A Generalized Search Method for Multiple Competing Hypotheses in Visual Tracking

Muhammad H. Khan, Michel F. Valstar, and Tony P. Pridmore
Computer Vision Laboratory, School of Computer Science,
University of Nottingham, UK.
{psxmhk,michel.valstar,tony.pridmore@nottingham.ac.uk}

***Abstract*—Visual tracking frameworks have traditionally relied upon a single motion model such as Random Walk, and a fixed, embedded search method like Particle Filter. As a single motion model can't reliably handle various target motion types, the interest toward multiple motion models has grown over the years. The existence of multiple competing hypotheses or predictions by the multiple motion models opens up the possibility of a wider range of search methods. To search for the target in a fixed grid of equal sized cells, an integration of the Wang-Landau method and the Markov Chain Monte Carlo (MCMC) method has recently been introduced. In this paper, we generalize this search method to cells of variable size and location, where the cells are formed around the predictions generated by multiple motion models. The effectiveness of the proposed method is tested by adopting a multiple motion model tracker. Experiments show that the modified tracker has improved accuracy and better consistency over different runs compared to its original, and superior performance over state-of-the-art trackers in challenging video sequences.**

I. Introduction

Visual tracking in image sequences is an important task in computer vision. It has many practical applications such as human computer interaction, automatic traffic control, security and surveillance, and medical imaging. Usually, a target to be tracked is initialized with a bounding box in the first frame, and it is required to estimate the trajectory of the target through subsequent frames. While easy to imagine, it is not so easy to accurately maintain track of the target in unconstrained environments due to challenges such as occlusions, rapid motion variations, illumination changes, and pose variations.

To address these challenges, many successful tracking algorithms have been proposed in the recent past. Generally speaking, a tracking algorithm builds on a combination of a matching function and a search strategy. The matching function weighs how well a certain hypothesis matches the target model, while the search strategy finds the optimal hypothesis through maximising or minimising a certain objective function, which itself is a function of the matching function. In this paper, we generalize a search method to obtain the best hypothesis from multiple competing hypotheses arbitrarily positioned in our search space.

Two different search strategies are commonly used by visual trackers: gradient descent and stochastic methods. Gradient descent methods [1],[2] remain popular due to their fast convergence rate and low computational cost, but can become trapped in local modes of the filtering distribution due to e.g. background clutter or rapid motion of a target. Stochastic methods such as particle filters (PF) [3],[4],[5] have enjoyed much success in tracking, as they can handle non-Gaussianity and multi-modality of a target distribution. PF is computationally impractical for high dimensional spaces, typically found in multi-object tracking. In the recent past, many methods [6] have been proposed to reduce the computational expense and improve the efficiency of PF. Among them, Markov Chain Monte Carlo (MCMC) methods gained popularity as efficient search methods [7],[8]. While simulating a target distribution with deep local maxima, these methods can, however, become stuck at a local maximum, leading to an inaccurate Bayesian inference. This is also known as the local trap problem.

Adaptive MCMC algorithms [9] provide an automatic way of tuning the proposal variance to maintain a certain acceptance rate of the sampler, and thus can better mix between different modes of a target distribution. However, it doesn't provide a systematic way of escaping local maxima. Kwon and Lee [10], combined the Wang-Landau Monte Carlo method with the MCMC method to escape local maxima in a complex target distribution, while searching in a regular grid that divides the image space in a number of equally sized cells. Towards a similar goal, a Stochastic approximation Monte Carlo (SAMC) based tracking algorithm was proposed by [11] to search for the optimal target state in a regular grid.

An important ingredient of visual tracking is the motion model. In stochastic methods, its purpose is to guide the search method towards the correct modes of the target distribution while avoiding search in areas with local traps. Since it is difficult to produce an accurate motion model for a large variety of tracking environments, visual tracking frameworks have conventionally depended on a single general purpose motion model like Random Walk (RW) or Nearly Constant Velocity (NCV). Their generality, resulting in an inability to model complex motion, becomes a drawback and results in poor tracking accuracy in situations where a target can display complex motion variations.

