Ph.D. DISSERTATION

Robust Visual Tracking with Uncertainty Analysis of Probabilistic Models

확률 모델의 불확실성을 고려한 강인한 물체 추적 기법

BY

JUNSEOK KWON

FEBRUARY 2013

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Abstract

In this dissertation, several robust tracking algorithms are proposed, which can work robustly in a real-world scenario. A visual tracker is defined as combination of four important ingredients, namely, appearance model, motion model, state representation type, and observation type. These ingredients are associated with the probability distributions. These probability distributions compose a single posterior, where the objective of visual tracking is to fine the best configuration of the target given the observation, which maximizes the posterior probability. However, our information about the tracking system to be modeled may not allow us to characterize those ingredients with precise distributions, because the distributions cannot be perfectly known in many practical cases. Thus, an any single posterior necessarily includes estimation error. We tackle this uncertainty of probabilistic distributions in the visual tracking problem, and present novel tracking algorithms to accurately track the targets under very challenging tracking environment. Two approaches are adopted to address the uncertainty of probabilistic distributions, which are Bayesian Model Averaging (BMA) and Interval Analysis (IA). The philosophy of these two approaches is that, due to the uncertainty of probabilistic distributions, the posterior cannot be uniquely determined and multiple candidates of the posteriors should be used. Then, depending on approaches to solve the uncertainty and ingredients to be considered, six different tracking methods are proposed: Wang-Landau

Monte Carlo (WLMC) tracker, Basin Hopping Monte Carlo (BHMC) tracker, Visual Tracker Sampler (VTS) tracker, Minimum Uncertainty Gap (MUG) tracker, Soft Bounding Box (SBB) tracker and Interval Tracker (IT).

Following the BMA approach, the WLMC tracker averaged multiple motion models to track smooth and abrupt motions at the same time. The BHMC tracker averaged multiple state models. Multiple state models enable the method to successfully describe the non-rigid targets, whose the geometric appearance changes over time. The VTS tracker averaged multiple observation and state models as well as appearance and motion models. Each appearance and motion model covers a specific appearance of the object and a different type of motion, respectively. Multiple observation and state models help be robust to the noise and motion blur. Following the IA approach, the MUG tracker obtained the lower and upper bounds of the estimated likelihood. With the likelihood interval, the likelihood estimation error is reduced by minimizing the gap between two bounds. The SBB tracker obtained the lower and upper bounds of the estimated state. Using the state interval, non-rigid targets are efficiently represented for the robust visual tracking. IT obtained the lower and upper bounds of both the estimated likelihood and the estimated state. The best state is achieved by maximizing a posterior probability and, at the same time, minimizing uncertainty of the posterior, which induces M4 (**M**MSE-**M**AP-**M**L-**M**UP) estimation.

In IA and BMA approaches, IA is the superset of BMA because IA has advanced two properties. First, IA utilizes an infinite number of candidates in interval, while BMA only utilizes a finite number of probabilistic model candidates. Second, IA can perform any arbitral operations using candidates, while BMA only averages probabilistic model candidates. The tracking methods, WLMC, BHMC, VTS, MUG,

SBB and IT have been developed toward tracking the targets under more complex and combinatorial tracking environment and toward using varying number of multiple trackers. IT can be combined with WLMC, BHMC, and VTS, which improve the tracking performance of original WLMC, BHMC, and VTS. In addition, IT includes MUG and SBB because IT utilizes the intervals of both the state in SBB and appearance models in MUG.

Experimental results show that the proposed methods efficiently solve the aforementioned uncertainty of probabilistic distributions. In several challenging realistic videos, the proposed methods track the targets accurately and reliably where the tracking environments are drastically changing over time. They outperform the recent state-of-the-art tracking methods.

Key words: Visual Tracking, Uncertainty Analysis of Probabilistic Distributions, Bayesian Model Averaging, Interval Analysis, Markov Chain Monte Carlo

Student number: 2008-30208

Contents

List of Figures

List of Tables

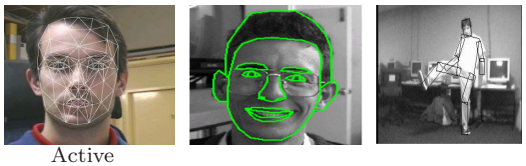
Chapter 1

Introduction

The objective of the tracking problem is to track the target accurately in the real-world scenario [4, 5]. In this scenario, it is a very challenging task to track an object since the scenario typically includes severe appearance or motion changes of the object. The appearance changes cover geometric and photometric variations of an object such as occlusion, pose, or illumination changes [6, 7, 8, 9, 10, 11, 12, 13, 1, 14]. Severe motion changes usually occur when a video has a low frame rate or when an object moves abruptly [15, 16, 17].

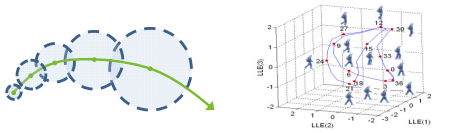
1.1 Background and Research Issues

To robustly track the target in the real-world scenario, most conventional tracking methods design the tracking problem as the Bayesian formulation. In the Bayesian tracking approach, the goal of the tracking problem is changed to find the best state, which maximizes the posterior probability $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$. This is called as the



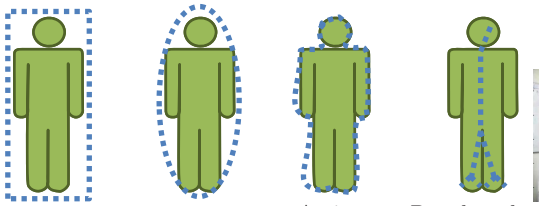
Active appearance model | Active shape model | Pictorial shape model

(a) Appearance model: $p(\mathbf{Y}_t|\mathbf{X}_t)$



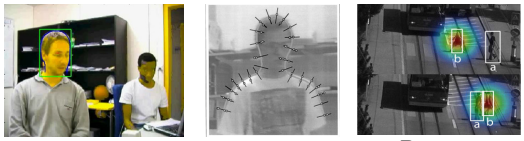
Linear transformation | Non-linear dynamics

(b) Motion model: $p(\mathbf{X}_t|\mathbf{X}_{t-1})$



Rectangle | Ellipse | Active contour | Part-based Models

(c) State model: $\mathbf{X}_t$



Color | Contour | Detector Confidence

(d) Observation model: $\mathbf{Y}_t$

Figure 1.1: **Four important ingredients of visual tracker** from Kevin Smith's slide. (a) The tracking system can measure the likelihood in many ways. (b) The tracking system can predict the next state based on current state in many ways. (c) The state can parameterize configuration (e.g. position, size) of the object in many ways. (d) The tracking system can obtain observation from images in many ways.

Maximum a Posterior (MAP) estimation:

$$\hat{\mathbf{X}}_t = \arg\max_{\mathbf{X}_t} p(\mathbf{X}_t|\mathbf{Y}_{1:t}), \tag{1.1}$$

where $\hat{\mathbf{X}}_t$ denotes the best state (MAP state) at time t given the observation $\mathbf{Y}_{1:t}$. To achieve this goal of the tracking problem, conventional tracking methods have tried to obtain the optimal MAP state using advanced stochastic [18] or deterministic [19] optimizers.

For example, the stochastic optimizers [18] approximately obtain the Maximum

a Posteriori (MAP) state over the N number of samples at each time t.

$$\hat{\mathbf{X}}_t = \underset{\mathbf{X}_t^{(l)}}{\arg\max}\, p(\mathbf{X}_t^{(l)}|\mathbf{Y}_{1:t}) \quad for\ l = 1, \ldots, N, \tag{1.2}$$

where $\hat{\mathbf{X}}_t = (\hat{X}_t^x, \hat{X}_t^y, \hat{X}_t^s)$ denotes the best state at time t given the observation $\mathbf{Y}_{1:t}$, and N is the total number of samples. In (1.2), the best sample $\hat{\mathbf{X}}_t$ is chosen among the N number of samples $\{\mathbf{X}_t^{(l)}\}_{l=1}^N$, which produces the highest posterior probability value $p(\mathbf{X}_t^{(l)}|\mathbf{Y}_{1:t})$.

One of popular stochastic optimizers is the Metropolis Hastings (MH) algorithm [20]. The MH algorithm defines a single Markov Chain and acquires samples over the chain. To obtain samples, two main steps are performed, namely, the proposal step and the acceptance step. These two steps iteratively go on until the number of iterations reaches a predefined value. The proposal step proposes a new state given the previous state based on some prior knowledge about the motion. The most commonly used prior knowledge about the motion is that the transition is governed by the Gaussian distribution. Thus, the proposal density is designed by

$$Q(\mathbf{X}_t'; \mathbf{X}_t) = G(\mathbf{X}_t, \sigma^2), \tag{1.3}$$

where $\mathbf{X}_t'$ is the new state, and G denotes the Gaussian function with mean $\mathbf{X}_t$ and variance σ^2. The acceptance step determines whether the next state $\mathbf{X}_t'$ is accepted or not:

$$a = min\left[1, \frac{p(\mathbf{X}_t'|\mathbf{Y}_{1:t})Q(\mathbf{X}_t; \mathbf{X}_t')}{p(\mathbf{X}_t|\mathbf{Y}_{1:t})Q(\mathbf{X}_t'; \mathbf{X}_t)}\right], \tag{1.4}$$

where $p(\mathbf{X}_t'|\mathbf{Y}_{1:t})$ denotes the posterior probability over the state $\mathbf{X}_t'$, and $Q(\mathbf{X}_t'; \mathbf{X}_t)$ represents the proposal density in (1.3). With (1.3)(1.4), the MCMC-based tracking method bases a new approach on a different Monte Carlo approximation of the posterior, in terms of unweighted samples, which is introduced in the work of Khan et.

al. [18]. Based on [18], the method obtains the following Monte Carlo approximation to the exact Bayesian filtering distribution $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ in (1.7):

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) \approx p(\mathbf{Y}_t|\mathbf{X}_t) \sum_{l=1}^{N} p(\mathbf{X}_t|\mathbf{X}_{t-1}^{(l)}), \tag{1.5}$$

where $p(\mathbf{Y}_t|\mathbf{X}_t)$ denotes the likelihood term over the state $\mathbf{X}_t$ and $p(\mathbf{X}_t|\mathbf{X}_{t-1}^{(l)})$ represents the proposal density that proposes a new sampled state $\mathbf{X}_t$ at time t based on the l-th sampled state at time $t-1$. The predictive prior $\int p(\mathbf{X}_t|\mathbf{X}_{t-1})$ $p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1})$ $d\mathbf{X}_{t-1}$= $p(\mathbf{X}_t|\mathbf{Y}_{1:t-1})$ is also obtained by the Monte Carlo approximation.

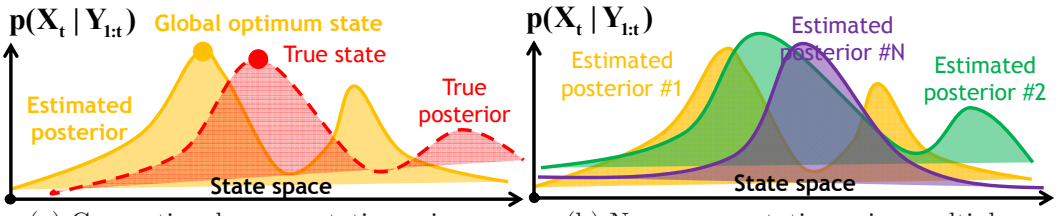
$$p(\mathbf{X}_t|\mathbf{Y}_{1:t-1}) \approx \sum_{l=1}^{N} p(\mathbf{X}_t|\mathbf{X}_{t-1}^{(l)}). \tag{1.6}$$

In (1.5)(1.6), the proposal density randomly selects a sample $\mathbf{X}_{t-1}^{(l)}$ and moves the selected sample according to the second-order autoregressive function, and use the result as the initial state of the $\mathbf{X}_t$ Markov chain. Then, the sampling procedure results in a unweighted particle approximation for the posterior $p(\mathbf{X}_t|\mathbf{Y}_{1:t}) \approx \{\mathbf{X}_t^{(l)}\}_{l=1}^{N}$, as derived in [18].

However, although the tracking methods can find the optimal MAP state using recent advanced stochastic optimizers, the MAP state does not always correspond to a true (ground-truth) state of a target. The posterior probability in (1.1) is efficiently formulated as Bayesian filtering. Given the state at time t and the observation up to time t, the Bayesian filter updates the posteriori probability $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ with the following formula:

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) \propto p(\mathbf{Y}_t|\mathbf{X}_t) \times \int p(\mathbf{X}_t|\mathbf{X}_{t-1}) p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1}) d\mathbf{X}_{t-1}, \tag{1.7}$$

where $p(\mathbf{Y}_t|\mathbf{X}_t)$, $p(\mathbf{X}_t|\mathbf{X}_{t-1})$, $\mathbf{X}_t$,and $\mathbf{Y}_t$ denote the appearance, motion, state, and observation models, respectively.



(a) Conventional representation using a single posterior

(b) New representation using multiple posteriors

Figure 1.2: **Problem of conventional posterior representation and new representation using multiple posteriors** (a) The estimated posterior has some error. Hence, the global optimum state of the estimated posterior may not correspond to the global optimum state of a true posterior. (b) Due to the estimation error, the posterior cannot be uniquely determined. Instead, a true posterior should be represented as multiple candidates of estimated posteriors. Using the multiple posteriors, the proposed method reduces estimation errors with interval analysis.

- **Appearance model ($\mathbf{A}_t$):** $p(\mathbf{Y}_t|\mathbf{X}_t)$ describes the appearance of a target at time t while measuring the coincidence of the target appearance and observation at the proposed state, as illustrated in Figure 1.1(a).
- **Motion model ($\mathbf{M}_t$):** $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ models the characteristic of the target motion at time t by predicting the next state $\mathbf{X}_t$ based on the previous state $\mathbf{X}_{t-1}$, as illustrated in Figure 1.1(b).
- **State model ($\mathbf{S}_t$):** $\mathbf{X}_t$ designs the target configuration at time t, which is typically called the state, as illustrated in Figure 1.1(c).
- **Observation model ($\mathbf{O}_t$):** $\mathbf{Y}_t$ denotes visual cues in the video at time t, as illustrated in Figure 1.1(d).

As derived in (1.7), the posterior probability is calculated using the distributions associated with the appearance, motion, state, and observation models. The problem is that those distributions cannot be perfectly known. In many practi-

cal cases, our information about the tracking system to be modeled may not allow us to characterize these models and types with precise distributions [21]. Due to the imprecise distributions, an estimated posterior could be false, as illustrated in Fig.1.2(a). Hence, the MAP state does not always correspond to a true (ground-truth) state of a target, although the tracking methods can find the optimal MAP state using recent advanced stochastic or deterministic optimizers. An any single posterior necessarily has some estimation error. In addition, the posterior cannot be uniquely determined especially when distributions associated with the appearance, motion, state, and observation models are contaminated. To overcome this problem, a true posterior should be represented as multiple candidates of estimated posteriors, as illustrated in Fig.1.2(b). Using the multiple candidates of estimated posteriors, two ways, namely, Bayesian Model Averaging (BMA) and Interval Analysis (IA) have been devised for the visual tracking problem.

1.2 Outline of the Dissertation

In this dissertation, the BMA and IA based tracking methods are proposed to robustly track the targets in the real-world tracking environment by solving ambiguities in probabilistic models

In Chapter 2, the basic ideas and challenges of the BMA and IA approaches are presented. First, the property of three tracking methods (the WLMC tracker, the BHMC tracker, the VTS tracker) are addressed, which follows the BMA approach. Then, the property of three tracking methods (the MUG tracker, the SBB tracker, IT) are explained, which follows the IA approach.

In Chapter 3, the detail algorithms and experimental results of the BMA based

tracking methods are presented. In Chapter 3.1, the WLMC tracker is proposed, which is again divided into three tracking algorithms, WLMC, AWLMC, and NFWL. These algorithms efficiently address the tracking of abrupt motions, as well as the accurate tracking of smooth motions. WLMC and AWLMC robustly tracked the target whose position abruptly changes over time. Meanwhile, NFWL successfully tracked the target whose scale and position drastically changes with a smaller number of samples. In Chapter 3.2, the BHMC tracker is proposed to track the highly non-rigid targets. The tracker evolves a local patch-based appearance model by the analysis of LLM. With the model, the algorithm robustly tracked the object whose geometric appearance is drastically changing over time, while efficiently finding best state of the object with the BH sampling. By selecting robust features and using segmentation results, the tracking performance was further enhanced. In Chapter 3.3, the VTS tracker is proposed to track the targets when there are several types of changes in target at the same time. The VTS tracker presents an effective tracking framework with visual tracking decomposition and visual tracker sampler. In the framework, our method efficiently samples multiple good trackers from the tracker space, and tracks the target robustly and successfully by utilizing them in challenging tracking environments.

In Chapter 4, the detail algorithms and experimental results of the IA based tracking methods are presented. In Chapter 4.1, the MUG tracker is proposed, which tracks the target robustly by finding the best state minimizing the gap between the lower and upper bounds of the likelihood. Obtaining the likelihood bounds is the same as considering all possible target models during the tracking process. Therefore, the method finds the good state of the target by reflecting all possible appearance changes of the target. In Chapter 4.2, the SBB tracker is proposed. The

SBB tracker presents a new bounding box representation called the soft bounding box representation, which is capable of reliably tracking highly non-rigid targets. The proposed bounding box represents the target as a range of the bounding box and solves the inherent ambiguity of the conventional bounding box representation for non-rigid targets. Using the soft bounding box representation, the proposed method does not need to deal with the ambiguous regions, which include both the foreground and the background at the same time. Hence, the method greatly improves tracking accuracy without additional computational cost. In Chapter 4.3, IT proposes the M4 estimation, which combines MAP, MMSE, ML, and MUP estimators using the rigorous theoretical basis and exploits the beneficial complementary relationship among them. In the M4 estimation, the method represents the posterior as interval and explicitly measures uncertainty of the posterior estimation. Then, the best state is found, which maximizes a posterior probability and, at the same time, minimizes uncertainty of the posterior estimation.

Finally, we conclude the dissertation in Chapter 5.

Chapter 2

Uncertainties in Probabilistic Tracking Models

Depending on approaches to solve the uncertainty and ingredients to be considered, six different tracking methods are proposed in this dissertation: Wang-Landau Monte Carlo (WLMC) tracker [22, 23], Basin Hopping Monte Carlo (BHMC) tracker [24, 25], Visual Tracker Sampler (VTS) tracker [26, 27, 28], Minimum Uncertainty Gap (MUG) tracker [29], Soft Bounding Box (SBB) tracker [30], and Interval Tracker (IT) [31], as shown in in Figure 2.1.

2.1 Bayesian Model Averaging Approaches

The basic idea is to consider multiple candidates of probabilistic distributions and to average them according to some criterion [21]. By averaging multiple candidates of probabilistic distributions,the statistical error (uncertainty) of them decreases at the rate of the square root of the number of candidates. For example, a true posterior

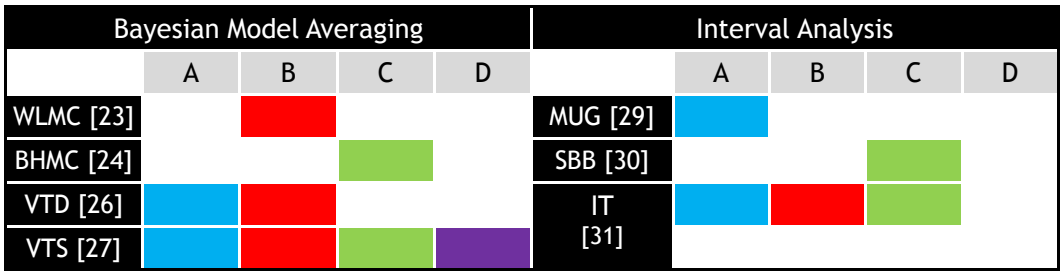


Figure 2.1: **Proposed tracking methods**. A,B,C, and D denotes the appearance model, motion model, state representation type, and observation type, respectively. Non-white color means that the tracking method considers uncertainty of the corresponding distribution.

Table 2.1: **Challenges in the Bayesian Model Averaging approach**.

Challenge	Content
1	Making candidates of the probabilistic distributions
2	Determining weights of candidates

$p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ is represented by sets of posterior candidates $\{p_i(\mathbf{X}_t|\mathbf{Y}_{1:t})\}_{i=1}^{N}$ as follows:

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) = \sum_{i=1}^{N} w_i p_i(\mathbf{X}_t|\mathbf{Y}_{1:t}), \tag{2.1}$$

where w_i is the weight of the i-th estimated posterior. In (2.1), the statistical error of the posterior $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ decreases at the rate of $\sqrt{N}$.

Then, there are two challenges in bayesian model averaging, as described in Table 2.1. The first challenge is to make candidates of probabilistic distributions. The candidates should well describe the target characteristics. The candidates should be as compact as possible. The relations among the candidates should be complementary. The second challenge is how to determine weights of candidates. Better candidates should have larger weights.

Following the BMA approach, three trackers were proposed to solve ambiguities in probabilistic models, while meeting aforementioned two challenges. the WLMC

Table 2.2: **Property of the WLMC tracker.**

	Approach
Solution	Bayesian Model Averaging
Uncertainty	Motion model: $p(\mathbf{X}_t\|\mathbf{X}_{t-1})$
Candidates	Manually make candidates of motion models using Gaussian functions with different variances: $p_1(\mathbf{X}_t\|\mathbf{X}_{t-1})$: small variance for smooth motions $p_2(\mathbf{X}_t\|\mathbf{X}_{t-1})$: large variance for abrupt motions
Weights	Wang-Landau Monte Carlo sampling
Goal	Robustly track both the smooth and abrupt motions

tracker [22, 23] considered uncertainty in motion models. The BHMC tracker [24, 25] considered uncertainty in state models. The VTS tracker [26, 27, 28] considered uncertainty in appearance, motion, state, and observation models.

2.1.1 Uncertainty in Motion Models

Table 2.2 shows property of the WLMC tracker [22, 23]. The WLMC tracker solves uncertainty in motion models using the BMA approach. The tracker manually makes two candidates of motion models using Gaussian functions with different variances. The first candidate has a small variance of the Gaussian function for smooth motions. The second candidate has a large variance of the Gaussian function for abrupt motions. The weight of the candidates are determined by the Wang-Landau Monte Carlo sampling method during the tracking process. By considering uncertainty in motion models, the tracker robustly tracks both the smooth and abrupt motions.

Abrupt motions cause conventional tracking methods to fail because they violate the motion smoothness constraint. To address this problem, we introduce the Wang-Landau sampling method and integrate it into a Markov Chain Monte

Carlo (MCMC)-based tracking framework. By employing the novel density-of-states term estimated by the Wang-Landau sampling method into the acceptance ratio of MCMC, our WLMC-based tracking method alleviates the motion smoothness constraint and robustly tracks the abrupt motions. Meanwhile, the marginal likelihood term of the acceptance ratio preserves the accuracy in tracking smooth motions. The method is then extended to obtain good performance in terms of scalability even on a high-dimensional state space. Hence, it covers drastic changes in not only position but also scale of a target. To achieve this, we modify our method by combining it with the N-fold way algorithm and present the N-Fold Wang-Landau (NFWL)-based tracking method. The N-fold way algorithm helps estimate the density-of-states with a smaller number of samples. Experimental results demonstrate that our approach efficiently samples the states of the target even in a whole state space without loss of time, and tracks the target accurately and robustly whose position and scale are severely changing.

2.1.2 Uncertainty in State Models

Table 2.3 shows property of the BHMC tracker [24, 25]. The BHMC tracker solves uncertainty in state models using the BMA approach. The tracker manually makes candidates of state models over time using the different number of local patches and using different topologies between local patches. The weight of the candidates are determined by the likelihood landscape analysis method during the tracking process. Then, the objective of the tracker is to robustly track non-rigid whose geometric appearance changes over time.

To track targets with drastically changing geometric appearances over time, we develop a local patch-based appearance model and provide an efficient online up-

Table 2.3: **Property of the BHMC tracker.**

	Approach
Solution	Bayesian Model Averaging
Uncertainty	State model: $\mathbf{X}_t$
Candidates	Automatically make candidates of state models using the different number of local patches and using different topologies between local patches: $\mathbf{X}_t^1$, $\mathbf{X}_t^2$,...
Weights	Likelihood landscape analysis
Goal	Robustly track non-rigid targets whose geometric appearance changes over time

dating scheme that adaptively changes the topology between patches. In the online update process, the robustness of each patch is determined by analyzing the likelihood landscape of the patch. Based on this robustness measure, the proposed method selects the best feature for each patch and modifies the patch by moving, deleting, or newly adding it over time. Moreover, a rough object segmentation result is integrated into the proposed appearance model to further enhance it. The proposed framework easily obtains segmentation results because the local patches in the model serve as good seeds for the semi-supervised segmentation task. To solve the complexity problem attributable to the large number of patches, the Basin Hopping (BH) sampling method is introduced into the tracking framework. The BH sampling method significantly reduces computational complexity, with the help of a deterministic local optimizer. Thus, the proposed appearance model could utilize a sufficient number of patches. The experimental results show that the present approach could track objects with drastically changing geometric appearance accurately and robustly.

Table 2.4: **Property of the VTS tracker.**

	Approach
Solution	Bayesian Model Averaging
Ingredients	Appearance model: $p(\mathbf{Y}_t\|\mathbf{X}_t)$, Motion model: $p(\mathbf{X}_t\|\mathbf{X}_{t-1})$ State model: $\mathbf{X}_t$, Observation model: $\mathbf{Y}_t$
Candidates	Automatically make candidates of Appearance models using SPCA: $p(\mathbf{Y}_t\|\mathbf{X}_t)$,... Motion models using KHM: $p(\mathbf{X}_t\|\mathbf{X}_{t-1})$,... State models using VPE: $\mathbf{X}_t$,... Observation models using GFB: $\mathbf{Y}_t$,...
Weights	Interacting MCMC
Goal	Tracking targets when several changes occur at the same time (deformation, abrupt motions, occlusion, illumination, noise, and motion blur)

2.1.3 Uncertainty in Appearance, Motion, State, and Observation Models

Table 2.4 shows property of the VTS tracker [26, 27, 28]. The VTS tracker solves uncertainty in appearance, motion, state, and observation models using the BMA approach. The tracker manually makes candidates of appearance models using the sparse principal component analysis (SPCA), motion models using the k-harmonic means (KHM) method, state models using the vertical projection of edge (VPE), and observation models using the Gaussian filter bank (GFB). The weight of the candidates are determined by the interacting MCMC method during the tracking process. By considering uncertainty in appearance, motion, state, and observation models at the same time, the tracker can robustly track targets when there are several changes in the tracking environment, such like deformation, abrupt motions, occlusion, illumination, noise, and motion blur.

The VTS tracker can work robustly in a challenging scenario, such that several kinds of appearance and motion changes of an object can occur simultaneously. The proposed tracking algorithm accurately tracks a target by searching for appropriate trackers in each frame. Since the real-world tracking environment varies severely over time, the trackers should be adapted or newly constructed depending on the current situation, so that each specific tracker takes charge of a certain change in the object. To do this, our method obtains several samples of not only the states of the target but also the trackers themselves during the sampling process. The trackers are efficiently sampled using the Markov Chain Monte Carlo (MCMC) method from the predefined tracker space by proposing new appearance, motion, state, and observation models, which are the important ingredients of visual trackers. All trackers are then integrated into one compound tracker through an Interacting MCMC (IMCMC) method, in which the trackers interactively communicate with one another while running in parallel. By exchanging information with others, each tracker further improves its performance, thus increasing overall tracking performance. Experimental results show that our method tracks the object accurately and reliably in realistic videos, where appearance and motion drastically change over time, and outperforms even state-of-the-art tracking methods.

2.2 Interval Analysis Approaches

When different reasonable posteriors yield substantially different answers, it is not reasonable to state that there is a single answer. The idea is then to deal with all possible candidates of posteriors. This leads to alternative models of representation of uncertainty based on a set of probability distributions [21]. For example, a true

Table 2.5: **Challenges in the Interval Analysis approach.**

Challenge	Content
1	Obtaining lower bound of the probabilistic distribution
2	Obtaining upper bound of the probabilistic distribution

posterior $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ is represented by interval as follows:

$$\underline{p(\mathbf{X}_t|\mathbf{Y}_{1:t})} \leq p(\mathbf{X}_t|\mathbf{Y}_{1:t}) \leq \overline{p(\mathbf{X}_t|\mathbf{Y}_{1:t})}, \tag{2.2}$$

where $\underline{p(\mathbf{X}_t|\mathbf{Y}_{1:t})}$ and $\overline{p(\mathbf{X}_t|\mathbf{Y}_{1:t})}$ are the lower and upper bounds of the estimated posterior, respectively. IA is different with BMA in the following aspect. IA utilizes an infinite number of candidates in interval, while BMA only utilizes a finite number of posterior candidates.

Then, there are two challenges in interval analysis, as described in Table 2.5. The first challenge is how to get optimal lower bound of the probabilistic distribution. The second challenge is how to get optimal upper bound of the probabilistic distribution.

Following the IA approach, three trackers were proposed to solve ambiguities in probabilistic models, while meeting aforementioned two challenges. the MUG tracker [29] considered uncertainty in appearance models. The SBB tracker [30] considered uncertainty in state models. IT [31] considered uncertainty in appearance and state models.

2.2.1 Uncertainty in Appearance Models

Table 2.6 shows property of the MUG tracker [29]. The MUG tracker solves uncertainty in appearance models using the IA approach. For this, the tracker optimally obtains the lower and upper bounds of the likelihood, which is the distribution as-

Table 2.6: **Property of the MUG tracker.**

	Approach
Solution	Interval Analysis
Uncertainty	Appearance model: $p(\mathbf{Y}_t\|\mathbf{X}_t)$
Lower Bound	Jensen's inequality: $\underline{p(\mathbf{Y}_t\|\mathbf{X}_t)} \leq p(\mathbf{Y}_t\|\mathbf{X}_t)$
Upper Bound	Gibbs' inequality: $p(\mathbf{Y}_t\|\mathbf{X}_t) \leq \overline{p(\mathbf{Y}_t\|\mathbf{X}_t)}$
Goal	Finding the best state that maximizes likelihood and, at the same time, minimizes gap between lower and upper bounds of the likelihood

sociated with appearance models. The tracker optimally obtains the lower bound of likelihood using Jensen's inequality. The tracker optimally obtains the upper bound of likelihood using Gibbs' inequality. Then, the goal of tracking problem is redefined to find the best state that maximizes likelihood and, at the same time, minimizes interval between lower and upper bounds of the likelihood.

The uncertainty of the likelihood is estimated by obtaining the gap between the lower and upper bounds of the likelihood. By minimizing the gap between the bounds, our method finds the confident state of the target. In the paper, the state that gives the Minimum Uncertainty Gap (MUG) between likelihood bounds is shown to be more reliable than the state which gives the maximum likelihood score, especially when there are severe illumination changes, occlusions, and pose variations. A rigorous derivation of the lower and upper bounds of the likelihood for the visual tracking problem is provided to address this issue. Additionally, an efficient inference algorithm using Interacting Markov Chain Monte Carlo is presented to find the best state that minimizes the gap between the bounds. Experimental results demonstrate that our method successfully tracks the target in realistic videos and outperforms conventional tracking methods.

Table 2.7: **Property of the SBB tracker.**

	Approach
Solution	Interval Analysis
Ingredients	State model: $\mathbf{X}_t$
Lower Bound	DempsterShafer theory: $\underline{\mathbf{X}_t} \leq \mathbf{X}_t$
Upper Bound	DempsterShafer theory: $\mathbf{X}_t \leq \overline{\mathbf{X}_t}$
Goal	solving inherent ambiguity in a single bounding box representation for highly non-rigid targets

2.2.2 Uncertainty in State Models

Table 2.7 shows property of the SBB tracker [30]. The MUG tracker solves uncertainty in state models using the IA approach. The tracker optimally obtains the lower and upper bounds of the state using DempsterShafer theory. The MUG tracker represents the highly non-rigid target as a soft bounding box (SBB), which describes the target with inner and outer bounding boxes. Then, It solves inherent ambiguity in a single bounding box representation for highly non-rigid targets.

In the soft bounding box representation, the target is described as a range of the bounding box, which is bounded by an inner bounding box and an outer bounding box. In the paper, the inner and outer bounding boxes are theoretically constructed based on the theory of evidence. With these bounding boxes, the proposed method can solve the inherent ambiguity in a single bounding box representation for highly non-rigid targets. In addition, the method does not deal with the ambiguous region directly, which includes the foreground and the background at the same time. Hence, it robustly tracks highly non-rigid targets. In the soft bounding box representation, the best state of the target is efficiently found using a new Constrained Markov Chain Monte Carlo sampling method, which uses the constraint in which the outer

Table 2.8: **Property of IT.**

	Approach
Solution	Interval Analysis
Ingredients	Appearance model: $p(\mathbf{Y}_t\|\mathbf{X}_t)$ State model: $\mathbf{X}_t$
Lower Bound	Interval Linearization: $\underline{p(\mathbf{Y}_t\|\mathbf{X}_t)} \leq p(\mathbf{Y}_t\|\mathbf{X}_t)$, $\underline{\mathbf{X}_t} \leq \mathbf{X}_t$
Upper Bound	Interval Linearization: $p(\mathbf{Y}_t\|\mathbf{X}_t) \leq \overline{p(\mathbf{Y}_t\|\mathbf{X}_t)}$, $\mathbf{X}_t \leq \overline{\mathbf{X}_t}$
Goal	measuring uncertainty (confidence) of the posterior estimation and incorporate the uncertainty into the Maximum a Posterior estimation

bounding box must include the inner bounding box. Experimental results show that our method can track non-rigid targets accurately and robustly, and outperform even state-of-the-art methods.

2.2.3 Uncertainty in Appearance and State Models

Table 2.8 shows property of IT [30]. IT solves uncertainty in appearance and state models using the IA approach. Using the Interval Linearization technique, the tracker obtains the lower and upper bounds of the likelihood, which is the distribution associated with appearance models. The tracker also obtains the lower and upper bounds of the state using the Interval Linearization technique. IT can measure uncertainty (confidence) of the posterior estimation. Then, it incorporate the uncertainty into the Maximum a Posterior estimation and proposes the M4 (**M**MSE-**M**AP-**M**L-**M**UP) estimation.

We tackle that an any single posterior necessarily includes estimation error and, thus, the posterior should be represented as interval. With interval of the posterior, our method can measure uncertainty (confidence) of the posterior estimation

and incorporate the uncertainty into the Maximum a Posterior (MAP) estimation. Then, its objective is to find the best state, which maximizes a posterior probability and, at the same time, minimizes uncertainty of the posterior (MUP) estimation. In this paper, the MAP estimation is improved to be robust to outliers by performing the MAP estimation over the state, which is obtained by the minimum mean-square error (MMSE) estimation. The MUP state is achieved by reducing the interval width of the posterior and simultaneously maximizing the likelihood with the Maximum Likelihood (ML) estimation. Our method combines all theses estimators with the rigorous theoretical basis and exploits the beneficial complementary relationship among them. Hence, the method is called the M4 (**M**MSE-**M**AP-**M**L-**M**UP) estimation. The experimental results demonstrated that the M4 estimation can be easily integrated into conventional tracking methods and greatly enhance the tracking accuracy of the conventional methods. In several challenging datasets, our method outperforms state-the-art tracking methods.

2.3 Related Works

Although there may be a long history of research on visual tracking, we only introduce in detail the works that are directly related to ours.

2.3.1 Sampling-based Tracking Methods

the particle filter shows its robustness for visual tracking by handling non-Gaussianity and multi-modality of a target density, and by reflecting the uncertainty of the target motion [32][33][16]. The particle filter is extended to the joint particle filter for the multi-object tracking problem [34][35]. However, the joint particle filter suffers

from exponential complexity as the state space increases. MCMC reduces the computational cost in high-dimensional state space, which has been intensively used by many tracking methods in recent years [18][36]. On the other hand, Data-Driven MCMC [37] provides quick convergence results with efficient proposals.

When the dimension of a solution space increases, however, these methods still suffer from the problem of being trapped in deep local optima and handling a vast number of samples. The BHMC tracker , based on the BH sampling method, solves these problems by combining a sampling method with a deterministic method and simplifying the landscape of a solution space. By doing this, the BHMC tracker can find a solution using smaller number of samples even in a very high-dimensional solution space. As the number of chains increases, the conventional sampling methods need more samples as many times as the number of chains. The VTS tracker solves this problem by utilizing IMCMC, which requires a relatively small number of samples by exchanging information between chains. Moreover, the conventional sampling methods only consider the uncertainty of the target state given a fixed tracker. The VTS tracker is an intuitively attractive solution to the problem of accounting for the uncertainty of the tracker by sampling the trackers themselves. In addition, the conventional sampling methods use the samples to obtain the state that minimizes the uncertainty of the target distribution, whereas the samples in the MUG tracker are used to obtain the maximum of the target distribution. In the MUG tracker, the proposed Constrained Markov Chain Monte Carlo (CMCMC) sampling method finds the best state while satisfying some constraints. It is a more difficult problem than the conventional sampling problem.

2.3.2 Tracking Methods to Deal with Abrupt Motions

To deal with the abrupt motion, most conventional approaches improve proposal models or inference methods. (i.e., Sequential belief propagation MC, quasi-random sampling, adaptive MCMC sampling, particle swarm optimization, and stochastic approximation). In this section, we focus on the weakness of the conventional approaches and show the advantages of our method.

Hua et al. [38] proposed sequential belief propagation MC to overcome abrupt motion such as the unexpected dynamics changes of the target. The method overcomes abrupt motion by combining a set of particle filters, in which searching and matching are done collaboratively in different scales. Philomin et al. [15] addresses the problem of tracking pedestrians from a moving car using quasi-random sampling To cope with abrupt changes in motion and shape, it combines the particle filter with quasi-random sampling. This method has two drawbacks. First, to track smooth motions, it chooses highly weighted particles and samples new states that are densely located around the states of those particles. However, if there are a few strong local maxima, most samples get trapped in those local maxima. Second, to capture the abrupt changes, the method uses uniform sampling over the entire state space. However, if the entire state space is very large, uniform sampling can be wasteful. Roberts et al. [39] presented the adaptive MCMC algorithm, which automatically changes the proposal variance of MCMC as the Markov Chain goes on. The proposal variance is tuned to produce an acceptance ratio as close as possible to the optimal value of 0.44 [40]. This adaptive scheme is highly helpful in tracking the abrupt motion. If the motion is abrupt, the acceptance ratio usually decreases because the proposal with a small variance cannot cover this motion. To

preserve the optimality of the acceptance ratio, the algorithm adaptively increases the proposal variance. However, this algorithm also has drawbacks. The algorithm does not provide a systematic way of escaping the local maxima and has no efficient sampling strategy to deal with large state space. Li et al. [41] employed Particle Swarm Optimization (PSO) to deal with the abrupt motion in contour tracking problems. For this, the method proposes a two-layer tracking framework in which PSO is successfully combined with a level set evolution. In the first layer, PSO is adopted to capture the global motion of the target and to help construct the coarse contour. In the second layer, level set evolution based on the coarse contour is carried out to track the local deformation. However, in the PSO method, there is a possibility that most samples will get trapped in a few strong local maxima. Hence, the PSO method fails to track highly abrupt motions. Zhou et al. [17] recently introduced Stochastic Approximation Monte Carlo (SAMC) based tracking scheme, which shows computationally efficient and effective performance in dealing with abrupt motion difficulties. However, the SAMC-based tracking method cannot cover abrupt changes in the scale and fails to track the target. Note that the abrupt scale changes of objects frequently occur in unconstrained videos under general settings.

Compared with the aforementioned methods, the WLMC tracker has three advantages as follows. The first advantage is that the tracker offers a systematic way to escape the local maxima and to reach the global maximum. If the tracker gets trapped in a certain subregion for a long time, the DOS term prevents the tracker from obtaining more samples at that subregion and forces it to get samples from other subregions. The second is to provide an efficient sampling schedule that can cope with the increased search space given a limited number of samples. With the

sampling schedule, the tracker obtains more samples at the subregions where local maxima might exist. Meanwhile, the exploration of other subregions is at least guaranteed to some degree. The last is that the tracker automatically increases or decreases the proposal variance. Depending on the DOS score, our method adaptively explores near subregions with small variances and distant subregions with large variances.

2.3.3 Tracking Methods for Non-Rigid Targets

Schindler et al. [42] represented an object as constellations of parts to track a bee accurately using the Rao-Blackwellized Particle Filter. This method focuses on fixing the topology of the constellation, whereas the proposed method evolves the topology via online updates. Ramanan et al. [43] proposed a tracking method operated by detecting the models of the target, the appearances of which should first be built. This method shows good results in tracking an articulated person. Using shape models of humans, Zhao et al. [37] successfully tracked humans in crowded environments where occlusion occurs persistently. Cehovin et al. [44] combined local appearance with global appearance of the target using a novel coupled-layer visual model. Godec et al. [45] employed a rough segmentation to describe global appearance of the target. The global appearance of these two methods help reduce noisy samples during the appearance updating process and thus help effectively prevent the trackers from drifting. By incorporating the global appearance, these two methods produced very accurate tracking results especially for the highly non-rigid objects. Nejhum et al. [46] described the target using multiple rectangular blocks, whose positions within the tracking window are adaptively determined.

The tracking methods of Schindler et al. and Ramanan et al., however, basically

assume that specific models of the targets are given. By contrast, the BHMC tracker utilizes *no* prior knowledge of the specific model for the target and *no* off-line training phase. The BHMC tracker also uses global appearance obtained from a rough segmentation. However, in contrast with the tracking methods of Cehovin et al. and Godec et al., the BHMC tracker includes the online feature selecting step. This enables that a different part of local appearance is described by a different feature. With the step, the BHMC tracker shows better tracking performance under the challenging tracking environments including illumination changes as well as severe deformation of the targets. Using multiple rectangular blocks and multiple local patches, the tracking method of Nejhum et al. efficiently tracked non-rigid objects undergoing large variation in appearance and shape. The SBB tracker is similar to that of Nejhum et al. because both methods use multiple bounding boxes. However, the main advantages of the SBB tracker are how to determine the optimal sizes of these bounding boxes and how to calculate likelihoods of them efficiently using the Dempster-Shafer theory.

2.3.4 Tracking Methods to Solve Ambiguities in the Real-World Tracking Problem

Ross et al. [1] proposed an adaptive tracking method that shows robustness to large changes in pose, scale, and illumination via incremental principal component analysis. Babenko et al. [6] successfully tracked an object in real time with the online multiple instance learning algorithm, where lighting conditions change and object occlusion occurs. Compared with these two works, the VTS tracker addresses more challenging scenarios for the tracking problem utilizing unstructured videos captured from broadcast networks.

Ross et al. [1] dealt with the ambiguities of target appearances as the method incrementally learns a low-dimensional subspace representation. Babenko et al. [6] handled the ambiguities by employing multiple instances of the appearance. Bao et al. [47], Mei et al. [48], and Zhang et al. [49] solved the ambiguities by finding a sparse approximation in a template subspace via L1 minimization. The tracking by detection approaches [8, 50, 51, 52, 14] overcame the ambiguities by using detection power and advanced machine learning algorithms. Compared with these methods, IT numerically measures the ambiguities and explicitly applies them in the visual tracking problem. Hence, it can accurately track the targets under the real-world tracking environment.

2.3.5 Tracking Methods with Online Appearance Learning and Feature Fusion

By approximately estimating the pixel-wise color density in a sequential manner, Han et al. [10] successfully tracked an object where lighting conditions, pose, scale, and view-point change over time. Ross et al. [1] presented an adaptive tracking method utilizing incremental principal component analysis. This adaptive tracking method shows robustness to large changes in pose, scale, and illumination. These two methods, however, do not consider extreme geometric changes of an object. The BHMC tracker explicitly tackles these changes using a local patch-based online appearance model.

Collins et al. [7] used multiple features and selected robust ones through an online feature-ranking mechanism to deal with changing appearances. Han et al. [53] presented a probabilistic sensor fusion technique. The method shows robustness to severe occlusion, clutter, and sensor failures. The method in [54] integrates

multiple cues, edge, and color in a probabilistic framework while the method in [55] fuses multiple observation models with parallel and cascaded evaluation. However, these methods do not consider extreme motion changes of an object. Additionally, only information related to target appearance are considered to improve tracking performance. In comparison, the VTS tracker exploits useful information both on the target motion and target representation.

2.3.6 Tracking Methods using Multiple Patches and Trackers

Adam et al. [56] presented a tracking method using multiple image fragments, where every fragment votes on the possible positions and scales of the object. By employing multiple fragments, the method is able to handle partial occlusions or pose changes efficiently. Nejhum et al. [46] modeled the constantly changing foreground shape using multiple rectangular blocks, whose positions within the tracking window are adaptively determined. Using multiple rectangular blocks, the algorithm efficiently tracks articulated objects undergoing large variations in appearance and shape. Yang et al. [4] proposed a novel attentional tracking method, which utilizes spatially attentional patches. The attentional patches include salient and discriminative regions of the targets. This method showed the robustness on a large variety of real-world video. Compared with these methods, the local patch-based appearance model of the BHMC tracker is more flexible, because the patch may be removed, newly added, and moved by affine transformation and transition. Thus, the BHMC tracker is able to track more severe non-rigid objects.

Badrinarayanan et al. [57] employed a novel randomized template tracker and a constant color model-based particle filter. Santner et al. [52] combined three different trackers in a cascade using the tracking-by-detection approach. However,

the number and types of trackers used in these methods are predefined by a user. On the other hand, the VTS tracker can replace current trackers with newly sampled ones during the tracking process and change the total number of trackers by adding good ones and removing the bad or redundant ones. To the best of our knowledge, the VTS tracker is the first attempt to define tracker space and sample trackers directly in this space.

2.3.7 Tracking Methods using Segmentation

Chockalingam et al. [58] represented the target by a Gaussian mixture model with multiple fragments and extracted accurate boundaries of the target using level sets. Then, the boundaries are used to learn the dynamic shape of the target over time. Lu and Hager [59] treated the tracking problem as an online binary classification one using dynamic foreground/background appearance models. Using a temporal adaptive importance re-sampling procedure, they maintained temporally changing appearance model for both foreground and background. Theses two methods demonstrated the effectiveness of their approach on several challenging sequences. However, the methods did not consider geometric structure of the targets such as relations between patches or fragments. Compared with aforementioned methods, the BHMC tracker covers the temporally changing geometric structure of the targets. Hence, the BHMC tracker robustly tracks the targets, of which geometric appearance severely change over time.

2.3.8 Tracking Methods using Likelihood Bound and DS Theory

Mei et al. [60] proposed an efficient L1 tracker with the Bounded Particle Resampling (BPR) technique which considers the upper bound of the likelihood. How-

ever, the method used the BPR technique to speed up the L1 tracker without sacrificing accuracy. The MUG tracker utilizes the likelihood bounds to measure uncertainty of the likelihood and enhances the accuracy of visual tracking.

