sequence 2 to test the SR under different step sizes. In this experiment, 300 samples were used. The results are shown in Fig. 8. It shows that, when $\varepsilon = 6.0$, the success rate is 86% and it begins to decrease severely when $\varepsilon > 6.0$. It reaches the lowest when $\varepsilon = 10.0$, about 18%. Thus, as to how to choose an appropriate ε, so long as the value of ε is <6.0, we can choose any possible value. Adaptively changing the step size during tracking is feasible for practical applications, which will be our further research.

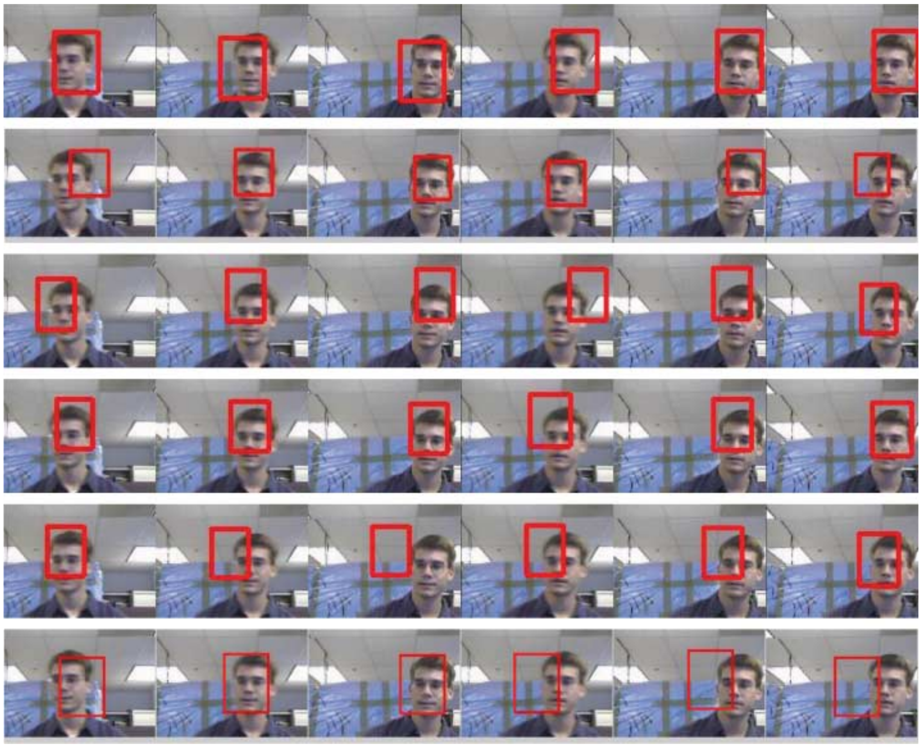

FIGURE 5. Tracking results of different methods over the *Face1* sequence. From top to bottom are the results of our proposed tracker, ASAMC, WLMC, AMCMC, MCMC and PF, respectively. The sample frames from left to right are #2, #3, #4, #19, #20 and #30, respectively.

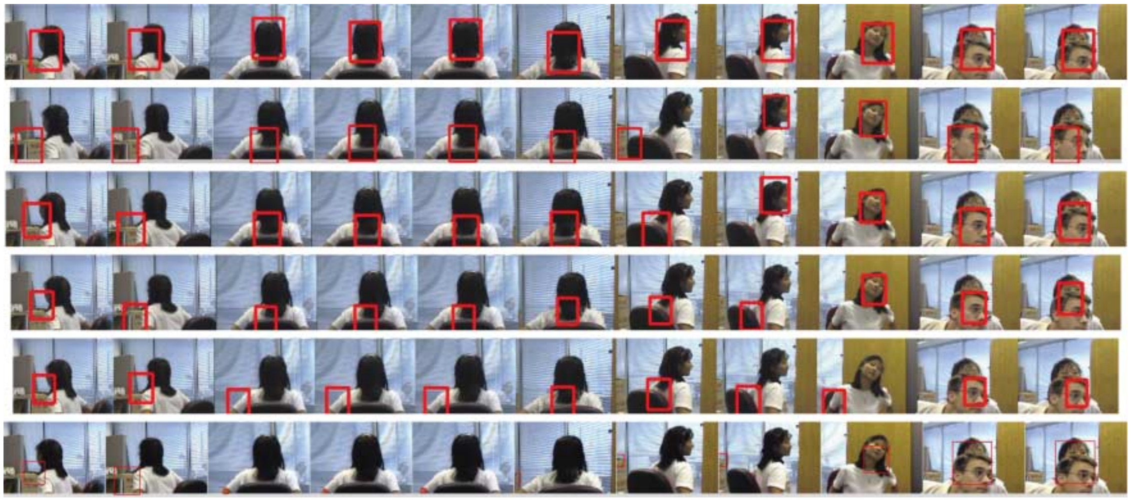

FIGURE 6. Tracking results of different methods over the *Face2* sequence. From top to bottom are the results of our proposed tracker, ASAMC, WLMC, AMCMC, MCMC and PF, respectively. The sample frames from left to right are #96, #100, #194, #196, #197, #238, #245, #247, #316, #440 and #441, respectively.