To capture different ways a target can move, some attention has been given to the notion of multiple motion models. Isard and Blake [12] learned a few distinct motion models, and a fixed finite state machine describing transitions among them from ground truth data. Later, North et al. [13] extended the work of [12] by learning more complex dynamics, and demonstrated the effectiveness of their approach on a juggling

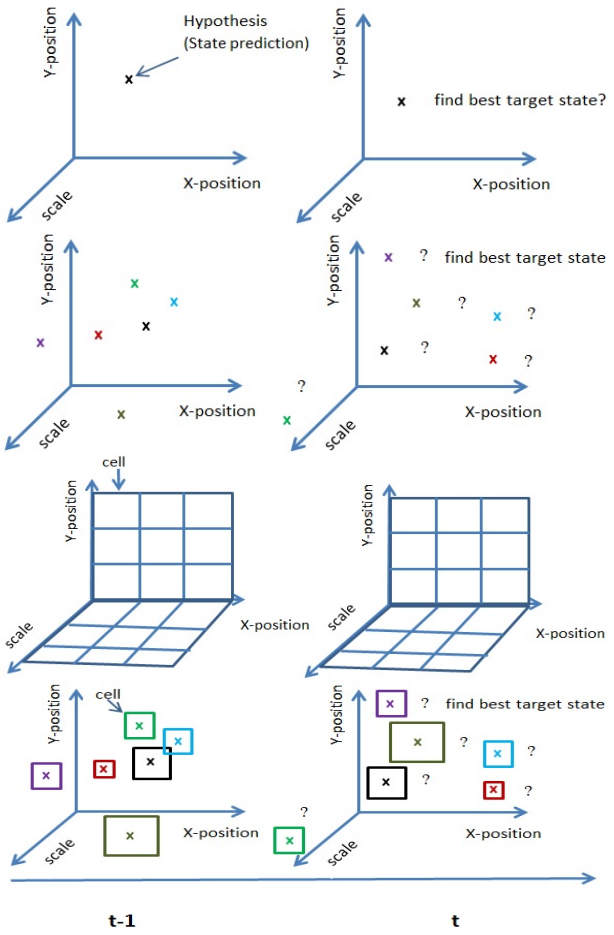


Fig. 1: Tracking frameworks relying on a single motion model produce a single state prediction, and use an embedded search strategy for finding the optimal target state around this state prediction. For instance, a linear motion model in a particle filter. First row of the figure shows a single state prediction in 3D state space at time t-1 and time t. In contrast, a multiple motion model framework generates multiple competing hypotheses or state predictions. The aim is to find the best target state from these predictions. Second row of the figure shows multiple state predictions in 3D state space at time t-1 and time t. We propose to model our search by allocating each state prediction a certain area, which we call a cell, in state space. The size of this cell is proportional to the confidence of its corresponding prediction. Last row of the figure shows cells of variable size in 3D state space at time t-1 and time t, where each cell is formed around a certain state prediction. The question of how to search for the optimal target state in these variable sized cells raises the possibility of a broader range of search strategies that can be introduced.

example. Instead of learning from some offline data, Kwon and Lee [14] sampled motion models from a pool, generated by utilizing the recent sampling history. This increased the accuracy and efficiency of the state sampling process. Kristan et al. [15] designed a two-stage dynamic model to improve the accuracy and efficiency of the bootstrap particle filter in handling various target motions. The method, however, fails when a target undergoes frequent non-constant motion. To handle complex target motion and occlusions, Khan et al. [16] proposed a multiple motion model tracker. It combines motion models learnt and applied over multiple temporal scales with an extension of the bootstrap particle filter.

Trackers employing multiple motion models such as [16] produce multiple competing hypotheses or state predictions as illustrated in the second row of Fig. 1. The question then becomes how to search for the optimal target state given these predictions. We propose to model our search by assigning each state prediction a certain area in state space, which we call a cell. The size of this cell is proportional to the uncertainty attached with its corresponding prediction, and its position in space depends on an estimator such as a motion model. The last row of Fig. 1 describes this more general problem in a 3D state space. We believe that the occurrence of the aforementioned problem in visual tracking invites the possibility of a wider range of search strategies for finding the optimal target state.

In this paper, we generalize a sampling based search method which integrates the Wang-Landau Monte Carlo method and MCMC method (WLMCMC sampling) [10]. WLMCMC operates on a regular grid of equal sized cells. We propose to generalize this to arbitrarily placed cells of variable size. In [10], the Wang-Landau method estimates the Density of States (DOS) term, which denotes the extent to which cells have been explored, and this term is used to generate moves to cells that have not been explored enough. This allows discovery of local maxima in specific cells, while jumping between them. The likelihood term in MCMC causes this method to spend more time in cells that contain highly probable target states. With this term, the method expends more samples around the current local maximum, which has already been well explored.