Faux et al. [61] presented a face tracking system using a pixel fusion process from three color sources within the framework of the Dempster-Shafer theory. Rafael et al. [62] resolved the multi-target problem by combining evidences from multiple sensors using the Dempster-Shafer theory. However, these methods employed the Dempster-Shafer theory only for the observation fusion. The SBB tracker is the first to use the theory for the bounding box representation.

Chapter 3

Bayesian Model Averaging (BMA) based Approaches

In this chapter, three tracking methods using the BMA approach are proposed to solve ambiguity in probabilistic models. In section 3.1, the WLMC tracker solves ambiguity in motion models to track smooth and abrupt motions at the same time. In section 3.2, the BHMC tracker solves ambiguity in state models to track highly non-rigid targets. In section 3.3, the VTS tracker solves ambiguity in appearance, motion, state, observation models to robustly track the target where several types of changes in targets occur.

3.1 The Wang-Landau Monte Carlo (WLMC) Tracker

In a complex outdoor scenario, tracking a target becomes very challenging because the scenario typically contains abrupt changes in the appearance and motion of the target. Recently, online learning techniques have started to tackle the problem of

(a) Smooth motion (b) Abrupt change in position (c) Abrupt change in scale

Figure 3.1: **Example of our tracking results** in *Snowboard* sequence. Our tracking method successfully tracks the target even though there are severely abrupt changes in the position and scale of the target.

abrupt changes in the appearance [63, 6, 7, 8, 10, 11, 24, 1, 64], wherein the appearance model is adapted with the online update to describe the changing appearance of the target. However, most conventional tracking methods rarely consider abrupt motions and easily fail to track the target when abrupt changes in the position and scale of the target occur [11, 1, 56, 19, 65, 12]. The objective of this work is to track the target accurately even though its position and scale severely change over time.

Although more generally applicable, this section focuses on the robust tracking of a target whose motion is mostly smooth, but rapidly changes over one or more small temporal intervals. This type of motion typically occurs in three challenging situations:

(1) A video that has a partially low frame rate.

(2) A target rapidly moves over one or more small temporal intervals.

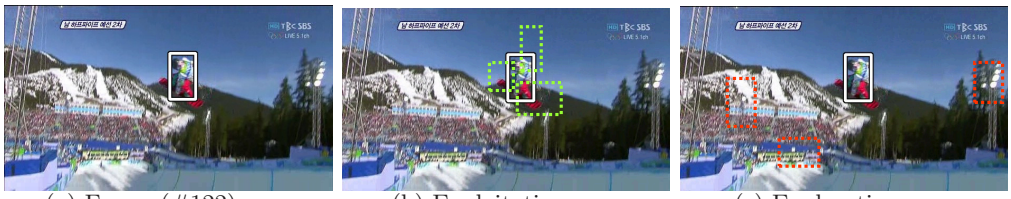
(3) A video that consists of edited clips acquired from several cameras.

Figure 3.1 illustrates the tracking results under the first situation. These situations are very challenging because they contain both smooth and abrupt motions, which cannot be accurately tracked at the same time by conventional tracking methods. If a tracking method focuses on improving performance in the smooth motions, it may miss the abrupt motions of the target. On the other hand, if the method tries

to track the abrupt motions robustly, the accuracy of tracking the smooth motions may decrease.

The philosophy of our method is that the smooth and abrupt motions can be efficiently tracked at the same time by trading off two factors in the acceptance ratio of Markov Chain Monte Carlo (MCMC), namely, the marginal likelihood (ML) term and the density-of-states (DOS) term. The ML term gives information about where the target object might be. It calculates the marginal likelihood estimate over the states within each disjoint-subregion of the state space. If the term has a high value at a certain subregion, then the subregion includes the states of the target with a high probability. In the sampling process, the ML term helps further simulate the seemingly good moves at the region near the current local maximum, which have been already explored enough. By this sampling strategy, our method can increase the accuracy of the smooth motions. We call this process as *exploitation*.

On the other hand, the DOS term informs our method which subregions are sufficiently searched. If the term has a high value at a certain subregion, then the method obtains sufficient samples from the subregion. Utilizing this term, the method simulates the moves at the region far from the current local maximum, which have not been greatly explored. Thus, the method has a chance to capture the abrupt motions of the target. This process is called as *exploration*. As it is intractable to accurately calculate the DOS in all subregions, the Wang-Landau sampling method approximately estimate them through a Monte Carlo simulation [66]. By making full use of the exploitation ability of the ML term with the exploration ability of the DOS term, our method tracks both smooth and abrupt motions accurately and robustly. Figure 3.2 shows examples of the two different types of moves: exploitation and exploration. Note that *exploitation* and *exploration* are well known concept in



(a) Frame (#122) (b) Exploitation (c) Exploration

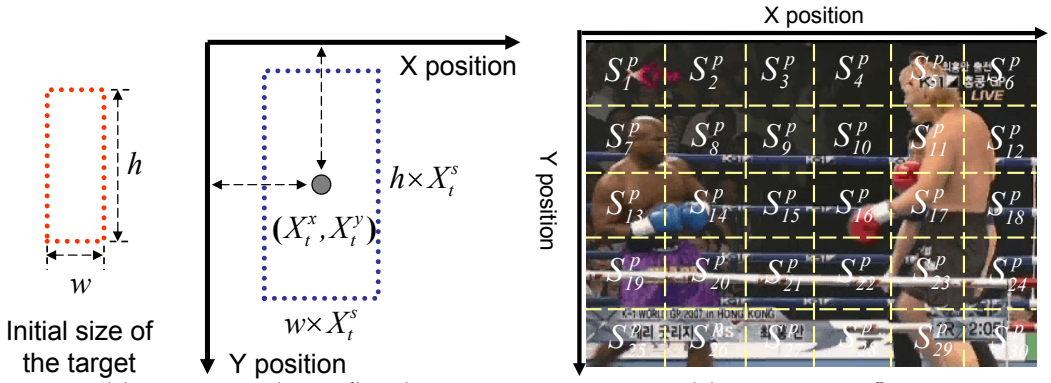
Figure 3.2: **Two different types of moves**. Exploitation moves deal with the smooth motions while exploration moves cover the abrupt motions of the target.

the statistics and applied mathematics literature [67].

In this section, we propose an effective Wang-Landau Monte Carlo-based object tracking framework to deal with the abrupt motions in unconstrained videos obtained from broadcasting media, such as music concerts, sports events, and movies.

3.1.1 WLMC-based Tracking algorithm

In the MCMC-based tracking method, the choice of the proposal variance in (1.3) is very critical to the success of tracking because the variance determines the range of movement at one proposal step. The variance has to be tuned to a larger value for rapid motions and conversely, a smaller value for slow ones. In that sense, the proposal in (1.3) cannot efficiently propose a new state especially in an abrupt motion case. Moreover, with the acceptance ratio in (1.4), the Markov Chain frequently gets trapped in the local maxima when the search space is increased due to the abrupt motions. Therefore, the conventional MCMC-based tracking methods are generally weak to abrupt motions, although they achieve great success in less challenging tracking problems [18, 36]. Our WLMC-based tracking method presented in the next subsection solves this problem.



(a) Example of (X_t^x, X_t^y, X_t^s) (b) Example of S_i^p

Figure 3.3: **Example of a state and subregion**. (a) (X_t^x, X_t^y) represents the center position of the target, and X_t^s indicates the scale where w and h are width and height of initial size of the target, respectively. (b) $\mathbf{S}^p$ is divided into 30 equal-size subregions (d=30).

3.1.1.1 Preliminary

The state space $\mathbf{S}$ is defined by a set of all possible states: $\mathbf{S} = \{(X_t^x, X_t^y, X_t^s) | (X_t^x, X_t^y, X_t^s) \in \mathbf{D}\}$, where $\mathbf{D}$ represents the domain of the states and Figure 3.3(a) illustrates an example of the states. The state space $\mathbf{S}$ consists of the Cartesian product of the state space of position and scale, $\mathbf{S} = \mathbf{S}^p \times \mathbf{S}^s$, where $\mathbf{S}^p = \{(X_t^x, X_t^y) | (X_t^x, X_t^y) \in \mathbf{D}_p\}$, $\mathbf{S}^s = \{X_t^s | X_t^s \in \mathbf{D}_s\}$, and $\mathbf{D}_p$ and $\mathbf{D}_s$ represent the domain of position and scale, respectively. As an abrupt motion occurs by the change of the position in many cases, in this section, we assume that the scale of the target is smooth over time, and only consider abrupt changes in the state space of position, $\mathbf{S}^p$. Then, to cope with abrupt motion, $\mathbf{S}^p$ is divided into d disjoint subregions: $\mathbf{S}^p = \bigcup S_i^p \; and \; \bigcap S_i^p = \phi \; for \; i = 1, \ldots, d$. A simple dividing strategy is to partition $\mathbf{S}^p$ into equal-size grids, as shown in Figure 3.3(b). This strategy empirically provides sufficient results for our tracking problem.

3.1.1.2 Algorithm

Our WLMC-based tracking method consists of three steps: proposal step, acceptance step, and estimation step. Compared with the conventional MCMC-based tracking method, our method inserts the novel DOS (Density-of-States) term into the acceptance ratio of MCMC, which is calculated by the estimation step.

• **Proposal Step:** As mentioned in the preliminary subsection, let us consider abrupt changes in position only and assume that the scale of the target is smoothly changing over time. With this assumption, we design two different types of proposal densities for the position and the scale. To cope with abrupt changes in position, we make the proposal density using Gaussian perturbation with a large variance:

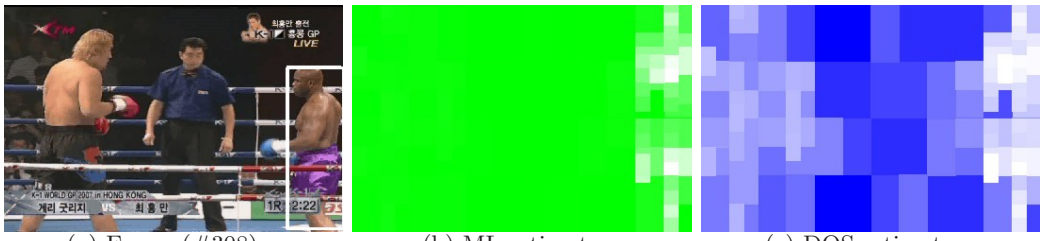
$$Q(X_t^{x'}; X_t^x) = G(X_t^x, \sigma_x^2),\ Q(X_t^{y'}; X_t^y) = G(X_t^y, \sigma_y^2), \tag{3.1}$$

where $X_t^{x'}$ and $X_t^{y'}$ represent the new x and y positions, respectively; and σ_x^2 and σ_y^2 denote the proposal variance of the x and y coordinates, respectively. The efficiency of the proposal step is very important for the success of tracking. Note that the proposal density in (3.1) can be wasteful due to large variance if it proposes many unnecessary states where the target might not exists. Our method solves this inefficiency in the acceptance step through the use of the DOS term.

For designing smooth changes in scale, we adopt the second-order autoregressive model [68], which proposes a new state $X_t^{s'}$ based on the previous states $X_{t-1}^{s'}, X_{t-2}^{s'}$ with a deterministic mapping function and a stochastic disturbance.

$$Q(X_t^{s'}; X_t^s) = 2X_{t-1}^s - X_{t-2}^s + G(0, \sigma_s^2), \tag{3.2}$$

where σ_s^2 is the proposal variance of the scale coordinate. The autoregressive process fits our smoothness assumption of scale well because it is a time series modeling



(a) Frame (#308) (b) ML estimate (c) DOS estimate

Figure 3.4: **Example of the ML and DOS (Density-of-States) estimates**. (b) The brighter the color, the higher the marginal-likelihood score. In the sample, the score is high around subregions where the target exists. (c) The brighter the color, the higher the DOS score. In the example, our method gets more samples at the subregions where the target might be while exploring all subregions at least to some degree.

strategy, which takes into account the historical data to predict a new state.

- **Estimation Step:** Although in practice, the estimation step is performed after the acceptance step, we introduce this step in advance because it updates the ML and the DOS terms, which are utilized in the acceptance ratio.

The ML term returns the marginal-likelihood score of a subregion, where the score is high if the subregion includes highly possible states of the target. We obtain the marginal-likelihood score of a subregion by integrating the likelihood values at all states of the subregion:

$$p(\mathbf{Y}_t|S_i^p) = \frac{1}{|S_i^p|}\int_{\mathbf{X}_t \in S_i^p} p(\mathbf{Y}_t|\mathbf{X}_t)d\mathbf{X}_t, \tag{3.3}$$

where $p(\mathbf{Y}_t|S_i^p)$ denotes the marginal-likelihood score of the i-th subregion S_i^p; $p(\mathbf{Y}_t|\mathbf{X}_t)$ indicates the likelihood at the state $\mathbf{X}_t$; and $|S_i^p|$ is the total number of states in S_i^p. In (3.3), $p(\mathbf{Y}_t|\mathbf{X}_t)$ is determined by

$$p(\mathbf{Y}_t|\mathbf{X}_t) = exp^{-\lambda DD(\mathbf{Y}_t, M_t)}, \tag{3.4}$$

where $\mathbf{Y}_t$ represents the observations obtained at the state $\mathbf{X}_t$; M_t denotes the model that describes the target at time t; and λ is the weighting parameter. In (3.4), the

DD function returns the diffusion distance between $\mathbf{Y}_t$ and M_t. We utilize the diffusion distance as a dissimilarity measure because it is robust to deformation and to quantization effects of the observation [69].

However, the integration in (3.3) is not feasible if a subregion includes a large number of states. Instead of considering all states within a subregion, we sample a few states from it and approximate the marginal-likelihood score using those samples:

$$p(\mathbf{Y}_t|S_i^p) \approx \frac{1}{m}\sum_{n=1}^{m} p(\mathbf{Y}_t|\mathbf{X}_t^n), \quad \mathbf{X}_t^n \in S_i^p, \tag{3.5}$$

where $\mathbf{X}_t^n$ is the n-th sampled state of the subregion S_i^p, and m is the total number of the sampled states. In (3.5), $\mathbf{X}_t^n$ is sampled using the proposal density in (3.1), and the former can be both accepted or rejected by the acceptance step. We update the marginal-likelihood as the Monte Carlo simulation goes on because the states are sequentially obtained. During the *(m+1)*-th state $\mathbf{X}_t^{m+1}$, is newly sampled from the subregion S_i^p, the marginal-likelihood score of S_i^p is updated as follows:

$$p(\mathbf{Y}_t|S_i^p) \leftarrow \frac{mp(\mathbf{Y}_t|S_i^p) + p(\mathbf{Y}_t|\mathbf{X}_t^{m+1})}{m+1}, \quad \mathbf{X}_t^{m+1} \in S_i^p. \tag{3.6}$$

The ML of each subregion is initially set to $\frac{1}{|S^p|}$, where $|S^p|$ denotes the number of subregions. If our method samples a state which belongs to the subregion S_i^p, it updates the ML of the subregion S_i^p using (3.6). Then, with the updated ML, it goes on sampling a new state and updating the ML of the corresponding subregion iteratively.

The DOS term returns the DOS score of a subregion, where the score is high if our method sufficiently gets the states from the subregion. As it is intractable to accurately calculate the DOS in all subregions, we approximately estimate it by using the Wang-Landau algorithm [66] through a Monte Carlo simulation. In the

algorithm, each subregion has its own histogram. If a state proposed by (3.1) is accepted at the acceptance step, and the state belongs to the subregion S_i^p, our method increases the histogram of S_i^p by one and modifies the DOS score of S_i^p by multiplying a modification factor. If not, our method does the same works for the subregion which includes the previous state:

$$h(S_i^p) \leftarrow h(S_i^p) + 1, \tag{3.7}$$

where $h(S_i^p)$ denotes the histogram value of the subregion S_i^p.

$$g(S_i^p) \leftarrow g(S_i^p) * f, \tag{3.8}$$

where $g(S_i^p)$ indicates the DOS score of the subregion S_i^p; and f is the modification factor, which is larger than one. The DOS of each subregion is initially set to 1 . If our method samples a state which belongs to the subregion S_i^p, the method updates the DOS of the subregion S_i^p using (3.8). Then, with the updated DOS, it goes on sampling a new state and updating the DOS of the subregion that includes the state iteratively. As the method progresses, it produces a semi-flat histogram. We consider a histogram as semi-flat if the value of the lowest bin is 80% larger than the average value of all bins in the histogram [66]. The semi-flat histogram indicates that our method explored all subregions at least to some degree. Therefore, the DOS estimates of all subregions are successfully obtained. Then, to obtain more accurate estimates, the modification factor in (3.8) is reduced by $f \leftarrow \sqrt{f}$, and the histogram is reset to 0. The method continues until the histogram becomes semi-flat again and restarts with a finer modification factor. The process is terminated when the factor becomes highly close to 1 or when the number of iterations reaches a predefined value.

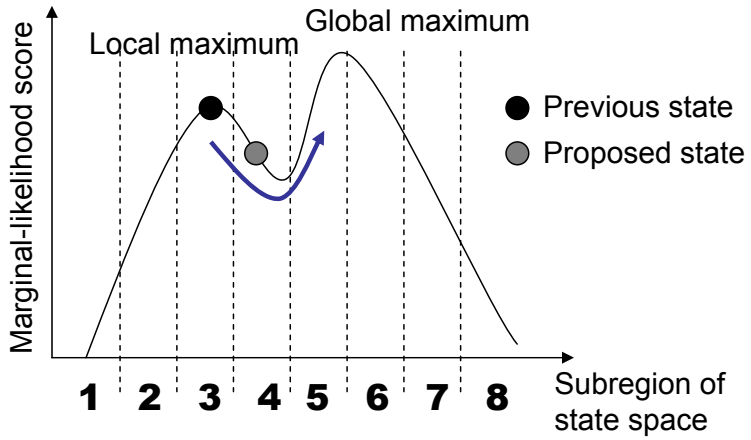


Figure 3.5: **Process of escaping local maxima**. The DOS term is used as the penalty term. If the DOS score in subregion 3 is much higher than that in subregion 4, the proposed state in subregion 4 can be accepted although the subregion has a lower marginal-likelihood score compared to that of the previous state.

The key point of the estimation step is that the chicken-egg-type problem is solved. To accurately estimate the DOS term, the acceptance ratio introduced by the next section should guide the Markov Chain in the direction of quickly producing the semi-flat histogram. Meanwhile, to calculate the acceptance ratio, the DOS term has to be known in advance. This estimation step provides an EM-style strategy to solve the aforementioned problem and to obtain the accurate DOS score. Figure 3.4 shows the ML and DOS score at each subregion.

- **Acceptance Step:** The acceptance ratio is calculated by the ML and DOS terms as follows:

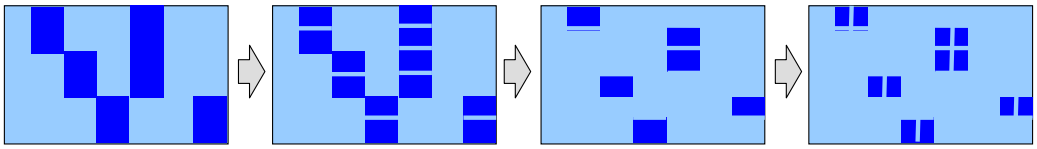
$$a = min\left[1, \frac{p(\mathbf{Y}_t|S_i^{p'})p(S_i^{p'})Q(\mathbf{X}_t;\mathbf{X}_t')}{p(\mathbf{Y}_t|S_i^{p})p(S_i^{p})Q(\mathbf{X}_t';\mathbf{X}_t)}\right] = min\left[1, \frac{p(\mathbf{Y}_t|S_i^{p'})\frac{1}{g(S_i^{p'})}Q(\mathbf{X}_t;\mathbf{X}_t')}{p(\mathbf{Y}_t|S_i^{p})\frac{1}{g(S_i^{p})}Q(\mathbf{X}_t';\mathbf{X}_t)}\right], \tag{3.9}$$

where S_i^p is the subregion that contains the previous state $\mathbf{X}_t$; $S_i^{p'}$ is the subregion

that includes the proposed state $\mathbf{X}'_t$; $p(\mathbf{Y}_t|S_i^p)$ is the marginal-likelihood score of the subregion S_i^p; and $g(S_i^p)$ is the DOS score of the subregion S_i^p. In (3.9), the second equality describes that $g(S_i^p)$ can be considered as prior knowledge about the subregion S_i^p such as $p(S_i^p) = \frac{1}{g(S_i^p)}$. This is because our method tries to accept the states in the subregion S_i^p more frequently if the DOS score of the subregion, $g(S_i^p)$ is very low. Reversely, the method tries to accept the states in the subregion S_i^p less frequently if $g(S_i^p)$ is very high. Thus, the DOS score $g(S_i^p)$ provides some knowledge about the subregion during the sampling process. Because the exact DOS score of the subregion is determined before the sampling process, the DOS score can be considered as the prior knowledge.

Our acceptance ratio in (3.9) has an advantage compared to that in (1.4). The first is that our acceptance ratio provides a systematic way to escape from the local maxima and reach the global maximum. This is crucial to the success of tracking abrupt motions. To capture abrupt motions, tracking methods should propose a new state with a large range of movement, which makes the Markov Chain deal with the increased number of local maxima. Therefore, getting trapped in the local maxima becomes more frequent. In our acceptance step, the Markov Chain is guided by the ML-DOS ratio, $\frac{p(\mathbf{Y}_t|S_i^p)}{g(S_i^p)}$. At a local maximum, this ratio initially has a large value because the marginal-likelihood score is very high around the local maximum. However, while the simulation goes on, the ratio decreases as the DOS score increases. The proposed state is then accepted when the ML-DOS ratio over the previous state sufficiently decreases compared to that over the proposed state. This process helps in easily escaping the local maxima, as described in Figure 3.5.

By mixing the ML term with the DOS term in the acceptance ratio, the WLMC-based tracking method efficiently combines the exploitation ability of the ML term



(a) Chosen subregions (b) Dividing horizontally (c) Chosen subregions (d) Dividing vertically

Figure 3.6: **Example of the dividing strategy**. Each chosen subregion (blue) is divided into two regions horizontally and vertically in turn.

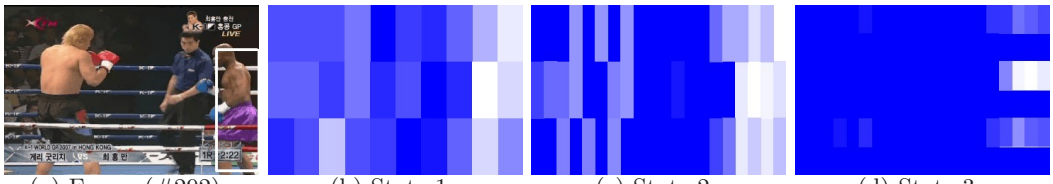
and the exploration ability of the DOS term. Moreover, the tracking problem that includes both abrupt motions and smooth motions is successfully solved. However, this method has a scalability problem. If the number of subregions of $\mathbf{S}^p$ increases, the states exponentially increase to obtain the accurate DOS score in (3.8). Our AWLMC-based tracking method presented in the next section solves this problem by a simple coarse-to-fine approach.

3.1.2 Adaptive WLMC (AWLMC)-based Tracking algorithm

To reduce the state space, we introduce a new annealing step into the WLMC-based tracking algorithm. We focus on explaining the annealing step here because other steps are the same as those in the WLMC-based tracking method.

3.1.2.1 Algorithm

• **Annealing step:** Our method starts the process over the state space $\mathbf{S}^p$, and performs the WLMC-based tracking algorithm explained in Section 3.1.1. If the histogram in (3.7) becomes semi-flat, the method decreases $\mathbf{S}^p$ by selecting half the number of subregions in $\mathbf{S}^p$. To choose the subregions, we utilize the DOS information. $\mathbf{S}^p$ initially consists of d disjoint subregions. We choose the $\frac{d}{2}$ number of subregions according to the score in descending order, as illustrated in Figure 3.6(a) and 3.6(c), because the DOS score indicates the probability of local maxima



(a) Frame (#292) (b) State 1 (c) State 2 (d) State 3

Figure 3.7: **Example of reducing the state space of position**. The AWLMC-based tracking method sequentially reduces $\mathbf{S}^p$ from (b) to (d) using the DOS value of each subregion. The method leaves the small size of the state space of position that contains robust candidates of the target position and eventually tracks the target as shown in (a).

in a subregion. Then, each chosen subregion is divided into two regions so that the total number of subregions reverts to d. If the subregions have been horizontally divided, they are then vertically divided at the next time, as shown in Figure 3.6(b) and 3.6(d). The annealing step is terminated when the number of iterations reaches a predefined value. The aforementioned dividing strategy empirically provides sufficient results for our tracking problem. Figure 3.7 describes the sequentially decreasing state space.

In comparison with the WLMC-based tracking method, the AWLMC-based tracking method obtains samples more efficiently when there are a larger number of subregions. However, if the number of subregions significantly increases, the AWLMC-based tracking method still suffers from the scalability problem. To solve this problem, we present the NFWL-based tracking method in the next section. This method works well although the number of subregions increases significantly, like in the case of the combined state space of scale, $\mathbf{S}^s$ and position, $\mathbf{S}^p$.

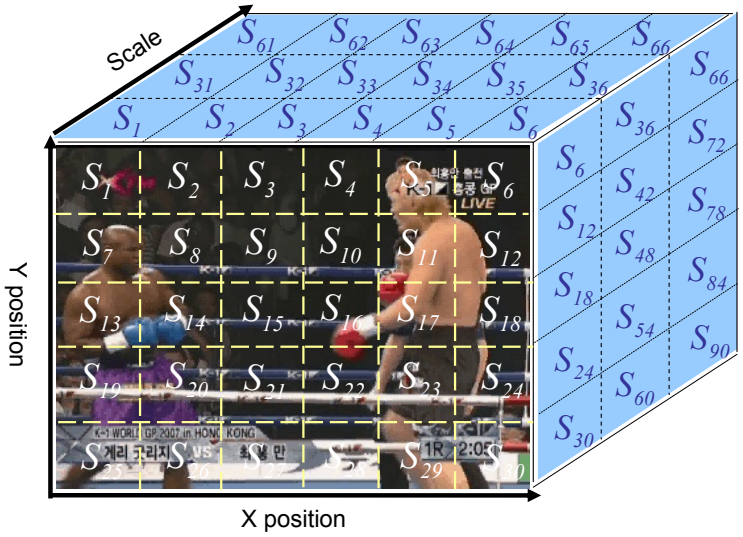


Figure 3.8: **Example of subregions**. **S** is divided into 90 equal-size subregions (d'=90).

3.1.3 N-Fold Wang-Landau (NFWL)-based Tracking Algorithm

3.1.3.1 Preliminary

To cope with abrupt changes in both position and scale, the NFWL-based tracking method considers the whole state space **S** that is constructed by the Cartesian product of the state space of position, $\mathbf{S}^p$, and scale, $\mathbf{S}^s$, and divides **S** into d' disjointed subregions, as shown in Figure 3.8: $\mathbf{S} = \bigcup S_i \; and \; \bigcap S_i = \phi \; for \; i = 1, \ldots, d'$. Note that in the WLMC- and AWLMC-based tracking methods, we divide the state space of position, $\mathbf{S}^p$, only into d disjointed subregions where $d \ll d'$. Compared with those methods, the NFWL-based tracking method copes with exponentially increased number of subregions. Therefore, a more efficient sampling strategy is needed to obtain good performance in estimating the ML and DOS terms with a limited number of samples.

3.1.3.2 Algorithm

The efficiency of the NFWL-based tracking method comes from two properties in the sampling process. The first property is that the method is a rejection-free algorithm. That is it accepts all proposed states without the acceptance step. The second is that the method employs a more efficient scheme to estimate the DOS term in the estimation step. It tries to obtain the DOS term with a smaller number of samples as much as possible.

• **Proposal step:** Since the method has no acceptance step, it should propose states with good quality. The states with good quality means the states that describe the target posterior distribution well enough. To well describe it, the states of high posterior probability should be more frequently sampled, and the states of low posterior probability should be less frequently sampled. For this, the method presents a new proposal density which consists of two parts. The first part is to choose a subregion of $\mathbf{S}$, and the second is to propose a state that belongs to the subregion.

When the previous state belongs to the i-th subregion S_i, the method chooses the j-th subregion S_j of the next state with the following probability:

$$p(S_j|S_i) = \frac{\frac{p(\mathbf{Y}_t|S_j)}{p(\mathbf{Y}_t|S_i)} \frac{g(S_i)}{g(S_j)} \left(\frac{d(S_i,S_0)}{d(S_j,S_0)}\right)^{\alpha}}{\sum_{j=1}^{d'} \frac{p(\mathbf{Y}_t|S_j)}{p(\mathbf{Y}_t|S_i)} \frac{g(S_i)}{g(S_j)} \left(\frac{d(S_i,S_0)}{d(S_j,S_0)}\right)^{\alpha}}, \tag{3.10}$$

where $p(\mathbf{Y}_t|S_j)$ is the ML score of the subregion S_j; $g(S_j)$ is the DOS score of the subregion S_j; S_0 is the subregion that includes the MAP state at the previous frame, $\hat{\mathbf{X}}_{t-1}$; $d(S_j, S_0)$ is the distance between S_j and S_0; d' is the total number of

subregions in $\mathbf{S}$; and α is the parameter. The method calculates $d(S_j, S_0)$ as follows:

$$d(S_j, S_0) = \frac{\|\bar{x}(S_j) - \bar{x}(S_0)\|_2}{MAX\|\bar{x}(\cdot) - \bar{x}(S_0)\|_2} + \frac{\|\bar{y}(S_j) - \bar{y}(S_0)\|_2}{MAX\|\bar{y}(\cdot) - \bar{y}(S_0)\|_2} + \frac{\|\bar{s}(S_j) - \bar{s}(S_0)\|_2}{MAX\|\bar{s}(\cdot) - \bar{s}(S_0)\|_2}, \tag{3.11}$$

where $(\bar{x}(S_j), \bar{y}(S_j))$ represents the center position of the subregion S_j, and $\bar{s}(S_j)$ indicates the average scale value in S_j. In (3.10), $\frac{p(\mathbf{Y}_t|S_j)}{p(\mathbf{Y}_t|S_i)}$, $\frac{g(S_i)}{g(S_j)}$, and $\frac{d(S_i,S_0)}{d(S_j,S_0)}$ describe that the method prefers to choose a new subregion that has higher ML score, lower DOS score, and lower distance from $\hat{\mathbf{X}}_{t-1}$ compared with the previous subregion, respectively. This means that the method frequently makes a move to the subregion 1) where the target might be, 2) that is insufficiently explored, and 3) close to the subregion where the target existed at the previous frame. Using $\frac{d(S_i,S_0)}{d(S_j,S_0)}$ in (3.10), the subregion which is close to the previous optimal state is proposed because the motion of the target is typically smooth. However, this assumption is not hold for abrupt motions. Therefore, the DOS term is additionally inserted by $\frac{g(S_i)}{g(S_j)}$ in (3.10). This term encourages to propose the subregions that have not been explored sufficiently. Since these subregions are far from the previous optimal state, they typically contain the states of abrupt motions. Thus, owing to the additional DOS term, our method can deal with the abrupt motions as well.

With the chosen subregion S_j according to the probability $p(S_j|S_i)$ in (3.10), the method then proposes a new state $\mathbf{X}'_t$ as follows:

$$\mathbf{X}'_t = \begin{cases} 2\mathbf{X}_{t-1} - \mathbf{X}_{t-2} + G(0, \sigma^2), & \text{if } i = j \\ U(S_j), & \text{if } i \neq j \end{cases}, \tag{3.12}$$

where $G(0, \sigma^2)$ indicates the Gaussian perturbation with mean 0 and variance σ^2, and $U(S_j)$ denotes the uniform sampling in the subregion S_j. In (3.12), if the chosen subregion is the same as the current one, a state is proposed by the second-order

autoregressive model [68] for further exploitation. On the other hand, if the chosen subregion is different, the method proposes a state by the uniform sampling in the chosen subregion for further exploration. This shows our proposal density makes full use of the exploitation and exploration abilities, adaptively and efficiently.

• **Estimation step:** To update the DOS with a smaller number of moves as much as possible, the method calculates the life-time of the previous subregion as follows:

$$\tau(S_i) = \frac{d'}{\sum_{j=1}^{d'} \frac{p(\mathbf{Y}_t|S_j)}{p(\mathbf{Y}_t|S_i)} \frac{g(S_i)}{g(S_j)} \left(\frac{d(S_i,S_0)}{d(S_j,S_0)}\right)^{\alpha}}, \tag{3.13}$$

where $\tau(S_i)$ denotes the life-time value of the subregion S_i. Unlike WLMC in which the histogram of a subregion is increased by 1 using (3.7) whenever it obtains a sample belonging to the subregion, NFWL estimates how many times a move to other subregions would be rejected on the average in the usual update scheme. This estimated value is the life-time in (3.13). The life-time value reflects how many samples are continually obtained by the method at the previous subregion. By estimating this value, our method can update the histogram of the previous subregion not by one in (3.7), but by the amount of the life-time value in (3.13):

$$h(S_i) \leftarrow h(S_i) + \tau(S_i), \tag{3.14}$$

where $h(S_i)$ indicates the histogram value of the subregion S_i. The DOS score is then adaptively estimated by

$$g(S_i) \leftarrow g(S_i) * \tau(S_i) * f, \tag{3.15}$$

where $g(S_i)$ indicates the DOS score of S_i, and f is a modification factor.

To make the flat histogram, the sampling method should visit all subregions sufficiently by moving a subregion to other subregions. The problem of WLMC is

that it needs many samples to move from a subregion to other subregions, which is called as tunneling time [70]. On the other hand, NFWL needs a small number of samples to move from a subregion to other subregions because it always accepts the moves. In this case, however, the histogram is biased due to the rejection-free property of NFWL. The life-time value $\tau(S_i)$ in (3.14) compensates for the bias of the histogram by measuring how many times a move to other subregions would be rejected on the average in the usual update scheme. Hence, by updating the amount of $\tau(S_i)$ at a time, NFWL quickly produces the flat histogram using a small number of samples. In addition, the life-time value of S_i, $\tau(S_i)$ is obtained by considering the DOS of all other subregions $\{S_j\}_{j=1}^{d'}$ in (3.13). Then, $\tau(S_i) * f$ represents how much the DOS of S_i should be modified compared to the DOS of other subregions. By considering the DOS of other subregions to modify the DOS of S_i, NFWL can reduce errors in the DOS at right edges of an interval, over which one wants to determine the DOS [70].

3.1.3.3 Converge Analysis of NFWL

The NFWL method eventually converges after sufficient time. The convergence of the method can be demonstrated by showing the convergence of error in the estimation of the DOS. To theoretically prove the convergence of the error, we define the estimation error $\epsilon(S_i^p, k)$ of the simulated DOS score $g(S_i^p, k)$ of the i-th subregion S_i^p at the k-th iteration:

$$\epsilon(S_i^p, t) = \left| 1 - \frac{\ln\left[g_n(S_i^p, k)\right]}{\ln\left[g_{ex}(S_i^p)\right]} \right| = \left| \frac{\ln\left[g_{ex}(S_i^p)\right] - \ln\left[g_n(S_i^p, k)\right]}{\ln\left[g_{ex}(S_i^p)\right]} \right|, \tag{3.16}$$

where $g_n(S_i^p, k)$ is the normalized version of $g(S_i^p, k)$ with respect to the exact DOS score $g_{ex}(S_*^p)$ of the subregion S_*^p, which includes the MAP state, and $g_{ex}(S_i^p)$ is the

exact DOS score of the subregion S_i^p. In (3.16), $g_n(S_i^p, k)$ is obtained by

$$g_n(S_i^p, k) = \frac{g(S_i^p, k) g_{ex}(S_*^p)}{g(S_*^p, k)}, \tag{3.17}$$

where $g(S_*^p, k)$ is the simulated DOS score of the subregion S_*^p at the k-th iteration. Then, the convergence of the error is proven theoretically with the following lemma based on [71]:

Lemma 1. The error $\epsilon(S_i^p, t)$ in (3.16) goes to a constant limit as the iteration increases to infinity.

Proof. By substituting $g_n(S_i^p, k)$ in (3.17) into $\epsilon(S_i^p, t)$ in (3.16), we get

$$\epsilon(S_i^p, t) = \frac{|\triangle \ln [g(S_i^p, k)] - \triangle \ln [g_{ex}(S_i^p)]|}{\ln [g_{ex}(S_i^p)]}, \tag{3.18}$$

where $\triangle \ln [g(S_i^p, k)] = \ln [g(S_i^p, k)] - \ln [g(S_*^p, k)]$ and $\triangle \ln [g_{ex}(S_i^p)] = \ln [g_{ex}(S_i^p)] - \ln [g_{ex}(S_*^p)]$. In (3.18), $\triangle \ln [g_{ex}(S_i^p)]$ is fixed during the simulation because it is not formulated as the function of the parameter k which is the number of iteration. In addition, $\triangle \ln [g(S_i^p, k)]$ becomes constant as $\ln [g(S_i^p, k)]$ and $\ln [g_n(S_*^p, k)]$ converge after the sufficient number of integrations k. This is because $\ln [g(S_i^p, k)]$ can be reformulated as

$$\ln [g(S_i^p, k)] = \sum_{i=1}^{k} [H(S_i^p, i) - H(S_i^p, i-1)] (\ln[f_{i-1}\tau_{i-1}]), \tag{3.19}$$

where $H(S_i^p, i)$ stands for the histogram of the subregion S_i^p at the i-th iteration, f_{i-1} denotes the modification factor at the *(i-1)*-th iteration, and τ_{i-1} indicates the life-time value at the *(i-1)*-th iteration. Then, $\ln [g(S_i^p, k)]$ in (3.19) is convergent because $H(S_i^p, i) - H(S_i^p, i-1)$ is finite and $\sum_{i=1}^{\infty} (\ln[f_i] + \ln[\tau_i])$ is also convergent.

By removing the acceptance step, we achieve a substantial amount of improvement on efficiency especially when the rejection rate is very high due to the rough

Algorithm 1 Our NFWL-based tracking method

Input: $\{\mathbf{X}_{t-1}^{(l)}\}_{l=1}^{N}$

Output: $\hat{\mathbf{X}}_t = (\hat{X}_t^x, \hat{X}_t^y, \hat{X}_t^s), \{\mathbf{X}_t^{(l)}\}_{l=1}^{N}$

1: **Bayesian Filtering step:** $p(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1}) \approx \{\mathbf{X}_{t-1}^{(l)}\}_{l=1}^{N}$ based on (1.5)(1.6). The method randomly selects a sample $\mathbf{X}_{t-1}^{(l)}$ and moves the selected sample according to the second-order autoregressive function, and use the result as the initial state of the $\mathbf{X}_t$ Markov chain.

2: **for** 1 to N **do**

3: **Proposal step:** Propose a new state using (3.10)(3.12).

4: **Estimation step:**

5: 1. Estimate the life-time using (3.13).

6: 2. Update the histogram using (3.14).

7: 3. Estimate the DOS using (3.15).

8: 4. Estimate the marginal-likelihoods using (3.3).

9: **Annealing step:** Reduce the state space by the process explained in Section 3.1.2.

10: **end for**

11: Estimate the MAP state $\hat{\mathbf{X}}_t$ using (1.2).

state space. Additionally, the adaptive update scheme of the DOS speeds up the sampling process and reduces the computational cost. These advantages in the NFWL-based tracking method permit it to cope with the abrupt changes of scale and to increase the accuracy of tracking results in real challenging videos. Algorithm 1 describes the whole procedure of our NFWL-based tracking method. Figure 3.9 illustrates similarly and dissimilarity among the WLMC-based tracking methods. Note that the N-fold algorithm cannot be combined with AWLMC. The main reason is that, in the N-fold algorithm, the proposal density function $p(S_j|S_i)$ in (3.10) and the life-time $\tau(S_i)$ in (3.13) are estimated by using original subregions $\{S_j\}_{j=1}^{d'}$.

Figure 3.9: **Similarly and dissimilarity among the WLMC-based tracking methods**. Same color means the corresponding methods include the same procedure. White color means that the corresponding method does not include the step.

Thus, the subregions cannot be modified like as AWLMC.

3.1.4 Experimental Details and Results

3.1.4.1 Experiment Settings

We tested eleven video sequences: *Boxing* and *Youngki* sequences for camera shot changes; *Singer*, *Snowboard*, *Elephant*, and *Bird* sequences for partially low-frame rate videos; and *Tennis*, *Animal*, *Badminton*, *Pingpong*, and *Football* sequences for rapid motions. For the comparative evaluation, we compared the proposed algorithms [WLMC, AWLMC, and NFWL] with eight different tracking methods [AWLMC$^+$, Standard MCMC (MCMC), Adaptive MCMC (AMCMC), Quasi-Random Sampling (QR), Particle Filter (PF), Mean Shift (MS), Stochastic Approximation Monte Carlo (SAMC), and Visual Tracking Decomposition (VTD)].

WLMC and AWLMC divided the state space of position into 6×4 subregions. On the other hand, NFWL divided the state space of scale as well and made a total of $6 \times 4 \times 6$ subregions. For the proposal density of WLMC and AWLMC, we set σ_x,σ_y in (3.1) and σ_s in (3.2) to 250, 250, and 1.414, respectively. In the case of NFWL, α in (3.10) and (3.13) were set to 0.5 for most sequences. Furthermore, σ in (3.12) was set to 2 and 1.414 for the x and y directions, respectively. The upper

bound of the scale is 5 and the lower bound is 0.1. The modification factor in (3.8) and (3.15) were initially set to 2.7. AWLMC$^+$ is a simple extension of AWLMC to cope with abrupt changes in scale. To do this, $6 \times 4 \times 6$ subregions were constructed and the ML and DOS scores were estimated for the six subregions of scale as well by utilizing the same process of AWLMC. In AWLMC$^+$, the scale s is sampled using the similar method in (3.1) instead of that in (3.2). MCMC is the tracking method based on [18]. Proposal variances of MCMC were set to 8 and 4 for the x and y directions, respectively. AMCMC is the Modified Adaptive Metropolis algorithm based on [39, 40], which increases or decreases the variance of proposal density to make the acceptance rate of proposal as close as possible to 0.44. For this, proposal variances of AMCMC were adaptively changed from 8 to 28 for the x direction and from 4 to 24 for the y direction. QR is the tracking method based on [15]. To track abrupt motions, 60 samples were quasi-randomly distributed over the whole state space. The rest of the samples were proposed around the local maxima states where each local maximum state included 20 samples. PF is the tracking method based on [33]. The motion model of PF utilized the second-order autoregressive process. The noise model was defined by the Gaussian function of which the variance was set to 250. MS [19] is the tracking method based on the implemented function in OpenCV. SAMC and VTD is the tracking method based on [17, 26], respectively. Note that our current method cannot handle rotation of the target, same as other tracking methods (PF, QR, MCMC, AMCMC, SAMC, and VTD).

3.1.4.2 Accuracy of tracking results

• **Coverage test:** The recall ρ and the precision ν measure the configuration errors between the ground truth state and the estimated state [36]: $\rho = \frac{A_t^X \cap A_t^G}{A_t^G}$, $\nu =$

Table 3.1: **Accuracy of tracking abrupt changes of position** (*F-measure*).

Seq.	WLMC	AWLMC	NFWL	MCMC	AMCMC	PF	QR	MS	VTD	SAMC
Boxing	0.82	**0.83**	**0.83**	0.30	0.37	0.59	0.70	0.14	0.14	0.81
Youngki	0.79	0.81	**0.81**	0.41	0.43	0.39	0.67	0.23	0.18	**0.81**
Tennis*	0.59	0.63	**0.75**	0.17	0.17	0.10	0.61	0.09	0.14	**0.72**
Animal*	0.41	0.47	**0.62**	0.20	0.20	0.28	0.38	0.02	0.07	**0.63**
Badminton	0.35	0.36	**0.82**	0.35	0.36	0.19	0.28	0.11	0.41	**0.42**
Pingpong	0.28	0.29	**0.72**	0.29	0.29	0.29	0.36	0.05	**0.36**	0.35
Football	0.17	0.19	**0.67**	0.61	**0.67**	0.47	0.33	0.11	0.39	0.18

$\frac{A_t^X \cap A_t^G}{A_t^X}$, where A_t^X denotes the estimated area and A_t^G indicates the ground truth area at time t. The recall ρ measures how much of the ground truth area A_t^G is covered by the estimated area A_t^X. The precision ν measures how much of A_t^X covers A_t^G. For good tracking quality, both the recall and precision should have high values. In information retrieval literatures, the *F-measure* is often used for evaluating this quantity:$F = \frac{2\nu\rho}{\nu+\rho}$. When the ground truth and estimated area perfectly overlap, the *F-measure* is 1.0.

Table 3.1 summarizes the tracking results of targets whose positions are abruptly changing. The numbers in the table indicate the *F-measure* where a higher number represents more accurate tracking results. In most sequences, NFWL most accurately tracked the target even though there were drastically abrupt changes in position. NFWL could find the best local maximum state of a target because with the equal number of samples, it searched larger portions of the state space compared with MCMC, AMCMC, PF, QR, and VTD. Additionally, the searching strategy of NFWL was more efficient than WLMC and AWLMC. WLMC and AWLMC showed the second-best performance on average. The performance of AWLMC was always better than WLMC due to the annealing step in AWLMC. VTD failed to track the

Table 3.2: **Accuracy of tracking the targets when there are only smooth motions** (*F-measure*).

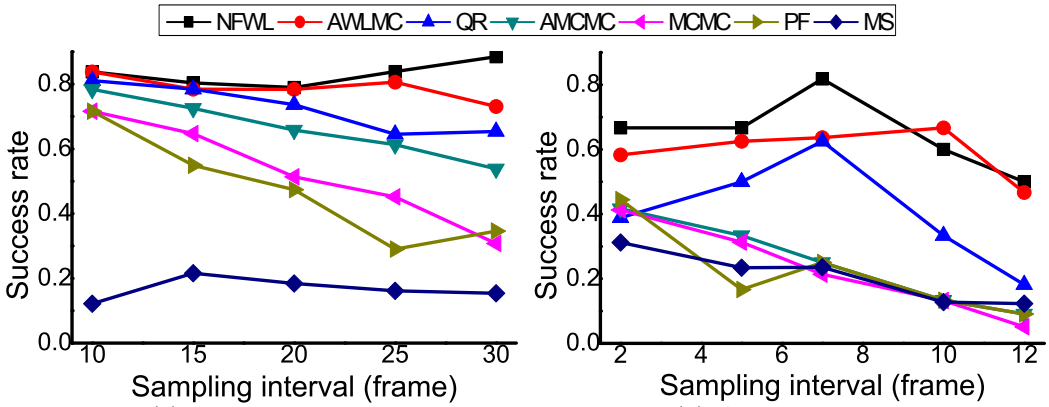
Seq.	WLMC	AWLMC	NFWL	MCMC	AMCMC	PF	QR	MS	VTD	SAMC
Boxing	0.82	**0.83**	**0.83**	0.80	0.80	0.71	0.70	0.32	0.81	0.82
Youngki	0.79	0.81	**0.81**	0.78	0.78	0.76	0.72	0.51	0.79	**0.81**

target in most sequences when there were drastically abrupt changes of position. In *Badminton* and *Pingpong* sequence, VTD accurately tracked the target only when the changes of position were relatively less abrupt. SAMC robustly tracked the targets in *Youngki* and *Animal** sequences, which include highly abrupt motions. However, SAMC frequently missed the targets when there were severe background clutter and occlusion as well as abrupt motions in *Pingpong* and *Football* sequences. Note that we obtained the ground truth states by manually drawing the bounding box around the target. For the sampling-based methods, we used the same number of samples, 1000. *Tennis** and *Animal** are down-sampled sequences of *Tennis* and *Animal* with the sampling interval of 25 and 5, respectively.