5. CONCLUSION

We have proposed an efficient tracking algorithm based on the H-MCMC within the Bayesian filtering framework. The H-MCMC is based on the HD, and can efficiently address the tracking of challenging scenarios avoiding being trapped in local maxima due to its prominent nature of ergodicity. Experimental results demonstrated that the proposed algorithm outperforms the other conventional sampling-based algorithms in some challenging situations. In our future research, we will further consider an adaptive step size-based tracking framework.

TABLE 2. Tracking performance of the six methods over the nine video sequences.

Tracking method	Samples	SR (%)
PF	1000	73.4
MCMC	1000	70.2
AMCMC	1000	71.75
WLMC	600	88.75
ASAMC	600	86.34
Ours	600	89.16

TABLE 3. Time consumption of different trackers over the *SonYeJin* sequence.

Tracking method	Samples	Time consumption (s/frame)
PF	600	0.5122
MCMC	600	0.5569
AMCMC	600	0.5676
WLMC	600	0.6546
ASAMC	600	0.6862
Ours	600	0.4052

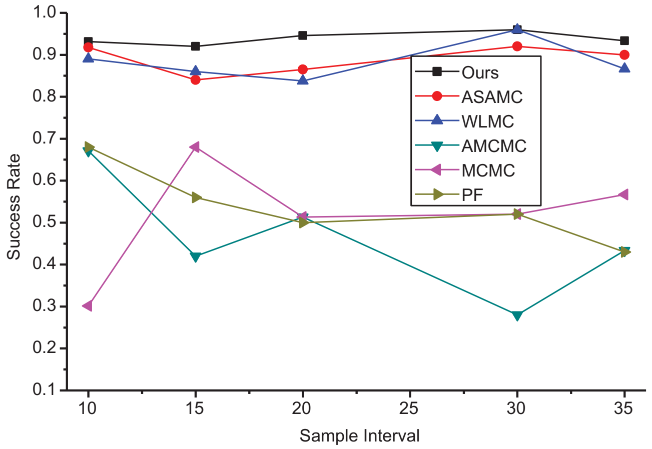

FIGURE 7. Comparison of the tracking performance affected by frame dropping on LFR videos. H-MCMC ran with 300 samples, while others with 600 samples.

FUNDING

This work was supported by the National Natural Science Foundation of China [61073133, 61175053, 60973067, 61175096, 61272369], the Innovation Group Project of China Education Ministry [2011ZD010], Fundamental research fund of DMU [2012QN030], Foundation of Scientific Planning Project of Dalian City [2011E15SF100fs] and the Natural Science Foundation of Liaoning Education Ministry [L2011241, L2010043].

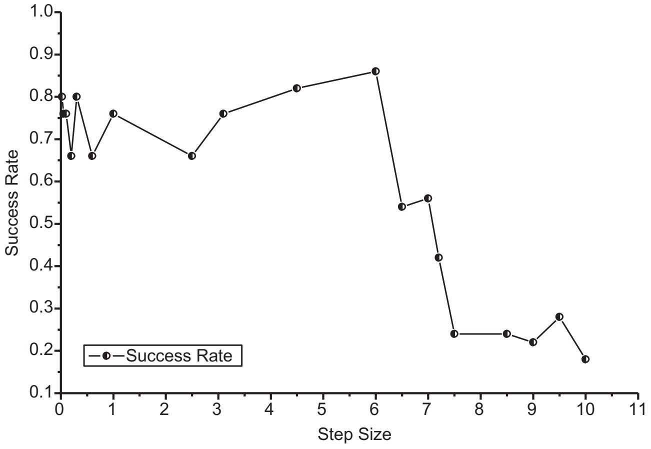

FIGURE 8. Changes of the success rate of the proposed tracker over different step sizes. This experiment is done on the *Tennis* sequence.

REFERENCES

[1] Yang, H., Shao, L., Zheng, F., Wang, L. and Song, Z. (2011) Recent advances and trends in visual tracking: a review. *Neurocomputing*, **74**, 3823–3831.

[2] Comaniciu, D., Ramesh, V. and Meer, P. (2000) Real-Time Tracking of Non-rigid Objects Using Mean Shift. *Proc. CVPR 00*, Hilton Head, SC, USA, June 13–15, Vol. 2, pp. 142–149. IEEE Computer Society, Los Alamitos, CA, USA.