WLMCMC sampling [10] is a powerful method for approximating a complex filtering distribution, as was demonstrated by searching for the best target state in a regular grid of equally sized cells. However, we believe that its full potential is yet to be exploited. In this regard, we make the following contributions:

- To search for the best target state in cells having variable size and arbitrary position in state space, we generalize WLMCMC sampling [10].
- We apply the proposed solution to the multiple motion model tracker [16]. The modified tracker shows improved accuracy and better consistency over multiple runs compared to the original, and outperforms state-of-the-art trackers in challenging video sequences.

After describing the Bayesian tracking algorithm in Section II, we outline the multiple motion model framework [16] in Section III, and then describe the generalization of WLMCMC sampling with respect to this framework in Section IV.

II. Bayesian Tracking Formulation

In our visual tracking formulation the aim is to find the best state of the target at time t given observations up to t. The state at time t is given by $\mathbf{X}_t = \{X_t^x, X_t^y, X_t^s\}$, where X_t^x, X_t^y, and X_t^s represent the x,y location and scale of the target, respectively. The posterior distribution $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$, given the state $\mathbf{X}_t$ at time t and observations $\mathbf{Y}_{1:t}$ up to t, is estimated

using the Bayesian formulation

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) \propto p(\mathbf{Y}_t|\mathbf{X}_t) \int p(\mathbf{X}_t|\mathbf{X}_{t-1})p(\mathbf{X}_{1:t-1}|\mathbf{Y}_{1:t-1})d\mathbf{X}_{t-1}, \quad (1)$$

where $p(\mathbf{Y}_t|\mathbf{X}_t)$ denotes the observation model, and $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ is a motion model. Now the best state of the target $\hat{\mathbf{X}}_t$ is obtained using Maximum a Posteriori (MAP) estimation over the N particles which approximates the posterior distribution $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$:

$$\hat{\mathbf{X}}_t = \arg\max_{\mathbf{X}_t^{(i)}} p(\mathbf{X}_t^{(i)}|\mathbf{Y}_{1:t}) \; for \; i = 1, ..., N, \quad (2)$$

where $\mathbf{X}_t^{(i)}$ is the i_{th} particle. The analytical solution to Eq.(1) is intractable in practice if the filtering distribution is non-Gaussian. Conventional tracking frameworks typically use a single motion model and a fixed sampling based search strategy to approximate $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$. Here, we discuss a framework from the class of methods that employ multiple motion models and generate multiple competing hypotheses or state predictions. For such cases, PF [3] and Metropolis Hastings (MH) [17] are infeasible, and a broad range of search strategies needs to be explored.

III. A Multiple Motion Model Framework

The multiple motion model framework (M^{3}F) we apply here addresses the occlusion and non-constant motion problems typical of single target tracking. It learns motion models at different model-scales, and applies those models at multiple prediction-scales. The model-scale is the duration of a sequence of recently estimated target states. The prediction-scale is the temporal distance, measured in frames of the input image sequence, over which a prediction is made. The application of learned models at multiple prediction-scales generates multiple competing hypotheses or state predictions at each time point. To search for the best target state, M^{3}F extends PF [18], in which a fixed particle set with a certain spread is allocated around each state prediction.

To capture possibly complex motion patterns, M^{3}F learns simple motion models at different model-scales. A simple motion model is characterized by a polynomial function of order d, and represented by $\mathbf{M}$. $\mathbf{M}$ is learned at a given model-scale separately for the x-location, y-location, and scale s of the target's state. For instance, an $\mathbf{M}$ of order 1, learned at model-scale m, predicts a target's x-location at time t as:

$$\tilde{x_t} = \beta_o^m + \beta_1^m t, \quad (3)$$

where β_1 is the slope, and β_o the intercept.

These models are learned at each time t, and a set of these models is represented by $\mathbf{M}_t^{j=1,...,|\mathbf{M}_t|}$, where $|.|$ is the cardinality of the set. Each model predicts target state $l(\tilde{x}, \tilde{y}, \tilde{s})$ at T prediction-scales.

Suppose there are T sets of motion models available at time t, one from each of T previous time-steps. Each set of models at time t is represented by its corresponding set of predictions. The most suitable motion model from each set is selected as follows.