To cope with the abrupt motions, the sampling-based methods typically utilize a large variance in the proposal density. However, the large variance causes the tracking accuracy of the smooth motions to decrease severely. Hence, it is crucial to compare the tracking performance in the smooth motion case as well. For comparison, we used *Boxing* and *Youngki* sequences that mainly consist of smooth motions. To make the sequences include only smooth motions, we re-initialized the states of other tracking methods to the ground truth before they failed to track the abrupt motions. As shown in Table 3.2, AWLMC and NFWL accurately tracked the smooth motions as good as other tracking methods, even though we did not re-initialize the states of AWLMC and NFWL. In conclusion, Table 3.1 and 3.2

Table 3.3: **Accuracy of tracking abrupt changes of both position and scale** (*F-measure*).

Seq.	AWLMC	AWLMC$^+$	NFWL	MCMC	AMCMC	PF	QR	MS	VTD	SAMC
Singer	0.34	0.55	**0.87**	0.70	0.70	0.54	0.53	0.10	**0.83**	0.48
Snowboard	0.59	**0.60**	**0.76**	0.36	0.41	0.39	0.54	0.01	0.35	0.54
Elephant	0.16	0.15	**0.82**	0.61	**0.69**	0.30	0.20	0.10	0.16	0.43
Bird	0.19	0.01	**0.63**	0.19	0.19	0.09	0.01	0.01	**0.67**	0.01



(a) *Tennis* sequence (b) *Animal* sequence

Figure 3.10: **Success rate** at *Tennis* and *Animal* sequences as a function of down-sampling interval for different tracking methods.

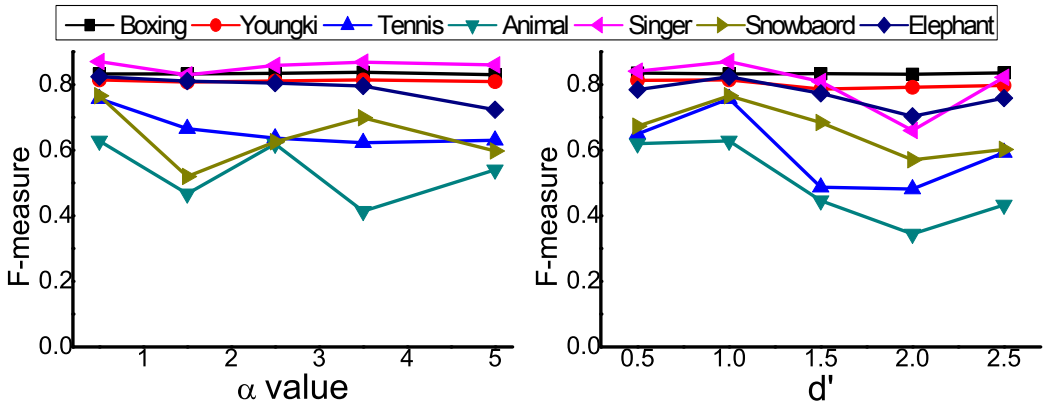
prove that AWLMC and NFWL make full use of the exploitation ability of the ML term with the exploration ability of the DOS term to track both smooth and abrupt motions efficiently.

When there were abrupt changes in both position and scale, NFWL drastically outperformed AWLMC and other tracking methods, as shown in Table 3.3. The performance of AWLMC decreased when there were severe changes in the scale, as with the case of *Singer*, *Snowboard*, *Elephant*, and *Bird* sequences because AWLMC assumed that scale changes are smooth over time. AWLMC$^+$ improved AWLMC by considering abrupt changes in the scale. However, AWLMC$^+$ performed worse due to the curse of dimensionality because the same process of AWLMC, in spite of increased subregions, is applied. SAMC failed to track the targets when there were

abrupt changes in both position and scale. This is because SAMC basically assumes that scale of the targets smoothly changes over time. VTD showed the good tracking performance when there were abrupt changes of both position and scale. However, the tracking results of NFWL were better than those of VTD because the sampling strategy utilizing the DOS in NFWL are more adequate to deal with the abrupt motions as comparison with sampling strategy utilizing the Interacting MCMC in VTD.

• **Success rate:** If the *F-measure* is larger than 0.5, the target is considered as correctly tracked at that frame. The success rate indicates the ratio between the number of correctly tracked frames and the number of total frames. If the methods failed to track the target, we re-initialized the state of the target with the ground truth. For the test, we down-sampled the *Tennis* sequence with the sampling interval from 10 frames to 30 frames, and the *Animal* sequence with the sampling interval from 2 frames to 12 frames. The results are depicted in Figure 3.10. In comparison with the other results, the success rates of AWLMC and NFWL are less affected by the change of the sampling interval, whereas those of other methods rapidly decrease as the sampling interval increases. Note that AWLMC and NFWL successfully tracked the target even in highly down-sampled videos which contained severely abrupt motions.

• **Parameter effects:** The tracking accuracy could be dependent on several parameters. To evaluate the robustness of the tracking methods, it is important to check how much of their results are affected by the parameters. In NFWL, two important parameters are α in (3.10) and (3.13), and d' described in Figure 3.8. If we increase the α value, the tracking performance of NFWL for smooth motions may be improved while the accuracy of tracking abrupt motions decreases. If we set



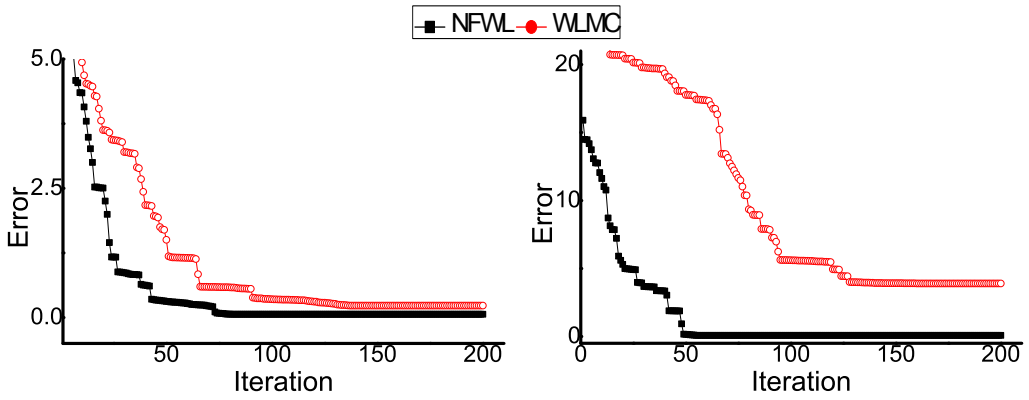
(a) Parameter α (b) Parameter d'

Figure 3.11: **Accuracy of tracking results with different parameter settings for NFWL.** (a) The α value controls the proposal variance. (b) The d' value determines the total number of subregions.

d' to a high value, NFWL could obtain more accurate states of the target. However, an increased number of samples is needed. As shown in Figure 3.11, NFWL was not sensitive to these parameters and robustly tracked the target with different parameter settings. Note that in Figure 3.11(b), X0.5 indicates that we divided the state space into $3 \times 2 \times 3$ subregions for NFWL, whereas the default setting is $6 \times 4 \times 6$.

3.1.4.3 Efficiency of sampling strategy

• **Accuracy of the DOS estimation:** Numerical experiment demonstrates that NFWL estimates the DOS more accurately with a small number of samples as comparison with WLMC. Figure 3.12 shows convergence of the error $\epsilon(S_i^p, t)$ defined in (3.16) for WLMC and NFWL. As illustrated in Figure 3.12, NFWL converges at the lower error values in most sequences, which means it estimates the DOS more accurately as comparison with WLMC. In the experiments, NFWL converged at the lower error values, 0.06501 and 0.08946 in *Animal* and *Tennis* sequences, respec-



(a) *Animal* sequence (b) *Tennis* sequence

Figure 3.12: **Convergence of the error** $\epsilon(S_i^p, t)$ defined in (3.16) for WLMC and NFWL.

tively, whereas WLMC converged at the higher error values 0.23372 and 3.90436 in *Animal* and *Tennis* sequences, respectively. Moreover, NFWL converges more rapidly in most sequences, which means it estimates the DOS with a small number of samples as comparison with WLMC. In the experiments, NFWL converged after shorter iterations, 82 and 55 in *Animal* and *Tennis* sequences, respectively, whereas WLMC converged after longer iterations, 138 and 160 in *Animal* and *Tennis* sequences, respectively. In the case of AWLMC, it converged at about $70 \sim 150$ iterations, which are between WLMC and AWLMC. Note that the ground truth of the DOS at each sequence was manually obtained according to $\ln[p(\mathbf{X}_t|\mathbf{Y}_{1:t})]$.

• **Number of samples:** Figure 3.13 demonstrates that NFWL needs a smaller number of samples compared with other methods to reach a similar tracking performance. For example, NFWL needed only 50 samples to get the tracking accuracy of 0.8 in the *Boxing* sequence, while AWLMC and WLMC needed more than 200 samples. Additionally, the figure shows that NFWL maintains the best performance even with a drastically small number of samples. In all sequences, NFWL produced the most accurate results regardless of the number of samples.

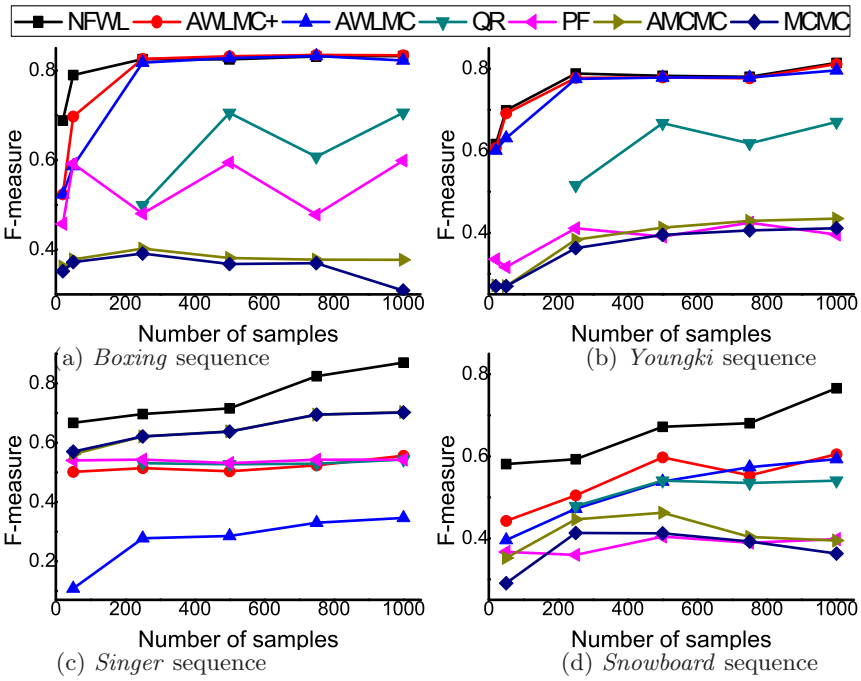
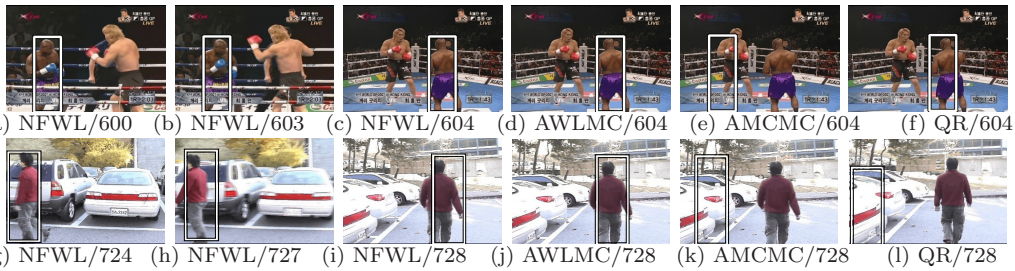


Figure 3.13: **Accuracy of tracking as the number of samples increases**.

Table 3.4: **Runtime of tracking methods** (*frame/second*).

WLMC	AWLMC	NFWL	MCMC	AMCMC	PF	QR	MS
7.0	7.0	7.1	7.1	7.6	9.5	9.7	18.2

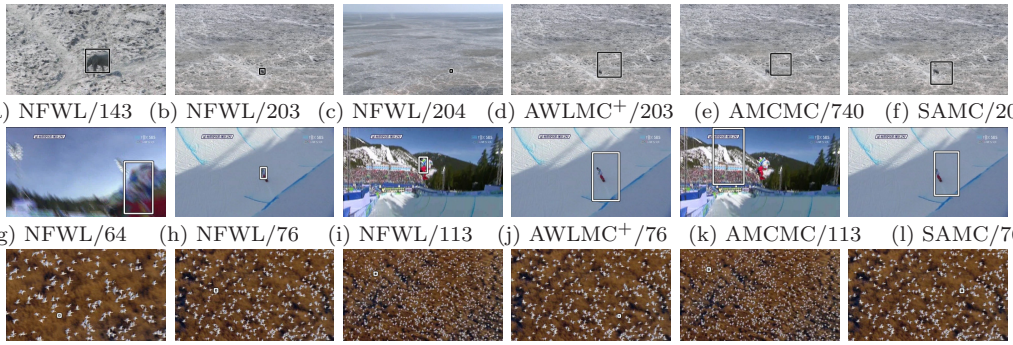
• **Computational cost:** As shown in Table 3.4, The runtime of the sampling-based tracking methods (WLMC, AWLMC, NFWL, MCMC, AMCMC, PF, and QR) were similar because they used the same number of samples. This result demonstrated that WLMC, AWLMC, and NFWL have no additive computational burden compared to the other sampling-based tracking methods because the DOS can be calculated at an extremely less computational cost. Note that our current implementation is not optimized, and it spends most computational time to get a likelihood score by measuring the diffusion distance in [69]. Thus, by properly optimizing the process of measuring the diffusion distance, we can greatly enhance the speed.



(a) NFWL/600 (b) NFWL/603 (c) NFWL/604 (d) AWLMC/604 (e) AMCMC/604 (f) QR/604

(g) NFWL/724 (h) NFWL/727 (i) NFWL/728 (j) AWLMC/728 (k) AMCMC/728 (l) QR/728

Figure 3.14: **Tracking results when the camera shot change occurs** in *Boxing* and *Youngki* sequences.



(a) NFWL/143 (b) NFWL/203 (c) NFWL/204 (d) AWLMC$^+$/203 (e) AMCMC/740 (f) SAMC/203

(g) NFWL/64 (h) NFWL/76 (i) NFWL/113 (j) AWLMC$^+$/76 (k) AMCMC/113 (l) SAMC/76

(m) NFWL/0 (n) NFWL/47 (o) NFWL/107 (p) AWLMC$^+$/47 (q) AMCMC/107 (r) SAMC/47

Figure 3.15: **Tracking results** in *Elephant*, *Snowboard*, and *Bird* sequences wherein both position and scale are drastically changing.

3.1.4.4 Qualitative Comparison

• **Camera shot changes:** Figure 3.14 presents the tracking results of test videos that contain the camera shot changes. In these videos, NFWL, AWLMC, and WLMC successfully tracked the target even though there are drastically abrupt changes in the position. QR also tracked the abrupt motions in the *Boxing* sequence (Figure 3.14(f)), whereas it failed to track the motions in the *Youngki* sequence (Figure 3.14(l)). On the other hand, the other methods failed to escape from the previous positions of the targets.

• **Partially low frame rate:** As another example of an abrupt motion, we

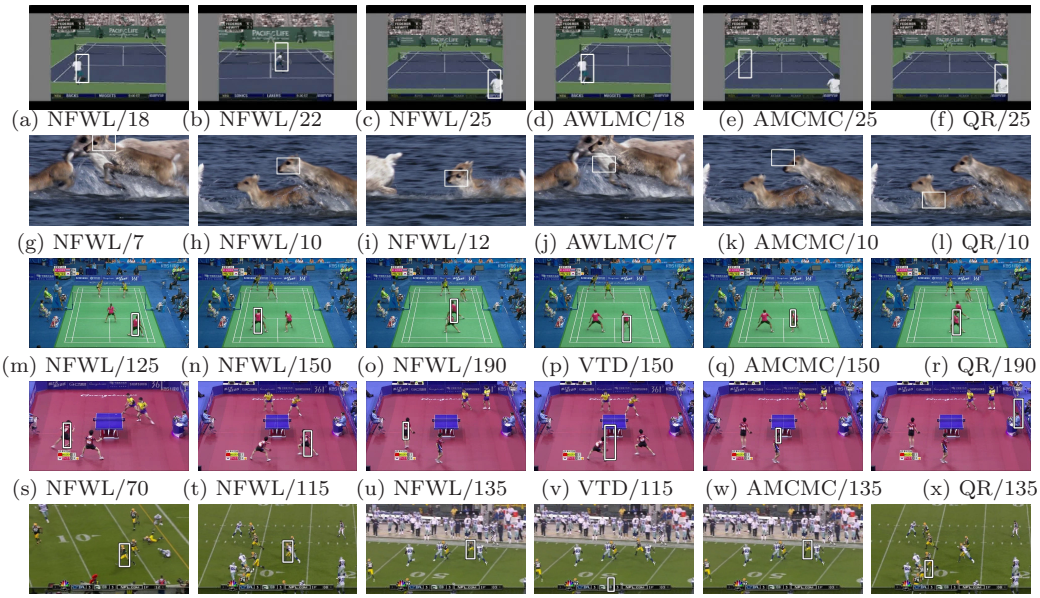


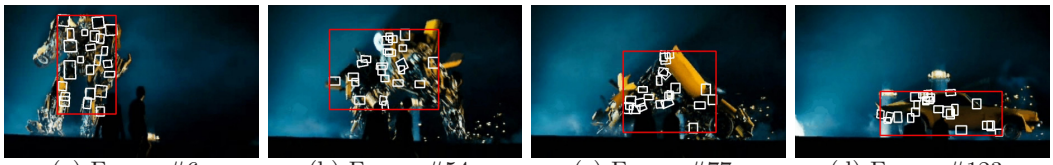
Figure 3.16: **Tracking results** in the down-sampled *Tennis*, *Animal*, *Badminton*, *Pingpong*, and *Football* sequences that include rapid motions.

tested *Singer* [1], *Snowboard*, *Elephant*, and *Bird* sequences which have partially low frame rates. In comparison with *Boxing* and *Youngki* sequences, these sequences are more challenging because they include the targets wherein both position and scale are drastically changing simultaneously. Moreover, the *Elephant* sequence contains severe background clutters of which appearance is similar to that of a target, the *Snowboard* sequence includes a highly deformable target, and the *Bird* sequence contains an extremely large number of objects that share a similar appearance to the target. The *Bird* sequence is also challenging because the size of the target becomes very small as time goes on. As shown in Figure 3.15, NFWL robustly tracked the targets in the background clutters even though the scale of the targets increased or decreased more than threefold in a short time. NFWL accurately tracked the

[1]The source code, test datasets, and result videos are available at http://cv.snu.ac.kr/research/nfwl/.

small object in *Bird* and *Elephant* sequences because the colors between the object and backgrounds is discriminable enough and robust similarity measure, diffusion distance, is adopted for our histogram-based appearance model. Because there is a substantial difference between the colors of the object and backgrounds, AMCMC also tracked the small object in *Elephant* sequence, while it failed to track the object just after the abrupt scale change. Owing to robust similarity measure, VTD also tracked the small object successfully in *Bird* sequence. On the other hand, all the other tracking methods failed to track the targets as illustrated in Figure 3.15. AWLMC$^+$ failed to track the abrupt motions because it needs a vast number of samples to cover the changes. The results of AWLMC and SAMC were also worse because they could not cope with the abrupt changes in scale.

- **Rapid motion:** Abrupt motion also occurs when the target moves rapidly. For the test, we down-sampled the *Tennis*, *Animal*, *Badminton*, *Pingpong*, and *Football* sequences, and made highly rapid motions whose directions and moving distances were quite unpredictable. NFWL efficiently addressed this motion uncertainty and accurately tracked the targets, as shown in Figure 3.16. *Badminton*, *Pingpong*, and *Football* sequences contain rapid objects that share a similar appearance to the targets. In addition, the sequences are more challenge because the targets and similar objects frequently occlude each other in the *Badminton* and *Pingpong* sequences, red color of the target is quite similar to that of background in the *Pingpong* sequence, and the target is very deformable in the *Football* sequence. Although it is extremely hard to track the targets in these tracking environments, NFWL successfully discriminated the targets from the similar objects in the background, and most accurately tracked the targets. On the other hand, the other methods including AWLMC and WLMC, frequently failed to track the targets when the targets



(a) Frame #6 (b) Frame #54 (c) Frame #77 (d) Frame #123

Figure 3.17: **Example of tracking results** in *transformer* seq. The proposed tracking algorithm successfully tracks a target even when the target's geometric appearance changes drastically. The white squares represent the affine transformed-local patches in the appearance model.

abruptly changed their positions, as shown in Figure 3.16.

3.2 The Basin Hopping Monte Carlo (BHMC) Tracker

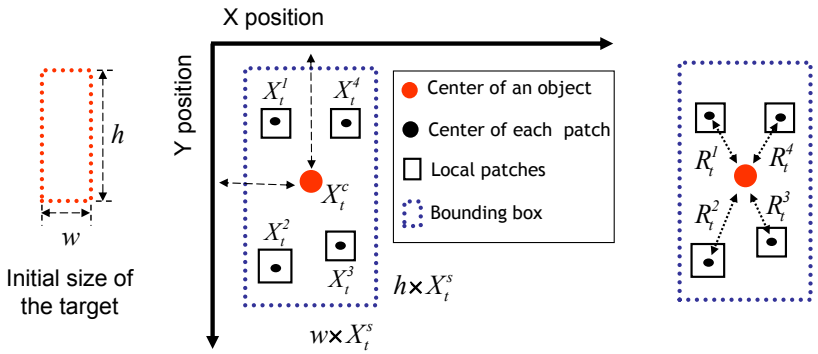
To track an object robustly under difficult real-world settings, tracking algorithms need to consider the target object's appearance changes adaptively. Recently, numerous online learning algorithms have been proposed to address *photometric appearance changes*, and these algorithms have shown promising results [63, 6, 7, 8, 10, 11, 51, 24, 1, 64, 72]. However, few studies focus on the geometric appearance changes in target objects. In this paper, we address the problem of tracking non-rigid objects, the geometric appearances of which change drastically over time. Although more generally applicable, the results in the present paper focus specifically on tracking the objects in scenes from movies and sports, which usually exhibit a large amount of *extreme geometric appearance changes*. Under aforementioned scenes, conventional tracking methods frequently fail to track the target object. Figure 3.17 shows a tracking example of such an object by the proposed method.

The philosophy of the proposed method lies in taking the advantages of both the histogram-based appearance [19, 33] and pixel-wise models [73, 10]. Note that the histogram-based appearance model covers geometric variations to some degree,

but loses spatial information of target objects. On the other hand, the pixel-wise model preserves all spatial information, but typically fails to capture the extreme geometric changes of target objects. To cover the geometric changes without losing spatial information of target objects, we propose a local patch-based appearance model as in [74, 75, 12] and present a new strategy for its online construction. The proposed local patch-based appearance model comprises a number of local patches and the topology between the patches. In the proposed model, the patch contains the local information of the target appearance. The topology exploits spatial relations of the patches. The model then preserves the spatial information of the target object using the topology between local patches. The model could also cover geometric variations well because the topology between local patches evolves through online update over time.

In the MCMC based tracking method, the design of the state, $\mathbf{X}_t$, is crucial to the success of visual tracking because it significantly affects the performance of the MCMC sampling method during the proposal step in (1.3) and the acceptance steps in (1.4). Ordinarily, the state at time t is represented as a three-dimensional vector, $\mathbf{X}_t = (X_t^x, X_t^y, X_t^s)$, where X_t^x, X_t^y, and X_t^s indicate the x, y position and scale of the target object, respectively. However, with conventional state representation, the tracking methods typically fail if the target is highly non-rigid because the representation cannot completely describe the geometric appearance of the target and cannot robustly cover its changes. To overcome these problems, the tracking methods require more advanced state representation, which describes the target's geometric appearance well. Thus, the proposed method presents a novel local patch-based appearance model in Section 3.2.1.

However, the local patch-based appearance model generates serious problems in



(a) Example of $\mathbf{X}_t = (\mathbf{X}_t^c, X_t^s, \mathbf{X}_t^1, \mathbf{X}_t^2, \mathbf{X}_t^3, \mathbf{X}_t^4)$ (b) Example of $\mathbf{R}_t = (\mathbf{R}_t^1, \mathbf{R}_t^2, \mathbf{R}_t^3, \mathbf{R}_t^4)$

Figure 3.18: **Example of proposed local patch-based appearance model** (a) The figure shows an example of the state, $\mathbf{X}_t$. (b) The figure describes an example of the topology between local patches, defined by $\mathbf{R}_t$.

conventional MCMC-based tracking approaches because the model should be represented as a very high-dimensional vectors. In the high-dimensional state space, the conventional MCMC method explained in Section 1.1 suffers from higher computational complexity because it requires an exponentially large number of samples to reach the global optimum. Additionally, the method becomes trapped in local optima more frequently because functions usually become rougher in a high-dimensional state space. To solve this problem, the proposed method presents the advanced MCMC method, the BH sampling method, to obtain samples efficiently, especially from the high-dimensional state space in Section 3.2.3.

3.2.1 Design of the Appearance Model

An object is represented by a local patch-based dynamic graph model as shown in Figure 3.18. In the proposed model, the object state $\mathbf{X}_t$ at time t is defined by $\mathbf{X}_t = (\mathbf{X}_t^c, X_t^s, \mathbf{X}_t^1, \cdots, \mathbf{X}_t^i, \cdots, \mathbf{X}_t^m)$ where $\mathbf{X}_t^c$ and X_t^s denotes the center position and scale of an object, respectively, $\mathbf{X}_t^i$ indicates the center position of the i-th local

patch, and m is the total number of local patches. Each local patch is assumed to be dependent only on the center of an object. This assumption is similar to those of the star model [75, 12]. The star model is frequently used because of its efficiency, which only considers the relation between center and each patch. As reported in [75], the star model has complexity $O(NP)$, while the fully connected model that considers all relations among patches has complexity $O(N^P)$, where N and P denote the number of features and patches, respectively. Although the fully connected model may completely utilizes the geometric information of the target, the complexity of the model exponentially increase, as the number of patches in the model increase. The k-pan model in [74] has advantages of both the star and fully connected models. Although this model showed good performance for the object recognition problem, the model did not produced good results for our tracking problem. The object recognition problem only considers a static scenario, in which the relations between patches never change over time. On the other hand, the tracking problem should cover all changes over time in relations between patches, especially if the target is highly non-rigid. To cover all changes in the model accurately and efficiently, the model should be designed with as simple relations between patches as possible, such like the star model. In our tracking problem, the star model produced more accurate results with lower complexity as comparison with the k-pan model.

Using the star model, the center position of the local patch, $\mathbf{X}_t^i$, is determined by $\mathbf{R}_t^i$, which represents the relative position between $\mathbf{X}_t^c$ and $\mathbf{X}_t^i$. In this manner, the topology of all local patches is constructed by $\mathbf{R}_t = (\mathbf{R}_t^1, \cdots, \mathbf{R}_t^i, \cdots, \mathbf{R}_t^m)$, as described in Figure 3.18(b). The objective of our tracking method is then finding the best sample of the object state, $\hat{\mathbf{X}}_t = (\hat{\mathbf{X}}_t^c, \hat{X}_t^s, \hat{\mathbf{X}}_t^1, \cdots, \hat{\mathbf{X}}_t^i, \cdots, \hat{\mathbf{X}}_t^m)$ using the MAP estimation in (1.2), where the l-th state sample is represented by $\mathbf{X}_t^{(l)} =$

$(\mathbf{X}_t^{c(l)}, X_t^{s(l)}, \mathbf{X}_t^{1(l)}, \cdots, \mathbf{X}_t^{i(l)}, \cdots, \mathbf{X}_t^{m(l)})$.

3.2.1.1 Photometric and Geometric Likelihoods

The likelihood is designed as:

$$p\left(\mathbf{Y}_t|O(\mathbf{X}_t^{(l)})\right) \approx \prod_{i=1}^{m}\left[p_p\left(\mathbf{Y}_t|O(\mathbf{X}_t^{i(l)})\right) p_g\left(O(\mathbf{X}_t^{i(l)})|\mathbf{X}_t^{c(l)}, \mathbf{R}_t^i\right)\right], \tag{3.20}$$

where p_p denotes the photometric likelihood and p_g indicates the geometric likelihood. In (3.20), $O(\mathbf{X}_t^{i(l)})$ returns the state vector of the local mode of the i-th patch centered on $\mathbf{X}_t^{i(l)}$, which indicates the locally best one among the states around $\mathbf{X}_t^{i(l)}$. To obtain the state of the local mode, the proposed method utilizes the image registration algorithm in [76]. In the image registration process, the i-th local patch is warped via affine transformation so that it best matches to the model image of the i-th patch, $\mathbf{M}_t^i$ at time t.

The photometric likelihood is then defined as:

$$p_p\left(\mathbf{Y}_t|O(\mathbf{X}_t^{i(l)})\right) = exp^{-\lambda_p F_1\left(I\left[O(\mathbf{X}_t^{i(l)})\right], \mathbf{M}_t^i\right)}, \tag{3.21}$$

where $I[O(\mathbf{X}_t^{i(l)})]$ indicates the patch image described by $O(\mathbf{X}_t^{i(l)})$, the F_1 function returns the normalized sum of squared differences between the patch at the state of the local mode and its model image, and λ_p denotes the weighting parameter set to 30. The i-th patch model $\mathbf{M}_t^i$ in (3.21) is updated by

$$\mathbf{M}_{t+1}^i = (1-\omega)\mathbf{M}^{i(ref)} + \omega\mathbf{M}_t^{i(dyn)}, \tag{3.22}$$

where $\mathbf{M}^{i(ref)}$ indicates the i-th reference local patch model in the initial frame and $\mathbf{M}_t^{i(dyn)}$ represents the model image obtained in the region of $O(\hat{\mathbf{X}}_t^i)$ at time t. The local patch model in the initial frame, $\mathbf{M}^{i(ref)}$, prevents from learning drastic appearance changes [77].

The geometric likelihood is defined by

$$p_g\left(O(\mathbf{X}_t^{i(l)})|\mathbf{X}_t^{c(l)}, \mathbf{R}_t^i\right) = exp^{-\lambda_g \left\| \left[O(\mathbf{X}_t^{i(l)}) - \mathbf{X}_t^{c(l)}\right] - \mathbf{R}_t^i \right\|_2}, \tag{3.23}$$

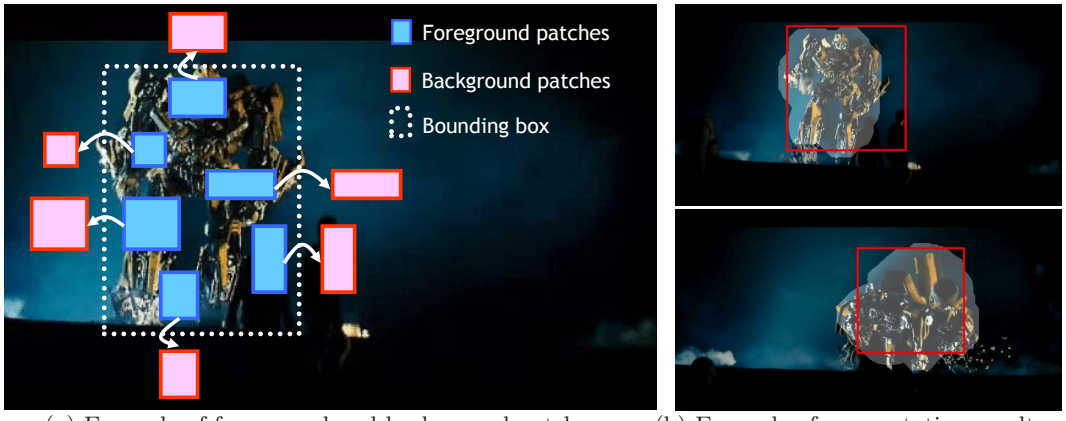
where $\|[O(\mathbf{X}_t^{i(l)}) - \mathbf{X}_t^{c(l)}] - \mathbf{R}_t^i\|_2$ returns the 2-norm distance between two vectors $[O(\mathbf{X}_t^{i(l)}) - \mathbf{X}_t^{c(l)}]$ and $\mathbf{R}_t^i$, and λ_g denotes the weighting parameter set to 1. In (3.23), $[O(\mathbf{X}_t^{i(l)}) - \mathbf{X}_t^{c(l)}]$ is the relative position of the local mode of the proposed i-th local patch with respect to the center of an object. $\mathbf{R}_t^i$ is the reference position of the i-th local patch with respect to the center of an object, which is updated by

$$\mathbf{R}_{t+1}^i = (1-\omega)\mathbf{R}_t^i + \omega\left(O(\hat{\mathbf{X}}_t^i) - \hat{\mathbf{X}}_t^c\right), \tag{3.24}$$

where $\hat{\mathbf{X}}_t^i$ and $\hat{\mathbf{X}}_t^c$ are the MAP states of $\mathbf{X}_t^i$ and $\mathbf{X}_t^c$, respectively. Note that this updating process is for unmodified local patches. For modified local patches, $\mathbf{M}_{t+1}^i$ and $\mathbf{R}_{t+1}^i$ are already determined in the modification process explained in Section 3.2.2.4.

3.2.1.2 Advanced Likelihood with Rough Segmentation

The local patch-based appearance model enables the proposed method to employ a segmentation technique very easily . As aforementioned, the model automatically produces several foreground patches over time. These patches could be good seeds for the segmentation algorithms. Figure 3.19(a) displays the construction of background patches. Since the number of background patches around the bounding box is large, our background model is similar to a conventional rectangular band. With the advantage of the conventional rectangular band, our background model also has good property of preserving the spatial information of background, while the rectangular band loses the spatial information of background.



(a) Example of foreground and background patches (b) Example of segmentation results

Figure 3.19: **Process of segmentation** in *transformer* seq. To construct a background patch, we firstly choose the nearest bounding line to each foreground patch. Then, we select a center position of the background patch outside this bounding line, which is in the perpendicular direction of the line, to place the patch 20 pixels away from the bounding box. The size of a background patch is equal to that of a foreground patch.

With the foreground and background patches, the proposed method segments the target using the random-walk algorithm in [78], as described in Figure 3.19(b). This algorithm finds out the target region probabilistically with a set of foreground patches and a set of background patches. Because the segmentation results are generative, the results are well integrated into our sampling based Bayesian tracking framework. The likelihood function is then enhanced by measuring the additional likelihood value on the segmented region, $p_s(\mathbf{Y}_t|S[O(\mathbf{X}_t^{(l)})])$:

$$p\left(\mathbf{Y}_t|O(\mathbf{X}_t^{(l)})\right) \approx p_s\left(\mathbf{Y}_t|S\left[O(\mathbf{X}_t^{(l)})\right]\right)\prod_{i=1}^{m}\left[p_p\left(\mathbf{Y}_t|O(\mathbf{X}_t^{i(l)})\right)p_s\left(O(\mathbf{X}_t^{i(l)})|\mathbf{X}_t^{c(l)},\mathbf{R}_t^i\right)\right],$$
$$p_s\left(\mathbf{Y}_t|S\left[O(\mathbf{X}_t^{(l)})\right]\right) = exp^{-\lambda_s F_2\left(I\left(S\left[O(\mathbf{X}_t^{i(l)})\right]\right),\mathbf{M}_t\right)}, \tag{3.25}$$

where $S[O(\mathbf{X}_t^{(l)})]$ represents the segmented region obtained using the seeds centered on $O(\mathbf{X}_t^{(l)})$, $\mathbf{M}_t$ indicates the whole model of the target, and λ_s denotes the weighting

parameter set to 5. The whole model $\mathbf{M}_t$ is the image patch inside the bounding box B, which is defined by

$$\begin{aligned} B_w &= \max\left[\{\hat{X}_{t-1}^{i(x)}\}_{i=1}^{m}\right] - \min\left[\{\hat{X}_{t-1}^{i(x)}\}_{i=1}^{m}\right], \\ B_h &= \max\left[\{\hat{X}_{t-1}^{i(y)}\}_{i=1}^{m}\right] - \min\left[\{\hat{X}_{t-1}^{i(y)}\}_{i=1}^{m}\right], \\ B_c &= \left(\min\left[\{\hat{X}_{t-1}^{i(x)}\}_{i=1}^{m}\right] + \frac{B^w}{2}, \min\left[\{\hat{X}_{t-1}^{i(y)}\}_{i=1}^{m}\right] + \frac{B^h}{2}\right), \end{aligned} \tag{3.26}$$

where B_w, B_h, and B_c denote the width, height, and center of the imaginary bounding box B constructed by all local patches $\{\hat{X}_{t-1}^{i}\}_{i=1}^{m}$, respectively, and; $\hat{X}_{t-1}^{i(x)}$ and $\hat{X}_{t-1}^{i(y)}$ indicate the x and y positions of the MAP state of the i-th local patch at time $t-1$, respectively. In (3.25), the function F_2 returns Bhattacharyya similarity [33] between HSV histogram of $S[O(\mathbf{X}_t^{(l)})]$ and $\mathbf{M}_t$. Using (3.25), the proposed method could cover severe scale changes in the target. Note that the segmentation algorithm reconstructs as much regions of the target as possible and provides the global information of the target appearance. Hence, the new likelihood function considers missed regions of the target while measuring the likelihood score, where the missed regions are typically made because of severe scale changes.

3.2.2 Update of the Appearance Model

In this process, the local patches in our appearance model are newly added, deleted or moved to a different position via online update.

3.2.2.1 Initialization of Patches

The initial position of patches has to be chosen to be good for image alignment. Thus, the condition number K of the Hessian Matrix H is used.

$$K = \frac{\sigma_{max}(H)}{\sigma_{min}(H)}, \tag{3.27}$$

(a) Bounding box (b) Good points (c) Chosen patches

Figure 3.20: **Example of patch initialization** in *diving* seq. (b) displays 50 points that have small K and (c) illustrates 15 initialized local patches.

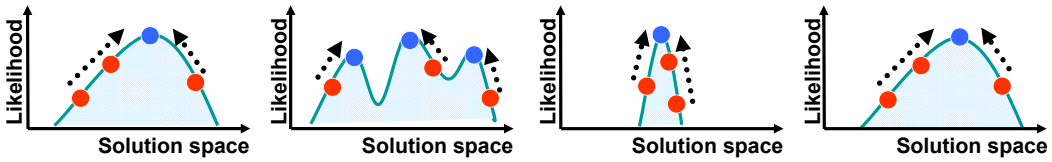
where $\sigma_{max}(H)$ and $\sigma_{min}(H)$ denote the maximal and minimal singular values of H respectively. In (3.27), the small K means that the matrix is numerically stable [2].

To initialize the patches, a bounding box around the target is drawn manually, and the center of the first local patch within the bounding box is chosen as the point with the least K value. The size of the patch is determined randomly. The second patch is chosen as the point with the next least K value. Hence, this patch does not to overlap with the existing local patches. The procedure is repeated and terminated only when there is no space to make local patches or the number of local patches reaches a predefined value. Figure 3.20 shows the patch initialization process.

3.2.2.2 Examination of Patches by LLA

When the *landscape of the local mode (LLM)* of each patch has good properties, the proposed appearance model reliably estimates the likelihood in (3.25), which is important for tracking success. Smoothness and steepness are used to measure good

[2]The Hessian matrix is defined by $H = \sum_x [\nabla I \frac{\partial W}{\partial p}]^T [\nabla I \frac{\partial W}{\partial p}]$ where W is the warp matrix, p is the warping parameter, and ∇I is the image gradient [76]. To update the warping parameter during image alignment, the inverse Hessian matrix H^{-1} must be used. Therefore, numerical stability is important.



(a) Smooth (b) Rough (c) Steep (d) Gradual

Figure 3.21: **Example of the likelihood landscape type** The green curve indicates the likelihood landscape of the local patches. Red circles denote samples of a local patch. Blue circles represent local modes of these samples.

Table 3.5: **The status of local modes.**

Smoothness	Status	Steepness	Status
$D_{sm} \geq \theta_{sm}$	The landscape of local modes is smooth.	$D_{st} \geq \theta_{st}$	The shape of local modes is steep.
$D_{sm} < \theta_{sm}$	The landscape of local modes is rough.	$D_{st} < \theta_{st}$	The shape of local modes is gradual.

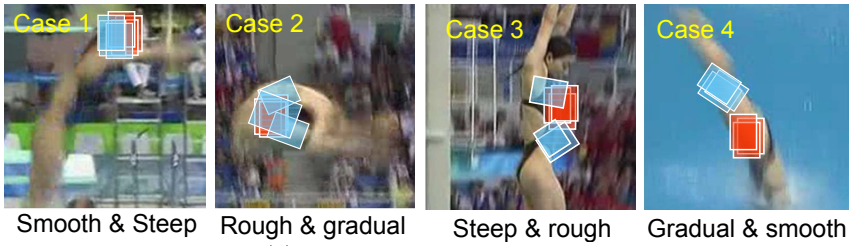
LLM. Smooth (rough) LLM means that local modes are gathered (scattered)in a narrow (wide) region of a solution space, whereas steep (gradual) LLM indicates that these local modes have a steep (gradual) shape. Both smooth and steep LLM guarantee that there is a very strong local mode for the patch. Figure 3.21 shows examples of different LLM types.

To measure these properties quantitatively, a new method of measurement inspired by [10] is designed. The degree of smoothness D_{sm} of the i-th local patch is measured as the variance on the positions of the local modes of the patch:

$$D_{sm}(i) = \frac{1}{\left\| \frac{1}{N} \sum_{l=1}^{N} \left[O(\mathbf{X}_t^{i(l)}) - \frac{1}{N} \sum_{l'=1}^{N} O(\mathbf{X}_t^{i(l')}) \right]^2 \right\|_2}, \tag{3.28}$$

where $O(\cdot)$ finds local modes for the N number of samples of the i-th local patch. The degree of steepness D_{st} of the i-th local patch is measured as the mean of the distance between the positions of samples and of local modes:

$$D_{st}(i) = \frac{1}{\frac{1}{N} \sum_{l=1}^{N} \| O(\mathbf{X}_t^{i(l)}) - \mathbf{X}_t^{i(l)} \|_2}. \tag{3.29}$$



(a) Qualitative results of LLM

Case	D_{sm}	D_{st}	Description
1	20.0	1.85	The local patch has the dominant appearance.
2	0.08	0.19	There are severe background clutters around the patch.
3	0.62	3.20	There are many of similar textures around the patch.
4	5.64	0.12	There exists homogeneous regions around the patch.

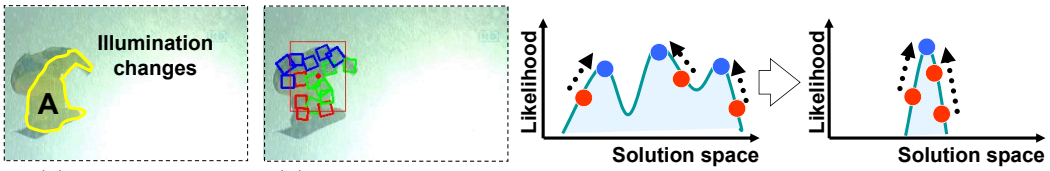
(b) Quantitative results of LLM

Figure 3.22: **Experimental results of the likelihood landscape analysis** Red squares denote samples of a local patch. Blue squares represent the local modes of these samples.

With the new methods of measurement in (3.28) and (3.29), the status of local modes, such as that in Table 3.5, could be determined. Figures 3.22(a) and 3.22(b) respectively show the qualitative and quantitative results of LLM for 4 different cases in *diving* seq. In case 1, the proposed method can track the patch robustly because there are no ambiguous regions around it, whose appearance is similar to that of the patch. However, the method frequently fails to track the patches in cases 2,3, and 4 because of background clutter, similar textures, and homogeneous regions around the patches. Therefore, these patches should be modified. This modification is explained in Section 3.2.2.4.

3.2.2.3 Online Feature Selection via LLA

In the real-world tracking environment, the photometric appearance of a target object also changes severely because of varying illumination conditions, as described in Figure 3.23(a). To track the target robustly in this environment, the proposed



(a) Illumination changes (b) Feature selection results (c) Advantage of the feature selection

Figure 3.23: **Feature selection process** at frame #81 in *snowboard* seq. (a) Region A is under severe the illumination changes. (b) Red, green, and blue squares denote local patches, which take hue, saturation, and value as a feature, respectively. As shown in the figure, the proposed method adaptively chooses a different feature for each local patch. (c) To describe region A, the hue and saturation features are better than the value feature because they make the likelihood landscape smooth and steep.

method uses locally different features to describe the target, as shown in Figure 3.23(b). Thus, some portions of the target are expressed as a feature and other portions as other features. This can be done efficiently by allowing the local patches to have different features. Note that a different feature makes a different LLM. Therefore, the proposed method can choose a robust feature for each patch by analyzing the LLM of the patch. If the chosen feature describes the target's current appearance well, the corresponding LLM of the patch should be smooth and steep. Otherwise the current appearance is rough and gradual, as illustrated in Figure 3.23(c). With this robust feature, the proposed method can cope with local appearance changes of the target efficiently and track it robustly, even with local illumination changes.

The improvement of the LLM of the patches by selecting features in the proposed method can be measured by $S_{LLM} = \frac{1}{m}\sum_{i=1}^{m} D_{sm}(i) + \frac{1}{m}\sum_{i=1}^{m} D_{st}(i)$, where $D_{sm}(i)$ in (3.28) and $D_{st}(i)$ in (3.29) return the degree of smoothness and steepness of the i-th local patch, respectively. In Figure 3.23, S_{LLM} increases from 7.14 to 61.96 by adaptively selecting the *Hue*, *Saturation*, and *Value* features, compared with the

Value feature only.