[3] Cai, Y., Freitas, N. and Little, J. (2006) Robust Visual Tracking for Multiple Targets. *Proc. ECCV 06*, Graz, Austria, May 7–13, pp. 107–118. Springer, Berlin.

[4] Martinez-del-Rincon, J., Orrite, C. and Medrano, C. (2011) Rao–Blackwellised particle filter for color-based tracking. *Pattern Recognit. Lett.*, **32**, 210–220.

[5] Khan, Z., Balch, T. and Dellaert, F. (2005) MCMC-based particle filter for tracking a variable number of interacting targets. *IEEE Trans. Pattern Anal.*, **27**, 1805–1819.

[6] Zhao, T. and Nevatia, R. (2004) Tracking Multiple Humans in Crowded Environment. *Proc. CVPR 04*, Washington, DC, USA, June 27–July 2, pp. 406–413. IEEE Computer Society, Los Alamitos, CA, USA.

[7] Lin, C. and Wolf, W. (2009) MCMC-Based Feature-Guided Particle Filtering for Tracking Moving Objects from a Moving Platform. *Proc. ICCV 09*, Kyoto, Japan, September 27–October 4, pp. 828–833. IEEE, Piscataway, NJ, USA.

[8] Li, Y., Ai, H., Yamashita, T., Lao, S. and Kawade, M. (2007) Tracking in Low Frame Rate Video: A Cascade Particle Filter with Discriminative Observers of Different Life Spans. *Proc. CVPR 07*, Minneapolis, MN, USA, June 18–23, pp. 1–8. IEEE, Piscataway, NJ, USA.

[9] Huang, Y. and Essa, I. (2005) Tracking Multiple Objects Through Occlusions. *Proc. CVPR 05*, San Diego, CA, USA, June 20–26, pp. 1182–1190. IEEE Computer Society, Los Alamitos, CA, USA.

[10] Mei, X., Ling, H., Wu, Y., Blasch, E. and Bai, L. (2011) Minimum Error Bounded Efficient *l*1 Tracker with Occlusion Detection. *Proc. CVPR 11*, Colorado Springs, CO, USA, June 20–25, pp. 1257–1264. IEEE, Piscataway, NJ, USA.

[11] Kwon, J. and Lee, K. (2008) Tracking of Abrupt Motion Using Wang–Landau Monte Carlo Estimation. *Proc. ECCV 08*, Marseille, France, October 12–18, pp. 387–400. Springer, Berlin.

[12] Zhou, X., Lu, Y., Lu, J. and Zhou, J. (2012) Abrupt motion tracking via intensively adaptive Markov-chain Monte Carlo sampling. *IEEE Trans. Image Process.*, **21**, 789–801.

[13] Duane, S., Kennedy, A., Pendleton, J. and Roweth, D. (1987) Hybrid Monte Carlo. *Phys. Lett. B*, **195**, 216–222.

[14] Ishwaran, H. (1999) Applications of hybrid Monte Carlo to generalized linear models: quasicomplete separation and neural networks. *J. Comput. Graph Stat.*, **8**, 779–799.

[15] Akhmatskaya, E. and Reich, S. (2011) New hybrid Monte Carlo methods for efficient sampling: from physics to biology and statistics. *Prog. Nucl. Sci. Technol.*, **2**, 447–462.

[16] Schimidt, N. (2009) Function Factorization Using Warped Gaussian Processes. *Proc. ICML 09*, Montreal, QC, Canada, June 14–18, pp. 921–928. Omnipress, Madison, WI, USA.

[17] Choo, K. and Fleet, D. (2001) People Tracking Using Hybrid Monte Carlo Filtering. *Proc. ICCV 01*, Vancouver, Canada, July 7–14, pp. 321–328. IEEE, Los Alamitos, CA, USA.

[18] Brooks, S., Gelman, A., Jones, G. and Meng, X. (2011) *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC press, USA.

[19] Hanson, K. (2001) Markov chain Monte Carlo posterior sampling with the Hamiltonian method. *Proc. SPIE*, **4322**, 456–567.

[20] Subbey, S., Alfaki, M. and Haugland, D. (2008) The Hamiltonian Monte Carlo Algorithm in Parameter Estimation and Uncertainty Quantification. *Proc. ECMOR 08*, Bergen, Norway, September 8–11. EAGE, The Netherlands.

[21] Perez, P., Hue, C., Vermaak, J. and Gangnet, M. (2004) Color-Based Probabilistic Tracking. *Proc. ECCV 04*, Prague, Czech Republic, May 11–14, pp. 661–675. Springer, Berlin.

[22] Roberts, O. and Rosenthal, S. (2009) Examples of adaptive MCMC. *J. Comput. Graph Stat.*, **18**, 349–367.