Let us denote $G = |\mathbf{M}_t|$, and let $\mathbf{l}_t^k = \{l_t^{j,k}|j = 1, ..., G\}$ represent a set of states predicted by G motion models learnt at time $t-k$, where $l_t^{j,k}$ denotes the predicted state by j_{th} motion model learned at k_{th} previous time-step. As $k = 1, ..., T$, there are T sets of predicted states at time t. Now the most suitable motion model $\mathbf{R}_t^k$ is selected from each set using the following criterion:

$$\hat{l}_t^k = \arg\max_{l_t^{j,k}} p(\mathbf{Y}_t|l_t^{j,k}) \quad (4)$$

where $\hat{l}_t^k$ is the most suitable state prediction from the set $\mathbf{l}_t^k$, and $p(\mathbf{Y}_t|l_t^{j,k})$ measures the visual likelihood at the predicted state $l_t^{j,k}$. In other words, $\hat{l}_t^k$ is the most suitable state prediction of the most suitable motion model $\mathbf{R}_t^k$. After this selection process, the T sets of motion models are reduced to T individual models.

There exist T most suitable state predictions at time t. We model our search by allocating each state prediction $\hat{l}_t^k$ a certain area in the state space, which we call a cell. The size of this cell is equivalent to the uncertainty attached to its corresponding hypothesis (see the last row of Fig. 1). For instance, the size of the cell around $\hat{l}_t^k$ will be $2 \times \sigma_x \times k$ pixels, and $2 \times \sigma_y \times k$ pixels, respectively. Along the third dimension, scale, the uncertainty would be $\sigma_s \times k$ around the predicted state $\hat{l}_t^k$. Here σ_x, σ_y, and σ_s are the standard deviations of a zero-mean Gaussian noise acting on translation x and y, and scale s between two consecutive time-steps, respectively.

Given T cells of variable size, which might overlap, the aim is to search for the best target state $\hat{\mathbf{X}}_t$ in these cells. An intuitive, and pragmatic approach would be to visit all the cells to some extent, and spend more time in those where there are highly probable target states. Moreover, these cells can be far apart in space. Sampling based search methods such as the MH algorithm [17], used conventionally in tracking frameworks, cannot be used here. While searching a large area with large proposal variance, the MH algorithm has the tendency to become stuck in local maxima. With a smaller variance, it would require many samples to search a large area, and again get trapped in a local maximum if there are deep valleys between the different modes of the target distribution. We generalize WLMCMC sampling to solve these problems in the next section.

The original instantiation of M^{3}F extends PF to allocate a fixed number of particles with a certain spread around each state prediction to obtain $\hat{\mathbf{X}}_t$. This spread is equal to the uncertainty associated with the corresponding state prediction computed as described above. We hypothesize that compared to the search based on extended PF, the proposed generalization of WLMCMC will not only improve the accuracy of the M^{3}F, but will result in more consistent tracking across different runs.

IV. A Generalized WLMCMC Sampling

WLMCMC sampling was introduced by [10] to search for the target in a whole image after dividing it into a fixed grid of

equal sized cells (see the third row of Fig. 1). It is composed of the Wang-Landau estimation and Markov Chain Monte Carlo (MCMC) method. Wang-Landau estimation is a Monte Carlo algorithm that was introduced in physics literature for calculating the density of states (DOS) by performing a set of random-walks in different energy cells [19].

A. Wang Landau Monte Carlo (WLMC) method

The aim is to estimate the DOS score for every cell, where the DOS score is high for a cell if it contains highly probable target states. As it is intractable to accurately calculate the DOS score for every cell, the WLMC method is used to estimate it. This method maintains a histogram h, and each bin of this histogram corresponds to a specific cell C_k. When C_k is visited, its bin count $h(C_k)$ is increased by 1, and its DOS score $g(C_k)$ is modified by multiplying a modification factor $f > 1$.

$$g(C_k) \leftarrow g(C_k) * f, \tag{5}$$

where $g(C_k)$ is initially set to 1 for all k. As the simulation progresses, the random-walk generates a semi-flat histogram. A histogram is considered semi-flat if the value of the lowest bin is larger than 80% of the average value of all bins in h [19]. The semi-flat histogram denotes that the method has explored all the cells to at least some degree. Now the method performs the next random-walk in a coarse-to-fine manner to obtain more accurate DOS estimates. For this, the f factor is reduced to $f \leftarrow \sqrt{f}$ and the histogram is reset to 0. The method continues until the histogram becomes semi-flat again; then restarts the random-walk with a finer modification factor. The algorithm terminates when the modification factor becomes close to 1 or the number of iterations reaches a pre-defined value.