3.2.2.4 Online Modification of Patches

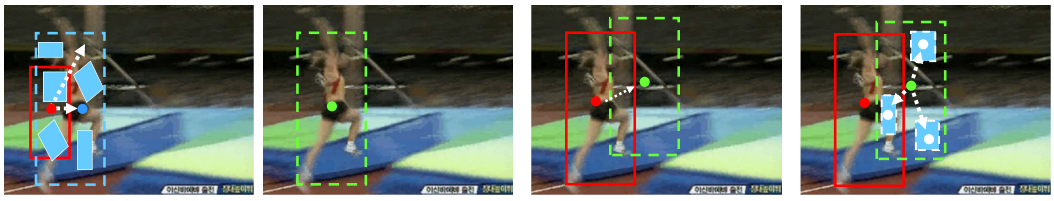
According to the results of the likelihood landscape analysis in Sections 3.2.2.2 and 3.2.2.3, the bad patches can be identified as those with rough or gradual LLM. These bad patches are modified online. Two criteria for modification are provided, such that

• **Criterion 1:** A modified local patch has to be similar to the foreground and dissimilar to the background.

• **Criterion 2:** A modified local patch has to be far from other local patches.

The first criterion prevents local patches from drifting away from an object and into a background. On the other hand, the second criterion allows local patches to cover as different parts of the object as possible. The first criterion is formulated by $\frac{exp^{-\lambda F_2\left(\tilde{\mathbf{x}}_t^i, FM\right)}}{exp^{-\lambda F_2\left(\tilde{\mathbf{x}}_t^i, BM\right)}} \geq \theta_{C1}$, where $\tilde{\mathbf{X}}_t^i$ denotes the modified i-th local patch, F_2 returns Bhattacharyya similarity between two HSV histograms, and λ indicates the weighting parameter. The foreground model FM is constructed by the average of HSV histograms of unmodified local patches and the background model BM is made by the HSV histogram of a background local patch. The process of constructing the background local patch is illustrated in Figure 3.19(a). The second criterion is formulated by $\|\tilde{\mathbf{X}}_t^i - \mathbf{X}_t^j\|_2 \geq \theta_{C2}$, for $\forall j$ and $j \neq i$.

When the above two criteria are satisfied, modifications are performed by adding new patches, or by deleting or moving bad patches. First, the proposed method makes several attempts to find a patch that satisfies the above criteria, whereby a bad patch is moved via the Gaussian perturbation centered on the current position. If the moved patch satisfies the above criteria within the predefined number of



(a) Proposal Q_1 (b) Proposal Q_2 (c) Proposal Q_3

Figure 3.24: **Process of the proposal step** in *high-jump* seq. (a) At the start of frame t, our method proposes a new center and scale of the object (green circle and dotted green rectangle) based on the positions of local patches (blue rectangles) at frame $t-1$ using Q_1 in (3.30). In the example, a new center is proposed in the right direction because the centroid of the local patches (blue circle) is located to the right of the object center (red circle). A new scale is proposed in the growing direction because the imaginary bounding box constructed by the local patches (dotted blue rectangle) is bigger than the current bounding box (red rectangle). (b) Within frame t, a new center and scale of the object (green circle and dotted green rectangle) are sampled by Q_2 in (3.32). (c) After proposing a new center and scale of the object, a new center of each local patch (white circle) is determined by Q_3 in (3.33).

iterations, the bad patch is replaced with the moved patch. If not, the bad patch is deleted. Second, a new patch that satisfies the aforementioned criteria is added by choosing a new position for the patch using the condition number explained in Section 3.2.2.1. If the new patch satisfies the aforementioned criteria within half the number of predefined iterations, the new patch is added.

3.2.3 Inference via the ABHMC sampling

The solution space generally becomes large as the number of local patches in our appearance model increase. Thus, the conventional MCMC method is inefficient for computing the integration in (1.7). Therefore the BH sampling method [79] is introduced in the tracking problem to provide a better performance in such high-dimensional solution spaces. The BH sampling method consists of two main steps,

similar to the conventional MCMC method, namely, the proposal and acceptance steps.

• **Proposal Step:** For the proposal step, three different proposal densities are used, namely, Q_1, Q_2, and Q_3, as illustrated in Figure 3.24. The proposal density Q_1 is used once at the start of each frame to connect the previous frame to the current one. In Q_1, the center of an object is assumed to be near the centroid formed by all local patches. And the scale of the object should be determined; thus, the bounding box contains all local patches compactly. With these assumptions, the proposal density Q_1 is designed, whose proposal variance changes adaptively according to the states of local patches. Then the adaptive proposal is

$$\begin{aligned} Q_1(X_t^{x(1)}; \hat{X}_{t-1}^x) &= G(\hat{X}_{t-1}^x, \sigma_x^2) + \gamma_x \delta_x, \\ Q_1(X_t^{y(1)}; \hat{X}_{t-1}^y) &= G(\hat{X}_{t-1}^y, \sigma_y^2) + \gamma_y \delta_y, \\ Q_1(X_t^{s(1)} \hat{X}_{t-1}^s) &= G(\hat{X}_{t-1}^s, \sigma_s^2) + \gamma_s \delta_s, \end{aligned} \tag{3.30}$$

where $Q_1(X_t^{x(1)}; \hat{X}_{t-1}^x)$ and $Q_1(X_t^{y(1)}; \hat{X}_{t-1}^y)$ indicate that the first sample of the object center $(\mathbf{X}_t^{x(1)}, \mathbf{X}_t^{y(1)})$ at the current frame is proposed based on the MAP state of the object center $(\hat{X}_{t-1}^x, \hat{X}_{t-1}^y)$ at the previous frame. In (3.30), δ_x, δ_y, and δ_s denote the adapting constant set to 0.3, 0.3, and 0.01, respectively, and γ_x, γ_y,

and γ_s represent the adapting parameters defined by

$$\gamma_x = \begin{cases} \gamma_x + 1 & \text{if } \hat{X}^x_{t-1} \ll \frac{1}{m}\sum_{i=1}^m \hat{\mathbf{X}}^{i(x)}_{t-1} \\ \gamma_x - 1 & \text{if } \hat{X}^x_{t-1} \gg \frac{1}{m}\sum_{i=1}^m \hat{\mathbf{X}}^{i(x)}_{t-1} \\ \gamma_x & \text{otherwise .} \end{cases}, \gamma_y = \begin{cases} \gamma_y + 1 & \text{if } \hat{X}^y_{t-1} \ll \frac{1}{m}\sum_{i=1}^m \hat{\mathbf{X}}^{i(y)}_{t-1} \\ \gamma_y - 1 & \text{if } \hat{X}^y_{t-1} \gg \frac{1}{m}\sum_{i=1}^m \hat{\mathbf{X}}^{i(y)}_{t-1} \\ \gamma_y & \text{otherwise .} \end{cases}.$$

$$\gamma_s = \begin{cases} \gamma_s + 1 & \text{if } \hat{X}^s_{t-1} \ll \frac{\sqrt{B_w^2+B_h^2}}{B_0} \\ \gamma_s - 1 & \text{if } \hat{X}^s_{t-1} \gg \frac{\sqrt{B_w^2+B_h^2}}{B_0} \\ \gamma_s & \text{otherwise .} \end{cases}, \tag{3.31}$$

where B_w and B_h respectively denote the width and height of the bounding box B defined by (3.26) and B_0 denotes the initial diagonal size of the bounding box of the target. In (3.31), γ_x, γ_y, and γ_s initially have zero value and ranges from -5 to 5.

Within each frame, the proposal density Q_2 proposes a new sample of the object center $\mathbf{X}_t^{c(l+1)}$ and scale $X_t^{s(l+1)}$, given the previous sample $\mathbf{X}_t^{c(l)}$ and $X_t^{s(l)}$, respectively, via Gaussian perturbation:

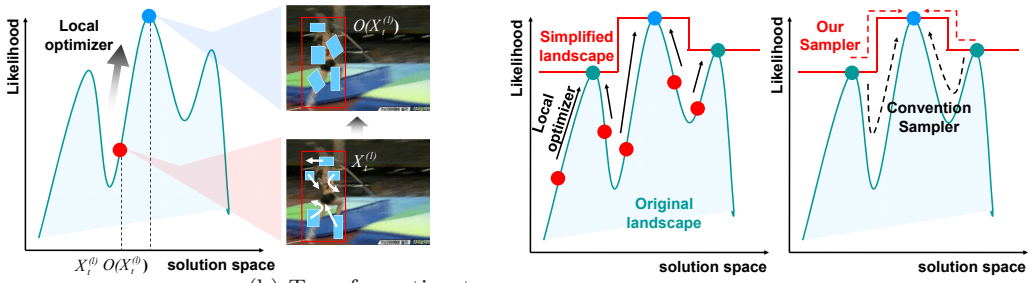
$$Q_2(\mathbf{X}_t^{c(l+1)}; \mathbf{X}_t^{c(l)}) = G(\mathbf{X}_t^{c(l)}, \Sigma_c^2),\ Q_2(X_t^{s(l+1)}; X_t^{s(l)}) = G(X_t^{s(l)}, \sigma_s^2), \tag{3.32}$$

where G denotes the Gaussian distribution with mean $\mathbf{X}_t^{c(l)}$ and variance Σ_c^2 to propose the new center, as well as mean $X_t^{s(l)}$ and variance σ_s^2 to propose the new scale.

The center position of each local patch is determined using the proposal density Q_3:

$$Q_3(\mathbf{X}_t^{i(l+1)}; \mathbf{R}_t^i) = \mathbf{X}_t^{c(l+1)} + \mathbf{R}_t^i, \quad for \ \ i = 1, \ldots, m, \tag{3.33}$$

where $\mathbf{X}_t^{i(l+1)}$ is a new sample of the center position for the i-th local patch, $\mathbf{X}_t^{c(l+1)}$ is a new sample of the object center obtained by (3.32), and $\mathbf{R}_t^i$ is the parameter estimated by (3.24).



(a) State of the local mode $O(\mathbf{X}_t^{(l)})$ (b) Transformation to the simplified landscape (c) Advantage of the simplified landscape

Figure 3.25: **Process of the acceptance step** in *high-jump* seq. (a) A state (red circle) is moved to the state of the local mode (blue circle) using a local optimizer. The states of the local modes represent the states of the local patches changed via affine transformation (blue rectangles). (b) After all states (red circles) proposed by Q in (3.30),(3.32), and (3.33) are moved to the states of local modes (blue and green circles), the original landscape (green curve) is transformed into a simpler one (red line). (c) In the simplified landscape, our sampler can easily reach the global optimum (blue circle) from the local optima (green circles) via the shorter path (dotted red arrows). On the other hand, the conventional sampler has difficulty reaching the global optimum because the longer path (dotted black arrows) contains the down direction.

• **Acceptance Step:** Most performance in the BH sampling method comes from the novel acceptance step. The primary difference between the conventional acceptance ratio in (1.4) and that of the BH sampling method is that the acceptance ratio of the BH sampling method is calculated by the likelihood ratio at the *local mode* of the state. Thus, the acceptance ratio is defined by

$$a = min\left[1, \frac{p\left(\mathbf{Y}_t|O(\mathbf{X}_t^{(l+1)})\right) Q(\mathbf{X}_t^{(l)}; \mathbf{X}_t^{(l+1)})}{p\left(\mathbf{Y}_t|O(\mathbf{X}_t^{(l)})\right) Q\left(\mathbf{X}_t^{(l+1)}; \mathbf{X}_t^{(l)}\right)}\right], \tag{3.34}$$

where $Q(\mathbf{X}_t^{(l+1)}; \mathbf{X}_t^{(l)})$ represents the proposal density defined in (3.30)(3.32)(3.33), and $p(\mathbf{Y}_t|O(\mathbf{X}_t^{(l)}))$ in (3.25) returns the likelihood value at the state of the local mode. The state of the local mode is easily found by the mode-seeking method such

Algorithm 2 The ABHMC-FS

Input: $\mathbf{X}_{t-1} = (\mathbf{X}^c_{t-1}, X^s_{t-1}, \mathbf{X}^1_{t-1}, \cdots, \mathbf{X}^i_{t-1}, \cdots, \mathbf{X}^m_{t-1})$

Output: $\hat{\mathbf{X}}_t = (\hat{\mathbf{X}}^c_t, \hat{X}^s_t, \hat{\mathbf{X}}^1_t, \cdots, \hat{\mathbf{X}}^i_t, \cdots, \hat{\mathbf{X}}^m_t)$

1: Initialize patches using (3.27) in an initial frame.
2: Propose $\mathbf{X}^{c(1)}_t = (X^{x(1)}_t, X^{y(1)}_t)$ and $X^{s(1)}_t$ using Q_1 in (3.30).
3: **for** $l = 1$ to $N - 1$ **do**
4: Propose $\mathbf{X}^{c(l+1)}_t$ and $X^{s(l+1)}_t$ using Q_2 in (3.32).
5: Determine $\mathbf{X}^{i(l+1)}_t$ for all i using Q_3 in (3.33).
6: Obtain $S(\mathbf{X}^{(l+1)}_t)$ with the process described in Section 3.2.1.2.
7: Calculate the likelihood score using (3.25).
8: Accept $\mathbf{X}^{(l+1)}_t$ with probability (3.34).
9: **end for**
10: Estimate the MAP state $\hat{\mathbf{X}}_t$ using (1.7).
11: Select patches to be modified using (3.28) and (3.29).
12: Choose features of the patches using the method described in Section 3.2.2.3.
13: Modify the patches using the criterion 1 and 2 introduced in Section 3.2.2.4.
14: Obtain updated parameters $\mathbf{M}^i_{t+1}$ and $\mathbf{R}^i_{t+1}$ using (3.22) and (3.24).

as the Lucas-Kanade image registration method [76], as shown in Figure 3.25(a).

The BH sampling method transforms the rough likelihood landscape of the original solution space into a simpler one using robust local optimization techniques in the sampling process, as depicted in Figure 3.25(b). In a new transformed landscape, the minima of the original landscape are no longer of concern in the sampling process. Hence, a greater chance exists for reaching the global optimum with a smaller number of samples. In all experiments, 20 samples are sufficient to obtain the MAP estimate. Figure 3.25(c) illustrates the advantage of our simplified landscape.

The proposed method robustly tracks a highly non-rigid target with the advanced

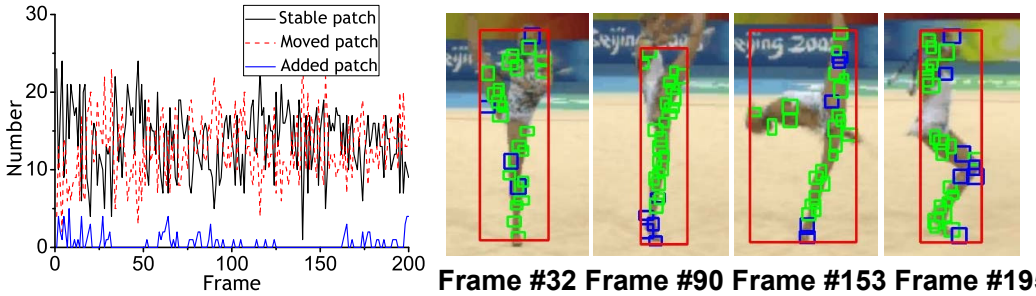
appearance model using the topology between local patches, new online-updating scheme using the likelihood landscape analysis, and the efficient inference method using the Basin Hopping sampling. Algorithm 2 illustrates the whole process of our tracking method, including local patch-based dynamic appearance modeling and adaptive Basin Hopping Monte Carlo sampling.

3.2.4 Experimental Results

17 video sequences were tested, namely, *snowboard*, *diving*, *high-jump*, *transformer*, and *gymnastic* sequences in [24]; *femaleskater*, *maleskater*, *indiandancer*, and *dancer* sequences in [46]; *dinosaur* and *hand2* sequences in [44]; *motocross2* and *skiing* sequences in [45]; *coke*, *girl*, *tiger1*, and *tiger2*, sequences in [6, 50, 80]. For comparative evaluation, the proposed algorithms (ABHMC, ABHMC-F, ABHMC-FS) are compared with 8 different tracking methods, namely, Mean-Shift tracker (MS) based on [19, 81], Standard MCMC (MCMC) based on [18], Incremental learning for Visual Tracking (IVT) in [1], Fragment-based tracker (FRAGT) in [56], Block Histogram-based Tracker (BHT) in [46], Multiple Instance Learning tracker (MIL) in [6], Local Global Tracker (LGT) in [44], and Hough-based Tracker (HT) in [45].

ABHMC denotes our original method in [24]. ABHMC-F improves the ABHMC with adaptive feature selection. ABHMC-FS is the final version, which also utilizes rough segmentation results. **All parameters of the proposed method are fixed for all experiments.** λ_p in (3.21), λ_g in (3.23), ω in (3.22)(3.24), λ_s in (3.25), θ_{sm}, θ_{st} in Table 3.5, θ_{C1}, θ_{C1} in Section 3.2.2.4, and M are set to 30, 1.0, 0.5, 5.0, 1.0, 0.25, 2.5, 5.0, and 50, respectively. Although the parameters were determined empirically, the proposed method was not sensitive to all these parameters,

MCMC uses an HSV color histogram for the appearance model, as in [33], while



(a) Number of stable, moved and newly added patches

(b) Geometric appearance of the object when the number of moved patches peak in (a)

Figure 3.26: **Number of stable, moved, and newly added patches** in *gymnastics* seq. (b) Among 27 local patches, 23 patches are moved at frame #32, 22 at frame #90, 22 at frame #153 and, 20 at frame #195, where green squares denote the moved patches and blue squares denote the stable ones.

dividing an object into the upper and lower body. Proposal variances of the MCMC are set to 8 and 4 for the x and y directions, respectively. The parameters of other trackers are adjusted to produce the best tracking performance. Note that the software used for the methods was provided by the authors in [6, 1, 44, 45, 56, 46]. We obtained ground truth from authors for the publicly available datasets and manually made ground truth for our datasets.

3.2.4.1 Efficiency of the Proposed Appearance Model

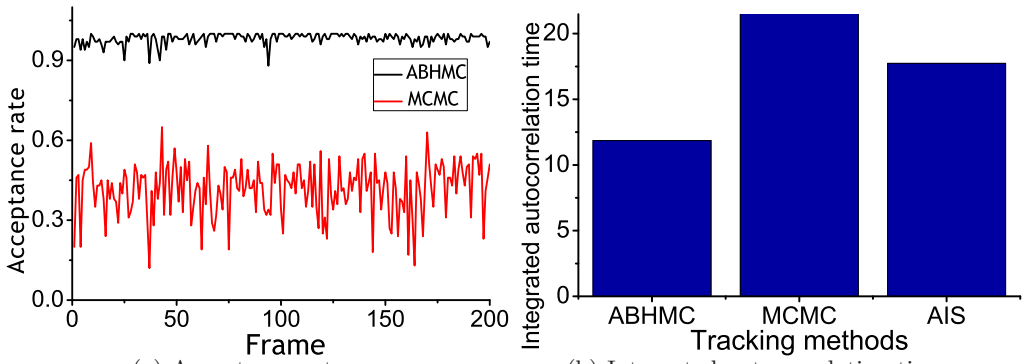
To evaluate the performance of the dynamic appearance modeling scheme qualitatively, the number of modified and unmodified local patches in each frame were verified. As illustrated in Figure 3.26(a), our appearance model actively moves, deletes, or adds patches based on the likelihood landscape analysis at each frame. This means that the topology between the local patches in the model evolves as time goes on. Figure 3.26(b) shows that the proposed appearance model adaptively modifies the position and number of patches, particularly when geometric appearance

Table 3.6: **Likelihood landscape analysis of the proposed appearance model. A step**: Local patch-based appearance modeling step, **B step**: Online updating step after A step, **C step**: Feature selecting step after B step. **C* step**: Other features (Gabor filters) were used for the step C. The numbers indicate S_{LLM} in Section 3.2.2.3. Larger S_{LLM} indicates better likelihood landscape.

Step \ Seq.	*diving*	*high-jump*	*gymnastics*	*transformer*	*Snowboard*
A	31.98	21.54	98.65	11.23	9.71
B	44.74	87.34	101.43	12.28	47.11
C	**76.94**	**150.675**	**202.43**	**35.34**	**85.03**
C*	45.23	36.22	150.32	17.98	36.92

of the object is drastically changing. The proposed method successfully captures the movements of the head, legs, and arms without a specific model for the target object.

We further evaluated the efficiency of the dynamic appearance modeling scheme using the likelihood landscape analysis. Table 3.6 demonstrates how our dynamic appearance modeling scheme enhances the LLM of the patches in the appearance model. As aforementioned in Section 3.2.2.3, the robustness of the proposed appearance model can be measured as S_{LLM}. For example, the model is evaluated as a larger value S_{LLM} if it is composed of good local patches with smoother and steeper LLMs. To obtain good local patches, our dynamic appearance modeling scheme adopts the online update step in Section 3.2.2.4 and the feature selection step in Section 3.2.2.3. After the online update and feature selection steps, the LLM of the patches in the appearance model have become very smooth and steep, indicating that the model discriminates the object from the background well. Therefore, with this model, our tracker tracks the object robustly despite severe photometric and geometric appearance changes. Instead of color features, other features such as Ga-



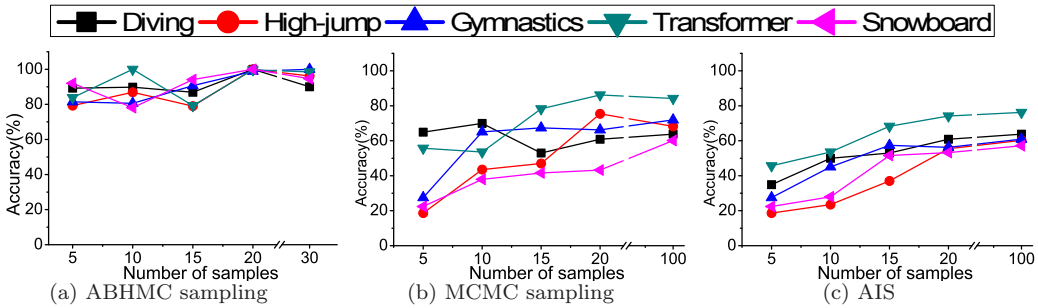
(a) Acceptance rate (b) Integrated autocorrelation time

Figure 3.27: **Property of the adaptive Basin Hopping Monte Carlo sampling** in *car4* seq. in [1] (a) Acceptance rate is defined by the number of accepted samples over the total number of samples in each frame. (b) To derive τ_{int}, each method used 500 samples. AIS denotes the Annealed Importance Sampling method in [2].

bor filters could be utilized in the C step. However, the Gabor filters made the LLM of the patches have bad properties, in which the LLM of the patches became very rough and gradual, as demonstrated by the C* step in Table 3.6. This means that the sampling method using Gabor filters should spend long time to find the global optimum as comparison with the method using color features. Hence, with the limited number of samples, the sampling method using Gabor filters has difficulty in reaching the global optimum.

3.2.4.2 Efficiency of the Proposed ABHMC Sampling

• **Property of Sampling Strategy:** To evaluate the performance of the ABHMC sampling in the proposed tracking algorithm more analytically and qualitatively, it was compared with the standard MCMC-based tracking algorithm [18]. In this experiment, the test video only contained a rigid object for fair comparison. An equal number of samples, equal appearance model, and equal transition model for both methods were set. Note that One particularly good property of the ABHMC



(a) ABHMC sampling (b) MCMC sampling (c) AIS

Figure 3.28: **Efficiency of the adaptive Basin Hopping Monte Carlo sampling.** Figures (a) and (b) illustrate the tracking accuracy as the number of samples increases. In the experiments, the tracking accuracy was obtained by $\frac{\text{Pascal score using the current number of samples}}{\text{Best Pascal score}} * 100$. AIS denotes the Annealed Importance Sampling method in [2].

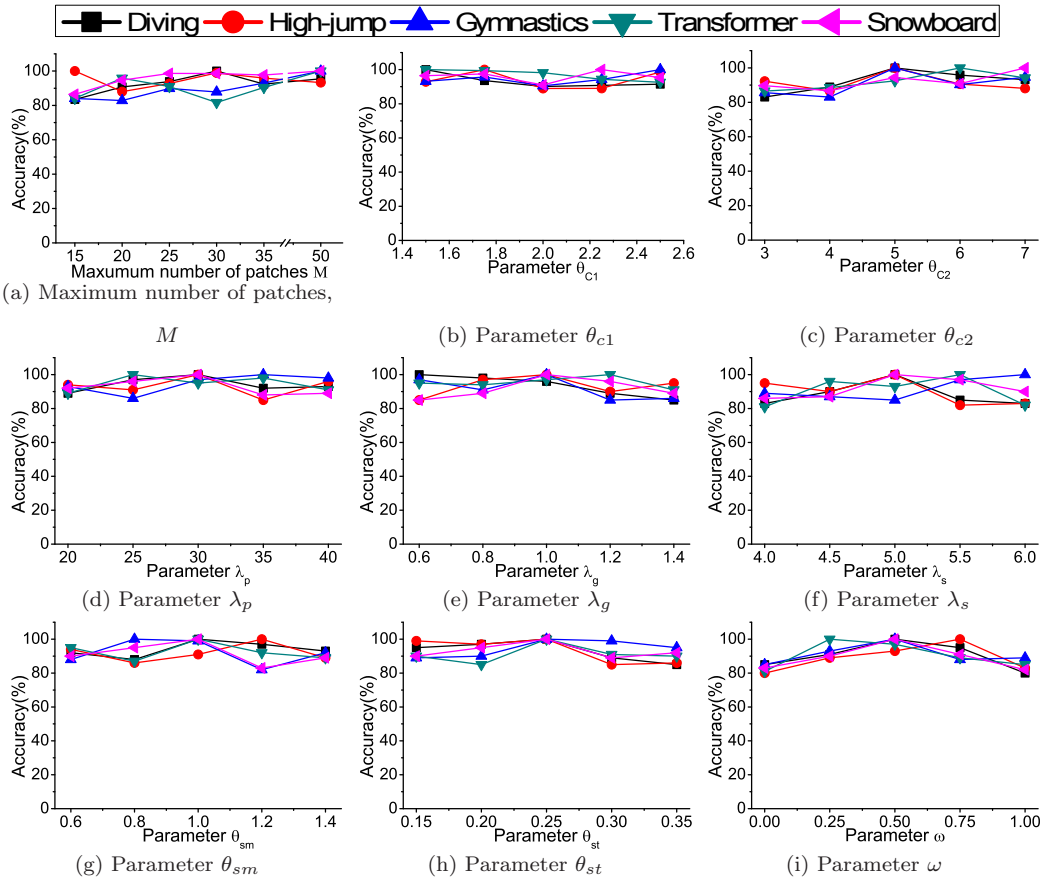
sampling is that it can easily jump over a deep basin by transforming a likelihood landscape into a simpler one. Thus, the depth of basins are lowered, and the ABHMC sampling can frequently accept the proposed samples. As shown in Figure 3.27(a), our tracking algorithm had higher acceptance rates than the standard MCMC method. This means that the proposed method easily escapes from the local optima and obtains more diverse samples.

Autocorrelation time measures the degree of statistical independence between samples [82]. This independence property is important in reducing the statistical error. If the samples are highly correlated, the statistical error does not decrease at the rate of the square root of the number of samples. Figure 3.27(b) and (c) illustrate the integrated autocorrelation time τ_{int}, where τ_{int} of the ABHMC sampling was smaller than that of the MCMC sampling and the AIS. This finding suggests that the ABHMC sampling method produces highly uncorrelated samples, which sufficiently minimizes the statistical error of the MAP estimate in (1.2).

• **Efficiency of Sampling Strategy:** The appearance model generally consists of 20 to 50 local patches, indicating a very large solution space. The proposed

tracking method, however, uses a very small number of samples, 20, in all experiments for tracking an object. This performance typically benefits from the ABHMC sampling. Figure 3.28 quantitatively demonstrates that the ABHMC sampling requires a smaller number of samples, compared with the MCMC sampling and the AIS method, to reach a similar tracking performance. For example, the ABHMC sampling needed only 5 samples to obtain the tracking accuracy of 0.7 in *Gymnastics* seq., whereas the MCMC needed more than 100 samples. Additionally, the figure shows that the ABHMC sampling maintains the best performance even with a drastically small number of samples. In all sequences, the ABHMC sampling produced the most accurate results, regardless of the number of samples. The main advantage of the ABHMC sampling is that it combines the stochastic method with the deterministic method. Hence, it has good properties from both the stochastic and deterministic methods. Using the deterministic method, the ABHMC sampling quickly finds the several local minima and thus needs the small number of samples to get the solution. The stochastic method prevents the method from getting trapped in a certain local minimum. On the other hand, the MCMC sampling and the AIS method require an enough number of samples to obtain the good solution because it only employs the stochastic process.

In the theoretical aspect, the simulated annealing method and its variants such as the AIS method suffer from the notorious "freezing" problem, as reported in [83]. The freezing problem occurs because the escape rate from local minima diverges with decreasing temperature. The ABHMC sampling ameliorates this problem and simplifies the original likelihood landscape by replacing the likelihood of each conformation with the likelihood of a nearby local minima. This replacement eliminates high likelihood barriers in the stochastic search that are responsible for the freezing



(a) Maximum number of patches, M (b) Parameter θ_{c1} (c) Parameter θ_{c2}

(d) Parameter λ_p (e) Parameter λ_g (f) Parameter λ_s

(g) Parameter θ_{sm} (h) Parameter θ_{st} (i) Parameter ω

Figure 3.29: **Experiments on the parameter settings.** In the experiments, the tracking accuracy was obtained by $\frac{\text{Pascal score using current parameters}}{\text{Best Pascal score}} * 100$.

problem in simulated annealing, as reported in [84].

3.2.4.3 Accuracy of the Proposed Tracking Methods

• **Effect of Parameter Settings:** Tracking accuracy could be dependent on several parameters. To evaluate the robustness of the tracking system, how much of their results are affected by the parameters must be checked. As shown in Figures 3.29(a) to 3.29(i), the proposed method was not sensitive to all these parameters. Figure 3.29(a) illustrates the tracking accuracy as the maximum number

of local patches increases. If this value is increased, the local patches can cover a larger portion of the object region. However, the risk that the local patches may cover background regions increases. Figure 3.29(b) shows the tracking accuracy as the parameter θ_{C1} increases. The parameter θ_{C1} is the threshold value of $\frac{likelihood\ over\ foreground}{likelihood\ over\ background}$, which is described in Section 3.2.2.4. If θ_{C1} is increased, the method reduces the chances of creating a false-positive of the local patches. However, the chances of creating a true-positive are also reduced. Figure 3.29(c) describes the tracking accuracy as the parameter θ_{C2} increases. The parameter θ_{C2} is the threshold value of the distance between neighbor patches, which is also described in Section 3.2.2.4. If θ_{C2} is increased, the ability to explore a missed object region is improved. However, the ability of exploiting the fine object region is decreased. Figure 14(d) to 14(f) show the tracking accuracy as likelihood parameters, λ_p, λ_g, and λ_s, increase. The parameters, λ_p, λ_g, and λ_s adjust weights of photometric, geometric, and segmentation factors in the likelihood function, respectively. The likelihood models have the different weight by assigning different values to λ_p, λ_g, and λ_s in (3.20) and (3.25). During the course of tracking, these weights could change depending on the tracking environment. However, these weight are fixed to simplify the algorithm of the proposed method. This is feasible because the weights does not significantly affect the tracking performance, as demonstrated in Figure 14(d) to 14(f). Figure 14(g) and 14(h) illustrate the tracking accuracy as LLM parameters, θ_{sm} and θ_{st}, increase. The parameter θ_{sm} is the threshold value to determine smoothness of LLM. The parameter θ_{st} is the threshold value to determine steepness of LLM. If θ_{sm} and θ_{st} increase, local patches are more frequently updated because LLM of local patches are considered as bad (rough and gradual) ones. Hence, the proposed method can cover drastic appearance changes of the targets. However, the method

has difficult in handling gradual appearance changes of the targets. Figure 14(i) describes the tracking accuracy as the appearance updating parameter ω increases. The parameter ω adjusts weights of old and new appearance of the targets to construct the appearance model. If ω increases, the appearance model is constructed by reflecting a current appearance rather than an old appearance of the targets. Hence, the tracking method accurately tracks the targets although the appearance frequently changes. However, the constructed appearance model is easily affected by noisy appearance because current appearance may be obtained from erroneous tracking results.

• **Comparison within the proposed methods:** Using multiple features (ABHMC-F) and additional segmentation results (ABHMC-FS), we enhanced the performance of the ABHMC tracker, as shown in Table 3.7. The ABHMC-FS tracker always outperformed the ABHMC and ABHMC-F trackers. Note that the gray test sequences, including *dancer*, *femaleskater*, *indiandancer*, and *maleskater* do not provide multiple (color) features. Hence, the results of ABHMC-F for these sequences are unavailable.

As shown in Table 3.7, most important steps which contribute to the final tracking performance are the online updating step and the segmentation step. These results demonstrate that our original ABHMC tracker proposed in [24] is robust because it includes the online updating step and our ABHMC-FS tracker is more robust because it includes the segmentation step as well. According to the experiment, to track the highly non-rigid object accurately, the local patch-based appearance model should be modified via online update like the ABHMC tracker. And the appearance model should include both the local and global appearance of the target like the ABHMC-FS tracker.

Table 3.7: **Quantitative analysis of individual component within the proposed method.** The numbers indicate tracking accuracy, which are evaluated by the Pascal score [3]. The Pascal score is defined by the overlap ratio between the predicted bounding box B_p and ground truth bounding box B_{gt}: $\frac{area(B_p \bigcap B_{gt})}{area(B_p \bigcup B_{gt})}$. **A step**: Base-line step (**MCMC**, **B step**: Local patch-based appearance modeling step after A step (Section 3.2.1.1), **C step**: Online updating step after B step (**ABHMC**, Section 3.2.2.1, 3.2.2.2, and 3.2.2.4), **D step**: Feature selecting step after C step (**ABHMC-F**, Section 3.2.2.3), **E step**: Segmentation step after D step (**ABHMC-FS**, Section 3.2.1.2)

Seq. \ Step	A step	B step	C step	D step	E step
diving	0.32	0.37	0.47	0.58	**0.64**
high-jump	0.33	0.35	0.37	0.41	**0.51**
gymnastics	0.39	0.45	0.61	0.65	**0.71**
transformer	0.49	0.42	0.59	0.63	**0.72**
snowboard	0.16	0.12	0.14	0.43	**0.58**
dancer	0.45	0.47	0.55	N/A	**0.58**
femaleskater	0.51	0.49	0.50	N/A	**0.58**
indiandancer	0.41	0.44	0.52	N/A	**0.59**
maleskater	0.40	0.43	0.49	N/A	**0.56**
coke	0.08	0.17	0.21	N/A	**0.35**
dinosaur	0.23	0.26	0.33	0.37	**0.63**
girl	0.61	0.35	0.41	0.42	**0.52**
hand2	0.19	0.40	0.49	0.52	**0.76**
motocross2	0.47	0.53	0.60	0.61	**0.72**
skiing	0.03	0.21	0.30	0.30	**0.55**
tiger1	0.21	0.26	0.37	N/A	**0.69**
tiger2	0.17	0.20	0.28	N/A	**0.62**
Average	0.32	0.35	0.43	0.49	**0.61**
Improvement (%)	0	+5	+13	+10	+20
Time (sec/frame)	0.04	+0.01	+0.04	+0.01	+1.90

Table 3.7 also shows the computation time of each individual component in the ABHMC-FS tracker. The tracker approximately takes 2 seconds per frame using the Pentium 4 quad core 2.4GHz CPU. It is notable that our ABHMC tracker without the segmentation procedure (before the E step in Table 3.7) is real time. It approximately takes 0.1 seconds per frame. In spite of such computational overhead, the segmentation procedure in the ABHMC-FS tracker is very useful because it greatly improves the tracking accuracy.

Table 3.8: **Quantitative comparison with other methods.** The numbers indicate mean and standard deviation of tracking accuracy, which are evaluated by the Pascal score [3]. These numbers were obtained by running each algorithm 5 times. The Pascal score is defined by the overlap ratio between the predicted bounding box B_p and ground truth bounding box B_{gt}: $\frac{area(B_p \bigcap B_{gt})}{area(B_p \bigcup B_{gt})}$. The red mark represents the best results, whereas the blue mark represents the second-best results. All parameters of the proposed method (ABHMC-FS) are fixed for this experiment.

Seq. \ Method	MS	MCMC	IVT	FRAGT	BHT	MIL	LGT	HT	ABHMC-FS
diving	0.41	0.32 (0.09)	0.18 (0.08)	0.21	0.17 (0.08)	0.20 (0.11)	**0.48** (0.11)	0.41 (0.10)	**0.64** (0.08)
high-jump	0.35	0.33 (0.09)	0.07 (0.11)	0.07	0.11 (0.10)	0.06 (0.11)	**0.40** (0.08)	0.38 (0.09)	**0.51** (0.10)
gymnastics	0.45	0.39 (0.11)	0.36 (0.13)	0.43	**0.56** (0.12)	0.33 (0.15)	0.44 (0.11)	0.47 (0.10)	**0.71** (0.09)
transformer	0.45	0.49 (0.15)	0.33 (0.11)	0.46	0.50 (0.11)	0.37 (0.19)	0.24 (0.17)	**0.51** (0.11)	**0.72** (0.12)
snowboard	0.10	0.16 (0.25)	0.16 (0.23)	0.15	0.15 (0.15)	0.15 (0.25)	0.18 (0.19)	**0.20** (0.27)	**0.58** (0.21)
dancer	0.39	0.45 (0.15)	**0.63** (0.13)	**0.61**	0.53 (0.13)	0.55 (0.13)	0.40 (0.13)	0.53 (0.15)	0.58 (0.13)
femaleskater	0.55	0.51 (0.13)	0.48 (0.12)	**0.57**	0.56 (0.11)	0.54 (0.10)	0.52 (0.12)	0.46 (0.10)	**0.58** (0.11)
indiandancer	0.37	0.41 (0.12)	0.64 (0.13)	**0.68**	0.63 (0.13)	**0.69** (0.10)	0.50 (0.15)	0.57 (0.13)	0.59 (0.12)
maleskater	0.39	0.40 (0.14)	0.13 (0.15)	0.55	**0.56** (0.09)	0.24 (0.09)	0.40 (0.14)	0.28 (0.12)	**0.56** (0.10)
coke	0.17	0.08 (0.17)	0.16 (0.14)	0.07	0.07 (0.12)	**0.35** (0.11)	0.24 (0.10)	0.30 (0.09)	**0.34** (0.09)
dinosaur	0.39	0.23 (0.20)	0.23 (0.13)	0.16	0.21 (0.14)	0.42 (0.15)	**0.46** (0.15)	0.23 (0.15)	**0.63** (0.13)
girl	0.50	**0.61** (0.15)	0.03 (0.11)	**0.63**	0.51 (0.13)	0.56 (0.14)	0.06 (0.11)	0.52 (0.13)	0.52 (0.15)
hand2	0.15	0.19 (0.12)	0.13 (0.15)	0.20	0.30 (0.08)	0.11 (0.12)	**0.65** (0.13)	0.60 (0.11)	**0.76** (0.09)
motocross2	0.46	0.47 (0.15)	0.43 (0.11)	0.45	0.03 (0.10)	0.59 (0.09)	0.50 (0.09)	**0.78** (0.10)	**0.72** (0.11)
skiing	0.10	0.03 (0.07)	0.05 (0.06)	0.05	0.20 (0.05)	0.04 (0.10)	0.04 (0.07)	**0.53** (0.06)	**0.55** (0.05)
tiger1	0.25	0.21 (0.12)	0.10 (0.09)	0.17	0.22 (0.08)	**0.64** (0.07)	0.12 (0.09)	0.40 (0.10)	**0.69** (0.08)
tiger2	0.28	0.17 (0.11)	0.08 (0.08)	0.20	0.13 (0.10)	**0.65** (0.10)	0.22 (0.09)	0.35 (0.09)	**0.62** (0.09)

• **Comparison with other tracking methods:** Table 3.8 summarizes the tracking results of nine different test sequences that include objects whose geometric appearances are changing drastically over time. The proposed ABHMC-FS tracker most accurately tracked the objects with the fixed parameters. The LGT and HT trackers also robustly tracked the targets and showed the second-best performance, where they are most recent tracking methods especially for highly non-rigid objects. However, these trackers were weak to severe illumination changes in *snowboard* and *dinosaur* sequences. BHT and FRAGT showed the good tracking performance. Both, however, were weak when the geometric appearance changes is more severe. As comparison with these methods, ABHMC-FS produced accurate tracking results even though there are illumination changes and deformation of targets at the same time. With the online-feature selection process, ABHMC-FS constructed an good appearance model using the selected local patches, which are robust to both the il-

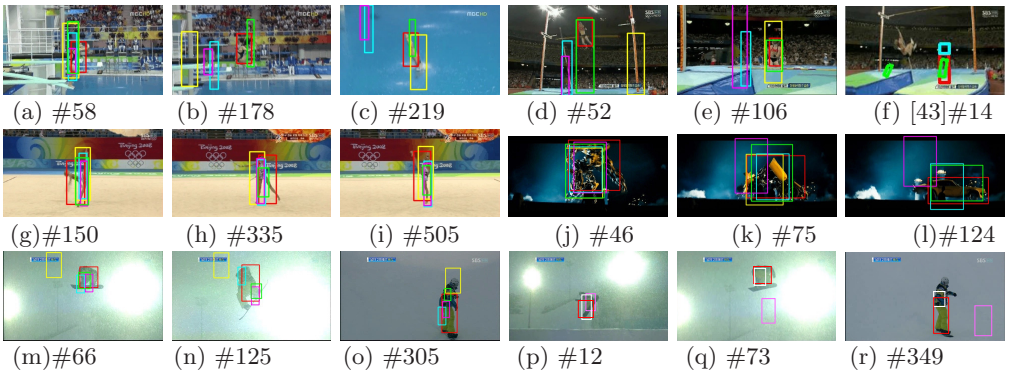


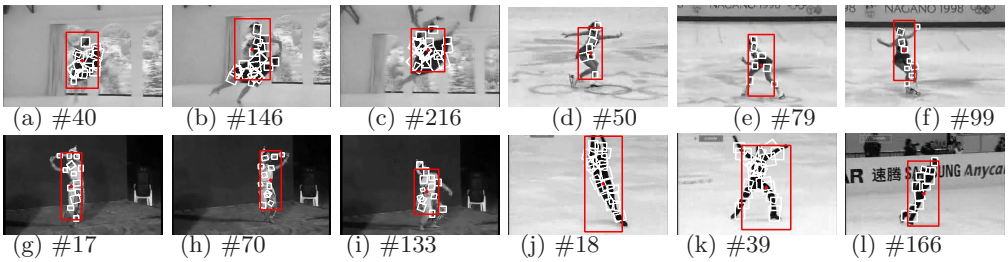
Figure 3.30: **Qualitative comparison with other methods using color sequences.** In (a) to (o), the green, magenta, cyan, yellow and red rectangles denote the bounding boxes of MCMC, IVT, FRAGT, BHT, and ABHMC-FS, respectively. In (p) to (r), the pink, white, and red rectangles denote the bounding boxes of ABHMC, ABHMC-F, and ABHMC-FS, respectively.

lumination changes and the deformation of targets. With the segmentation process, ABHMC-FS employed global appearance information and prevented local patches from drifting into the background. As comparison with the adaptive color-based tracking algorithm, the MS and MCMC trackers, the ABHMC-FS tracker produced better tracking results Since the non-rigid object severely changes its color appearance caused by deformation, the MS and MCMC trackers have difficult in covering drastic color changes. The ABHMC-FS tracker solved this problem by considering geometry appearance as well as color appearance of the object using the local patch-based appearance modeling. In addition, ABHMC-FS successfully tracked the targets in benchmark datasets such as *coke*, *girl*, *tiger1*, and *tiger2*. It is notable that ABHMC-FS showed small standard deviation, which means that the method produces stable tracking results. The main reason of small standard deviation is that ABHMC-FS partly utilizes the deterministic method, while the deterministic methods produce zero standard deviation.

In Figures 3.30(a) to 3.30(c), videos that include background clutter similar to

the object were tested. In the case of other methods, a trajectory was easily hijacked by the background clutter, with colors similar to those of the object, when the object changes its geometric appearance. On the other hand, our ABHMC-FS robustly tracked the object despite the background clutter and geometric appearance changes. Figures 3.30(d) and 3.30(e) demonstrate how the proposed method outperformed conventional tracking algorithms during drastic geometric appearance changes. The conventional tracking algorithms failed to track the object when the positions of the head and legs are reversed. Figure 3.30(f) shows that the specific model of the object occasionally cannot capture the drastic geometric changes of the object. The proposed method also tracked thin parts of the object (e.g. arms or legs) and covered the greater parts of the object area, whereas other methods failed to track such objects accurately, as shown in Figures 3.30(g) to 3.30(l). The test video used in Figures 3.30(m) to 3.30(o) includes the illumination and scale changes of an object. For tracking an object that grows larger over time, the proposed method extended the range between the center of an object and each local patch, and added new patches. Moreover, by changing the features of the patches adaptively, the proposed method tracked the object successfully despite severe illumination changes. Figures 3.30(p) to 3.30(r) shows the comparison within the proposed methods (ABHMC, ABHMC-F, and ABHMC-FS). The improvement in tracking performance was significant, especially in *snowboard* seq., which includes severe illumination and scale changes. In the sequence, the ABHMC-F efficiently dealt with the illumination change using multiple features and ABHMC-FS robustly handled the scale change with the segmentation results.

Figures 3.31(a) to 3.31(l) show the tracking results of the proposed method when only intensity values of gray sequences are available. In this case, the method neither



(a) #40 (b) #146 (c) #216 (d) #50 (e) #79 (f) #99

(g) #17 (h) #70 (i) #133 (j) #18 (k) #39 (l) #166

Figure 3.31: **Tracking results of the proposed method using gray sequences.** The red rectangles denote the bounding boxes of ABHMC-FS. White squares describe the local modes of patches in our appearance model.

(a) #154 (b) #230 (c) #302 (d) #41 (e) #49 (f) #114

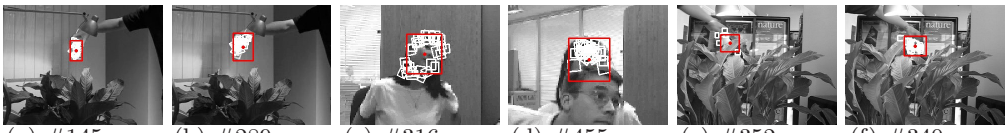
(g) #11 (h) #18 (i) #22 (j) #26 (k) #59 (l) #79

Figure 3.32: **Tracking results of the proposed method using recent challenging sequences.** The red rectangles denote the bounding boxes of ABHMC-FS. White squares describe the local modes of patches in our appearance model.

uses color features nor chooses robust ones. The method, however, robustly tracked the object in these sequences as well, while our appearance model well described the geometric appearance of the object using local patches and their local modes. Figures 3.32 and 3.33 demonstrates that the ABHMC-FS tracker accurately tracks the targets in the recent challenging datasets including highly non-rigid objects and in the benchmark datasets including pose variations and occlusions of the objects.