B. Proposal Step

The proposal step defines how the transition from the current state to a new state will occur based on some previous knowledge of the target motion. In this case, the previous knowledge of the target motion is that it can move from the current cell to any of the cells within one proposal step. The proposal density is defined as

$$Q(\mathbf{X}_t^{'}; \mathbf{X}_t) = Q_c(\mathbf{X}_t^{'}) \tag{6}$$

Q_c proposes a new state $\mathbf{X}_t^{'}$ in two stages. In the first stage, a cell C_k is chosen randomly from the T available cells. In the second stage, the x-location and y-location of $\mathbf{X}_t^{'}$ are uniformly drawn from the chosen cell C_k, and the scale part of $\mathbf{X}_t^{'}$ is proposed by adding zero-mean Gaussian noise with standard deviation $\sigma_s \times k$ to the scale part of $\hat{l}_t^k$.

C. Acceptance Step

The acceptance ratio decides whether the proposed state is accepted or not using the likelihood ratio between the proposed state $\mathbf{X}_t^{'}$ and the current state $\mathbf{X}_t$

$$a = min\left[1, \frac{p(\mathbf{Y}_t|\mathbf{X}_t^{'})Q(\mathbf{X}_t; \mathbf{X}_t^{'})}{p(\mathbf{Y}_t|\mathbf{X}_t)Q(\mathbf{X}_t^{'}; \mathbf{X}_t)}\right] \tag{7}$$

The WLMCMC algorithm integrates the density of states term with the acceptance ratio in Eq.(7). Let D be a mapping function from the state $\mathbf{X}_t$ to the cell C_k, which contains the state $\mathbf{X}_t$.

$$D : X_t \rightarrow C_k \tag{8}$$

Then the acceptance ratio becomes

$$a = min\left[1, \frac{p(\mathbf{Y}_t|\mathbf{X}_t^{'})\frac{1}{g(D(\mathbf{X}_t^{'}))}Q(\mathbf{X}_t; \mathbf{X}_t^{'})}{p(\mathbf{Y}_t|\mathbf{X}_t)\frac{1}{g(D(\mathbf{X}_t))}Q(\mathbf{X}_t^{'}; \mathbf{X}_t)}\right], \tag{9}$$

where $g(D(\mathbf{X}_t^{'}))$ denotes the density of states in the cell which contains $\mathbf{X}_t^{'}$. Eq.(9) has two important advantages over Eq.(7). The first advantage is that it provides a systematic way for a Markov Chain to escape local maxima and capture the global maximum. This is required because the cells could be positioned far apart from each other in the space. And in these situations, the Markov Chain has a higher probability of meeting local maxima. The second advantage of Eq.(9) is that it enables Markov Chain to spend more time around local maxima, while guaranteeing to visit all the cells to some extent. Again, this is desirable in this scenario because there could be any number of cells containing highly probable target states, and this should be discovered by visiting each of them to some degree.

The DOS score $g(D(\mathbf{X}_t^{'}))$ in Eq.(9) is calculated exactly in the same way as described in subsection A. For instance, if a state proposed by Eq.(6) is accepted by the acceptance ratio in Eq.(9), and the state belongs to cell C_k, then the DOS score of the cell $g(C_k)$ is modified by the factor f, and its bin count $h(C_k)$ is increased by 1. Otherwise, the same procedure is applied to the cell which contains the current state. Algorithm 1 details relevant steps of the generalized WLMCMC method given variable sized cells, where each cell is formed around a certain state prediction.

Algorithm 1 Generalized WLMCMC method for variable sized cells.

Input: A set of variable sized cells at time t: $\mathbf{C_t} = \{C_k | k = 1, ...T\}$

Output: Best state of the target at time t: $\hat{\mathbf{X}}_t$

Initialize the DOS score for each cell $g(C_k) = 1$, and the bin count for each cell $h(C_k) = 0$, where $k = 1, ..., T$.