3.3 The Visual Tracking Sampler (VTS) Tracker

To track the object robustly in aforementioned scenario, several tracking methods proposed advanced appearance and motion models [6, 85, 19, 8, 9, 10, 11, 51,



(a) #145 (b) #289 (c) #316 (d) #455 (e) #352 (f) #340

Figure 3.33: **Tracking results of the proposed method using benchmark sequences.** The red rectangles denote the bounding boxes of ABHMC-FS. White squares describe the local modes of patches in our appearance model.

(a) Abrupt motion (b) Occlusion (c) Illumination change

Figure 3.34: **Example of our tracking results** in the *skating1*^{L} sequence. Our tracking algorithm successfully tracks a target even though there are severe pose variations, abrupt motions, occlusion, and illumination changes combinatorially.

22, 26, 12, 16, 13, 1, 52, 14, 86, 64, 87]. However, these methods are insufficient to cope with the complicated real-world tracking environment. To deal with all possible changes of an object simultaneously, tracking methods require more complex appearance and motion models, and should adopt more complex state representation and observation types. Moreover, given that the tracking environment severely varies from frame to frame, trackers should not be fixed, but should be dynamically generated depending on the current tracking environment. This paper thus focuses on how to design the complex models efficiently, how to construct the appropriate trackers automatically and how to integrate the constructed trackers for successful tracking under challenging real-world scenarios. Figure 3.34 shows the tracking results of our method in the real-world tracking environment.

The philosophy of our method is to use multiple basic trackers instead of a single complex tracker to solve combinatorial and realistic tracking problems. In multiple basic trackers, a tracker is only robust to a specific type of changes in the

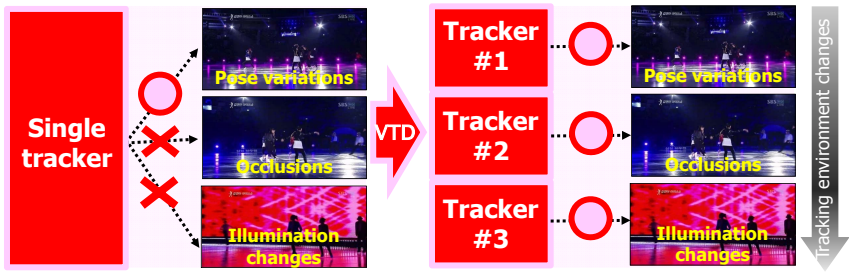
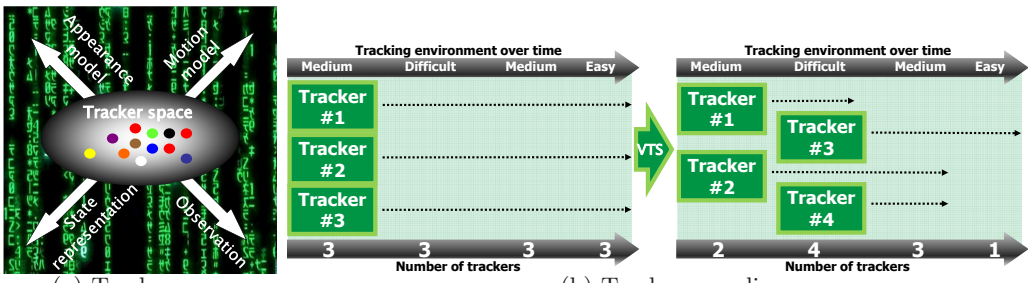


Figure 3.35: **Primary advantage of the proposed method using VTD** A single tracker has difficulty in covering several appearance and motion changes at the same time. We successfully cover these changes in our multiple-tracker approach, where each tracker deals with a specific type of object change.

object. However, because each tracker takes charge of a different type of changes, the trackers as a whole can cover various object changes at the same time, by communicating each other, as illustrated in Figure 3.35. For the communication, we introduce Interacting Markov Chain Monte Carlo (IMCMC) technique [88] , which comprises multiple interactive chains. In our tracking system, one chain corresponds to one basic tracker. Although each basic tracker is simple, by allowing the exchange of information among basic trackers with unique advantages, our tracking method efficiently fuses all the complementary advantages.

In this case, the multiple basic trackers are constructed by extracting the basic distinctive components of appearance model, motion model, state representation type, and observation type of a tracker. We call the process of determining these basic components as *Visual Tracking Decomposition* (VTD). We obtain the basic components of the appearance model using the sparse principal component analysis (SPCA) [89] to determine object models that comprise different feature combinations, Each object model is then mapped to a basic component of the appearance model. To obtain the basic components of the motion model, our method summa-



(a) Tracker space (b) Tracker sampling

Figure 3.36: **Secondary advantage of our method using VTS** (a) The figure describes our four-dimensional tracker space, in which the axes are the appearance model, motion model, state representation type, and observation type. A tracker is determined by sampling a point in the tracker space, where each circle represents a different tracker. (b) Compared with conventional tracking approaches that always use a fixed number of trackers over time, our method chooses appropriate trackers during the tracking process for the robust tracking of the object. In our approach, the number of trackers changes adaptively depending on the degree of difficulty in tracking the target.

rizes motion vectors produced by a moving object into a few representative clusters using the k-harmonic means (KHM) method [90]. Each basic component of the motion model is then modeled by the Gaussian function with the mean as the center of each cluster. For the basic components of the state representation type, the method represents the target as a mixture of multiple fragments, which are obtained by the vertical projection of edge (VPE) [66]. A different mixture of multiple fragments subsequently produces a different basic component of the state representation type. Finally, the method exploits several different observations using the Gaussian filter bank (GFB) [91], where a different observation indicates a different basic component of the observation type.

The second philosophy of our method is that the basic trackers can be constructed probabilistically, as illustrated in Figure 3.36(a). Using sampling methods, the trackers themselves are sampled, as well as the states of the targets. We call the process of sampling trackers as *Visual Tracker Sampler* (VTS). In our framework,

a sample represents information not only about a proposed state, but also about a proposed tracker. During the sampling process, our method obtains multiple trackers and their states as samples and then determines whether they will be accepted. By choosing an accepted sample that gives the highest value for the Conditional Maximum a Posteriori (CMAP) estimate, the method simultaneously finds a highly possible tracker and a highly possible state, with the former indicating the best tracker for the target and the latter denoting the best state where the target might be located.

3.3.1 Decomposition of Bayesian Tracker

We adopt a Bayesian approach to target tracking. This approach offers a systematic way of combining prior knowledge of target positions, modeling assumptions, and observation information to the problem of tracking targets [18]. As derived in (1.7), a Bayesian tracker consists of four important ingredients, namely, Appearance model $\mathbf{A}_t$:$p(\mathbf{Y}_t|\mathbf{X}_t)$, Motion model $\mathbf{M}_t$:$p(\mathbf{X}_t|\mathbf{X}_{t-1})$, State model $\mathbf{S}_t$:$\mathbf{X}_t$, Observation type $\mathbf{O}_t$:$\mathbf{Y}_t$. These ingredients form the sets: $\mathbf{A}_t$=$\{\mathrm{A}^i|i = 1,\ldots,|\mathbf{A}_t|\}$, $\mathbf{M}_t$=$\{\mathrm{M}^i|i = 1,\ldots,|\mathbf{M}_t|\}$, $\mathbf{S}_t$=$\{\mathrm{S}^i|i = 1,\ldots,|\mathbf{S}_t|\}$, and $\mathbf{O}_t$=$\{\mathrm{O}^i|i = 1,\ldots,|\mathbf{O}_t|\}$, where $\mathbf{A}_t$, $\mathbf{M}_t$, $\mathbf{S}_t$, and $\mathbf{O}_t$ indicate the set of appearance models, motion models, state representation types, and observation types at time t, respectively, and $|\cdot|$ indicates cardinality of the set. And A^i, M^i, S^i, and O^i denote the i-th component of $\mathbf{A}_t$, $\mathbf{M}_t$, $\mathbf{S}_t$, and $\mathbf{O}_t$, respectively. The i-th tracker at time t, T_t^i is then constructed by choosing a specific component of appearance model A^j, motion model M^k, state representation type S^l, and observation type O^m from the sets, $\mathbf{A}_t$, $\mathbf{M}_t$, $\mathbf{S}_t$, and $\mathbf{O}_t$, respectively. Hence, we derive $\mathrm{T}_t^i = (\mathrm{A}^j, \mathrm{M}^k, \mathrm{S}^l, \mathrm{O}^m)$. In a similar manner, our method finally creates the $|\mathbf{T}_t|$ number of trackers at time t, $\mathbf{T}_t = \{\mathrm{T}_t^i|i = 1,\ldots,|\mathbf{T}_t|\}$ by fully

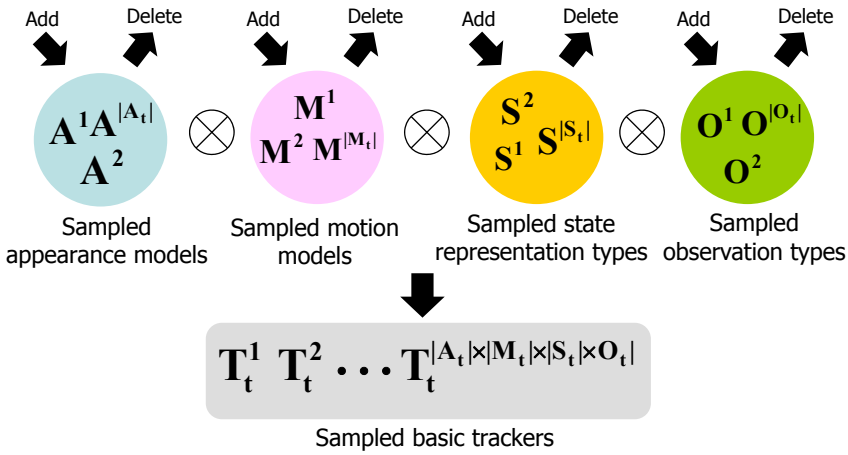


Figure 3.37: **Multiple basic trackers** Different associations of the four ingredients in the sets produce different basic trackers.

associating the four ingredients in the sets, as illustrated in Figure 3.37, where $|\mathbf{T}_t| = |\mathbf{A}_t| \times |\mathbf{M}_t| \times |\mathbf{S}_t| \times |\mathbf{O}_t|$.

3.3.1.1 Decomposed Posterior Probability

Using the aforementioned ingredients, the original posterior probability in (1.7) can be efficiently estimated by the weighted linear combination of the decomposed posterior probabilities, which depends on the i-th tracker, $\mathrm{T}_t^i = (\mathrm{A}^j, \mathrm{M}^k, \mathrm{S}^l, \mathrm{O}^m)$:

$$p(\mathbf{X}_t|\mathbf{Y}_{1:t}) \approx \sum_{i=1}^{|\mathbf{T}_t|} p(\mathrm{T}_t^i|\mathbf{Y}_{1:t}) p(\mathbf{X}_t|\mathrm{T}_t^i, \mathbf{Y}_{1:t}), \tag{3.35}$$

where $p(\mathbf{X}_t|\mathrm{T}_t^i, \mathbf{Y}_{1:t})$ represents the i-th decomposed posteriori probability, and $p(\mathrm{T}_t^i|\mathbf{Y}_{1:t})$ indicates its weight.

Compared with a direct estimation of the posterior probability, the decomposition strategy in (3.35) produces better performance under the following logarithmic scoring criterion.

Lemma 2. Averaging the decomposed posterior probabilities is optimal under the

logarithmic scoring criterion in [92]:

$$E\left[log\left\{\sum_{i=1}^{|\mathbf{T}_t|} p(\mathrm{T}_t^i|\mathbf{Y}_{1:t})p(\mathbf{X}_t|\mathrm{T}_t^i,\mathbf{Y}_{1:t})\right\}\right] \geq E\left[log\ p(\mathbf{X}_t|\mathbf{Y}_{1:t})\right], \tag{3.36}$$

for any distribution $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ where the expectation is with respect to $\sum_{i=1}^{|\mathbf{T}_t|} p(\mathrm{T}_t^i|\mathbf{Y}_{1:t})\ p(\mathbf{X}_t|\mathrm{T}_t^i,\mathbf{Y}_{1:t})$.

Proof. Inequality follows from the non-negative property of the Kullback-Leibler information divergence [3].

Then, to decompose the posterior probability efficiently while reflecting various changes in visual tracking, each decomposed posterior probability $p(\mathbf{X}_t|\mathrm{T}_t^i,\mathbf{Y}_{1:t})$ should be conditioned on the tracker, T_t^i, which runs robustly in the current tracking environment. The next section explains how we obtain the set of trackers and use them to identify the best target state.

3.3.2 Conditional Maximum a Posteriori Estimate

Our method determines the best state of the target, $\hat{\mathbf{X}}_t$, at time t using the Conditional Maximum a Posteriori (CMAP) estimate:

$$\hat{\mathbf{X}}_t \equiv \arg\max_{\mathbf{X}_t} p(\mathbf{X}_t|\mathbf{T}_t,\mathbf{Y}_{1:t}). \tag{3.37}$$

The posterior probability in (3.37) is conditioned on the set of trackers, $\mathbf{T}_t$. Thus, we should search all possible trackers and their states to obtain the CMAP estimate. However, this task is unfeasible because the search space is drastically large and highly dimensional.

We solve this problem by approximately estimating the posterior probability in (3.37) with the samples of trackers and states. To do this, our method first

[3] The decomposition strategy of the posterior probability is directly related to the Bayesian Model Averaging approach in [93].

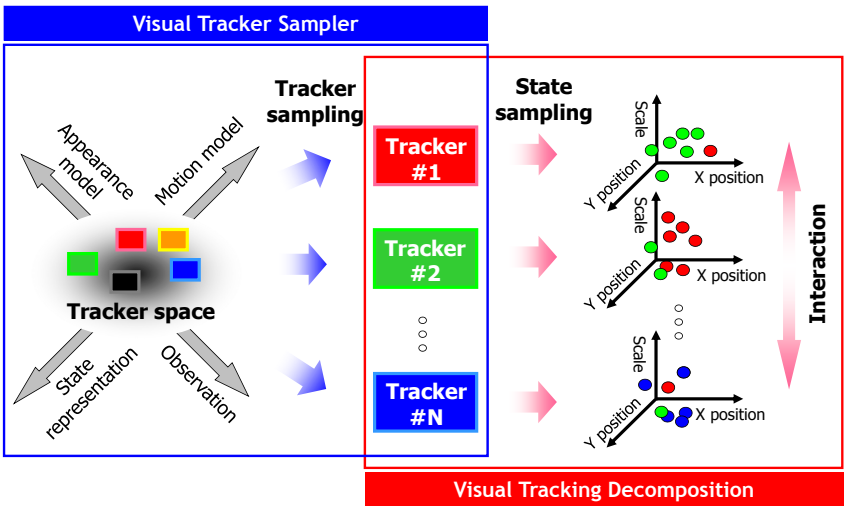
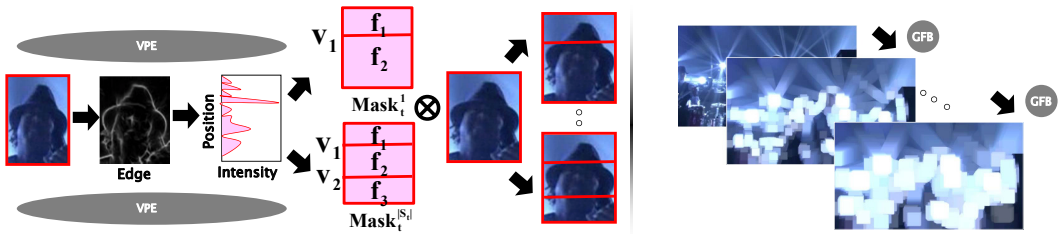


Figure 3.38: **General procedure of our method** The trackers are constructed by sampling. These trackers are then operated in parallel and interactively. Samples of the target state are then obtained utilizing the trackers.

obtains the samples of trackers and then uses them to determine samples for states. Among these sampled states, our method chooses the best one $\hat{\mathbf{X}}_t$, which provides the highest value on (3.37). The remaining tasks is to obtain samples of trackers (tracker sampling process in Section 3.3.3) and states (state sampling process in Section 3.3.4) simultaneously. Figure 3.38 describes the general procedure of our method.

3.3.3 Tracker Sampling Process

This section describes how tracker samples are obtained. We first define the visual tracker space to sample a basic tracker itself. The aforementioned four ingredients characterize trackers. Thus, sampling a tracker can be viewed as then sampling of its basic ingredients, as illustrated in Figure 3.37. During sampling, the basic ingredients should be considered together because they are inter-related. Notably, however,



(a) State model (b) Observation model

Figure 3.39: **Candidates of the state and observation models** Candidates of the state and observation models are made via VPE and GFB, respectively.

considering all ingredients at the same time is intractable. Thus, in our work, we use the Gibbs sampling strategy, through which we determine one ingredient at a time, whereas other ingredients are fixed on the best ones. To determine each ingredient properly, following aspects should be considered: The sampler should choose the required basic components of ingredients by accepting the ones that help track the target under the current environment, while avoiding unnecessarily complex components of ingredients through the acceptance ratio. The sampler thereby maintains the number of basic components of ingredients as small as possible and provides good performance in terms of scalability. Additionally, to find good models or types efficiently in the extremely vast tracker space and to reduce convergence time, the sampler should utilize the proposal that sufficiently exploits underlying cues in the video. The following sub-sections describe how the proposal and acceptance ratio for each ingredient is designed to achieve these goals.

3.3.3.1 State Model

• **Proposal Step:** A state model should be designed to preserve the target's spatial information while also covering its geometric variations to some degree. Our sampler thus represents the target as a combination of multiple fragments by adopting the

philosophy of [56]. The i-th state model is then defined by:

$$\mathrm{S}^i \equiv \mathbf{X}_t = BB_{\{x_t,y_t,s_t\}} \bigotimes \mathbf{Mask}_t^i, \tag{3.38}$$

where x_t, y_t, and s_t indicate the x, y center position and scale of the bounding box of the target, respectively. In (3.38), the bounding box described by $\{x_t, y_t, s_t\}$, $BB_{\{x_t,y_t,s_t\}}$, is divided into several fragments using the i-th mask, $\mathbf{Mask}_t^i$, where $\bigotimes$ denotes the convolution operator. The mask comprises vertical sub-indices of the bounding box, $\mathbf{Mask}_t^i = \{\mathrm{v}_j | j = 1, \ldots, |\mathbf{Mask}_t^i|\}$, where the vertical sub-indexes are normalized to have a value ranging from 0 to 1. Using the mask, our sampler produces an $|\mathbf{F}_t^i|$ number of image fragments, $\mathbf{F}_t^i = \{\mathrm{f}_j | j = 1, \ldots, |\mathbf{Mask}_t^i| + 1\}$, by dividing the bounding box horizontally at each vertical sub-index, v_j. Figure 3.39(a) shows that the vertical sub-index v_j is efficiently achieved by the vertical projection of edge (VPE) [66]. The randomly chosen model S^i is then added into $\mathbf{S}_t$ by the proposal function, $Q_S(\mathbf{S}_t^*; \mathbf{S}_t)$, which proposes the new set of state representation types, $\mathbf{S}_t^*$:

$$\mathbf{S}_t^* \sim Q_S(\mathbf{S}_t^*; \mathbf{S}_t) = \mathbf{S}_t \bigcup \mathrm{S}^i. \tag{3.39}$$

To remove a model from the current set, the sampler randomly selects a model, and proposes a new set that does not include itself:

$$\mathbf{S}_t^* \sim Q_S(\mathbf{S}_t^*; \mathbf{S}_t) = \mathbf{S}_t / \mathrm{S}^i. \tag{3.40}$$

• **Acceptance Step:** Given the proposed set of state representation types, $\mathbf{S}_t^*$, our sampler decides on acceptance or rejection using the acceptance ratio. This ratio is designed so that the state representation types in $\mathbf{S}_t^*$ reduces target appearance

variations for the most recent five frames:

$$\begin{aligned} a_S &= min\left[1, \frac{p(\mathbf{S}_t^*|\mathbf{X}_t, \mathbf{Y}_{1:t})Q(\mathbf{S}_t; \mathbf{S}_t^*)}{p(\mathbf{S}_t|\mathbf{X}_t, \mathbf{Y}_{1:t})Q(\mathbf{S}_t^*; \mathbf{S}_t)}\right] \\ \text{where} \quad &- \log p(\mathbf{S}_t^*|\mathbf{X}_t, \mathbf{Y}_{1:t}) \propto \sum_{i=1}^{|\mathbf{S}_t^*|} \sum_{j=1}^{|\mathbf{F}_t^i|} VAR(\mathrm{f}_j) + \lambda_S \log |\mathbf{S}_t^*|. \end{aligned} \tag{3.41}$$

In (3.41), $VAR(\mathrm{f}_j)$ returns variance of the j-th image fragment, f_j, for the most recent five frames; $\log |\mathbf{S}_t^*|$ prevents the set $\mathbf{S}_t^*$ from having large numbers of state representation types; and λ_S is the weighting parameter.

3.3.3.2 Observation Model

• **Proposal Step:** Biological evidence shows that the human visual system uses the response of multiple filters called the filter bank to observe visual information. Similarly, more robust observation types can be achieved using the Gaussian filter bank (GFB) [91], as shown in Figure 3.39(b). The i-th observation model is constructed by the convolution between image $\mathbf{I}_t$ and the Gaussian distribution with mean $\{x_t, y_t\}$ and variance Σ_i^2 for all $\{x_t, y_t\}$ of $\mathbf{X}_t$.

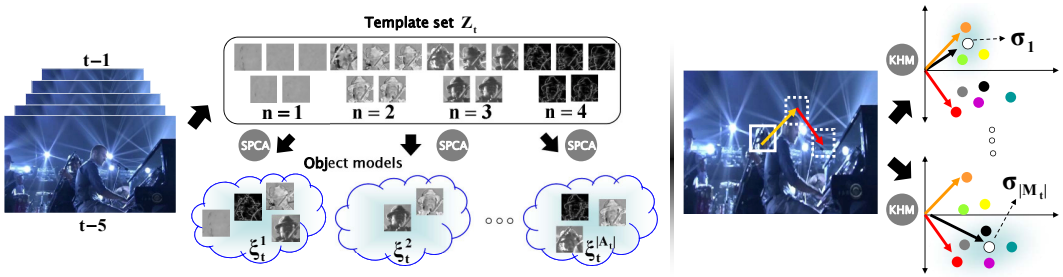
$$\mathrm{O}^i \equiv \mathrm{Y}_t = \mathbf{I}_t * G(\{x_t, y_t\}, \Sigma_i^2), \ \forall \{x_t, y_t\}, \tag{3.42}$$

where Σ_i is selected randomly from the uniform distribution, $U[0, 10]$, in a component-wise manner for x_t and y_t. The randomly chosen model O^i is inserted into $\mathbf{O}_t$ by the proposal function, $Q_O(\mathbf{O}_t^*; \mathbf{O}_t)$, which proposes the new set of observation types, $\mathbf{O}_t^*$:

$$\mathbf{O}_t{}^* \sim Q_O(\mathbf{O}_t^*; \mathbf{O}_t) = \mathbf{O}_t \bigcup \mathrm{O}^i. \tag{3.43}$$

To remove a model from the current set, the sampler randomly selects a model, and proposes a new set that does not include itself:

$$\mathbf{O}_t{}^* \sim Q_O(\mathbf{O}_t^*; \mathbf{O}_t) = \mathbf{O}_t / \mathrm{O}^i. \tag{3.44}$$



(a) Appearance model (b) Motion model

Figure 3.40: **Candidates of the appearance and motion models** We make candidates of the appearance model and motion model utilizing SPCA and KHM, respectively.

• **Acceptance Step:** The acceptance ratio is designed so that the response of observation types in $\mathbf{O}_t^*$ become more similar among foreground images, but more different between foreground and background images for the most recent five frames. Foreground and background images are obtained by cropping images within and around the bounding box of the target, respectively. The acceptance ratio is then defined by:

$$\begin{aligned} a_O &= min\left[1, \frac{p(\mathbf{O}_t^*|\mathbf{X}_t, \mathbf{Y}_{1:t})Q(\mathbf{O}_t; \mathbf{O}_t^*)}{p(\mathbf{O}_t|\mathbf{X}_t, \mathbf{Y}_{1:t})Q(\mathbf{O}_t^*; \mathbf{O}_t)}\right] \\ \text{where} \quad &- \log p(\mathbf{O}_t^*|\mathbf{X}_t, \mathbf{Y}_{1:t}) \propto \\ &\frac{\sum_{i=1}^{|\mathbf{O}_t^*|}\sum_{j,k=t-5}^{t-1} DD(\phi_j^i, \phi_k^i)}{\sum_{i=1}^{|\mathbf{O}_t^*|}\sum_{j,k=t-5}^{t-1} DD(\phi_j^i, \psi_k^i)} + \lambda_O \log|\mathbf{O}_t^*|, \end{aligned} \tag{3.45}$$

where λ_O is the weighting parameter; and ϕ_j^i and ψ_k^i represent the foreground and background image of the i-th observation model at times j and k, respectively. In (3.45), the $DD(\phi_j^i, \psi_k^i)$ function [69] returns the diffusion distance between ϕ_j^i and ψ_k^i.

3.3.3.3 Appearance Model

• **Proposal Step:** An appearance model should cover most appearance changes of the target. Such a model can be efficiently obtained by sparse principal component

analysis (SPCA). SPCA finds several sparse principal components, each of which is composed of a mixture of templates that describe the target appearance, as shown in Figure 3.40(a). In this paper, we employ the mixture of template model for object representation. For this model, we define set $\mathbf{Z}_t$, as expressed below, which consists of different types of feature templates of an object up to time t:

$$\mathbf{Z}_t = \{z_m^n | m = 1, \ldots, t, n = 1, \ldots, u\}, |\mathbf{Z}_t| = tu, \tag{3.46}$$

where z_m^n denotes the n-th type of the feature template at time m, and $|\mathbf{Z}_t|$ indicates the total number of feature templates in $\mathbf{Z}_t$. In (3.46), the different types of feature templates z_m^n are obtained using different types of feature extractors FE^n for the image patch $\mathbf{Y}_t(\hat{\mathbf{X}}_m)$ at each time:

$$z_m^n = \frac{FE^n(\mathbf{Y}_t(\hat{\mathbf{X}}_m))}{\|FE^n(\mathbf{Y}_t(\hat{\mathbf{X}}_m))\|}, m = 1, \ldots, t, n = 1, \ldots, u, \tag{3.47}$$

where $\mathbf{Y}_t(\hat{\mathbf{X}}_m)$ represents the image patch at time m, which is described by $\hat{\mathbf{X}}_m$ in (3.37); and FE^n indicates the feature extractor for obtaining the n-th type of the feature template.

An appearance model takes one subset of $\mathbf{Z}_t$ as its own object model ξ_t^i at time t.

$$\xi_t^i \subset \mathbf{Z}_t. \tag{3.48}$$

Then, the appearance model is determined by

$$\mathrm{A}^i \equiv p(\mathbf{Y}_t|\mathbf{X}_t) = exp^{-\gamma DD(\mathbf{Y}_t(\mathbf{X}_t), \xi_t^i)}, \tag{3.49}$$

where γ denotes the weighting parameter, and $\mathbf{Y}_t(\mathbf{X}_t)$ indicates the image patch described by $\mathbf{X}_t$. In (3.49), the DD function returns the diffusion distance between $\mathbf{Y}_t(\mathbf{X}_t)$ and ξ_t^i at time t. We utilize diffusion distance as a dissimilarity measure

because it is robust for the deformation and quantization effects of the observation [69]. Given that the object model ξ_t^i comprises multiple templates, $DD(\mathbf{Y}_t(\mathbf{X}_t), \xi_t^i)$ is computed as the sum of the dissimilarity between the image patch $\mathbf{Y}_t(\mathbf{X}_t)$ and each template in ξ_t^i. To complete the design of the appearance model, the remaining task is to obtain a subset of $\mathbf{Z}_t$ as an object model ξ_t^i, which is efficiently performed by the SPCA method.

Note that the appearance model is inter-related to the state representation and the observation types. To deal with this inter-relation, we use the Gibbs sampling strategy, where the state representation and the observation types are fixed to the best ones during the sampling of appearance models.

$$p(\mathbf{Y}_t|\mathbf{X}_t) = \prod_{m=1}^{|\hat{\mathbf{F}}_t|} exp^{-\gamma \frac{DD(\hat{\mathrm{Y}}_t(\hat{\mathrm{f}}_m), \xi_t^i)}{|\hat{\mathbf{F}}_t|}}, \tag{3.50}$$

where $\hat{\mathbf{F}}_t$ denotes the best set of fragments for the state model, $\hat{\mathrm{f}}_m$ indicates the m-th fragment in the best set, and $\hat{\mathrm{Y}}_t$ represents the best observation model. Using (3.50), the final likelihood is obtained by averaging the likelihoods of all fragments, where the likelihood of each fragment is calculated by measuring the diffusion distance between the best observation of the fragment $\hat{\mathrm{Y}}_t(\hat{\mathrm{f}}_m)$ and the object model ξ_t^i .

Three conditions for the object model ξ_t^i are considered to be ideal in terms of tracking performance and efficiency. The first condition is that ξ_t^i has to cover most appearance changes in an object over time. The second condition is that the formation of the object model should be as compact as possible, while preserving good performance. The last condition is that the relations among other object models should be complementary. To satisfy all of these conditions, our method adopts the SPCA method to construct ξ_t^i. Given a Gramian matrix A_t, the original SPCA method [89] seeks out sparse principal components c, which only have a

limited number of non-zero entries, while capturing a maximum amount of variance, as expressed by

$$maximize \quad c^T G_t c - \rho|c|^2 \quad subject\ to \quad \|c\|_2 = 1, \tag{3.51}$$

where $|c|$ is the number of nonzero entries in c, and ρ controls the penalty on the nonzero entries of c. As the ρ value increases, we have more sparse principal components c [4]. For our tracking problem, the Gramian matrix G_t at time t is constructed as:

$$G_t = g^T g, \ g = \begin{pmatrix} z_1^1 & \dots & z_t^1 & \dots & z_1^u & \dots & z_t^u \end{pmatrix}, \tag{3.52}$$

where the size of G_t is $|\mathbf{Z}_t| \times |\mathbf{Z}_t|$ because the column size of the matrix g is $|\mathbf{Z}_t|$.

With conventional convex optimization tools [89], we can efficiently obtain the approximate principal components c in (3.51). Each principal component then composes each object model ξ_t^i in (3.48) as follows:

$$\xi_t^i = \{z_m^n | z_m^n = g(x), c_i(x) \neq 0\}. \tag{3.53}$$

If the x-th element of the i-th principal component c_i has a nonzero value, the i-th object model ξ_t^i includes the template z_m^n located at the x-th column of the matrix g in (3.52). Hence, each object model captures significant appearance changes in an object because each model is constructed by each significant eigenvector. The sparsity of the eigenvector gives compactness to the model while making it have a small number of templates. Because the eigenvectors have an orthogonal property, the object models have a complementary relationship with one another.

As the last step, the proposal function, $Q_A(\mathbf{A}_t^*; \mathbf{A}_t)$ chooses a new model A^i with a higher eigenvalue. The new model is then added into $\mathbf{A}_t$, suggesting a new set of

[4] We set ρ to 90 in all of the experiments.

appearance models, $\mathbf{A}_t^*$:

$$\mathbf{A}_t^* \sim Q_A(\mathbf{A}_t^*; \mathbf{A}_t) = \mathbf{A}_t \bigcup \mathrm{A}^i. \tag{3.54}$$

To remove a model from the current set, the sampler randomly selects a model, and proposes a new set that does not include itself:

$$\mathbf{A}_t^* \sim Q_A(\mathbf{A}_t^*; \mathbf{A}_t) = \mathbf{A}_t / \mathrm{A}^i. \tag{3.55}$$

• **Acceptance Step:** Our sampler accepts the proposed set of appearance models, $\mathbf{A}_t^*$, with high probability if the appearance models in $\mathbf{A}_t^*$ produce higher likelihood scores than those in $\mathbf{A}_t$ at the CMAP state, $\hat{\mathbf{X}}_t$, for the most recent five frames, in which the CMAP state at time t found by (3.37) indicates the best state of the target at time t:

$$\begin{aligned} a_A &= min\left[1, \frac{p(\mathbf{A}_t^*|\hat{\mathbf{X}}_t, \mathbf{Y}_{1:t})Q(\mathbf{A}_t; \mathbf{A}_t^*)}{p(\mathbf{A}_t|\hat{\mathbf{X}}_t, \mathbf{Y}_{1:t})Q(\mathbf{A}_t^*; \mathbf{A}_t)}\right] \\ \text{where} \quad &-\log p(\mathbf{A}_t^*|\hat{\mathbf{X}}_t, \mathbf{Y}_{1:t}) \propto \sum_{i=1}^{|\mathbf{A}_t^*|} \sum_{j=t-5}^{t-1} DD(Y_j(\hat{\mathbf{X}}_j), \xi_t^i) + \lambda_A \log |\mathbf{A}_t^*|. \end{aligned} \tag{3.56}$$

In (3.56), $Y_j(\hat{\mathbf{X}}_j)$ indicates the observation at the MAP state, $\hat{\mathbf{X}}_j$ at time j, and λ_A is the weighting parameter.

3.3.3.4 Motion Model

• **Proposal Step:** A motion model has to describe the representative characteristics of the target motion over time. This model is efficiently found by the K-Harmonic Means (KHM) method, which clusters data and finds centers of the clusters, as shown in Figure 3.40(b), where KHM is known to be insensitive to the initialization of the centers [90]. The data, $\mathbf{D}_t$, for KHM is acquired by gathering velocity vectors between accepted neighbor states in the most recent five frames. By

selecting the i-th cluster center, $\mathbf{\Sigma}_i = [\sigma_i^x, \sigma_i^y, \sigma_i^s]^T$ of $\mathbf{D}_t$, the i-th motion model is constructed as:

$$\mathrm{M}^i \equiv p(\mathbf{X}_t^*|\mathbf{X}_t) = G(\mathbf{X}_t, \mathbf{\Sigma}_i^2) = G(\{x_t, y_t, s_t\}, \Sigma_i^2), \tag{3.57}$$

where G denotes the Gaussian function with mean $\mathbf{X}_t$ and variance Σ_i^2. Note that, for the motion model, our method utilizes the state before the masking process, as explained in Section 3.3.3.1. Because all fragments in the bounding box are consistently moved by the motion model, our method does not need to consider the specific state model. Using the motion model in (3.57), a new state without masking, $\{x_t^*, y_t^*, s_t^*\}$ is proposed based on the previous state without masking, $\{x_t, y_t, s_t\}$. Subsequently, the proposal function, $Q_M(\mathbf{M}_t^*; \mathbf{M}_t)$ chooses the new model, M^i, with a higher confidence value, adds the new model into $\mathbf{M}_t$, and proposes the new set of motion models, $\mathbf{M}_t^*$:

$$\mathbf{M}_t^* \sim Q_M(\mathbf{M}_t^*; \mathbf{M}_t) = \mathbf{M}_t \bigcup \mathrm{M}^i. \tag{3.58}$$

To remove a model from the current set, the sampler randomly selects a model, and proposes a new set that does not include itself:

$$\mathbf{M}_t^* \sim Q_M(\mathbf{M}_t^*; \mathbf{M}_t) = \mathbf{M}_t / \mathrm{M}^i. \tag{3.59}$$

• **Acceptance Step:** Our sampler accepts $\mathbf{M}_t^*$ with high probability if the motion models in $\mathbf{M}_t^*$ have more accurate cluster centers σ_i than those in $\mathbf{M}_t$:

$$\begin{aligned} a_M &= min\left[1, \frac{p(\mathbf{M}_t^*|\mathbf{X}_t, \mathbf{Y}_{1:t})Q(\mathbf{M}_t; \mathbf{M}_t^*)}{p(\mathbf{M}_t|\mathbf{X}_t, \mathbf{Y}_{1:t})Q(\mathbf{M}_t^*; \mathbf{M}_t)}\right] \\ \text{where} \quad &- \log p(\mathbf{M}_t^*|\mathbf{X}_t, \mathbf{Y}_{1:t}) \propto \sum_{i=1}^{|\mathbf{M}_t^*|} VAR(\mathbf{D}_t, \sigma_i) + \lambda_M \log |\mathbf{M}_t^*|. \end{aligned} \tag{3.60}$$

In (3.60), $VAR(\mathbf{D}_t, \sigma_i)$ returns the variance of data, $\mathbf{D}_t$, that belongs to the cluster centered on σ_i, and λ_M is the weighting parameter.

3.3.4 State Sampling Process

In the previous section, our method proposed new trackers and determined whether they were accepted or not. Given the sampled trackers, new states of the target are obtained by the state sampling process. The state sampling process comprises two modes, namely, parallel and interacting. In the parallel mode, our method acts as parallel Metropolis Hastings algorithms. When the method is in the interacting mode, the trackers communicate with others and make leaps to better states of an object. The best state $\hat{\mathbf{X}}_t$ is then chosen among the sampled states by the CMAP criterion in (3.37).

3.3.4.1 Parallel Mode

Each sampled tracker $\mathrm{T}_t^i, i = 1, \ldots, |\mathbf{T}_t|$ constructs its own Markov Chain, runs in parallel with others, and produces samples of the state from the Markov Chain via the Metropolis Hastings algorithm, to estimate each decomposed posteriori probability, $p(\mathbf{X}_t|\mathrm{T}_t^i, \mathbf{Y}_{1:t})$ in (3.35). The sampling process comprises two main steps: the proposal and acceptance steps. In the proposal step, a new state is proposed by the proposal density function. For example, using the tracker constructed by the i-th appearance model, the j-th motion model, the k-th state model, and the l-th observation model, our method proposes a new state $\mathbf{X}_t^{j^*}$:

$$\mathbf{X}_t^{j^*} \sim Q_j(\mathbf{X}_t^{j^*}|\mathbf{X}_t^j) = G(\mathbf{X}_t^j, \Sigma_j^2), \tag{3.61}$$

where G denotes the Gaussian function with mean $\mathbf{X}_t^j$ and variance Σ_j^2. Our method also determines whether the proposed state is accepted or not using the following

acceptance ratio:

$$a_P = min\left[1, \frac{p(\mathbf{Y}_t|\mathrm{A}^i, \mathrm{S}^k, \mathrm{O}^l, \mathbf{X}_t^{j^*})Q_j(\mathbf{X}_t^j; \mathbf{X}_t^{j^*})}{p(\mathbf{Y}_t|\mathrm{A}^i, \mathrm{S}^k, \mathrm{O}^l, \mathbf{X}_t^j)Q_j(\mathbf{X}_t^{j^*}; \mathbf{X}_t^j)}\right]$$
$$\text{where } \; p(\mathbf{Y}_t|\mathrm{A}^i, \mathrm{S}^k, \mathrm{O}^l, \mathbf{X}_t^{j^*}) = \prod_{m=1}^{|\mathbf{F}_t^k|} exp^{-\gamma \frac{DD(\mathrm{Y}_t^l(\mathrm{f}_m), \xi_t^i)}{|\mathbf{F}_t^k|}}. \tag{3.62}$$

In (3.62), $p(\mathbf{Y}_t|\mathrm{A}^i, \mathrm{S}^k, \mathrm{O}^l, \mathbf{X}_t^{j^*})$ is the modified appearance model of (3.49), which further considers the k-th state model, and the l-th observation model, where $\mathbf{Y}_t^l(\mathrm{f}_m)$ indicates the l-th observation at the m-th image fragment. These two steps iteratively continue until the number of iterations reaches a predefined value.

3.3.4.2 Interacting Mode

During the sampling process, the basic trackers communicate with other basic trackers regarding the good configuration of an object. Because each basic tracker utilizes different components of the appearance, motion, state, and observation models, exchanging information results in the fusion of all components and in the implicit estimation of their weights. A component is implicitly considered to have a heavy weight if the basic tracker with the component produces numerous states that are frequently propagated to other basic trackers. To allow the trackers to communicate with one another, we introduce IMCMC [88] to our tracking problem. Using IMCMC, the trackers communicate with others and make leaps to better states of an object. A tracker accepts the state of another tracker, which is constructed by the i-th appearance model, the j-th motion model, the k-th state model, and the l-th observation model as its own state with the following probability:

$$a_I = \frac{p(\mathbf{Y}_t|\mathrm{A}^i, \mathrm{S}^k, \mathrm{O}^l, \mathbf{X}_t^{j^*})}{\sum_{i=1}^{|\mathbf{A}_t|}\sum_{j=1}^{|\mathbf{M}_t|}\sum_{k=1}^{|\mathbf{S}_t|}\sum_{l=1}^{|\mathbf{O}_t|} p(\mathbf{Y}_t|\mathrm{A}^i, \mathrm{S}^k, \mathrm{O}^l, \mathbf{X}_t^{j^*})}. \tag{3.63}$$

Our method operates in an interacting mode with the probability α_t at each iteration, which linearly decreases from 1.0 to 0.0 as the simulation goes on.

3.3.4.3 Property of State Sampling

In the state sampling process using parallel interacting Markov Chains, the samples from several decomposed posterior probabilities are fair samples from the original target posterior probability, $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$.

Lemma 3. The probability that the $|\mathbf{T}_t|$ number of parallel interacting Markov Chains visit every possible state between interaction times converges to unity, as $\alpha_t \to 0$ and $t \to \infty$.

Proof. Based on [94], we examine the behavior of each of the $|\mathbf{T}_t|$ parallel interacting processes. With t_i and t_{i+1} as the time of the i-th and *(i+1)*-th interacting modes, respectively, $\tau_i = t_{i+1} - t_i - 1$ is formulated, which refers to the number of iterations between two adjacent interacting modes. With n_{max} as the number of maximum iterations for the Markov Chain to visit every state. the probability that τ_i is equal or greater than n_{max} converges to unity for very small $\epsilon > 0$ and sufficiently large i, as follows:

$$\begin{aligned} p(\tau_i \geq n_{max}) &= 1 - p(\tau_i < n_{max}) = 1 - \sum_{t=0}^{n_{max}-1} p(\tau_i = t) \\ &= 1 - \sum_{t=0}^{n_{max}-1} \left(\alpha_{t_i+t+1} \prod_{s=t_i+1}^{t_i+t} (1 - \alpha_s) \right) \geq 1 - \sum_{t=0}^{n_{max}-1} \alpha_{t_i+t+1} \\ &\geq 1 - n_{max} max\{\alpha_t | t = t_i + 1, \ldots, t_i + n_{max}\} = 1 - \epsilon \end{aligned} \tag{3.64}$$

The last equality is supported since $\alpha_t \to 0$ for sufficiently large i.

3.3.5 Experimental Results

3.3.5.1 Implementation Details and Settings

We tested 18 video sequences [5] : *shaking*, *soccer*, *animal*, *skating1*, *skating1*^{L}, *skating2*, *singer1*, *singer1*^{L}, *singer2*, *basketball*, and *football* sequences in [26]; *soccer*^{N}, *skating1*^{N}, *iron-man*, and *matrix* sequences in [27]; and *tiger1*, *david*, and *occlface* in [56, 6, 1]. Using the datasets, our method (VTD, VTS) were compared with five different tracking methods: standard MCMC (MC) based on [18, 33, 36], Incremental Visual Tracking (IVT) in [1], Multiple Instance Learning (MIL) in [6], Fragment-based tracker (FRAGT) in [56], and Context-tracker (CT) in [95]. The same initializations were set to all methods for fair comparison. The parameters of all methods were adjusted to show the best performance. To obtain the tracking results of IVT, MIL, FRAGT, and CT, we used the software provided by the authors.

In our methods, VTD [26] denotes the tracking method which employs a *fixed* number of trackers with different appearance models and motion models only. On the other hand, VTS [27, 28] indicates the tracking method that uses a *varying* number of sampled trackers with different state representation types and observation types, as well as appearance models and motion models. For the experiment, we utilized hue, saturation, intensity, and edge template for the features FE^n in (3.47). The hue template describes the chrominance characteristic of an object. The intensity template represents the brightness status of the object [33]. And the edge template gives relatively consistent information on the shape of the object even under severe illumination changes [96]. With four different types of features,

[5]The codes of VTD and VTS are available at http://cv.snu.ac.kr/softwares.

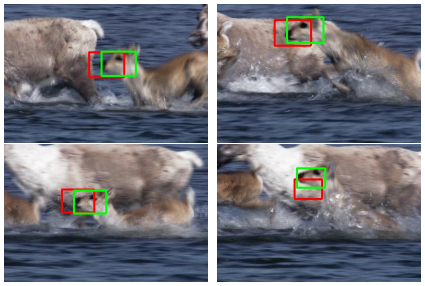
Table 3.9: **Performance of IMCMC and SPCA.** VTDĨ denotes our method without interaction between trackers, whereas VTDS̃ indicates our method without sparse principal component analysis. The numbers indicate average center location errors in pixels.

	tiger1	*david*	*face*	*shaking*	*soccer*	*animal*	*skating1*
VTD	**13**	**7**	**7**	**5**	**21**	**11**	**7**
VTDĨ	35	24	8	7	22	13	8
VTDS̃	54	70	8	68	96	40	219

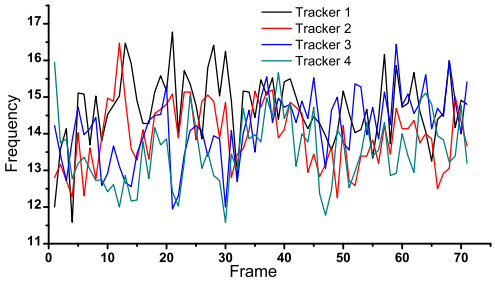
the set $\mathbf{Z}_t$ in (3.46) was created using five image patches obtained at the initial frame and four recent frames where $|\mathbf{Z}_t|$ is 20. In all experiments, we set λ_S,λ_O,λ_A, and λ_M in (3.41),(3.45),(3.56), and (3.60) to 0.05 and γ in (3.49)(3.62) to 5, which hardly affects on the tracking results. Note that our current implementation is not optimized, and it spends majority of the computational time to obtain a likelihood score by measuring the diffusion distance in [69]. Thus, by properly optimizing the process of measuring diffusion distance, we can significantly enhance the speed although it takes $0.2 \sim 1$ seconds per frame at the current state.

3.3.5.2 Quantitative Evaluation

Performance of IMCMC: VTD had a better performance than VTDĨ, as shown in Table 3.9, where VTDĨ denotes VTD without interaction between trackers. The results show that the interaction process in VTD is important in improving tracking performance, especially in the *tiger1* sequence. The sequence contains several kinds of appearance and motion changes. In the VTD method, a proper tracker among multiple ones covered these changes each time and propagated its state to the other trackers. Thus, VTD typically provided more accurate results than VTDĨ. Figure 3.41(b) describes how frequently each tracker exchanges information about its state in the *animal* sequence, which includes drastically abrupt motions of the object. In



(a) #5,#12,#22,#25



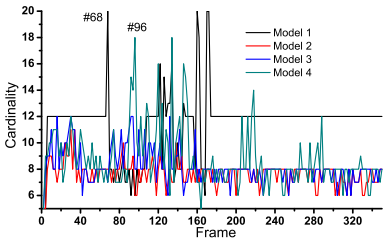
(b) The frequency of interaction between 4 trackers

Figure 3.41: **Interaction among multiple trackers** in the *animal* sequence.

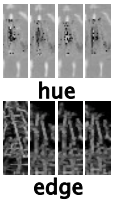
this sequence, each basic tracker actively interacted with the rest while helping other basic trackers to make leaps to a better state. Although some basic trackers failed to track the object, our method successfully found the proper state of the object, as shown in Figure 3.41(a), where the red rectangle denotes the tracking result of the failed basic tracker. Meanwhile, the green rectangle indicates the leapt state of the failed tracker with the help of other good trackers.

Performance of SPCA: We designed VTD$\tilde{\text{S}}$, such that different appearance models employ different types of features. On the other hand, each appearance model of VTD includes several types of feature templates obtained by SPCA. As shown in Table 3.9, the performance of VTD drastically improved in comparison with VTD$\tilde{\text{S}}$, indicating that the appearance models constructed by SPCA are very useful in our tracking problem. Figure 3.42(b) shows how SPCA adaptively constructs object models at each frame under severe illumination changes from frame #60 to #170 in the *singer1* sequence. The changes of cardinality in each model indicate that SPCA transforms each model into a different one to cover the specific appearance changes in an object. At frame #68, to represent the illuminated object, SPCA added hue and edge templates to Model 1, as shown in Figure 3.42(b)(c), which are relatively robust for the illumination changes [5]. Similarly, at frame #96, Model 4 is severely

(a) #1,#68,#96,#300



(b) Cardinality of 4 observation models



(c) Added templates at #68

Figure 3.42: **Adaptiveness of the observation models** in the *singer1* sequence.

Table 3.10: **Performance of sampling trackers.** The numbers denote the center location errors in pixels, where the green numbers indicate the total number of samples utilized to track the target.

	tiger1	*david*	*face*	*shaking*	*soccer*	*animal*	*skating1*
VTD	15	10	9	20	23	22	8
VTS	**13**	8	8	**5**	**17**	**10**	8
#N	514	562	356	616	408	696	304
VTD	13	**7**	**7**	5	21	11	**7**
#N	800	800	800	800	800	800	800

modified to deal with these changes. With the help of SPCA, VTD accurately tracks the object despite severe illumination changes as illustrated in Figure 3.42(a).

Performance of sampling trackers: To evaluate the performance of the tracker sampling process of VTS, we compared VTD with VTS. For this experiment, we modified the VTS method to construct trackers by changing the appearance and motion models only. If we test VTS directly, fair comparison with VTD is not achieved, because VTS considers additional tracker elements, such as the state representation and observation types. Table 3.10 summarizes the location errors. Because VTS automatically utilizes different numbers of trackers according to the tracking environment, we adjusted the total number of samples of VTD to be the same as that of VTS. VTS showed better tracking accuracy than VTD when the same number of samples were utilized. Moreover, with a smaller number of samples, VTS produced

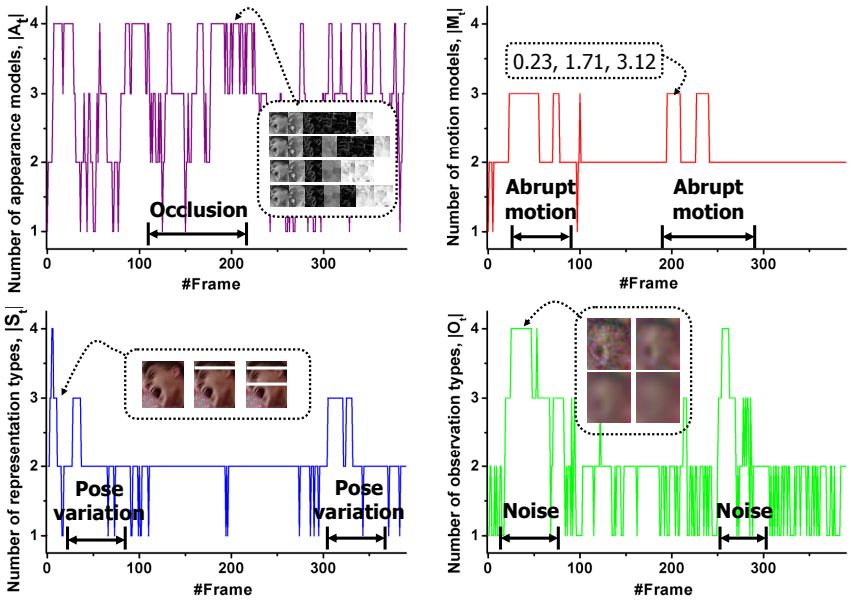
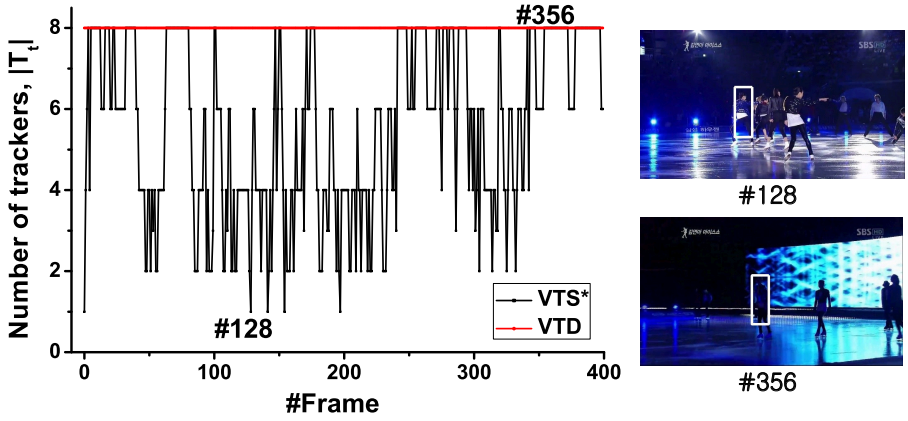


Figure 3.43: **Number of ingredients** as time goes on in the $soccer^N$ sequence

similar and even better tracking results compared with the original VTD, which always utilized more samples, 800. The better performance of VTS comes from the tracker sampling process, in which VTS changes the number of trackers and maintains only the required trackers by adaptively selecting appropriate ones depending on the current tracking environment.

To demonstrate how VTS produces accurate tracking results with an understanding of its mechanisms, we provide intermediary results of the four ingredients in VTS. As shown in Figure 3.43, VTS increased the number of each ingredient appropriately when there were specific changes in appearance or motion. For example, VTS constructed three motion models, of which proposal variances are 0.23, 1.71, and 3.12, and successfully tracked complex motions. To overcome severe noise in the sequence, VTS automatically employed four observation types, in which the degree of Gaussian blur differed. When pose variations manifested, VTS made appearance



(b) Number of tracker

Figure 3.44: **Number of trackers** as time goes on in the *skating1*^{N} sequence.

of the target insensitive to the variations, as passible, by utilizing three state representation types. Using the four appearance models, VTS described both occluded and non-occluded targets and robustly tracked them.

As illustrated in Figure 3.44, the tracker sampling process of VTS adaptively changed the number of trackers according to the tracking environment over time. For example, the tracker sampling process decreased the number of trackers and saved the resources at frame #128, because the frame included almost no movements and appearance changes of the target. At frame #356, VTS increased the number of trackers to capture appearance variations attributed to severe illumination changes. On the other hand, VTD wasted the resources by always using 8 trackers. Thus, it inaccurately tracked the target with a small number of samples.

Performance of the whole tracking system: We compared conventional methods with VTS by evaluating tracking accuracy. For the evaluation, we constructed highly challenging video sequences. We manually added noise and motion blur into the *soccer* and *skating1* sequences, and made new sequences, such as, *soccer*^{N} and *skating1*^{N}. Then, the sequences simultaneously exhibit severe illumination, view-

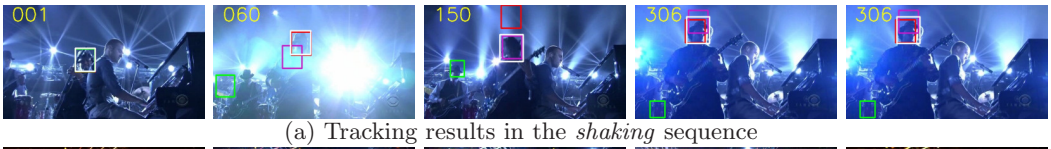
Table 3.11: **Comparison of tracking accuracy.** The numbers denote the center location errors in pixels, where red is the best result and blue is the second-best result. *singer1*^{L} and *skating1*^{L} represent the modified version of the original sequences to have partially low frame rate. *soccer*^{N} and *skating1*^{N} indicates the modified version of the original sequences to have noise and blur.

	singer1	*tiger1*	*david*	*face*	*shaking*	*soccer*	*animal*	*skating1*	*soccer*^{N}	*skating1*^{N}	*singer1*^{L}	*skating1*^{L}	*iron*	*matrix*
MC	43	32	41	19	98	53	26	172	72	126	156	162	**78**	123
IVT	**12**	83	**5**	20	150	116	21	213	225	291	226	215	104	**50**
FRAGT	27	40	46	**6**	**30**	82	97	93	**71**	105	67	**53**	113	92
MIL	59	15	23	27	38	**41**	30	**85**	147	**87**	**66**	142	122	57
CT	41	**7**	**5**	12	213	141	**9**	161	145	124	91	177	167	175
VTS	**3**	**12**	**7**	**8**	**5**	**17**	**10**	**8**	**24**	**8**	**11**	**8**	**15**	**12**
#N	712	642	576	408	616	408	696	304	1224	976	848	832	1188	1036

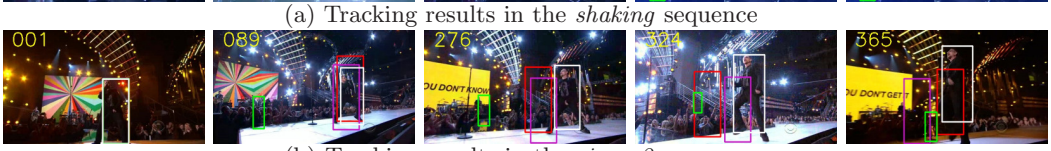
point changes, occlusions, noise, and motion blur. Moreover, we obtained new tracking sequences captured from real movies, such as *iron-man* and *matrix* sequences, where challenging appearance and motion changes exist. In these sequences, VTS most accurately tracked the targets as shown in Table 3.11. VTS robustly handled noise and motion blur by constructing the robust trackers that can cope with the current tracking environment and by further considering state representation and observation types to construct trackers. During the tracking process, VTS found the appropriate observation types by determining the variances of the Gaussian filter toward making the observation robust to noise, and identified the appropriate state model by separating the target into several fragments, which, when combined, are robust to motion blur. VTS also produced accurate tracking results in the conventional *tiger1*, *david*, and *occlface* sequences.

3.3.5.3 Qualitative Evaluation

Illumination change and pose variation: Figure 3.45 presents the tracking results in the *shaking* and *singer2* sequences. Although the stage lighting condition drastically changed and the pose of the object severely varied because of head-shaking or dancing, our method successfully tracked the object. Becasue our

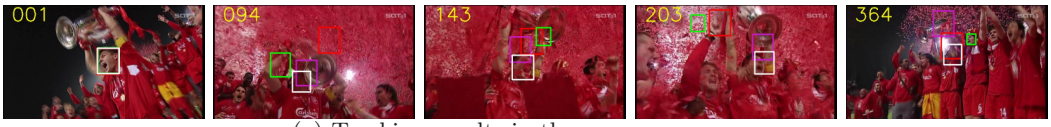


(a) Tracking results in the *shaking* sequence



(b) Tracking results in the *singer2* sequence

Figure 3.45: Tracking results when there are **severe illumination changes** and **pose variations**. White, green, red, and purple rectangles represent tracking results of VTS, CT, FRAGT, and MIL, respectively.



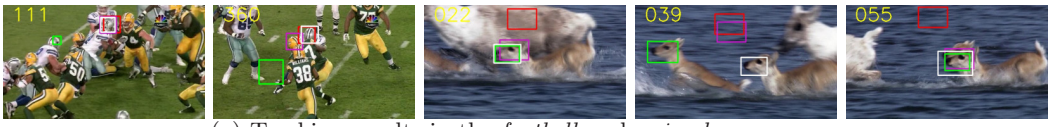
(a) Tracking results in the *soccer* sequence

(b) Tracking results in the *tiger1* and *occlface* sequences.

Figure 3.46: Tracking results when there are **severe occlusions** and **pose variations**. White, green, red, and purple rectangles represent tracking results of VTS, CT, FRAGT, and MIL, respectively.

appearance models evolve themselves by online update, our method efficiently covered pose variations. Additionally, the method was robust to illumination change because the appearance models utilized a mixture of templates. However, other methods failed to track the object when these changes occurred in combination as illustrated in Figure 3.45.

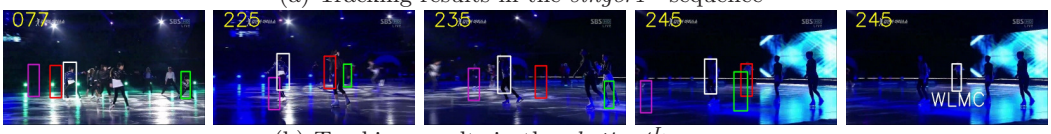
Occlusion and pose variation: Figure 3.46 demonstrates how the proposed method outperforms conventional tracking algorithms when the target is severely occluded by other objects. As shown in Figure 3.46, our method robustly tracked the object in the *soccer*, *tiger1*, and *occlface* sequences. The method was robust to occlusion because it constructed multiple observation models. Each model kept a



(a) Tracking results in the *football* and *animal* sequences.

Figure 3.47: Tracking results when there is **severe background clutter**. White, green, red, and purple rectangles represent tracking results of VTS, CT, FRAGT, and MIL, respectively.

(a) Tracking results in the *singer1*^{L} sequence



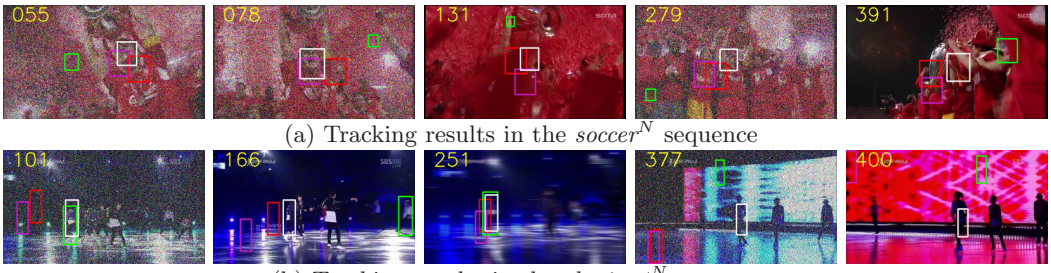
(b) Tracking results in the *skating1*^{L} sequence

Figure 3.48: Tracking results when there are **abrupt motions** and **severe illumination changes**. White, green, red, and purple rectangles represent tracking results of VTS, CT, FRAGT, and MIL, respectively.

different history of the object's appearance over time, which included the occluded and non-occluded appearance, as well as a mixture of both. Each model handled a different degree of occlusion. On the other hand, other methods failed to track the object accurately, as depicted in Figure 3.46.

Background clutters: In Figure 3.47, we tested the *football* and *animal* sequences which had severe background clutter, the appearance of which is similar to that of the target. In the case of other tracking methods, a trajectory was hijacked by a football player wearing a similar helmet of the target when the two players collided with each other at frame #360 in the *football* sequence. Our method resolved this problem and successfully tracked the target.

Abrupt motion and illumination change: For the tests, we made original videos, *singer1* and *skating1*, to have partially low frame rate. In the converted videos, the position and the scale of an object are drastically changed. Moreover,



(a) Tracking results in the $soccer^N$ sequence

(b) Tracking results in the $skating1^N$ sequence

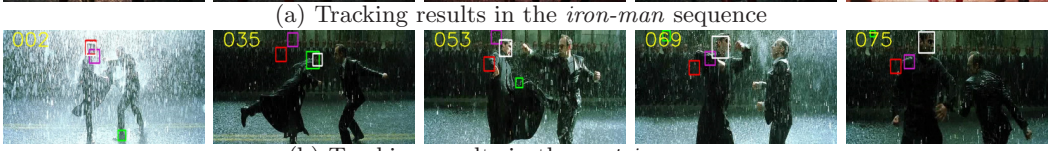
Figure 3.49: Tracking results when there are **abrupt motions** and **severe illumination changes**. White, green, red, and purple rectangles represent tracking results of VTS, CT, FRAGT, and MIL, respectively.

severe illumination changes translate the appearance of the object into different one. As shown in Figure 3.48, our method covered these changes and reliably tracked the object. However, other methods including WLMC [22] failed to track the object as described in Figure 3.48. Note that WLMC comprises the most recent state-of-the-art tracking methods that can cope with abrupt motions. However, WLMC could not deal with severe illumination changes at the same time.

Motion blur and noise: Figure 3.49 illustrates the tracking results in highly challenging sequences that have severe noise and motion blur as well. VTS accurately and robustly tracked the targets although severe types of appearance changes occurred simultaneously. Note that VTD successfully tracked the targets when only noise or motion blur existed. Otherwise, VTD failed to track the targets when noise and motion blur occurred with illumination changes, such as frame #377 of the $skating1^N$ sequence, and occurred with occlusions, such as frame #279 of the $soccer^N$ sequence.

Real movies: Figure 3.50 presents the tracking results under the real-world tracking environment utilizing the *iron-man* and *matrix* sequences. As shown in the figure, VTS covered most variations occurring in the sequences and robustly tracked

(a) Tracking results in the *iron-man* sequence



(b) Tracking results in the *matrix* sequence

Figure 3.50: Tracking results in **real movies**. White, green, red, and purple rectangles represent tracking results of VTS, CT, FRAGT, and MIL, respectively.

the target. However, MIL and CT failed to track the target accurately because of the severe appearance changes at frame #101 in the *iron-man* sequence and at frame #53 in the *matrix* sequence. Moreover, MIL and CT trackers were frequently hijacked by other objects at frame #69 in the *matrix* sequence, the appearance of which is similar to the target.

Algorithm 3 illustrates the whole process of our tracking method.

Algorithm 3 Proposed method

Input: $\mathbf{X}_{t-1}$, $\mathbf{T}_{t-1}$ **Output:** $\hat{\mathbf{X}}_t$, $\mathbf{T}_t$

$\alpha_t = 1, p_A = 0.1, p_M = 0.1, p_S = 0.1, p_O = 0.1$.
for 1 to n **do**
  1. State sampling
  Choose mode. Sample $\theta \sim U[0, 1]$.
  if $\theta < \alpha_t$ **then**
    for 1 to $|\mathbf{T}_t|$ **do**
      Accept $\mathbf{X}_t^j$ with the probability (3.63).
    end for
  else
    for 1 to $|\mathbf{T}_t|$ **do**
      Propose $\mathbf{X}_t{}^*$ using (3.61)and accept $\mathbf{X}_t{}^*$ with the probability (3.62).
    end for
  end if
  α_t linearly decreases from 1 to 0.0.
  2. Tracker sampling
  Choose move. Sample $\theta \sim U[0, 1]$.
  if $\theta < p_S$ **then**
    - Propose $\mathbf{S}_t{}^*$ using (3.39) or (3.40) with the probability 0.5, respectively.
    - Accept $\mathbf{S}_t{}^*$ with the probability (3.41).
  else if $\theta < p_S + p_O$ **then**
    - Propose $\mathbf{O}_t{}^*$ using (3.43) or (3.44) with the probability 0.5, respectively.
    - Accept $\mathbf{O}_t{}^*$ with the probability (3.45).
  else if $\theta < p_S + p_O + p_A$ **then**
    - Propose $\mathbf{A}_t{}^*$ using (3.54) or (3.55) with the probability 0.5, respectively.
    - Accept $\mathbf{A}_t{}^*$ with the probability (3.56).
  else if $\theta < p_S + p_O + p_A + p_M$ **then**
    - Propose $\mathbf{M}_t{}^*$ using (3.58) or (3.59) with the probability 0.5, respectively.
    - Accept $\mathbf{M}_t{}^*$ with the probability (3.60).
  end if
end for
Estimate $\hat{\mathbf{X}}_t$ using (3.37).

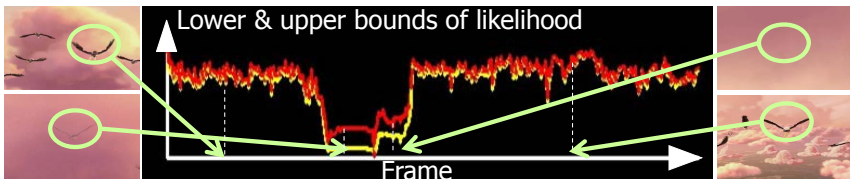
Chapter 4

Interval Analysis (IA) based Approaches

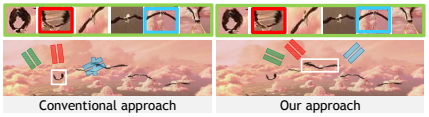
In this chapter, three tracking methods using the IA approach are proposed to solve ambiguity in probabilistic models. In section 4.1, the MUG tracker solves ambiguity in appearance models by minimizing uncertainty gap between the lower and upper bounds of the likelihood. In section 4.2, the SBB tracker solves ambiguity in state models to track highly non-rigid targets. In section 4.3, IT solves ambiguity in appearance and state models and proposes the M4 estimation.

4.1 The Minimum Uncertainty Gap (MUG) Tracker

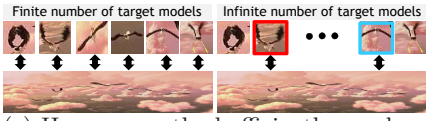
To robustly track the target, most conventional tracking methods design the tracking problem as the Bayesian formulation [10, 97, 33, 14, 98]. In the Bayesian tracking approach, the goal of the tracking problem is typically changed to find the best state, which maximizes the posterior probability $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$. To obtain the MAP



(a) Why our method considers the likelihood bounds?



(b) Why our method minimizes the likelihood bounds?



(c) How our method efficiently employs infinite number of target models?

Figure 4.1: **Basic idea of the proposed method**. (a) The likelihood bounds (uncertainty) are formed inevitably in real tracking situations due to different target models that are employed via different updating strategies during the tracking process. (b) A large gap between the upper and lower bounds indicates that the corresponding state gives very different answers (likelihoods) depending on the target models used (red and blue), although the average likelihood obtained by using set of all target models (green) is high. That means the likelihood estimation over that state is uncertain and unreliable. So, our method tries to find the state that has minimum gap (uncertainty), which gives consistent answers (likelihoods), regardless of the target models. And by maximizing the average likelihood bound at the same time, our method gets the state, which confidently maximizes the likelihood. (c) The proposed method only compares two target models with observations while utilizing the infinite number of target models, which generate lower and upper bounds of the likelihood. On the other hand, other methods compares all the finite number of target models to evaluate the likelihood.

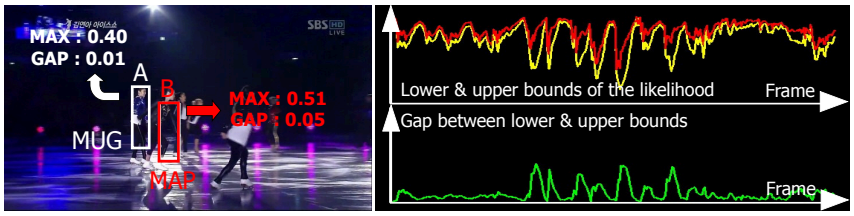
state, the method searches for the state that maximizes the likelihood $p(\mathbf{Y}_t|\mathbf{X}_t)$, which is near the previous state as a prior. In this case, the likelihood is typically calculated by measuring the similarity between the observation $\mathbf{Y}_t$ at the state $\mathbf{X}_t$ and the target model M_t at time t.

In this case, the MAP estimation assumes that the best state produces the highest likelihood score near by the previous state. However, in the real-world scenario, this assumption is not valid unless the target model M_t is always correct. In

practice, the target model is easily corrupted and distorted during the online update. To deal with severe appearance changes, many tracking algorithms evolve the target model with online update. However, because of the tracking error, the target model includes more background and becomes erroneous as time goes on. Eventually, the conventional trackers drift into the background and fail to track the target. This drift phenomenon frequently occurs even though the methods find the optimal MAP state because of the noisy target model. According to this fundamental inherent problem, the conventional MAP-based tracking approach need to be reconsidered.

Thus, we redefine the goal of tracking problem as to find *the best state that maximizes the average bound of the likelihood and, at the same time, minimizes the gap between bounds of the likelihood*. We call this the *Minimum Uncertainty Gap* (MUG) estimation. Note that in general tracking problem, the upper and lower bounds or the uncertainties of the likelihoods are naturally formed since many different likelihoods are made by different target models that are the reference appearances of the target. The different target models are usually constructed due to the different updating strategies during the tracking process [77]. Specially when there exist severe occlusions, illumination changes, and so on, the likelihood uncertainty becomes larger, as empirically demonstrated in Fig.4.1(a). This is because the distractors such as occlusions and illumination changes usually make the target models to be much different with each other.

Since the different likelihoods can be generated by different target models, obtaining the likelihood bounds is the same as considering all possible target models that could be constructed. Using the likelihood bounds, the proposed method can find the good target state because it implicitly covers all possible appearance changes of the target with all possible target models. Thus, as illustrated in Fig.4.1(b), the



(a) MAP and MUG (b) Likelihood bounds and gap

Figure 4.2: **Example of our tracking results** in *skating1* seq. Our method successfully tracks the target using the MUG estimation, whereas the conventional methods fail to track it using the MAP estimation. The MUG estimation finds the true state A of the target because the gap between the likelihood bounds in State A is smaller than that in State B. On the other hand, the MAP estimation finds the wrong state B because the posterior probability in State B is larger than that in State A.

large gap between the upper and lower bounds indicates that the corresponding state can have either a very good likelihood or a very bad likelihood depending on the employed target model. In this case, the likelihood estimation over the state is easily affected by the noisy target models and the estimated likelihood is uncertain and not reliable. Hence, by minimizing the gap between the two likelihood bounds, the proposed method can find the confident state of the target. MUG is also affected by aforementioned distractors (outliers) in the target model. Nevertheless, MUG is more robust to them. This is because MUG provides the confidence score about the likelihood estimation, whereas MAP cannot. Note that the outliers usually produce low confidence scores [99]. Therefore, MUG can easily identify and avoid them by estimating the confidence values of them. To measure the confidence of the likelihood, our method estimates the lower and upper bounds of the likelihood, minimizes the gap between the bounds, and accurately tracks the target, as shown in Fig.4.2.

4.1.1 New Objective of the Bayesian Tracker

To find the best state, $\hat{\mathbf{X}}_t$, our method obtains the Minimum Uncertainty Gap (MUG) at each time t as follows:

$$\hat{\mathbf{X}}_t = \arg\min_{\mathbf{X}_t} \frac{p_u(\mathbf{Y}_t|\mathbf{X}_t) - p_l(\mathbf{Y}_t|\mathbf{X}_t)}{p_u(\mathbf{Y}_t|\mathbf{X}_t) + p_l(\mathbf{Y}_t|\mathbf{X}_t)}, \tag{4.1}$$

where $p_l(\mathbf{Y}_t|\mathbf{X}_t)$ and $p_u(\mathbf{Y}_t|\mathbf{X}_t)$ denote the lower and upper bounds of the likelihood, respectively. In (4.1), our method finds the state that maximizes the average bound $[p_u(\mathbf{Y}_t|\mathbf{X}_t) + p_l(\mathbf{Y}_t|\mathbf{X}_t)]$ and minimizes the gap between bounds, $[p_u(\mathbf{Y}_t|\mathbf{X}_t) - p_l(\mathbf{Y}_t|\mathbf{X}_t)]$, at the same time. The best state (MUG state) at time t is represented as a three-dimensional vector $\hat{\mathbf{X}}_t = (\hat{X}_t^x, \hat{X}_t^y, \hat{X}_t^s)$, where $\hat{X}_t^x$, $\hat{X}_t^y$, and $\hat{X}_t^s$ indicate the x, y position and the scale of the target, respectively.

To obtain the MUG state, we need to estimate the lower and upper bounds of the likelihood. First, we define the likelihood as

$$p(\mathbf{Y}_t, \theta|\mathbf{X}_t) = exp^{-\lambda dist\left(\theta, \mathbf{Y}_t(\mathbf{X}_t)\right)}, \tag{4.2}$$

where $\mathbf{Y}_t(\mathbf{X}_t)$ denotes the observation at the state $\mathbf{X}_t$, $dist\big(\theta, \mathbf{Y}_t(\mathbf{X}_t)\big)$ represents the dissimilarity measure between the target model θ and the observation $\mathbf{Y}_t(\mathbf{X}_t)$, and λ is a weighting parameter. The observation and the target model are modeled by HSV histogram. The dissimilarity measure is designed by Bhattacharyya similarity coefficient [33]. As aforementioned, the main cause of the tracking failures is the noisy target models. Therefore, our method integrates out the target model θ in (4.2) and estimates the log marginal likelihood: $\int_\Theta \ln p(\mathbf{Y}_t, \theta|\mathbf{X}_t) d\theta$, where Θ denotes the whole target model space. To approximate the integral numerically, we obtain the mathematical lower (Jensen's inequality) and mathematical upper bounds (Gibbs'

inequality) of the marginal likelihood based on [100].

$$\ln p_l(\mathbf{Y}_t|\mathbf{X}_t,\gamma) = \int_\Theta q(\theta|\gamma,\mathbf{X}_t)\ln\frac{p(\mathbf{Y}_t,\theta|\mathbf{X}_t)}{q(\theta|\gamma,\mathbf{X}_t)}d\theta, \tag{4.3}$$

$$\ln p_u(\mathbf{Y}_t|\mathbf{X}_t,\gamma) = \int_\Theta p(\theta|\mathbf{Y}_t,\mathbf{X}_t)\ln\frac{p(\mathbf{Y}_t,\theta|\mathbf{X}_t)}{q(\theta|\gamma,\mathbf{X}_t)}d\theta, \tag{4.4}$$

where $q(\theta|\gamma,\mathbf{X}_t)$ is the prior distribution of the target model θ and γ is the hyper parameter of the distribution. In this paper, the target model θ is the HSV histogram [33]. Because $\mathbf{Y}_t$ is given and θ is marginalized out in (4.3)(4.4), the lower and upper bounds of the likelihood is the function of $\mathbf{X}_t$ and γ, which are a variable and a parameter, respectively. Then, the goal of our method is to find both the parameter and variable, which reduce gap between the likelihood bounds. Thus, our method composes two main parts as follows:

• **Parameter estimation (Section 4.1.2):** Using the MUG states, $\{\hat{\mathbf{X}}_i\}_{i=1}^t$, our method learns the parameter γ for time $t+1$. In our method, the parameter is not set empirically but is obtained analytically to maximize the lower bound in (4.3) and to minimize the upper bound in (4.4). Moreover, the parameter is not fixed to constant but is adaptively varied at each time t by the process in Section 4.1.2.

• **State inference (Section 4.1.3):** Given the parameter γ estimated at time $t-1$, our method finds the MUG state $\hat{\mathbf{X}}_t$ at time t, which produces the minimum uncertainty gap. To achieve the goal, the method searches states that maximize the average bound by increasing the denominator in (4.1). Thus, the method can obtain good quality of states with high likelihood scores. This advantage is similar to that of the MAP estimation. In addition, it prevents the best state with the minimum uncertainty gap from having a low likelihood score. Our method simultaneously searches states that minimize uncertainty of the likelihood estimation by decreasing the numerator in (4.1). Then, the method can avoid outliers which have a large

uncertainty gap, even though they have high likelihood scores. This advantage cannot be achieved in the MAP estimation.

4.1.2 Parameter Estimation

4.1.2.1 Parameter γ_u to Minimize Upper Bound

We learn the best parameter γ_u which minimizes the upper bound (4.4): $\gamma_u = \underset{\gamma}{\operatorname{argmin}} \ln p_u(\mathbf{Y}_t|\mathbf{X}_t, \gamma)$. Then, γ_u also minimizes the KL divergence $D(p \parallel q)$ because $\ln p_u(\mathbf{Y}_t|\mathbf{X}_t, \gamma) = D(p \parallel q) + \ln p(\mathbf{Y}_t|\mathbf{X}_t)$, where

$$D(p \parallel q) = \int_{\Theta} p(\theta|\mathbf{Y}_t, \mathbf{X}_t) \ln \frac{p(\theta|\mathbf{Y}_t, \mathbf{X}_t)}{q(\theta|\gamma, \mathbf{X}_t)} d\theta. \tag{4.5}$$

The parameter γ_u that minimizes $D(p \parallel q)$ satisfies $\mathbb{E}_{q(\theta|\gamma_u, \mathbf{X}_t)}[v(\theta)] = \mathbb{E}_{p(\theta|\mathbf{Y}_t, \mathbf{X}_t)}[v(\theta)]$, as derived in Appendix. This means that the minimization of KL divergence is equivalent to Moment Matching (MM) of θ [101]: the first and second moments of θ under the distribution $q(\theta|\gamma_u, \mathbf{X}_t)$ is equal to those under the distribution $p(\theta|\mathbf{Y}_t, \mathbf{X}_t)$. In (4.5), the prior $q(\theta|\gamma, \mathbf{X}_t)$ is designed as

$$q(\theta|\gamma, \mathbf{X}_t) = q(\theta|\mu, \sigma, \mathbf{X}_t) = \frac{1}{\sqrt{2\pi}\sigma} exp^{-\left(\frac{\theta-\mu}{\sqrt{2}\sigma}\right)^2}, \tag{4.6}$$

so the parameter $\gamma_u = (\mu_u, \sigma_u)$ is obtained for each bin in the HSV histogram:

• **1st MM:** Since the first moment of θ under $q(\theta|\gamma_u, \mathbf{X}_t)$ and under $p(\theta|\mathbf{Y}_t, \mathbf{X}_t) \approx p(\mathbf{Y}_t, \theta|\mathbf{X}_t)$ in (4.2) is μ_u and $\int_{\Theta} \theta p(\theta|\mathbf{Y}_t, \mathbf{X}_t) d\theta$, respectively, the following can be taken:

$$\mu_u = \int_{\Theta} \theta p(\theta|\mathbf{Y}_t, \mathbf{X}_t) d\theta. \tag{4.7}$$

In (4.7), the integration over θ is approximated using the Z samples of θ, $\{\theta_i\}_{i=1}^{Z}$, where θ_i is designed as $\mathbf{Y}_i(\hat{\mathbf{X}}_i)$, which indicates the observation around the MUG

state at the i-th recent frame. By substituting $\mathbf{X}_t = \hat{\mathbf{X}}_t$ and $p(\theta|\mathbf{Y}_t, \mathbf{X}_t) \approx p(\mathbf{Y}_t, \theta|\mathbf{X}_t)$ in (4.2) into (4.7), we get $\mu_u = \int_\Theta \theta p(\theta|\mathbf{Y}_t, \mathbf{X}_t)d\theta \approx \frac{1}{Z}\sum_{i=t-Z}^{t-1} \theta_i exp^{-\lambda dist\left(\theta_i, \mathbf{Y}_t(\hat{\mathbf{X}}_t)\right)}$.

• **2nd MM:** Since the second moment of θ under $q(\theta|\gamma_u, \mathbf{X}_t)$ and $p(\theta|\mathbf{Y}_t, \mathbf{X}_t)$ is σ_u and $\int_\Theta \left(\theta - \mu_u\right)^2 p(\theta|\mathbf{Y}_t, \mathbf{X}_t)d\theta$, respectively, the following can be taken:

$$\sigma_u = \int_\Theta \left(\theta - \mu_u\right)^2 p(\theta|\mathbf{Y}_t, \mathbf{X}_t)d\theta, \tag{4.8}$$

where $\int_\Theta \left(\theta - \mu_u\right)^2 p(\theta|\mathbf{Y}_t, \mathbf{X}_t)d\theta \approx \frac{1}{Z}\sum_{i=t-Z}^{t-1} \left(\theta_i - \mu_u\right)^2 exp^{-\lambda dist\left(\theta_i, \mathbf{Y}_t(\hat{\mathbf{X}}_t)\right)}$.

Finally, the global minimum of the upper bound of the likelihood at the state $\mathbf{X}_t$ in (4.4) is estimated based on [100]:

$$\ln p_u(\mathbf{Y}_t|\mathbf{X}_t, \gamma_u) \approx \frac{1}{Z}\sum_{i=t-Z}^{t-1} \ln \frac{p(\theta_i, \mathbf{Y}_t|\mathbf{X}_t)}{q(\theta_i|\gamma_u, \mathbf{X}_t)}, \tag{4.9}$$

where the integration in (4.4) is approximately obtained using the Z samples of θ, $\{\theta_i\}_{i=1}^{Z}$, where θ_i indicates the observation around the MUG state at the i-th recent frame.

4.1.2.2 Parameter γ_l to Maximize Lower Bound

We obtains the parameter γ_l which maximizes the lower bound $\ln p_l(\mathbf{Y}_t|\mathbf{X}_t, \gamma)$ in (4.3): $\gamma_l = \underset{\gamma}{\text{argmax}} \ln p_l(\mathbf{Y}_t|\mathbf{X}_t, \gamma)$. For this purpose, the gradient of $\ln p_l(\mathbf{Y}_t|\mathbf{X}_t, \gamma)$ in (4.3) is taken with respect to γ to zero:

$$\frac{d}{d\gamma} \ln p_l(\mathbf{Y}_t|\mathbf{X}_t, \gamma) = -\int_\Theta h(\theta|\gamma, \mathbf{X}_t) q(\theta|\gamma, \mathbf{X}_t)d\theta = 0, \tag{4.10}$$

where

$$\begin{aligned} h(\theta|\gamma, \mathbf{X}_t) &= \left[1 + \ln \frac{q(\theta|\gamma, \mathbf{X}_t)}{p(\mathbf{Y}_t, \theta|\mathbf{X}_t)}\right] \frac{d}{d\gamma} q(\theta|\gamma, \mathbf{X}_t) = \\ h(\theta|\mu, \sigma, \mathbf{X}_t) &= \left[1 + \ln \frac{q(\theta|\mu, \sigma, \mathbf{X}_t)}{p(\mathbf{Y}_t, \theta|\mathbf{X}_t)}\right] \begin{bmatrix} \frac{\partial}{\partial \mu} q(\theta|\mu, \sigma, \mathbf{X}_t) \\ \frac{\partial}{\partial \sigma} q(\theta|\mu, \sigma, \mathbf{X}_t) \end{bmatrix}. \end{aligned} \tag{4.11}$$

[1] To find the parameter $\gamma_l = (\mu_l, \sigma_l)$ that satisfies (4.10), the quasi-optimized lower bound is estimated by Stochastic Approximation Monte Carlo (SAMC) in [102] to define the recursive approximation of the solution of $\frac{d}{d\gamma} \ln p_l(\mathbf{Y}_t|\mathbf{X}_t, \gamma) = 0$. The SAMC algorithm then iteratively updates a sequence of values via the recursion:

$$\gamma^{(n+1)} = \gamma^{(n)} + s^{(n+1)}\omega(\gamma^{(n)}), \omega(\gamma^{(n)}) = -\int_{\Theta} h(\theta|\gamma^{(n)}, \mathbf{X}_t)d\theta, \tag{4.12}$$

where $\int_{\Theta} h(\theta|\gamma^{(n)}, \mathbf{X}_t)d\theta \approx \frac{1}{Z}\sum_{i=t-Z}^{t-1} h(\theta_i|\gamma^{(n)}, \hat{\mathbf{X}}_t)$. $\gamma^{(n+1)}$ denotes approximation of the γ value at *(n+1)*-th iteration, and $s^{(n+1)}$ indicates the modification factor at *(n+1)*-th iteration, which linearly decreases from 0.5 to 0.1 as time goes on. After the predefined iterations N, we get $\gamma^{(N)} = \gamma_l = (\mu_l, \sigma_l)$.

Then, the final estimate of the lower bound at the state $\mathbf{X}_t$ in (4.3) is

$$\ln p_l(\mathbf{Y}_t|\mathbf{X}_t, \gamma_l) \approx \frac{1}{Z}\sum_{i=t-Z}^{t-1} \ln \frac{p(\theta_i, \mathbf{Y}_t|\mathbf{X}_t)}{q(\theta_i|\gamma_l, \mathbf{X}_t)}. \tag{4.13}$$

4.1.3 State Inference

To find the best state that satisfies (4.1) with the fixed parameters γ_l and γ_u, our method utilizes the IMCMC sampling method [27]. In the IMCMC sampling method, two markov chains are designed. The first chain finds the state that maximizes the average likelihood bound. The second finds the state that minimizes the gap between the bounds.

The IMCMC sampling method consists of two modes, parallel and interacting. In the parallel mode, our method acts as the parallel Metropolis Hastings algorithm and separately obtains samples over those chains via two main steps: the proposal

[1] In (4.11), $[1 + \ln \frac{q(\theta|\mu,\sigma,\mathbf{X}_t)}{p(\mathbf{Y}_t,\theta|\mathbf{X}_t)}] = [\lambda dist\big(\theta, \mathbf{Y}_t(\mathbf{X}_t)\big) - \big(\frac{\theta-\mu}{\sqrt{2}\sigma}\big)^2 - \ln\sqrt{2\pi\sigma^2} + 1]$, $\frac{\partial}{\partial\mu} q(\theta|\mu, \sigma, \mathbf{X}_t) = \frac{1}{\sqrt{2\pi}\sigma} exp^{-\big(\frac{\theta-\mu}{\sqrt{2}\sigma}\big)^2} \frac{1}{\sigma^2}(\theta - \mu)$, and $\frac{\partial}{\partial\sigma} q(\theta|\mu, \sigma, \mathbf{X}_t) = -\frac{1}{2\sigma^2} + \frac{(\theta-\mu)^2}{2\sigma^2}$.

step and the acceptance step. At the proposal step, a new state is proposed by the proposal density function.

$$Q_(\mathbf{X}_t^*; \mathbf{X}_t) = G(\mathbf{X}_t, \sigma_p^2), \tag{4.14}$$

where Q denotes the proposal density function, G represents the Gaussian distribution with mean $\mathbf{X}_t$ and variance σ_p^2, and $\mathbf{X}_t^*$ represents a new state at time t. Given the proposed state, each chain decides whether the state is accepted or not with the acceptance ratio in the acceptance step:

$$\begin{aligned} a_1^p &= min\left[1, \frac{\left[p_u(\mathbf{Y}_t|\mathbf{X}_t^*, \gamma_u) + p_l(\mathbf{Y}_t|\mathbf{X}_t^*, \gamma_l)\right] Q(\mathbf{X}_t; \mathbf{X}_t^*)}{\left[p_u(\mathbf{Y}_t|\mathbf{X}_t, \gamma_u) + p_l(\mathbf{Y}_t|\mathbf{X}_t, \gamma_l)\right] Q(\mathbf{X}_t^*; \mathbf{X}_t)}\right], \\ a_2^p &= min\left[1, \frac{\left[p_u(\mathbf{Y}_t|\mathbf{X}_t, \gamma_u) - p_l(\mathbf{Y}_t|\mathbf{X}_t, \gamma_l)\right] Q(\mathbf{X}_t; \mathbf{X}_t^*)}{\left[p_u(\mathbf{Y}_t|\mathbf{X}_t^*, \gamma_u) - p_l(\mathbf{Y}_t|\mathbf{X}_t^*, \gamma_l)\right] Q(\mathbf{X}_t^*; \mathbf{X}_t)}\right], \end{aligned} \tag{4.15}$$

where $p_l(\mathbf{Y}_t|\mathbf{X}_t^*, \gamma_l)$ in (4.13) and $U_{\gamma_u}(\mathbf{X}_t^*)$ in (4.9) denote the estimated lower and upper bounds of the likelihood at the state $\mathbf{X}_t^*$, respectively. These steps iteratively proceed until the number of iterations reaches the predefined value R.

When the method is in the interacting mode, the trackers communicate with the others and make leaps to better states of the target. Due to the interaction mode, our method can find the common state, which maximizes the the average likelihood bound and, at the same time, minimizes the gap between bounds. A chain accepts the state of the chain 1 as its own state with the probability a_1^i, which maximizes the average likelihood bound. Similarly, a chain accepts the state of the chain 2 as

its own state with the probability a_2^i, which minimizes the gap between bounds:

$$\begin{aligned} a_1^i &= \frac{[p_u(\mathbf{Y}_t|\mathbf{X}_t,\gamma_u)+p_l(\mathbf{Y}_t|\mathbf{X}_t,\gamma_l)]-\Delta_1}{2p_l(\mathbf{Y}_t|\mathbf{X}_t,\gamma_l)-\Delta_1+\Delta_2}, \\ a_2^i &= \frac{\Delta_2-[p_u(\mathbf{Y}_t|\mathbf{X}_t,\gamma_u)-p_l(\mathbf{Y}_t|\mathbf{X}_t,\gamma_l)]}{2p_l(\mathbf{Y}_t|\mathbf{X}_t,\gamma_l)-\Delta_1+\Delta_2}, \\ \Delta_1 &= \mathrm{MAX}\left(\left[p_u(\mathbf{Y}_t|\hat{\mathbf{X}}_{t-1})+p_l(\mathbf{Y}_t|\hat{\mathbf{X}}_{t-1})\right]-\frac{1}{4},0\right), \\ \Delta_2 &= \mathrm{MIN}\left(\left[p_u(\mathbf{Y}_t|\hat{\mathbf{X}}_{t-1})-p_l(\mathbf{Y}_t|\hat{\mathbf{X}}_{t-1})\right]+\frac{1}{4},1\right). \end{aligned} \tag{4.16}$$

In (4.16), $[p_u(\mathbf{Y}_t|\mathbf{X}_t,\gamma_u)+p_l(\mathbf{Y}_t|\mathbf{X}_t,\gamma_l)]-\Delta_1$ indicates the increased quantity of the average likelihood bound and $\Delta_2-[p_u(\mathbf{Y}_t|\mathbf{X}_t,\gamma_u)-p_l(\mathbf{Y}_t|\mathbf{X}_t,\gamma_l)]$ represents the decreased quantity of the gap between bounds. $p_l(\mathbf{Y}_t|\hat{\mathbf{X}}_{t-1})$ and $p_u(\mathbf{Y}_t|\hat{\mathbf{X}}_{t-1})$ are the lower and upper bounds of the likelihood on the best state at the previous frame with the best parameter, respectively. Our method operates in the interacting mode with the probability α, which linearly decreases from 1.0 to 0.0 as the simulation goes on. Notably, the IMCMC method [27] typically converges to the invariant distribution $\frac{p_u(\mathbf{Y}_t|\mathbf{X}_t)-p_l(\mathbf{Y}_t|\mathbf{X}_t)}{p_u(\mathbf{Y}_t|\mathbf{X}_t)+p_l(\mathbf{Y}_t|\mathbf{X}_t)}$ in (4.1). Algorithm 4 illustrates the whole process of our tracking method.