Set $f = 2.7$

for $q = 1$ *to N*, where N is the total number of particles

- Given the current state $\mathbf{X}_t^q$, propose a new state $\mathbf{X}_t^{'}$ using the Eq.(6).
- Use Eq.(9) to compute the acceptance ratio.
- if the proposed state is accepted then set $\mathbf{X}_t^{q+1}$ to $\mathbf{X}_t^{'}$, else set $\mathbf{X}_t^{q+1}$ to $\mathbf{X}_t$.
- $g(D(\mathbf{X}_t^{q+1})) \leftarrow g(D(\mathbf{X}_t^{q+1})) * f$
- $h(D(\mathbf{X}_t^{q+1})) \leftarrow h(D(\mathbf{X}_t^{q+1})) + 1$
- If h is semi-flat then reset $h(C_k) = 0, \forall k$ and $f \leftarrow \sqrt{f}$.

end

Compute the best state $\hat{\mathbf{X}}_t$ using Eq.(2).

V. Experimental Details and Results

We used ten video sequences for experiments. The sequences are *TUD-Campus*[20], *TUD-Crossing*[20],*ball2* [16], *PETS 2001 Dataset 1*[1],*toy1* [2], *Person*[21], *ball1* [16], *car*[22], *Person2* [3], and *squash* [16].

We denote the multiple motion model framework (M^3F) based on generalized WLMCMC by M^3F-GWL. M^3F-GWL was compared to the original instantiation of M^3F denoted by M^3F-PF [16], and six state-of-the-art trackers. The state-of-the-art trackers are WLMCMC [10], Visual Tracking Decomposition (VTD) [23], Fragment-based Tracker (FragT) [24], Incremental Subspace Visual Tracker (IVT) [25], Real-Time Robust L1-Tracker using Accelerated Proximal Gradient (L1-APG) [26], and Semisupervised Boosting Tracker (Semi) [27].

In terms of search methods, M^3F-PF, IVT, and L1-APG are based on particle filters, FragT and Semi utilize dense sampling methods, WLMCMC and VTD use MCMC. The minimum and maximum number of samples used for WLMCMC, VTD, IVT, and L1-APG was 600 and 640, respectively. The minimum and the maximum size of the cell in terms of half width and half height in image space were 1 pixel and 30 pixels, respectively. For M^3F-PF and M^3F-GWL, model-scales of 2,3,4, and 5 frames were used, and at each model-scale a linear motion model was learned. M^3F-PF and M^3F-GWL utilized the HSV colour histogram as the observation model and Bhattacharyya coefficient as the distance measure [3].

A. Performance Evaluation

Table 1 reports tracking accuracy of 8 trackers on 10 video sequences in terms of centre location error in pixels, averaged over 5 runs of each tracker. The first eight videos involve occlusions of varying lengths, and the last two contain both occlusions and rapid motion variations. IVT and SemiBoost did not perform well in any of the sequences as the former updates its holistic appearance model in a blind manner, and the latter relies on a naive detection strategy after the target is occluded. An efficient sampling scheme combined with an annealing procedure allows WLMCMC to perform best in the *TUD-Campus* sequence. FragT achieved top and second best performance in *car* and *TUD-Crossing* sequences, respectively, which involved partial occlusions. Likewise, L1-APG produced the lowest center location error in the *TUD-Crossing* sequence. FragT uses a part-based appearance model, while L1-APG relies on an explicit occlusion detection mechanism whose output is linked to a robust minimization model.

M^3F-PF showed better accuracy than the state-of-the-art trackers in 6 out of 10 sequences in handling occlusions and motion variations. The last column shows that M^3F-GWL improved the accuracy of M^3F-PF in almost every sequence using the same number of particles, although the only difference between M^3F-PF and M^3F-GWL is the search method. Given multiple cells of variable size formed around state predictions, M^3F-GWL produces more samples from the cells containing higher probability of local maxima that increases chances of reaching to the global maximum. On the other hand, M^3F-PF just allocates a fixed number of particles with a certain spread around each state prediction, and thus it can miss the global maximum more often. For instance in Fig. 2, when the target re-appears after occlusion in *person2* sequence, M^3F-GWL displays more accurate tracking than M^3F-PF by capturing the global maximum more often than M^3F-PF.

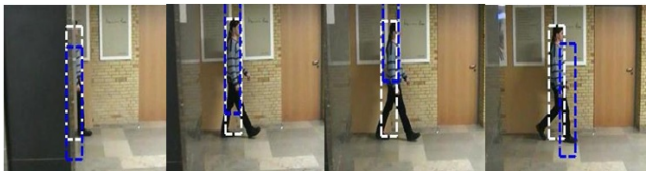


Fig. 2: A comparison of tracking accuracy between M^3F-GWL(white) and M^3F-PF(blue). M^3F-GWL shows more accuracy in tracking a person when it becomes visible after occlusion in comparison to M^3F-PF.