4.1.4 Experimental Results

For the experiments, publicly available video sequences obtained from [56, 24, 27, 46, 98, 52] were utilized. Using the sequences, the proposed method (MUG) was analyzed and compared with five state-of-the-art tracking methods. In all experiments, λ in (4.2) is set to 5. Z in (4.9) and (4.13) is set to 15. σ_p in (4.14) is set to $\sigma_p^x=\sqrt{4}, \sigma_p^y=\sqrt{2}$, and $\sigma_p^s=\sqrt{0.01}$, where σ_p^x, σ_p^y, and σ_p^s denote the variance of the x translation, y translation and the scale, respectively. Please note that our method always use the same parameters throughout the experiments and the parameters of

Algorithm 4 Minimum Uncertainty Gap tracker

Input: $\mathbf{X}_{t-1} = (\mathbf{X}^x_{t-1}, \mathbf{X}^y_{t-1}, \mathbf{X}^s_{t-1}), \alpha = 1$

Output: $\hat{\mathbf{X}}_t = (\hat{\mathbf{X}}^x_t, \hat{\mathbf{X}}^y_t, \hat{\mathbf{X}}^s_t)$

while all frames **do**
  for 1 to R **do**
    Choose mode. Sample $\rho \sim U[0, 1]$.
    if $\rho < \alpha$ **then**
      Chain 1 changes its state into that of Chain 2 with probability a^i_2 in (4.16).
      Chain 2 changes its state into that of Chain 1 with probability a^i_1 in (4.16).
    else
      Chain 1 proposes a state using Q_1 in (4.14) and accepts the state with probability a^p_1 in (4.15).
      Chain 2 proposes a state using Q_2 in (4.14) and accepts the state with probability a^p_2 in (4.15).
    end if
    Decrease the α value.
  end for
  Estimate the MUG state, $\hat{\mathbf{X}}_t$ using (4.1).
  Determine γ_u using (4.7) and (4.8).
  Determine γ_l using (4.12).
end while

other methods were adjusted to show the best tracking performance. Same initializations were set to all methods for fair comparison. The software provided by the authors were used to obtain the tracking results of IVT, MIL, FRAGT, and VTS.

4.1.4.1 Analysis of the Proposed Method

Lower and Upper Bounds of the Likelihood: The tracking environments are examined when the gaps between the likelihood bounds are maximized. As illus-

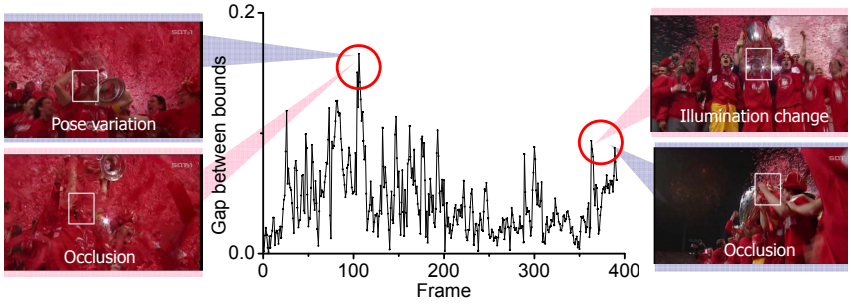
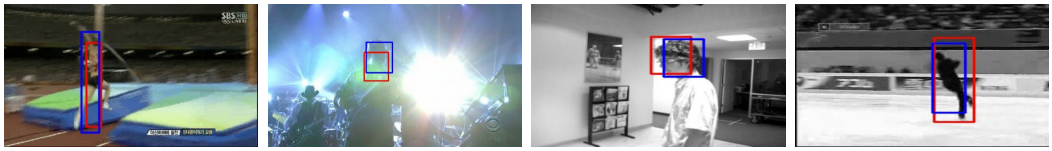


Figure 4.3: **Tracking environments** when the gaps between the likelihood bounds are maximized in *soccer* seq.



(a) *high-jump* (b) *shaking* (c) *david* (d) *skater*

Figure 4.4: **States** of the target, which produce the maximum lower bound (blue rectangle) and the minimum upper bound (red rectangle) of the likelihood at a frame.

trated in Figure 4.3, the gaps between the likelihood bounds were maximized when there were severe occlusions, pose variations, or illumination changes. These changes caused the target appearance to become noisy. Because of the noisy appearance, the estimated likelihood became very uncertain. This uncertainty of the likelihood produced the large gap between the lower and upper bounds of the likelihood in our method. In other words, the likelihood cannot be uniquely determined, especially when the tracking environments include the aforementioned appearance changes. Therefore, the proposed method considers the uncertainty of the likelihood to track the target robustly in real-world situations.

The states of the target that produce the maximum lower bound or the minimum upper bound of the likelihood are also checked. In (4.7)(4.8), the target model for the minimum upper bound is constructed by averaging the target appearance in

Table 4.1: **Comparison of tracking results** using MAP. ML, and MUG. The numbers indicate the average center location errors in pixels. The improvement score is calculated by dividing the tracking error of ML3 by that of MUG.

	bird1	*bird2*	*lemming*	*woman*	*soccer*	*skating1*	*diving*	*high-jump*	*skater*
MAP	199	45	11	127	51	115	26	70	30
ML1	208	47	12	137	56	110	43	65	47
ML2	201	51	16	138	61	107	46	71	51
ML3	210	42	11	101	49	150	27	71	37
MUG	13	11	16	14	32	17	14	30	17
Score	**16**	**3.8**	**0.7**	**7.2**	**1.5**	**8.8**	**1.9**	**2.3**	**2.2**

the recent frames. So, the model is adequate to track the target whose appearance smoothly changes over time. Then, the state that produces the minimum upper bound is the best state when the smooth changes in the target appearance are assumed, as shown in Figure 4.4(a)(c). In (4.10)(4.11), the target model for the maximum lower bound is heavily updated if the current observation is vastly different from the model. So, the model is robust to track the target whose appearance abruptly changes at a certain time. Then, the state that produces the maximum lower bound is the best state when the abrupt changes in the target appearance are assumed, as shown in Figure 4.4(b)(d). Therefore, the state that reduces the gap between two bounds in our method is the best state for both smooth and abrupt changes in target appearance.

Performance of MUG and IMCMC: To evaluate the performance of MUG, we used the same likelihood function that employs the Bhattacharyya coefficient as the similarity measure and the HSV color histogram as the feature. The only difference is how to determine the best state. The best states estimated by MAP is the one which maximize the posterior probability. The best states estimated by ML1, ML2, and ML3 are the ones which maximize the lower likelihood bound, upper likelihood bound, and average likelihood bounds, respectively. The best state obtained by

Table 4.2: **Comparison of tracking results** using MCMC and IMCMC. The improvement score is calculated by dividing the tracking error of MCMC by that of IMCMC.

	bird1	*bird2*	*lemming*	*woman*	*soccer*	*skating1*	*diving*	*high-jump*	*skater*
MCMC	24	31	52	40	47	89	32	65	51
IMCMC	13	11	16	14	32	17	14	30	17
Score	**1.8**	**2.8**	**3.3**	**2.9**	**1.5**	**5.2**	**2.3**	**2.2**	**3.0**

MUG is the one which minimizes the gaps between the lower and upper bounds of the likelihood. As shown in Table 4.1, the best state obtained by MUG gives more accurate tracking results. These results demonstrated the state which produces the maximum likelihood score and the maximum posterior probability do not always correspond to the true target state in real-world settings. Additionally, the results shows the tracking methods should consider the uncertainty of the estimated likelihood by measuring the gaps between the likelihood bounds like our method.

To evaluate the performance of IMCMC, the same MUG strategy is used to determine the best state. The only difference is the procedure in finding the best state. The best state in MCMC is found by using a single chain, in which the chain finds the state that maximizes the lower bound and minimizes the upper bound simultaneously. The best state in IMCMC is obtained by employing two chains, in which one chain only searches for the state that maximizes the lower bound and the other only searches for the state that minimizes the upper bound. Then, two chains exchange information about good states to reduce the gap between the lower and upper bounds. As indicated in Table 4.2, using two chains shows better tracking performance because the tracking methods using a single chain get trapped in local optima more frequently as the target distribution becomes complex. The target distribution is complex because the different two types of the likelihood distribution are mixed in a single distribution. Our method divides a complex distribution into

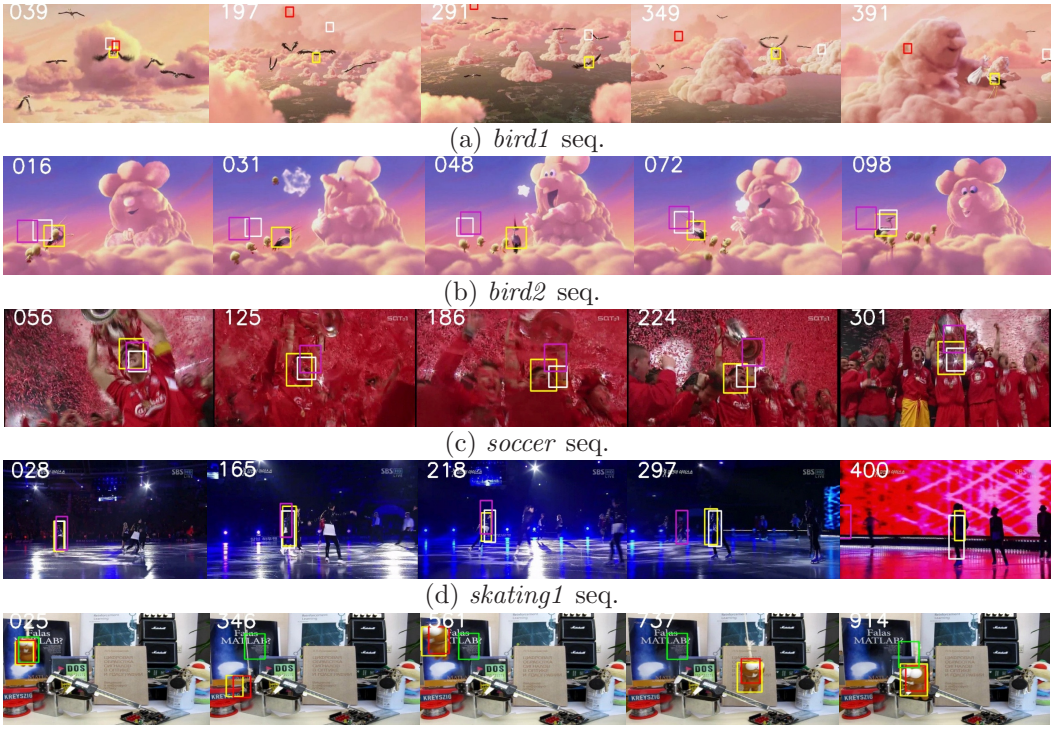
Table 4.3: **Comparison of tracking results.** The numbers indicate the average center location errors in pixels. Red is the best result and blue is the second-best. Other numbers in () indicate the percent of successfully tracked frames, where tracking is success when the overlap ratio between the predicted bounding box A_p and ground truth bounding box A_g: $\frac{area(A_p \bigcap A_g)}{area(A_p \bigcup A_g)}$.

	MC [18, 33]	**IVT [1]**	**FRAGT [56]**	**MIL [6]**	**VTS [27]**	**MUG**
bird1	215 (16)	230 (13)	228 (13)	270 (11)	**119** (13)	**13** (43)
bird2	40 (18)	115 (11)	24 (67)	**13** (81)	81 (23)	**11** (86)
lemming	**12** (85)	**14** (79)	84 (26)	**14** (51)	70 (45)	16 (71)
woman	138 (11)	133 (11)	112 (19)	120 (15)	**111** (19)	**14** (62)
soccer	53 (15)	116 (9)	82 (11)	41 (17)	**15** (35)	**32** (20)
skating1	172 (14)	213 (11)	93 (26)	85(31)	**8**(93)	**17**(40)
diving	**27** (24)	79 (20)	64 (20)	73 (20)	80 (20)	**14** (24)
high-jump	73 (15)	79 (15)	**69** (15)	91 (15)	143 (14)	**30** (17)
skater	28 (47)	86 (41)	**23** (61)	85 (41)	25 (66)	**17** (66)
Speed(fps)	**7**	**6**	**2**	**17**	**0.4**	**3**

two simple ones using IMCMC, where each distribution is constructed by each bound of the likelihood.

4.1.4.2 Comparison with other Tracking methods

As summarized in Table 4.3, our method (MUG) most accurately tracked the targets in terms of the center location error and the success rate, even though there are several types of appearance changes. VTS showed the second-best tracking performance. Our method was robust to the geometric appearance changes of the non-rigid target in *diving* seq., *high-jump* seq., and *skater* seq.; the occlusions in *bird1* seq., and *woman* seq.; and the motion blur in *bird2* seq. In this paper, we wanted to demonstrate that our method can produce better tracking results by utilizing a very simple likelihood function and its lower and upper bounds. In using the simple likelihood function, the method was much faster and more accurate than VTS. The tracking performance of the proposed method can be further enhanced if more advanced likelihood functions are employed. Additionally, with the simple likelihood function, our method produced more accurate tracking results than other



(a) *bird1* seq.

(b) *bird2* seq.

(c) *soccer* seq.

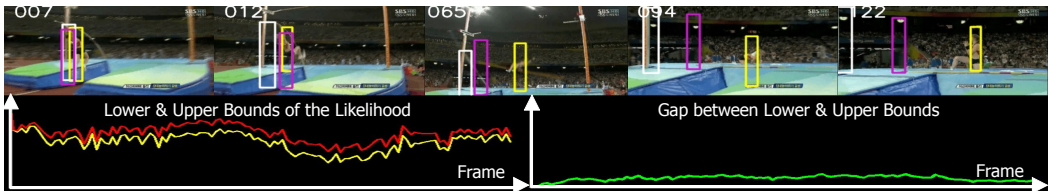
(d) *skating1* seq.

(e) *lemming* seq.

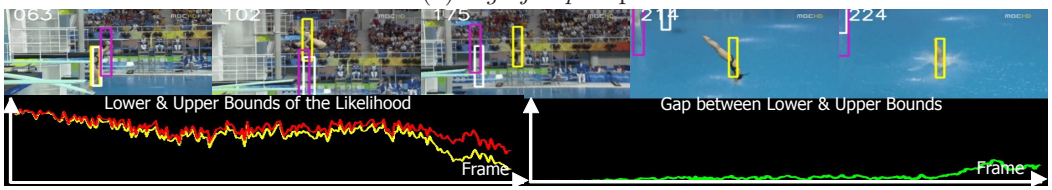
Figure 4.5: **Tracking results** in several challenging sequences. Yellow, white, purple, green and red rectangles represent tracking results of MUG, VTS, MIL, FRAGT, and MC, respectively. Yellow and red curves represent lower and upper bounds of the likelihood over time in MUG, respectively. Green curve represents gap between the bounds over time in MUG.

state-of-the-art methods (IVT, FRAGT, MIL), where they are robust to the pose variations, occlusions, and illumination changes. Note that, for the sampling-based methods MC, IVT, FRAGT, VTS, and MUG, we used the same number of samples to track the target. We adjusted the total number of samples to that of VTS.

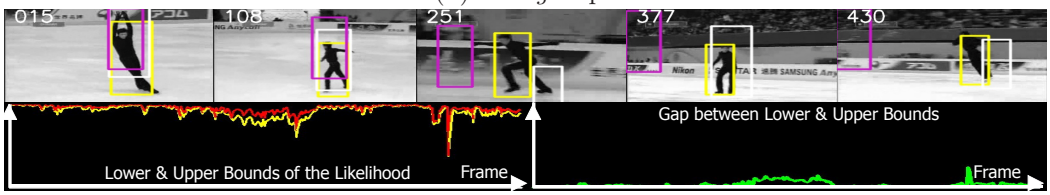
Figure 4.5 and 4.6 demonstrate how the proposed method outperforms the state-of-the-art methods in several challenging sequences. In Figure 4.5, the tracking performance under the severe occlusions and background clutters was tested. When the sequence contained several appearance changes of the target at the same time, our method robustly tracked the target over time, while other tracking methods fre-



(a) *high-jump* seq.



(b) *diving* seq.



(c) *skater* seq.

(d) *woman* seq.

Figure 4.6: **Tracking results** with lower and upper bounds of the likelihood obtained by MUG.

quently missed the targets. The tracking results of MIL drifted into the background when the aforementioned changes transformed the target appearance into a different one. Our method overcame this problem and successfully tracked the target by evaluating the target configuration with several likelihoods. In Figure 4.6, our method did not miss the target in all frames, although the sequences include the severe geometric appearance changes of the target. On the other hand, MIL and VTS frequently failed to track the target when the target was severely deformed. Our method was more efficient than VTS in terms of the computational cost because it utilized two likelihoods only for evaluating the configuration by estimating the lower and upper bounds of the likelihood. Additionally, our method utilized the gap

between different likelihood bounds, so the method was free from the normalization problem in VTS caused when the several different likelihood are used directly to evaluate the target appearance. Note that our method is highly applicable because conventional tracking methods can be also enhanced by replacing the ML and MAP with the MUG estimation. This scenario has been proven for a simple MCMC-based tracking framework with the experiment in Table 4.1.

4.1.5 Appendix

To minimize the KL divergence in (4.5), $q(\theta|\gamma, \mathbf{X}_t)$ is assumed to be exponential and $D(p \parallel q)$ is a linear functional of $q(\theta|\gamma, \mathbf{X}_t)$. Then, $\ln q(\theta|\gamma, \mathbf{X}_t)$ is convex with respect to γ. Hence, the global minimum of the KL divergence can be found. As aforementioned, we design $q(\theta|\gamma, \mathbf{X}_t)$ as the exponential family of distributions with the following form:

$$q(\theta|\gamma, \mathbf{X}_t) = g(\gamma)h(\theta)exp(\gamma^T v(\theta)), \tag{4.17}$$

where $h(\theta)$ and $v(\theta)$ are functions from the space of possible values of θ to the real numbers and $g(\gamma)$ is a normalization factor:

$$g(\gamma) \int_{\Theta} h(\theta)exp(\gamma^T v(\theta))d\theta = 1. \tag{4.18}$$

By taking the gradient of both sides with respect to γ, the following can be taken:

$$\nabla g(\gamma) \int_{\Theta} h(\theta)exp(\gamma^T v(\theta))d\theta \int_{\Theta} g(\gamma)h(\theta)exp(\gamma^T v(\theta))v(\theta)d\theta = 0, \tag{4.19}$$

Since $g(\gamma)h(\theta)exp(\gamma^T v(\theta)) = q(\theta|\gamma, \mathbf{X}_t)$ from (4.17) and $\int_{\Theta} h(\theta)exp(\gamma^T v(\theta))d\theta = \frac{1}{g(\gamma)}$ from (4.18), (4.19) is changed into

$$-\nabla \ln g(\gamma) = \mathbb{E}_{q(\theta|\gamma, \mathbf{X}_t)} \left[v(\theta) \right]. \tag{4.20}$$

By substituting $q(\theta|\gamma, \mathbf{X}_t)$ in (4.17) into the KL divergence in (6), we get

$$D(p \parallel q) = -\ln g(\gamma) - \gamma^T \mathbb{E}_{p(\theta|\mathbf{Y}_{1:t}, \mathbf{X}_t)} [v(\theta)] + const. \tag{4.21}$$

To minimize KL divergence in (4.21), the gradient of $D(p \parallel q)$ is taken with respect to γ to zero. And we get

$$-\nabla \ln g(\gamma) = \mathbb{E}_{p(\theta|\mathbf{Y}_{1:t}, \mathbf{X}_t)} [v(\theta)] . \tag{4.22}$$

Since $-\nabla \ln g(\gamma)$ in (4.20) is equal to $-\nabla \ln g(\gamma)$ in (4.22), we finally get

$$\mathbb{E}_{q(\theta|\gamma, \mathbf{X}_t)} [v(\theta)] = \mathbb{E}_{p(\theta|\mathbf{Y}_{1:t}, \mathbf{X}_t)} [v(\theta)] . \tag{4.23}$$

4.2 The Soft Bounding Box (SBB) Tracker

The bounding box representation is widely used because it allows the easy inference of the best configuration using a low-dimensional vector [47, 10, 26, 13]. In addition, with the representation, tracking methods in the tracking by detection approach easily train a classifier [6, 7, 50, 51, 52, 55].

However, the single bounding box representation has inherent ambiguity. Note that in general no single bounding box representation can cover the whole region of the target while excluding all the background region, as shown in Fig.4.7(a) and (b). In this case, the tracking methods may choose either inner bounding box representation or outer bounding box representation. However, although the inner bounding box representation in Fig.4.7(a) can deliver the pure target region, it looses lots of useful information of the target by excluding large parts of it. While, the outer bounding box representation in Fig.4.7(b) includes the whole target region, but it inevitably contains unwanted background region. Now, then the natural question

(a) Inner bounding box representation (b) Outer bounding box representation (c) Soft bounding box representation

Figure 4.7: **Example of the different bounding box representation for non-rigid objects**. (a) This representation only includes the foreground regions but excludes some parts of the target. (b) This representation includes whole foreground regions but also includes some background regions. (c) In our method, the bounding box of the target is represented as a range from the inner bounding box (red) to the outer bounding box (blue).

is *which box representation can best describe a general and non-rigid target*. The present paper aims to resolve this ambiguity in the bounding box representation, and to track a non-rigid target robustly using a new bounding box representation.

We note that the best single bounding box representation might exist somewhere between the inner bounding box and the outer bounding box. However, the best representation cannot be uniquely determined because of the inherent ambiguity. So, the bounding box of the target should be represented as a range from the inner bounding box to the outer bounding box. We call this representation as the soft bounding box (SBB) representation. By describing the bounding box as a range, the proposed method solves the inherent ambiguity in the bounding box representation. In addition, this representation improves tracking performance because it does not consider the ambiguous regions (blue region in Fig.4.7(c)) in determining the probable configuration of the target, which contain mixed target and background regions. Instead, our method only considers the *maximal pure foreground region*, inside of the inner bounding box (red region in Fig.4.7(c)) and the *pure background region*,

outside of the outer bounding box (green region in Fig.4.7(c)) in determining the probable configuration of the target.

Our method finds the the optimal inner and outer bounding boxes using the Dempster-Shafer (DS) theory [103, 104]. Following the DS theory, the optimal inner bounding box is obtained by maximizing the similarity between the red region and the target model; and the optimal outer bounding box is obtained by maximizing the dissimilarity between the green region and the target model in Section 4.2.1.2. Then, the objective of our method is to find the best state of the inner and outer bounding boxes using (4.34) in Section 4.2.2.1, which maximizes the posterior probability with the following constraint. Our method inferences the outer bounding box to include the inner bounding box using the Constrained Markov Chain Monte Carlo (CMCMC)-based sampling method in Section 4.2.2.2. This is called Constrained Maximum a Posteriori (CMAP) estimate.

4.2.1 Design of the Soft Bounding Box

4.2.1.1 Dempster-Shafer Theory in the Bounding Box Representation

In the tracking method, a state $\mathbf{X}_t$ describes a bounding box at t-th frame, so the bounding box can also be denoted by $\mathbf{X}_t$. Using the Shafer's framework, the probability density of the bounding box $p(\mathbf{X}_t)$ can be represented as intervals bounded by two values, i.e., belief and plausibility:

$$bel(\mathbf{X}_t) \leq p(\mathbf{X}_t) \leq pl(\mathbf{X}_t), \tag{4.24}$$

where $bel(\mathbf{X}_t)$ and $pl(\mathbf{X}_t)$ denote the belief and the plausibility of the bounding box, respectively. In (4.24), the belief is constituted by the sum of all the likelihood scores from all sub-regions enclosed by the bounding box. It is the amount of belief that

directly supports the bounding box at least in part, forming a lower bound [105].

$$bel(\mathbf{X}_t) = \sum_{\mathbf{R}_t|\mathbf{R}_t \subset \mathbb{R}(\mathbf{X}_t)} p(\mathbf{Y}_t|\mathbf{R}_t), \tag{4.25}$$

where $\mathbb{R}(\mathbf{X}_t)$ returns the whole region enclosed by the bounding box, $\mathbf{R}_t$ denotes the sub-region enclosed by the bounding box, and $p(\mathbf{Y}_t|\mathbf{R}_t)$ indicates the likelihood of the sub-region given the observation, $\mathbf{Y}_t$. In (4.25), the likelihood is represented by

$$p(\mathbf{Y}_t|\mathbf{R}_t) = exp^{-\lambda dist\left(M_t, \mathbf{Y}_t(\mathbf{R}_t)\right)}, \tag{4.26}$$

where $\mathbf{Y}_t(\mathbf{R}_t)$ denotes the observation of the region $\mathbf{R}_t$, $dist\big(M_t, \mathbf{Y}_t(\mathbf{R}_t)\big)$ indicates the distance between the observation and the target model M_t, and λ is the weighting parameter. For the observation and the distance measure, we utilize the HSV color histogram and the Bhattacharyya similarity coefficient in [33]. In (4.26), the likelihood is normalized, such that the sum of the likelihood scores of all possible regions is 1:

$$\sum_{\mathbf{R}_t|\mathbf{R}_t \subset \mathbf{U}_t} p(\mathbf{Y}_t|\mathbf{R}_t) = 1, \tag{4.27}$$

where $\mathbf{U}_t$ denotes a universal set that contains all possible regions.

The plausibility in (4.24) is one minus the sum of the likelihood scores of all the regions whose intersection with the region enclosed by the bounding box is empty. It is an upper bound on the possibility that the bounding box could be true up to that value, because there is only so much evidence which contradicts that bounding box [105]:

$$pl(\mathbf{X}_t) = \sum_{\mathbf{R}_t|\mathbf{R}_t \cap \mathbb{R}(\mathbf{X}_t) \neq \phi} p(\mathbf{Y}_t|\mathbf{R}_t), \tag{4.28}$$

where the plausibility can be efficiently estimated by $pl(\mathbf{X}_t) = 1 - bel(\overline{\mathbf{X}}_t)$, where $\overline{\mathbf{X}}_t$ indicates all regions whose intersection with the region enclosed by the bounding

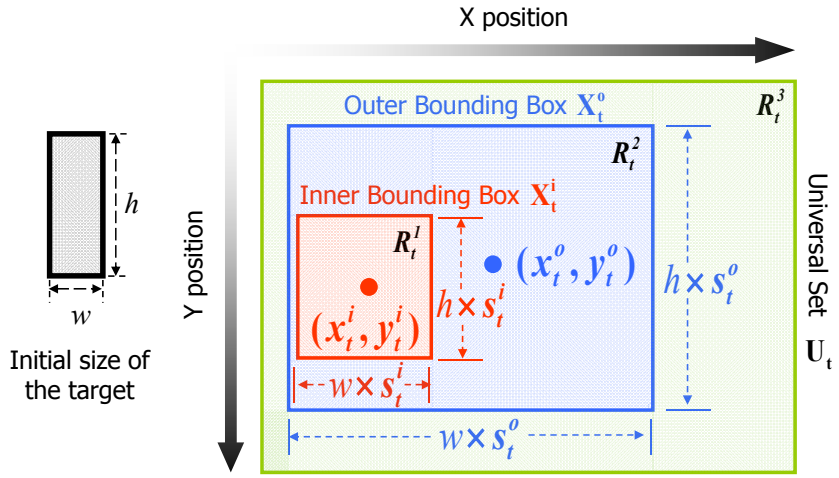


Figure 4.8: **Notation of the SBB representation**

box is empty. With the belief and the plausibility of the bounding box estimated by (4.25) and (4.28), respectively, our method determines the inner and outer bounding boxes, as explained in the next section.

4.2.1.2 Inner and Outer Bounding Boxes

The inner bounding box $\mathbf{X}_t^i$ divides the whole region into the region R_t^1 and the region $\{R_t^2, R_t^3\}$, as shown in Figure 4.8. The outer bounding box $\mathbf{X}_t^o$ divides the whole region into the regions $\{R_t^1, R_t^2\}$ and the region R_t^3, as described in Figure 4.8. The regions $\{R_t^1, R_t^2\}$ denotes a union of regions R_t^1 and R_t^2. The universal set $\mathbf{U}_t$ is determined by the set $\{R_t^1, R_t^2, R_t^3\}$, and the power set of the universal set, $2^{\mathbf{U}_t}$ is determined by the set, $\{\phi, \{R_t^1\}, \{R_t^2\}, \{R_t^3\}, \{R_t^1, R_t^2\}, \{R_t^1, R_t^3\}, \{R_t^2, R_t^3\}, \mathbf{U}_t\}$. Using (4.27), we get $p(\mathbf{Y}_t|\{R_t^1\})+p(\mathbf{Y}_t|\{R_t^2\})+p(\mathbf{Y}_t|\{R_t^3\})+p(\mathbf{Y}_t|\{R_t^1, R_t^2\})+p(\mathbf{Y}_t|\{R_t^2, R_t^3\})=1$, where the likelihoods of the regions ϕ and $\{\mathbf{U}_t\}$ are zero because they are meaningless, and the likelihoods of the regions $\{R_t^1, R_t^3\}$ is zero because R_t^1 and R_t^3 are not adjacent regions.

Then, the belief of the inner bounding box is estimated by (4.25):

$$bel(\mathbf{X}_t^i) = p(\mathbf{Y}_t|\{R_t^1\}). \tag{4.29}$$

The plausibility of the inner bounding box is estimated by (4.28):

$$pl(\mathbf{X}_t^i) = p(\mathbf{Y}_t|\{R_t^1\}) + p(\mathbf{Y}_t|\{R_t^1, R_t^2\}). \tag{4.30}$$

As $p(\mathbf{Y}_t|\{R_t^1, R_t^2\})$ is positive, the inner bounding box satisfies (4.24): $bel(\mathbf{X}_t^i) \leq p(\mathbf{X}_t^i) \leq pl(\mathbf{X}_t^i)$. Similarly, the belief of the outer bounding box is

$$bel(\mathbf{X}_t^o) = p(\mathbf{Y}_t|\{R_t^1\}) + p(\mathbf{Y}_t|\{R_t^2\}) + p(\mathbf{Y}_t|\{R_t^1, R_t^2\}). \tag{4.31}$$

The plausibility of the outer bounding box is

$$\begin{aligned} pl(\mathbf{X}_t^o) = p(\mathbf{Y}_t|\{R_t^1\}) + p(\mathbf{Y}_t|\{R_t^2\}) + p(\mathbf{Y}_t|\{R_t^1, R_t^2\}) \\ + p(\mathbf{Y}_t|\{R_t^2, R_t^3\}) = 1 - p(\mathbf{Y}_t|\{R_t^3\}). \end{aligned} \tag{4.32}$$

As $p(\mathbf{Y}_t|\{R_t^2, R_t^3\})$ is positive, the outer bounding box also satisfies (4.24): $bel(\mathbf{X}_t^o) \leq p(\mathbf{X}_t^o) \leq pl(\mathbf{X}_t^o)$. Then, with (4.29),(4.30),(4.31), and (4.32), we obtain

$$\begin{aligned} bel(\mathbf{X}_t^i) \leq pl(\mathbf{X}_t^i) \leq bel(\mathbf{X}_t^o) \leq pl(\mathbf{X}_t^o), \\ bel(\mathbf{X}_t^i) \leq p(\mathbf{X}_t^i), p(\mathbf{X}_t^o) \leq pl(\mathbf{X}_t^o). \end{aligned} \tag{4.33}$$

Finally, two theorems of the SBB representation are presented as follows.

Lemma 4. In terms of Dempster Shafer theory, the optimal inner bounding box $\hat{\mathbf{X}}_t^i$ is determined by maximizing the similarity between the whole region inside the bounding box, R_t^1 and the target model, whereas the optimal outer bounding box $\hat{\mathbf{X}}_t^o$ is determined by maximizing the dissimilarity between the region outside of the bounding box, R_t^3 and the target model.

Proof. Section 4.2.4 includes proof of lemma 4.

Lemma 5. The optimal inner bounding box $\hat{\mathbf{X}}_t^i$ forms the lower bound of the bounding box $\mathbf{X}_t$ in terms of a theory of evidence, whereas the optimal outer bounding box $\hat{\mathbf{X}}_t^o$ forms the upper bound of the bounding box $\mathbf{X}_t$ in terms of a theory of evidence.

Proof. Section 4.2.4 includes proof of lemma 5.

4.2.2 Visual Tracker using the Soft Bounding Box

4.2.2.1 Constrained Maximum a Posteriori

As illustrated in Figure 4.8, the state $\mathbf{X}_t^{soft}$ is represented as the combination of the sub-states of the inner and outer bounding boxes, $\mathbf{X}_t^{soft} = (\mathbf{X}_t^i, \mathbf{X}_t^o)$. The inner bounding box sub-state $\mathbf{X}_t^i$ consists of a three-dimensional vector, $\mathbf{X}_t^i = (x_t^i, y_t^i, s_t^i)$, where x_t^i, y_t^i, and s_t^i are the x, y coordinates and the scale of the inner bounding box, respectively. The outer bounding box sub-state $\mathbf{X}_t^o$ also consists of a three-dimensional vector, $\mathbf{X}_t^o = (x_t^o, y_t^o, s_t^o)$, where x_t^o, y_t^o, and s_t^o are the x, y coordinates and the scale of the outer bounding box, respectively. Thus, the total cardinality of $\mathbf{X}_t^{soft}$ is six. Then, the objective of our tracking problem is to find the best state $\hat{\mathbf{X}}_t^{soft}$ that maximizes the posterior $p(\mathbf{X}_t^{soft}|\mathbf{Y}_{1:t})$:

$$\begin{aligned} \hat{\mathbf{X}}_t^{soft} = (\hat{\mathbf{X}}_t^i, \hat{\mathbf{X}}_t^o) = \underset{\mathbf{X}_t^i, \mathbf{X}_t^o}{\arg\max}\, p(\mathbf{X}_t^i, \mathbf{X}_t^o|\mathbf{Y}_{1:t}) \\ \textit{subject to} \;\; \mathbb{R}(\hat{\mathbf{X}}_t^i) \subset \mathbb{R}(\hat{\mathbf{X}}_t^o) \subset \alpha\mathbb{R}(\hat{\mathbf{X}}_t^i), \end{aligned} \tag{4.34}$$

where $\hat{\mathbf{X}}_t^i$ and $\hat{\mathbf{X}}_t^o$ denote the best sub-states of the inner and outer bounding boxes, respectively, and α is a parameter. $\alpha\mathbb{R}(\hat{\mathbf{X}}_t^i)$ means that the width and height of the region $\mathbb{R}(\hat{\mathbf{X}}_t^i)$ are multiplied by α.

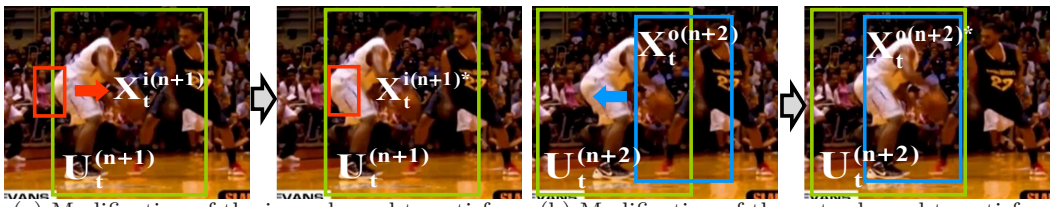
In (4.34), $p(\mathbf{X}_t^i, \mathbf{X}_t^o|\mathbf{Y}_{1:t}) \propto p(\mathbf{Y}_{1:t}|\mathbf{X}_t^i, \mathbf{X}_t^o)p(\mathbf{X}_t^i, \mathbf{X}_t^o) = p(\mathbf{Y}_{1:t}|\mathbf{X}_t^i)p(\mathbf{Y}_{1:t}|\mathbf{X}_t^o)p(\mathbf{X}_t^i, \mathbf{X}_t^o)$. The likelihood of the inner bounding box, $p(\mathbf{Y}_{1:t}|\mathbf{X}_t^i)$, is calculated by $p(\mathbf{Y}_{1:t}|\mathbf{R}_t = \mathbb{R}^1(\mathbf{X}_t^i))$ using (4.26)(4.29) based on Lemma 4, where $\mathbb{R}^1(\mathbf{X}_t^i)$ indicates the region

inside of the inner bounding box $\mathbf{X}_t^i$. The likelihood of the outer bounding box, $p(\mathbf{Y}_{1:t}|\mathbf{X}_t^o)$, is calculated by $1 - p(\mathbf{Y}_{1:t}|\mathbf{R}_t = \mathbb{R}^3(\mathbf{X}_t^o))$ using (4.26)(4.32) based on Lemma 4, where $\mathbb{R}^3(\mathbf{X}_t^o)$ indicates the region outside of the outer bounding box $\mathbf{X}_t^o$. The prior $p(\mathbf{X}_t^i, \mathbf{X}_t^o)$ is realized by the constraint in (4.34). The region described by the outer bounding box, $\mathbb{R}(\mathbf{X}_t^o)$, must include the region described by the inner bounding box, $\mathbb{R}(\mathbf{X}_t^i)$. The second constraint in (4.34) prevents the outer bounding box from becoming infinitely large. Compared with MAP, our CMAP in (4.34) is more difficult because it should satisfy the aforementioned constraints.

To obtain the best state, searching all states within the state space is impractical. Thus, the proposed method adopts the MCMC sampling method [18], which produces N number of sampled states. Among the sampled states, the sampling method easily chooses the best one that maximizes the posterior probability in (4.34). Based on the MCMC sampling method, we modify it to satisfy the aforementioned constraint and present a new CMCMC sampling method, which will be explained in the next section.

4.2.2.2 Constrained Markov Chain Monte Carlo

The proposed CMCMC sampling method defines a single Markov Chain and obtains samples over the chain. As we define two sub-states, $\mathbf{X}_t^i$ and $\mathbf{X}_t^o$, we get samples of the sub-states, alternately. For example, we get samples such like $\mathbf{X}_t^{i(n-1)}$, $\mathbf{X}_t^{o(n)}$, $\mathbf{X}_t^{i(n+1)}$,..., where $\mathbf{X}_t^{o(n)}$ and $\mathbf{X}_t^{i(n+1)}$ are the n-th sample for the outer bounding box and the *(n+1)*-th sample for the inner bounding box, respectively. First, the method obtains a sample of $\mathbf{X}_t^i$ by two main steps: (1) the proposal step and (2) the



(a) Modification of the inner bound to satisfy the constraint

(b) Modification of the outer bound to satisfy the constraint

Figure 4.9: **Example of the SBB constraint**

acceptance step. The proposal step suggests a new sample given a previous sample:

$$Q_i(\mathbf{X}_t^{i(n+1)}; \mathbf{X}_t^{o(n)}, \mathbf{X}_t^{i(n-1)}) = \begin{cases} Q_{xy}(\mathbf{X}_t^{i(n+1)}; \mathbf{X}_t^{o(n)}) = G(\mathbf{X}_t^{o(n)}, \sigma_{xy}^2) & \text{for } x, y \\ Q_s(\mathbf{X}_t^{i(n+1)}; \mathbf{X}_t^{i(n-1)}) = G(\mathbf{X}_t^{i(n-1)}, \sigma_s^2) & \text{for } s \end{cases}. \tag{4.35}$$

In (4.35), a new position of the inner bounding box is proposed through the Gaussian function G, with the current position of the outer bounding box as the mean and σ_{xy}^2 as the variance. A new scale of the inner bounding box is proposed through G, with the current scale of the inner bounding box as the mean and σ_s^2 as the variance. The proposal sample $\mathbf{X}_t^{i(n+1)}$ is modified into the sample $\mathbf{X}_t^{i(n+1)^*}$ bounded by the region $1.2\mathbb{R}(\mathbf{X}_t^{o(n)})$ to satisfy the relaxed constraint of (4.34), as shown in Figure 4.9(a):

$$\mathbb{R}(\mathbf{X}_t^{i(n+1)^*}) \subset 1.2\mathbb{R}(\mathbf{X}_t^{o(n)}) = \mathbb{R}(\mathbf{U}_t^{(n+1)}), \tag{4.36}$$

where $\mathbf{U}_t^{(n+1)}$ is the *(n+1)*-th universal set. In (4.36), the region $1.2\mathbb{R}(\mathbf{X}_t^{o(n)})$ has the same center as $\mathbf{X}_t^{o(n)}$ and is 1.2 times the scale of $\mathbf{X}_t^{o(n)}$, where 1.2 is set empirically. After the proposed step, the acceptance step determines whether the proposed sample $\mathbf{X}_t^{i(n+1)^*}$ is accepted or not with the following probability:

$$a = min\left[1, \frac{p(\mathbf{Y}_t|\mathbf{X}_t^{i(n+1)^*})Q_i(\mathbf{X}_t^{o(n)}, \mathbf{X}_t^{i(n-1)}; \mathbf{X}_t^{i(n+1)^*})}{p(\mathbf{Y}_t|\mathbf{X}_t^{i(n-1)})Q_i(\mathbf{X}_t^{i(n+1)^*}; \mathbf{X}_t^{o(n)}, \mathbf{X}_t^{i(n-1)})}\right]. \tag{4.37}$$

Algorithm 5 Soft Bounding Box Tracker

Input: $\mathbf{X}_{t-1} = (\mathbf{X}^i_{t-1}, \mathbf{X}^o_{t-1})$

Output: $\hat{\mathbf{X}}^{soft}_t = (\hat{\mathbf{X}}^i_t, \hat{\mathbf{X}}^o_t)$

1: **for** 1 to N **do**

2: 1. Propose a sample of the inner BB using (4.35) with (4.36).

3: 2. Accept the sample with the probability (4.37).

4: 3. Propose a sample of the outer BB using (4.38) with (4.39).

5: 4. Accept the sample with the probability (4.40).

6: **end for**

7: Estimate the CMAP state $\hat{\mathbf{X}}^{soft}_t$ using (4.34).

In (4.37), $p(\mathbf{Y}_t|\mathbf{X}^{i(n+1)^*}_t) = p(\mathbf{Y}_t|\{R^1_t\})$ based on Theorem 1, where R^1_t denotes the whole region in the inner bounding box $\mathbf{X}^{i(n+1)^*}_t$.

The method then proposes a sample of $\mathbf{X}^o_t$:

$$Q_o(\mathbf{X}^{o(n+2)}_t; \mathbf{X}^{i(n+1)}_t, \mathbf{X}^{o(n)}_t) = \left(\begin{array}{ll} Q_{xy}(\mathbf{X}^{o(n+2)}_t; \mathbf{X}^{i(n+1)}_t) = G(\mathbf{X}^{i(n+1)}_t, \sigma^2_{xy}) & \text{for } x, y \\ Q_s(\mathbf{X}^{o(n+2)}_t; \mathbf{X}^{o(n)}_t) = G(\mathbf{X}^{o(n)}_t, \sigma^2_s) & \text{for } s \end{array} \right), \tag{4.38}$$

where $\mathbf{X}^{o(n+2)}_t$ is the *(n+2)*-th samples of the outer bounding box. In (4.38), a new position of the outer bounding box is proposed through G, with the current position of the inner bounding box as the mean and σ^2_{xy} as the variance. A new scale of the outer bounding box is proposed through G, with the current scale of the outer bounding box as the mean and σ^2_s as the variance. The proposed sample $\mathbf{X}^{o(n+2)}_t$ is modified into the sample $\mathbf{X}^{o(n+2)^*}_t$ bounded by the region $5.0\mathbb{R}(\mathbf{X}^{i(n+1)}_t)$ to satisfy the constraint of (4.34), as shown in Figure 4.9(b):

$$\mathbb{R}(\mathbf{X}^{o(n+2)^*}_t) \subset 5.0\mathbb{R}(\mathbf{X}^{i(n+1)}_t) = \mathbb{R}(\mathbf{U}^{(n+2)}_t), \tag{4.39}$$

where $\mathbf{U}^{(n+2)}_t$ is the *(n+2)*-th universal set. In (4.39), the region $5.0\mathbb{R}(\mathbf{X}^{i(n+1)}_t)$

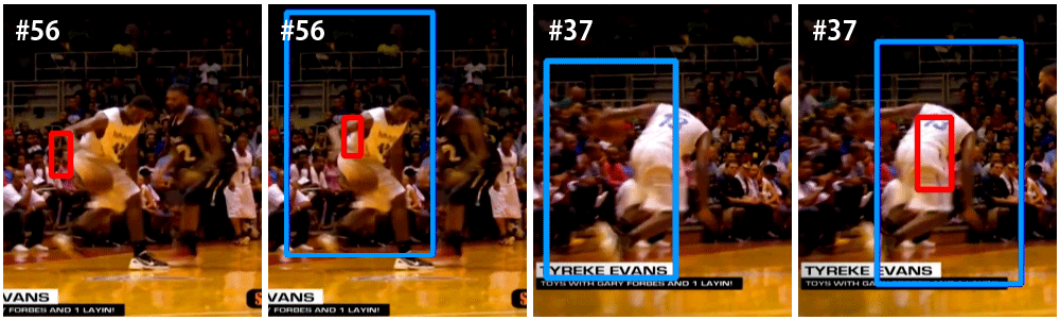
has the same center as $\mathbf{X}_t^{i(n+1)}$ and is 5.0 times the scale of $\mathbf{X}_t^{i(n+1)}$, where 5.0 is set empirically. After the proposal step, the method accepts the proposed sample $\mathbf{X}_t^{o(n+2)^*}$ with the following probability:

$$a = min\left[1, \frac{p(\mathbf{Y}_t|\mathbf{X}_t^{o(n+2)^*})Q_o(\mathbf{X}_t^{i(n+1)}, \mathbf{X}_t^{o(n)}; \mathbf{X}_t^{o(n+2)^*})}{p(\mathbf{Y}_t|\mathbf{X}_t^{o(n)})Q_o(\mathbf{X}_t^{o(n+2)^*}; \mathbf{X}_t^{i(n+1)}, \mathbf{X}_t^{o(n)})}\right]. \tag{4.40}$$

In (4.40), $p(\mathbf{Y}_t|\mathbf{X}_t^{o(n+2)^*}) = \left[1 - p(\mathbf{Y}_t|\{R_t^3\})\right]$ based on Theorem 1, where R_t^3 denotes the whole region outside of the outer bounding box $\mathbf{X}_t^{o(n+2)^*}$. These steps iteratively continue until the number of iterations reaches the predefined value N. Algorithm 5 illustrates the whole process of the proposed tracking method.