Fig.3 shows a comparison between M^3F-PF and M^3F-GWL in terms of tracking consistency over 5 runs with and without occlusions on five different sequences. The improved consistency of M^3F-GWL over M^3F-PF under both situations suggests that while the multiple motion model framework [16] has the potential to handle occlusions, its tracking consistency can be improved further with a sophisticated search method such as generalized WLMCMC.

Fig.4(a) shows tracking results in the *person* sequence, which is captured with a moving camera in an outdoor environment. VTD, FragT, L1-APG, and IVT failed to recapture the target after the first occlusion, while Semi reacquired the target a few frames after occlusion. In contrast, M^3F-PF, M^3F-GWL, and WLMCMC successfully tracked the target throughout this video sequence.

In the *ball2* sequence, the target undergoes occlusions of different lengths. Fig. 4(b) shows that M^3F-GWL recovers the target more quickly than M^3F-PF after occlusion (frame #

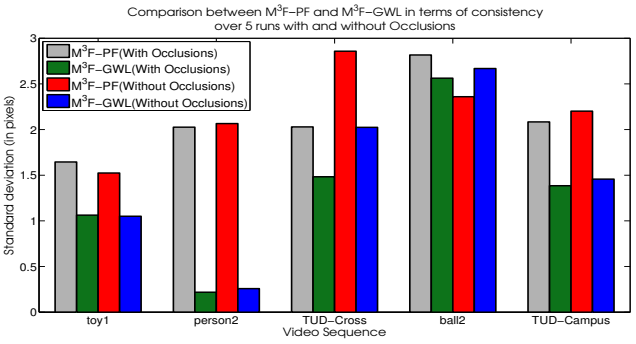


Fig. 3: Comparison of tracking consistency over 5 runs between M^3F-GWL and M^3F-PF with and without occlusions. Each bar in this figure is a standard deviation (in pixels) calculated over a set of five mean center location errors (in pixels). Each mean value in this set is computed by averaging the centre location error over all frames (with or without occlusion) for a video sequence.

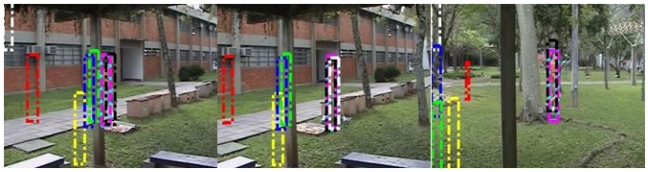
[1] *PETS 2001 Dataset 1* is available from http://ftp.pets.rdg.ac.uk/

[2] *toy1* is made by us.

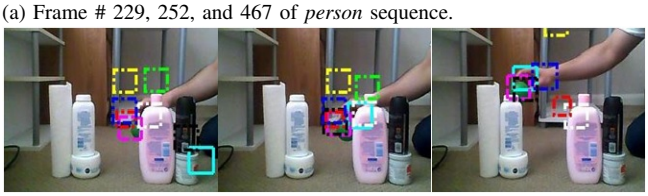
[3] *Person2* is available from http://www.iai.uni-bonn.de/ kleind/tracking/

TABLE I: **Tracking accuracy of 8 trackers on 10 video sequences.** Mean centre location error in pixels is given, averaged over all frames for a video sequence. Each tracker was run five times and the results were averaged. The best results are marked in bold. N is the number of particles used in M^{3}F-PF and M^{3}F-GWL.

Sequence	IVT	L1-APG	VTD	Semi	FragT	WLMCMC	M^{3}F-PF	M^{3}F-GWL	N
toy1	111.396	110.773	98.265	99.085	107.894	23.911	23.762	**21.388**	600
ball2	104.229	71.851	66.729	78.359	106.905	29.928	19.491	**18.708**	640
TUD-Ca	186.647	100.126	187.023	61.282	112.127	**22.034**	23.604	22.772	160
TUD-Cr	41.859	**2.351**	63.495	62.039	4.912	37.856	30.671	27.999	500
PETS'01	76.472	60.676	83.207	114.551	67.446	55.334	26.755	**25.440**	640
Person	83.579	115.822	85.080	177.683	83.956	19.141	10.218	**9.442**	400
car	81.342	31.280	47.504	38.533	**15.769**	27.915	27.842	27.033	400
Person2	12.236	26.231	18.861	49.854	**9.213**	12.042	11.732	10.460	400
squash	122.205	60.918	20.036	68.629	35.677	19.864	11.245	**10.322**	100
ball1	144.537	124.676	69.184	66.734	210.258	20.711	16.827	**15.470**	280



(a) Frame # 229, 252, and 467 of *person* sequence.



(b) Frame # 123, 126, and 188 of *ball2* sequence.

Fig. 4: Tracking results in *person* and *ball2* sequences. M^{3}F-GWL(magenta), M^{3}F-PF(cyan), WLMCMC(black), FragT(yellow), Semi(white), L1-APG(red), VTD(blue) and IVT(green).

123), and tracks more accurately (frame # 126) afterwards.

VI. Conclusion

In this paper, we generalize a search method for multiple competing hypotheses in visual tracking. Such hypotheses are usually state predictions generated in a multiple motion model framework. The search is modelled by assigning a certain area in state space, which we call a cell, to each state prediction. To search for the best target state in these cells, we generalize WLMCMC sampling to cells of variable size and location. To show the effectiveness of the proposed solution, we adapt the multiple motion model tracker of [16]. The modified tracker demonstrates improved accuracy and better consistency over different runs than the original, and shows superior performance over other trackers in challenging tracking environments.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *PAMI*, vol. 25, no. 5, pp. 564–577, 2003.

[2] C. Yang, R. Duraiswami, and L. Davis, "Efficient mean-shift tracking via a new similarity measure," in *CVPR, 2005.*

[3] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *ECCV, 2002.*

[4] M. Isard and A. Blake, "Condensationconditional density propagation for visual tracking," *IJCV*, vol. 29, no. 1, pp. 5–28, 1998.

[5] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans," *PAMI*, vol. 30, no. 10, pp. 1728–1740, 2008.

[6] O. Cappe, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential monte carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[7] Z. Khan, T. Balch, and F. Dellaert, "Mcmc-based particle filtering for tracking a variable number of interacting targets," *PAMI*, vol. 27, no. 11, pp. 1805–1819, 2005.

[8] K. Smith, D. Gatica-Perez, and J.-M. Odobez, "Using particles to track varying numbers of interacting people," in *CVPR, 2005.*

[9] G. O. Roberts and J. S. Rosenthal, "Examples of adaptive mcmc," *Journal of Computational and Graphical Statistics*, vol. 18, no. 2, pp. 349–367, 2009.

[10] J. Kwon and K. M. Lee, "Tracking of abrupt motion using wang-landau monte carlo estimation," in *ECCV, 2008.*

[11] X. Zhou, Y. Lu, J. Lu, and J. Zhou, "Abrupt motion tracking via intensively adaptive markov-chain monte carlo sampling," *CVPR, 2010.*

[12] M. Isard and A. Blake, "A mixed-state condensation tracker with automatic model-switching," in *ICCV, 1998.*

[13] B. North, A. Blake, M. Isard, and J. Rittscher, "Learning and classification of complex dynamics," *PAMI*, vol. 22, no. 9, pp. 1016–1034, 2000.

[14] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *ICCV, 2011.*

[15] M. Kristan, S. Kovacic, A. Leonardis, and J. Pers, "A two-stage dynamic model for visual tracking," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 6, pp. 1505–1520, 2010.

[16] M. H. Khan, M. Valstar, and T. Pridmore, "A multiple motion model tracker handling occlusion and rapid motion variation," in *British Machine Vision Workshop (BMVW) 2013*. British Machine Vision Association, 2013.

[17] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[18] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188, 2002.

[19] F. Wang and D. P. Landau, "Efficient, multiple-range random walk algorithm to calculate the density of states," *Phys. Rev. Lett.*, vol. 86, no. 10, p. 2050, 2001.

[20] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *CVPR, 2008.*

[21] L. Dihl, C. R. Jung, and J. Bins, "Robust adaptive patch-based object tracking using weighted vector median filters," in *SIBGRAPI, 2011.*

[22] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *PAMI*, vol. 34, no. 7, pp. 1409–1422, 2012.

[23] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *CVPR, 2010.*

[24] A. Adam, E. Rivlin, and I. S, "Robust fragments-based tracking using the integral histogram," in *CVPR, 2006.*

[25] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1-3, pp. 125–141, 2008.

[26] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *CVPR, 2012.*

[27] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *ECCV, 2008.*