4.2.3 Experimental Results

The proposed method (SBB) was compared with eight different state-of-the-art tracking methods: (1) standard MCMC (MC) based on [18], (2) Incremental Visual Learning (IVT) [106, 1], (3) Multiple Instance Learning (MIL) [6] (4) Basin Hopping Monte Carlo (BHMC) [24], (5) Blocks Histogram Tracker (BHT) [46], (6) Hough-based Tracking (HT) [45], (7) Local Global Tracker (LGT) [44], and (8) Visual Tracking Decomposition (VTD) [26]. In all the experiments, σ_{xy} is set for x, y to 0.5 and σ_s for s to 0.01 in (4.35) and (4.38). The inner and outer bounding boxes of the proposed method could have different width/height proportion initially. However, the proportion is fixed for each bounding box during the tracking process. The number of samples was fixed to 1000 for all sampling-based methods, including the proposed method. For fair comparison, the same initialization was used on all methods. The parameters of all methods were adjusted to produce the best tracking performance. Note that we utilize the codes provided by authors to evaluate IVT, MIL, BHMC, BHT, HT, LGT, and VTD.



(a) Inner bounding box (b) Proposed method (c) Outer bounding box (d) Proposed method

Figure 4.10: **Performance of the SBB** in *basketball* seq. which has abrupt motions and pose variations. The red and blue rectangles are the inner and outer bounding boxes, respectively.

4.2.3.1 Performance of the Proposed Method

Performance of the Soft Bounding Box: The performance difference between the single bounding box representation and the SBB representation were examined. The experiments were performed under the same conditions, differing only in the types of bounding box representation. As shown in Figure 4.10, either the inner bounding box alone or the outer bounding box alone is prone to drift away from the target. Figure 4.10(a) shows that the inner bounding box began to drift and to track the background region around the target, as the appearance of the target became severely deformed. The proposed method kept tracking the target because of the constraint provided by the outer bounding box. Thus, the outer bounding box serves as a weak constraint that gives an estimate of the position of the target to the inner bounding box, as it began sampling from the position of the inner bounding box of the previous frame. If the inner bounding box includes some parts of the background, then it will have a tendency to drift because it recognizes the background part as the foreground. However, in a SBB representation, the inner bounding box is sampled at the estimated position from the outer bounding box, pulling the inner bounding

Table 4.4: **Comparison of tracking results** using IMCMC and CMCMC. The numbers indicate the average center location errors in pixels. These numbers were obtained by running each algorithm five times and averaging the results.

	basketball	*lazysong*	*fx*	*diving*	*gymnastics*
IMCMC	209	42	44	56	109
CMCMC	36	17	25	16	16
	faceocc	*twinings*	*singer*	*skating*	**Average**
IMCMC	21	18	84	56	**71.0**
CMCMC	18	17	12	28	**20.5**

box to the center of the target and reducing the possibility of drifting. The outer bounding box can estimate the target position better than the inner bounding box, owing to its large region that makes the outer bounding box robust to noise.

Similarly, the inner bounding box also helps the outer bounding box. In Figure 4.10(c), the outer bounding box drifted, despite being insensitive to noise. As in the previous case, the inner bounding box complemented the outer bounding box, as shown in Figure 4.10(d). The inner bounding box usually has higher probability of only including the foreground than the outer bounding box. Thus, it also serves as a weak constraint, pulling the outer bounding box to the foreground region.

Performance of the Constrained MCMC: The performances of IMCMC in [26] and CMCMC were also compared to demonstrate the superiority of the proposed CMCMC. For this experiment, IMCMC was applied to incorporate purposely the same capability of pulling one box to another, similar to CMCMC. IMCMC initially verifies the constraint in which the inner bounding box must be inside the outer bounding box. If so, it separately samples each Markov Chain for each bounding box; otherwise, it provides an offset to one of the bounding boxes and probabilistically determines which one has to be adjusted. However, it cannot prevent itself from drifting away even if the two bounding boxes do satisfy the constraint. When the inner bounding box begins to drift, it forces the outer bounding box to drift as well

Table 4.5: **Quantitative comparison of tracking results** with other methods. The numbers indicate the average center location errors in pixels. In this experiment, other tracking methods utilize the **inner bounding box representation**. For our method, the mean of center positions of inner and outer bounding boxes is reported as the final tracking result. The best result is shown in red and the second-best in blue. N/W means that a method does not work at the corresponding dataset. Note that BHMC, VTD, and MC are state-of-the-art color-based trackers. BHMC,BHT,HT, and LGT are trackers, which are designed especially for highly non-rigid targets.

	BHMC	BHT	MIL	IVT	VTD	MC	HT	LGT	SBB
basketball	80	63	133	**50**	58	110	197	160	**36**
lazysong	55	142	38	38	**17**	30	56	42	**17**
fx	73	69	56	46	**33**	144	70	126	**25**
diving	41	N/W	76	68	23	20	76	**15**	**16**
gymnastics	29	N/W	42	62	22	**17**	108	99	**16**
faceocc	50	N/W	36	61	21	20	34	**19**	**18**
twinings	**5**	34	15	17	**9**	14	31	22	17
singer	24	51	17	13	8	22	**5**	**5**	12
skating	57	N/W	93	80	**42**	68	97	74	**28**

Table 4.6: **Quantitative comparison of tracking results** with other methods. In this experiment, other tracking methods utilize the **outer bounding box representation**.

	BHMC	BHT	MIL	IVT	VTD	MC	HT	LGT	SBB
basketball	**63**	157	101	237	177	133	169	170	**36**
lazysong	74	115	48	43	**24**	94	137	71	**17**
fx	29	N/W	75	**27**	39	37	70	70	**25**
diving	**18**	74	88	91	70	29	83	29	**16**
gymnastics	**7**	N/W	22	27	**10**	76	102	98	16
faceocc	30	29	25	27	**7**	31	**13**	19	18
twinings	29	43	**10**	32	**7**	27	36	24	17
singer	55	48	24	**3**	5	84	51	**4**	12
skating	52	49	95	157	**30**	36	174	160	**28**

because of the constraint, rendering the inner bounding box even worse. As shown in Table 4.4, IMCMC cannot outperform CMCMC, because the probability that the inner bounding box will pull the outer bounding box toward it and the vulnerability of the inner bounding box to noise are high.

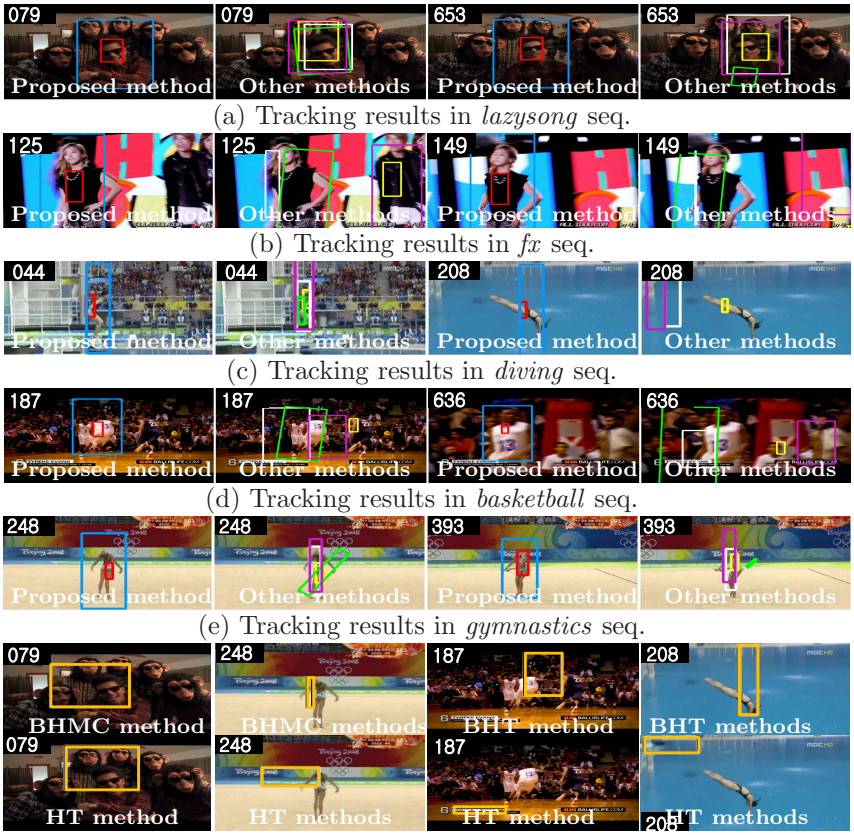
4.2.3.2 Comparison with Other Methods

For this comparison, nine test sequences (six of which were publicly available and the others were made by us) and eight different state-of-the-art tracking methods

Table 4.7: **Quantitative comparison of tracking results** with other methods. In this experiment, other tracking methods utilize the **regular bounding box representation**, which is defined by authors of the datasets.

	BHMC	BHT	MIL	IVT	VTD	MC	HT	LGT	SBB
basketball	**51**	63	123	292	161	71	199	180	**37**
lazysong	72	115	87	57	**22**	95	165	70	**17**
fx	**30**	69	43	37	31	36	67	70	**25**
diving	35	74	94	95	85	28	96	**14**	**16**
gymnastics	**7**	N/W	16	22	**10**	12	100	99	16
faceocc	27	29	32	**16**	**8**	19	35	19	18
twinings	29	34	**10**	20	**7**	21	31	22	17
singer	48	48	21	**7**	10	85	**9**	15	12
skating	49	49	102	176	**29**	31	164	170	**28**

were used. Tables 4.5, 4.6, and 4.7 show the quantitative evaluation of the tracking results. Our method always used the SBB representation whereas other methods used the inner, outer, and regular bounding box representations to produce results in Tables 4.5, 4.6, and 4.7, respectively. These tracking results indicate that our method is robust to track deformable target objects, as the drifting problem is effectively resolved using the SBB representation. Our method outperformed the recent state-of-the art tracking methods especially for highly non-rigid objects, which are BHMC, BHT, HT, and LGT. This demonstrates that a single bounding box representation is not adequate to represent highly non-rigid objects. In addition, our method produced better results than color-based tracking methods (VTD and MC) because our method did not consider the ambiguous regions while calculating color histograms. The tracking results also indicate that our method is not sensitive to the representation type of the bounding boxes because it describes the target as a range of the bounding box. Conversely, the other methods yielded very different tracking results depending on the representation type of the bounding boxes (i.e., inner bounding box in Table 4.5, outer bounding box in Table 4.6, and regular bounding box in Table 4.7). Our method approximately takes 0.1 sec per frame



(a) Tracking results in *lazysong* seq.

(b) Tracking results in *fx* seq.

(c) Tracking results in *diving* seq.

(d) Tracking results in *basketball* seq.

(e) Tracking results in *gymnastics* seq.

(f) Tracking results of the methods especially for highly non-rigid objects.

Figure 4.11: **Qualitative comparison of the tracking results using other methods.** The red and blue boxes give the results of the proposed method (the inner and outer bounding boxes). The yellow, white, green, and pink boxes give the results of MCMC method using the inner bounding box representation, VTD, IVT, and MIL using the outer bounding box representation, respectively.

because it uses the simple likelihood function and the simple SBB representation; the SBB representation greatly increases the tracking accuracy at a small computational cost.

In Figure 4.11(a), the *lazysong* seq., which includes some objects similar in appearance to the target object, is tested. In the case of other methods, their bounding boxes expanded and maintained their sizes, including some background objects. The MCMC method did not experience this problem. However, the pro-

posed method showed the most accurate tracking performance among these methods. Figure 4.11(b) shows the tracking results of *fx* seq. The target person was severely occluded by other person who wore clothes of the same color as that of the target. Whereas some trajectories were hijacked by the other person, the proposed method successfully tracked the target. Even with an abrupt camera motion and a similar background in frame #149, the proposed method showed good quality while others failed to do so. Figure 4.11(c) shows the tracking results of *diving* seq. In frame #44, all methods were able to track the woman. However, when the woman started spinning, the proposed method continued to track the woman while the other methods failed to track it. Figure 4.11(d) shows the tracking results in *basketball* seq. A basketball player dribbled past another player, scored, and was occluded by the referee in this sequence. In frame #187, the MCMC method was distracted by the background and the other player. In this frame, the other methods experienced drifting problems, as they had a larger background part than the foreground part in their bounding boxes. . In frame #636, after the referee passed the basketball player, the MCMC and the other methods were again distracted and could hardly track the target, but the proposed method maintained the trajectory of the target. Figure 4.11(e) shows the tracking results in *gymnastics* seq. In the sequence, the proposed method, VTD, and MCMC method were able to track the target. VTD showed the best result, but its distance from the target became wider when the gymnast turned and changed her pose fast. Figure 4.11(f) shows the results of BHMC, BHT, and HT, which are designed especially for highly non-rigid objects, and demonstrates that these method also frequently failed to track the targets when there are severe deformation of the targets.

4.2.4 Appendix

Lemma 4. In terms of Dempster Shafer theory, the optimal inner bounding box $\hat{\mathbf{X}}_t^i$ is determined by maximizing the similarity between the whole region inside the bounding box, R_t^1 and the target model, whereas the optimal outer bounding box $\hat{\mathbf{X}}_t^o$ is determined by maximizing the dissimilarity between the region outside of the bounding box, R_t^3 and the target model.

Proof. The optimal bounding box has the largest belief and plausibility values. Thus, (4.33) is maximized to obtain the best inner and outer bounding boxes: $\max bel(\mathbf{X}_t^i) \leq \max p(\mathbf{X}_t^i), \max p(\mathbf{X}_t^o) \leq \max pl(\mathbf{X}_t^o)$. Thereafter, the optimal inner bounding box $\hat{\mathbf{X}}_t^i$ is obtained using (4.29):

$$\hat{\mathbf{X}}_t^i = \underset{\mathbf{X}_t^i}{\arg\max}\, bel(\mathbf{X}_t^i) = \underset{\mathbf{X}_t^i}{\arg\max}\, p(\mathbf{Y}_t|\{R_t^1\}). \tag{4.41}$$

In (4.41), the optimal inner bounding box is determined by maximizing the likelihood of the region R_t^1. Similarly, the optimal outer bounding box $\hat{\mathbf{X}}_t^o$ is obtained using (4.32):

$$\hat{\mathbf{X}}_t^o = \underset{\mathbf{X}_t^o}{\arg\max}\, pl(\mathbf{X}_t^o) = \underset{\mathbf{X}_t^o}{\arg\max} \left[1 - p(\mathbf{Y}_t|\{R_t^3\})\right]. \tag{4.42}$$

In (4.42), the optimal outer bounding box is determined by minimizing the likelihood of the region R_t^3.

Lemma 5. The optimal inner bounding box $\hat{\mathbf{X}}_t^i$ forms the lower bound of the bounding box $\mathbf{X}_t$ in terms of a theory of evidence, whereas the optimal outer bounding box $\hat{\mathbf{X}}_t^o$ forms the upper bound of the bounding box $\mathbf{X}_t$ in terms of a theory of evidence.

Proof. Using (4.33),(4.41), and (4.42), we get

$$bel(\hat{\mathbf{X}}_t^i) \leq p(\mathbf{X}_t) \leq pl(\hat{\mathbf{X}}_t^o). \tag{4.43}$$

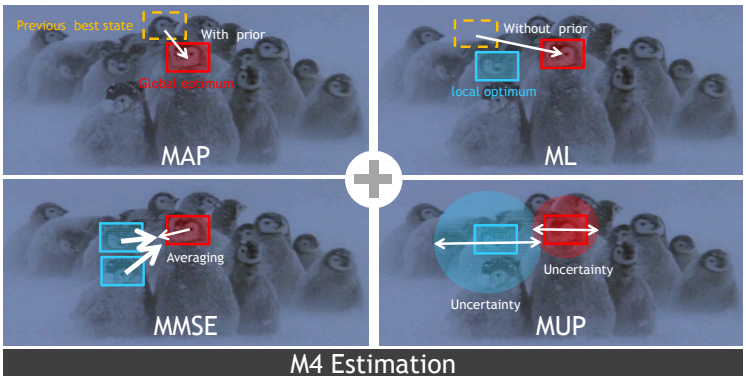


Figure 4.12: **Basic idea of the proposed tracker**. (1) **MAP** finds the state (red box), which maximizes likelihood with prior. The prior enforces the current state to be near the state at the previous time (orange box), which makes smooth motions to be tracked. (2) **ML** finds the state (red box), which maximizes likelihood without prior. The current state can be far from the state at the previous time although a strong local optimum state (blue box) is close to the previous state. Hence, the tracker can deal with abrupt motions. (3) **MMSE** finds the state around strong local optimum states. The tracker doesn't get trapped in a local optimum state (blue box) and can reach to the other states. However, it also makes the tracker to be driven away from the global optimum state (red box). To prevent this problem, MMSE should be guided by MAP, ML, and MUP like our M4 estimation. (4) **MUP** finds the confident state (red box), which minimizes the state estimation error. MUP numerically measures how the MAP, ML, and MMSE estimation is convincing. (5) *We propose the unified framework, which combines all theses estimators with the rigorous theoretical basis and exploits the beneficial complementary relationship among them.*

In (4.43), $\hat{\mathbf{X}}_t^i$ and $\hat{\mathbf{X}}_t^o$ are the lower and upper bounds of $\mathbf{X}_t$, respectively, in terms of a theory of evidence.

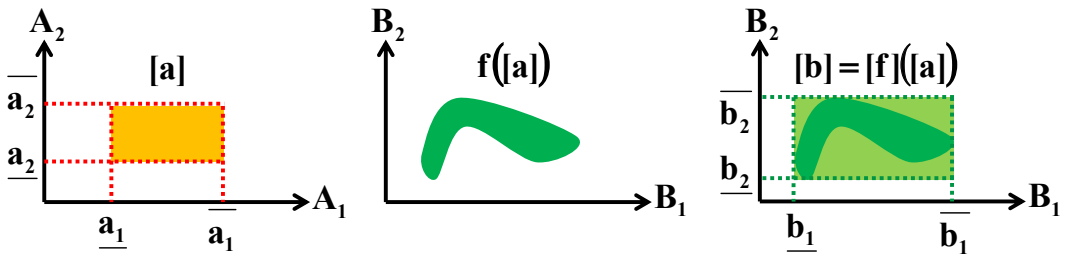
4.3 Interval Tracker

The proposed method follows the aforementioned interval analysis approach to employ the uncertainty in distributions associated with the appearance model (likelihood) and state (prior). This uncertainty occurs when information about the state

and the appearance of the target are initially insufficient or only partially available. For example, if the target is severely occluded in the initialization step, the tracking method cannot uniquely determine the state and the appearance of the target. Moreover, the uncertainty also occurs when information about the state and the appearance of the target is corrupted during the tracking process. In this case, the method cannot perfectly estimate and update distributions associated with the state and the appearance model. To solve theses problems, the proposed method makes the distributions to include uncertainty via interval analysis. In our method, the distributions are represented as interval. Then, the method transforms the tracking problem into the M4 estimation problem, which combines the MAP, MMSE, MUP, and ML estimations with the rigorous mathematical basis, as illustrated in Fig.4.12. With the M4 estimation, the method finds the best interval of the state, instead of the best state, where the best interval of the state maximizes the posterior and minimizes uncertainty of the posterior estimation. Then, for evaluation and comparison, our method gets the best state from the best interval of the state. Contribution of the proposed method is four-fold.

4.3.1 Interval Linearization of Posterior Probability

First, $[p]([\mathbf{X}_t]|\mathbf{Y}_{1:t})$ is different from $p([\mathbf{X}_t]|\mathbf{Y}_{1:t})$, as explained in Fig.4.13. $p([\mathbf{X}_t]|\mathbf{Y}_{1:t})$ only employs the uncertainty of the *state*. On the other hand, $[p]([\mathbf{X}_t]|\mathbf{Y}_{1:t})$ employs the uncertainty of the *posterior* as well. Second, interval of the posterior indicates a possible range that the estimated posterior probability can be changed due to estimation error, when $[\mathbf{X}_t]$ is set to a specific value. For example, $0.2 \leq [p]([\mathbf{X}_t] = 5|\mathbf{Y}_{1:t}) \leq 0.7$.



(a) (b) (c)

Figure 4.13: **Difference between** $f([a])$ **and** $[f]([a])$ Let assume 2-dimensional input space A and output space B. (a) A conventional input, a, is represented as a point in A. Our input with interval, [a], is represented as a range (orange rectangle). (b) Then, the function $f([a])$ just projects the orange rectangle into the output space B (dark green shape). Hence, $f([a])$ only employs the uncertainty of the input [a]. (c) On the other hand, the function $[f]([a])$ finds a range (green rectangle), which contains the dark green shape via the interval linearization technique. In this case, $[f]([a])$ employs the uncertainty of the function itself as well.

We formulate the posterior via interval analysis:

$$[p]\left([\mathbf{X}_t]|\mathbf{Y}_{1:t}\right) \propto [p]\left(\mathbf{Y}_t|[\mathbf{X}_t]\right) \times \int p\left([\mathbf{X}_t]|[\mathbf{X}_{t-1}]\right)[p]\left([\mathbf{X}_{t-1}]|\mathbf{Y}_{1:t-1}\right) d[\mathbf{X}_{t-1}], \quad (4.44)$$

where $[p]\left([\mathbf{X}_t]|\mathbf{Y}_{1:t}\right)$ denotes interval of the posterior. In (4.44), interval of the prior, $[\mathbf{X}_t]$, is defined as

$$[\mathbf{X}_t] = \left[\underline{\mathbf{X}}_t, \overline{\mathbf{X}}_t\right] = \left[(\underline{X}_t^1, \underline{X}_t^2, \underline{X}_t^3)^T, (\overline{X}_t^1, \overline{X}_t^2, \overline{X}_t^3)^T\right], \quad (4.45)$$

where $\underline{\mathbf{X}}_t \leq \mathbf{X}_t \leq \overline{\mathbf{X}}_t$ and; X_t^1, X_t^2, and X_t^3 indicate x-center, y-center positions, and scale of the target, respectively. Interval of the likelihood, $[p](\mathbf{Y}_t|[\mathbf{X}_t])$, is defined as

$$[p](\mathbf{Y}_t|[\mathbf{X}_t]) = \left[\underline{[p](\mathbf{Y}_t|[\mathbf{X}_t])}, \overline{[p](\mathbf{Y}_t|[\mathbf{X}_t])}\right], \quad (4.46)$$

where $\underline{[p](\mathbf{Y}_t|[\mathbf{X}_t])} \leq [p](\mathbf{Y}_t|[\mathbf{X}_t]) \leq \overline{[p](\mathbf{Y}_t|[\mathbf{X}_t])}$.

Interval of the likelihood, $[p](\mathbf{Y}_t|[\mathbf{X}_t])$, is decomposed into two terms by the

first-order interval Taylor extension in [107].

$$\begin{aligned}[p](\mathbf{Y}_t|[\mathbf{X}_t]) &= p(\mathbf{Y}_t|\dot{\mathbf{X}}_t) \oplus \sum_{i=1}^{3}\left[\frac{\partial}{\partial X_t^i}p(\mathbf{Y}_t|[\mathbf{X}_t]) \otimes ([X_t^i] \ominus \dot{X}_t^i)\right] \\ &= \underbrace{p(\mathbf{Y}_t|\dot{\mathbf{X}}_t)}_{\text{Single value}} \oplus \underbrace{\left[\sum_{i=1}^{3}\left[\lambda_i(\underline{X}_t^i - \dot{X}_t^i)\right], \sum_{i=1}^{3}\left[\lambda_i(\overline{X}_t^i - \dot{X}_t^i)\right]\right]}_{\text{Interval value}},\end{aligned} \tag{4.47}$$

where $\lambda_i = \text{MAX}\left(\left|\frac{\partial}{\partial X_t^i}p(\mathbf{Y}_t|[\mathbf{X}_t])\right|\right)$ and; $\ominus$[2], $\otimes$[3], and $\oplus$ indicates the element-wise minus, time, and plus operations, respectively. In (4.47), $\dot{X}_t^i$ can be any points which belong to $[X_t^i]$. In our method, we set $\dot{X}_t^i = \mathbb{E}_{p(\mathbf{Y}_t|[X_t^i])}[X_t^i]$, which has a following property.

Lemma 6. If $\dot{X}_t^i = \mathbb{E}_{p(\mathbf{Y}_t|[X_t^i])}[X_t^i]$, the likelihood at the state $\dot{X}_t^i$ is equal to expectation of $[p](\mathbf{Y}_t|[X_t^i])$ with respect to $p(\mathbf{Y}_t|[X_t^i])$:

$$p(\mathbf{Y}_t|\dot{\mathbf{X}}_t) = \mathbb{E}_{p(\mathbf{Y}_t|[\mathbf{X}_t])}[p](\mathbf{Y}_t|[\mathbf{X}_t]). \tag{4.48}$$

Lemma 7. $\dot{X}_t^i$ is the MMSE estimate of X_t^i over $[X_t^i]$:

$$\dot{X}_t^i = \hat{X}_{\text{mmse}} = \arg\min_{\hat{X}} \mathbb{E}_{p(\mathbf{Y}_t|[X_t^i])}\|[X_t^i] - \hat{X}\|^2. \tag{4.49}$$

Proof. Proof of Lemma 6 and 7 are described in Section 4.3.6. Since our method performs the MAP estimation over the MMSE state, it can decrease contribution of outlier states during the MAP estimation.

The first term in (4.47) has a single value, which is similar to the conventional likelihood representation. The second term in (4.47) has an interval value, where the interval width is equal to the estimation error (uncertainty) of the likelihood

[2] $[X_t^i] \ominus \dot{X}_t^i = \left[\underline{X}_t^i - \dot{X}_t^i, \overline{X}_t^i - \dot{X}_t^i\right]$.

[3] $\frac{\partial}{\partial X_t^i}p(\mathbf{Y}_t|[\mathbf{X}_t])([X_t^i] \ominus \dot{X}_t^i) = \left[\lambda_i(\underline{X}_t^i - \dot{X}_t^i), \lambda_i(\overline{X}_t^i - \dot{X}_t^i)\right]$.

$p(\mathbf{Y}_t|\dot{\mathbf{X}}_t)$. The interval width is defined by gap between two bounds in the second term:

$$\mathrm{W}\left[\sum_{i=1}^{3}\left[\lambda_i(\underline{X}_t^i - \dot{X}_t^i)\right], \sum_{i=1}^{3}\left[\lambda_i(\overline{X}_t^i - \dot{X}_t^i)\right]\right] = \sum_{i=1}^{3}\left(\lambda_i(\overline{X}_t^i - \underline{X}_t^i)\right). \tag{4.50}$$

If the interval width is large, the estimated likelihood $p(\mathbf{Y}_t|\dot{\mathbf{X}}_t)$ is drastically uncertain. In (4.47), $p(\mathbf{Y}_t|[\mathbf{X}_t])$ should be differentiable to obtain the second term. To make our method generally applicable, we approximate a derivative of the likelihood function, $\frac{d}{d\mathbf{X}_t}p(\mathbf{Y}_t|[\mathbf{X}_t])$, and, thus, our method can utilize general likelihood functions (e.g. bhattacharyya similarity coefficient [33], diffusion distance [69]).

4.3.2 Decomposition of Posterior Probability

To enable our method to evaluate the likelihood in (4.47), the likelihood is reformulated as

$$[p](\mathbf{Y}_t|[\mathbf{X}_t]) \approx \underbrace{\omega_C p_C(\mathbf{Y}_t|\dot{\mathbf{X}}_t)}_{\text{Conventional likelihood}} \oplus \underbrace{\omega_U [p]_U(\mathbf{Y}_t|[\mathbf{X}_t])}_{\text{Uncertainty of likelihood}}, \tag{4.51}$$

where ω_C and ω_U are normalization parameters, which make $[p](\mathbf{Y}_t|[\mathbf{X}_t])$ to be probability. These parameters can be automatically determined by the sampling method [27], of which process is presented in the section 4.3.3.3. In (4.51), the conventional likelihood is defined by

$$p_C(\mathbf{Y}_t|\dot{\mathbf{X}}_t) = exp^{-\gamma_C DD(\mathbf{Y}_t(\dot{\mathbf{X}}_t), M_t)}, \tag{4.52}$$

where γ_C denotes the weighting parameter, $\mathbf{Y}_t(\dot{\mathbf{X}}_t)$ represents the HSV color histogram at the image patch described by $\dot{\mathbf{X}}_t$, and M_t indicates the reference HSV color histogram of the target at time t. The HSV color histogram is obtained by the method in [33]. In (4.52), the DD function returns the diffusion distance [69]

between the observation $\mathbf{Y}_t(\dot{\mathbf{X}}_t)$ and the target model M_t at time t. The uncertainty of likelihood in (4.51) is defined as probability like:

$$[p]_U(\mathbf{Y}_t|[\mathbf{X}_t]) = exp^{-\gamma_U \sum_{i=1}^{3}\left[\lambda_i(\overline{X}_t^i - \underline{X}_t^i)\right]}, \tag{4.53}$$

where $\sum_{i=1}^{3}\left[\lambda_i(\overline{X}_t^i - \underline{X}_t^i)\right]$ in (4.50) is interval width of $[p](\mathbf{Y}_t|[\mathbf{X}_t])$ and γ_U is the weighting parameter.

By inserting (4.51) into (4.44), we get

$$\begin{aligned}[p]\left([\mathbf{X}_t]|\mathbf{Y}_{1:t}\right) \approx & \omega_C \underbrace{p_C(\mathbf{Y}_t|\dot{\mathbf{X}}_t)\int p([\mathbf{X}_t]|[\mathbf{X}_{t-1}])[p]([\mathbf{X}_{t-1}]|\mathbf{Y}_{1:t-1})d[\mathbf{X}_{t-1}]}_{p_C(\dot{\mathbf{X}}_t|\mathbf{Y}_{1:t})} + \\ & \omega_U \underbrace{[p]_U(\mathbf{Y}_t|[\mathbf{X}_t])\int p([\mathbf{X}_t]|[\mathbf{X}_{t-1}])[p]([\mathbf{X}_{t-1}]|\mathbf{Y}_{1:t-1})d[\mathbf{X}_{t-1}]}_{[p]_U([\mathbf{X}_t]|\mathbf{Y}_{1:t})}.\end{aligned} \tag{4.54}$$

Our method decompose an original posterior distribution into two posterior distributions. Then, the landscapes of the decomposed posteriors could be simplified and, thus, sampling methods prevent the Markov Chains from getting trapped in local minima and efficiently search for the best state [108]. These two decomposed posteriors in (4.54) induce two different visual trackers.

4.3.3 The M4 Estimation

Our objective of visual tracking is to find the best interval of the state, $[\hat{\mathbf{X}}_t]$, which maximizes the posterior probability and minimizes uncertainty of the posterior estimation at the same time.

$$[\hat{\mathbf{X}}_t] \equiv \underset{[\mathbf{X}_t]}{\arg\max}\, \omega_C p_C(\dot{\mathbf{X}}_t|\mathbf{Y}_{1:t}) + \omega_U [p]_U([\mathbf{X}_t]|\mathbf{Y}_{1:t}), \tag{4.55}$$

where $\dot{\mathbf{X}}_t = \mathbb{E}_{p(\mathbf{Y}_t|[\mathbf{X}_t])}[\mathbf{X}_t]$, $p_C(\dot{\mathbf{X}}_t|\mathbf{Y}_{1:t})$ indicates a conventional posterior, and $[p]_U([\mathbf{X}_t]|\mathbf{Y}_{1:t})$ denotes uncertainty of the posterior estimation. Two trackers induced by (4.54) are performed for the MMSE-MAP and ML-MUP estimations,

respectively. The first tracker searches for the best interval of the state, which maximizes the posterior probability over the state obtained by the MMSE estimation. The second tracker searches for the best interval of the state, which minimizes uncertainty of the posterior estimation. Minimizing uncertainty can be achieved by the ML estimation. Then, the IMCMC sampling method used in [88, 26, 27] finds the best common interval of the state for two trackers via interaction between two trackers. The states sampled by two trackers converge to a single state, although our method does not satisfy detailed balance [88].

4.3.3.1 Tracker 1: MMSE-MAP Estimation

The first posterior in (4.54) induces the first tracker. The goal of the tracker 1 is to find the best interval of the state, which maximizes $p_C(\dot{\mathbf{X}}_t|\mathbf{Y}_{1:t})$ in (4.55). Since our method maximizes the posterior $p_C(\dot{\mathbf{X}}_t|\mathbf{Y}_{1:t})$ over the MMSE state $\dot{\mathbf{X}}_t$ proved by **Lemma 7**, this estimation is called MMSE-MAP. MMSE-MAP can decrease contribution of outlier states during the MAP estimation. Notably, MMSE prevents the tracker from getting trapped in a local optimum state (outlier) and helps reach to the other states. The problem is that MMSE makes the tracker to be driven away from the global optimum state as well. However, the problem rarely occurs because MMSE is also enhanced by MAP in our M4 estimation.

To obtained the best MMSE-MAP state, our method obtains samples over the chain 1 via two steps: the proposal step and the acceptance step. In the proposal step, by the proposal density function $Q([\mathbf{X}_t^*];[\mathbf{X}_t])$, a new interval of the state, $[\mathbf{X}_t^*]$, is proposed based on the previous interval of the state, $[\mathbf{X}_t]$. Our proposal density function is different from conventional ones because the function consists of interval values. To handle interval values, we design three proposal density functions for x,y

positions and the scale, $Q([X_t^{i*}] = [\underline{X}_t^{i*}, \overline{X}_t^{i*}]; [X_t^i])$ for i=1,2,3, to have the lower bound $\underline{X}_t^{i*}$ and the upper bound $\overline{X}_t^{i*}$ of the proposed interval of the state such like:

$$\begin{aligned} &\underline{X}_t^{i*} = \text{MIN}(X_t^{i*}),\ \overline{X}_t^{i*} = \text{MAX}(X_t^{i*}), \\ &\text{where}\ \ X_t^{i*} \sim G(X_t^i, \sigma_i)\ \ \text{for all}\ \ X_t^i \in [X_t^i]. \end{aligned} \tag{4.56}$$

In (4.56), $G(X_t^i, \sigma_i)$ denotes the Gaussian function with mean X_t^i and variance σ_i. Then, $\dot{\mathbf{X}}_t^*$ is obtained by getting expectation of $[\mathbf{X}_t^*]$ with respect to $p(\mathbf{Y}_t|[\mathbf{X}_t^*])$. Note that the transition probability $p([\mathbf{X}_t^*]|[\mathbf{X}_t])$ in (4.54) is realized by $Q([\mathbf{X}_t^*]; [\mathbf{X}_t])$ in (4.56). Given the proposed interval of the state, the chain decides whether the proposed interval of the state is accepted or not with the acceptance ratio in the acceptance step. The acceptance ratio is designed like:

$$a_1^p = min\left[1, \frac{p_C(\mathbf{Y}_t|\dot{\mathbf{X}}_t^*)Q([\mathbf{X}_t]; [\mathbf{X}_t^*])}{p_C(\mathbf{Y}_t|\dot{\mathbf{X}}_t)Q([\mathbf{X}_t^*]; [\mathbf{X}_t])}\right], \tag{4.57}$$

where $p(\mathbf{Y}_t|\dot{\mathbf{X}}_t)$ in (4.52) is the likelihood without interval. These two steps iteratively proceed until the number of iterations reaches a predefined value.

4.3.3.2 Tracker 2: ML-MUP Estimation

The second posterior in (4.54) induces the second tracker. The goal of the tracker 2 is to find the best interval of the state, which maximizes $p_U([\mathbf{X}_t]|\mathbf{Y}_{1:t})$ in (4.55). To maximize $p_U([\mathbf{X}_t]|\mathbf{Y}_{1:t})$, the interval width $\overline{X}_t^i - \underline{X}_t^i$ and λ_i in (4.53) should be minimized. Minimizing the interval width $\overline{X}_t^i - \underline{X}_t^i$ results in decreasing the uncertainty of the posterior (MUP) estimation. To minimize λ_i, a derivative of the likelihood in $\lambda_i = \text{MAX}\left(\left|\frac{\partial}{\partial X_t^i} p(\mathbf{Y}_t|[\mathbf{X}_t])\right|\right)$ should be minimized. It is consistent with the ML (Maximum Likelihood) estimation. The ML estimation also finds the state where a derivative of the likelihood over the state becomes small. Hence, maximizing $p_U([\mathbf{X}_t]|\mathbf{Y}_{1:t})$ is called ML-MUP.

The best ML-MUP state can be achieved by obtaining samples over the chain 2 via two steps: the proposal step and the acceptance step. The proposal step is same with that of the tracker 1. In the acceptance step, our method frequently accepts the states with the following acceptance ratio:

$$a_2^p = min\left[1, \frac{p_U(\mathbf{Y}_t|[\mathbf{X}_t])Q([\mathbf{X}_t];[\mathbf{X}_t^*])}{p_U(\mathbf{Y}_t|[\mathbf{X}_t^*])Q([\mathbf{X}_t^*];[\mathbf{X}_t])}\right], \tag{4.58}$$

where $p_U(\mathbf{Y}_t|[\mathbf{X}_t])$ in (4.53) is uncertainty of the likelihood estimation.

4.3.3.3 Interaction between two Trackers

Our method consists of two modes, parallel and interacting. In the parallel mode, the method acts as the parallel Metropolis Hastings algorithms by using (4.56)(4.57) for tracker 1 and (4.56)(4.58) for tracker 2. When the method is in the interacting mode, the trackers communicate with the other and make leaps to better states of an object. In the interacting mode, two trackers accepts the state of the tracker 1 as their next states with the following probability:

$$a_1^i = \frac{p_C(\mathbf{Y}_t|\dot{\mathbf{X}}_t)}{p_C(\mathbf{Y}_t|\dot{\mathbf{X}}_t) + [p]_U(\mathbf{Y}_t|[\mathbf{X}_t])}. \tag{4.59}$$

Similarly, two tracker accepts the state of the tracker 2 as their next state with the following probability:

$$a_2^i = \frac{[p]_U(\mathbf{Y}_t|[\mathbf{X}_t])}{p_C(\mathbf{Y}_t|\dot{\mathbf{X}}_t) + [p]_U(\mathbf{Y}_t|[\mathbf{X}_t])}. \tag{4.60}$$

By the interaction between two trackers, common states can be evaluated by both tracker 1 and 2. Our method operates in an interacting mode with the probability α, which linearly decreases from 1.0 to 0.0 as the simulation goes on. Note that ω_C and ω_U in (4.55) are considered as $\frac{\text{\# of accepted states of tracker 1}}{\text{\# of proposed states}}$ and $\frac{\text{\# of accepted states of tracker 2}}{\text{\# of proposed states}}$ in the interacting mode, respectively.

4.3.4 Implementation Details

Initialization: At the initial frame, we manually draw the bounding box over the target region, which determines the x,y positions (X^1, X^2) and scale $X^3$of the target. Then, the initial interval of the state in (4.45), $[\mathbf{X}_0] = \left[(\underline{X}_0^1, \underline{X}_0^2, \underline{X}_0^3)^T, (\overline{X}_0^1, \overline{X}_0^2, \overline{X}_0^3)^T\right]$ is set to $\underline{X}_0^1 = X^1 - 0.25B_w, \overline{X}_0^1 = X^1 + 0.25B_w$, $\underline{X}_0^2 = X^2 - 0.25B_h, \overline{X}_0^2 = X^2 + 0.25B_h$, $\underline{X}_0^3 = X^3 - 0.05,$ and $\overline{X}_0^3 = X^3 + 0.05$, where B_w and B_h denote the width and height of the initial bounding box, respectively. The reference target model M_0 in (4.52) is made using the image patch inside of the initial bounding box. Then, at each frame, the interval of the state, $[\mathbf{X}_t]$, and the target model, M_t, is initialized with the similar manner based on the best state at the previous frame, $\hat{\mathbf{X}}_{t-1}$.

Final Representation: At each frame, the best state of the target, $\hat{\mathbf{X}}_t$, is represented as

$$\hat{\mathbf{X}}_t = \mathbb{E}_{p(\mathbf{Y}_t|[\mathbf{X}_t])}[\hat{\mathbf{X}}_t], \tag{4.61}$$

where $[\hat{\mathbf{X}}_t]$ indicates the best interval of the state, which is found by (4.55). The final representation of the target state in (4.61) enables our tracking results to be evaluated and to be compared with other tracking methods. This final representation can be justified both theoretically and empirically. Theoretically, the likelihood at the state $\hat{\mathbf{X}}_t$ is equal to expectation of $[p](\mathbf{Y}_t|[\mathbf{X}_t])$ at interval of the state, $[\hat{\mathbf{X}}_t]$, by Lemma 1. Empirically, the interval width decreases and usually converges into a single state, as demonstrated in the convergence part of Section 4.3.5.1. Therefore, the reference target model M_t in (4.52) can be updated at each time t using the image patch described by the best state $\hat{\mathbf{X}}_t$.

Approximation: To estimate $\dot{X}_t^i$ and λ_i in (4.47) and to get $\underline{X}_t^{i*}$ and $\overline{X}_t^{i*}$ in (4.56),

Algorithm 6 Interval based Tracking method (IT)

Initialization: Initialize interval $[\mathbf{X}_t]$ (Section 4.3.4).

Input: $[\hat{\mathbf{X}}_{t-1}]$ and $\alpha = 1$

Output: $[\hat{\mathbf{X}}_t]$ and $\hat{\mathbf{X}}_t$

for 1 to N **do**

  Choose mode. Sample $\rho \sim U[0, 1]$. T means a tracker.

  if $\rho < \alpha$ **then**

    T1 adopts interval of T2 with probability a_2^i in (4.60).

    T2 adopts interval of T1 with probability a_1^i in (4.59).

  else

    T1 and 2 propose interval by $Q([\mathbf{X}_t^*]; [\mathbf{X}_t])$ in (4.56).

    T1 accepts the interval with probability a_1^p in (4.57).

    T2 accepts the interval with probability a_2^p in (4.58).

  end if

  Decrease the α value.

end for

Find the best interval $[\hat{\mathbf{X}}_t]$ using (4.55).

Final representation: Find the best state $\hat{\mathbf{X}}_t$ using (4.61).

our method should consider all $X_t^i \in [X_t^i]$. Because it is intractable to consider all, our method samples the 10 number of X_t^i and approximately obtain $\dot{X}_t^i$, λ_i, $\underline{X}_t^{i*}$, and $\overline{X}_t^{i*}$. Our method also approximates the derivative of the likelihood function, $\frac{d}{d\mathbf{X}_t} p(\mathbf{Y}_t|[\mathbf{X}_t])$, in (4.50) by using finite differences[4]. Algorithm 6 describes the whole procedure of the proposed method.

[4] http://en.wikipedia.org/wiki/Finite_difference

Table 4.8: **Tracking results by plug-in**. The numbers indicate average center location errors in pixels. To get the numbers, we averaged tracking results of all datasets.

	MC	WLMC	BHMC	VTD	VTS
Baseline (A)	107	101	98	72	71
Combined by IT (B)	58	53	42	9	15
Improvement (A-B)	49	48	56	**63**	56

4.3.5 Experimental Results

The proposed method (IT) was compared with recent 11 tracking methods: MS [19], MC [18], WLMC [22], BHMC [24], FRAGT [56], IVT [1], MIL [6], L1T [47, 48], MTT [49], VTD [26], and VTS [27]. We adjusted parameters of each tracker to produce best tracking results, while our method utilized the *fixed parameter setting* in all experiments. γ_C in (4.52) and γ_U in (4.53) were set to 5 for all experiments. For the sampling-based tracking methods, we used the same number of samples, 800. For other methods, we used the authors' codes. The tracking methods were evaluated using totally 15 challenging datasets: 7 benchmark datasets from [1], [6], and [85]; 4 real-world datasets from [26] and; 4 our datasets, which are *mission*, *penguin*, *rhinoceros*, and *terminator*.

4.3.5.1 Analysis of the Proposed Method

Applicability via Plug-in: Our method is highly applicable because it can be easily combined with other tracking methods and it greatly improves the tracking performance of the original methods. Table 4.8 demonstrates that the accuracy of the original tracking methods increases as our method is plugged in. Our method combined with VTD is the best in terms of tracking accuracy. It reduced the center location errors by 87% ($\frac{B-A}{A}$). The tracking results of other tracking methods were improved to about 50%. VTD was well fitted with our method because using

(a) From #107 (b) From #85 (c) From #25 (d) From #58

Figure 4.14: **Example of different start frames**. The initial target appearances is corrupted by occlusion like (a), pose variation like (b), illumination like (c), and blur like (d). Our method represents this ambiguities of the target appearance and the state as interval and solves it with the M4 estimation.

Table 4.9: **Tracking results with different start frames.** The numbers indicate average center location errors in pixels. To get the numbers, we averaged tracking results of all datasets. IT* denote our method combined by VTD, which produces the most accurate tracking result among the methods in Table 4.8.

	MS	MC	FRAGT	IVT	MIL	L1T	MTT	VTD	VTS	IT*
Baseline (A)	96	107	93	89	88	58	45	72	71	9
Random (B)	151	149	138	119	117	87	70	130	127	11
Decrement (B-A)	55	42	45	30	29	29	25	58	56	**3**

multiple estimators boosted effectiveness of using multiple appearance and motion models. The speed of our method depends on the original tracking algorithms, which is combined with our method. For example, our method combined with MC shows real-time performance. Our method combined with VTD takes $1 \sim 5$ seconds per frame. The plug-in process is very simple if the original tracking methods are based on MCMC such like WLMC, BHMC, VTD and VTS. The plug-in process replaces each Markov Chain of the original methods with two Markov Chains constructed by our process, which is described in line 4-13 of Algorithm 6. Initialization and final representation processes are explained in Section 4.3.4.

Robustness to Different Start Frames: Additional advantage of our method is that it is robust to initialization of the state and the target appearance. To

Table 4.10: **Tracking results with several estimation methods**. The numbers indicate average center location errors in pixels. The improvement is error difference between two neighbor steps.

	A step:MAP	B:A+MMSE	C:B+MUP	D:C+ML
IT*	72	59	31	9
Improvement	N/A	13	**28**	22

demonstrate it, we randomly selected 10 frames for each dataset and set them as the starting frames. We initialized the target state using the ground-truth configuration at the starting frame and constructed the target appearance using the image patch described by the state. Then, 10 tracking results from different start frames for each dataset were averaged. Figure4.14 shows a few examples of the randomly selected starting frames. The initial target appearance and state are very ambiguous due to the severe occlusion, pose variation, illumination, and blur. Nevertheless, as reported in Table 4.9, the tracking accuracy of our method hardly depends on the starting frames. The method solved the ambiguities by first representing them as the likelihood and state intervals and by gradually decreasing the width of intervals as time goes on.

Fusion of Several Estimation Methods: Our method can have aforementioned advantages and can accurately track the targets because it efficiently fused the several estimation methods. Table 4.10 describes how much each estimation method plays a role to improve the tracking accuracy. For example, in the C step, our method fused MAP, MMSE, and MUP. By additionally inserting MUP into the B step, the 28 amount of tracking errors was reduced. Our method most greatly enhanced the tracking accuracy by employing MUP, which demonstrates that MUP makes the overall algorithm successful. Introducing MUP into the estimation process is significant because the estimation could not be perfect and, thus, the estimation error should be considered during the tracking process.

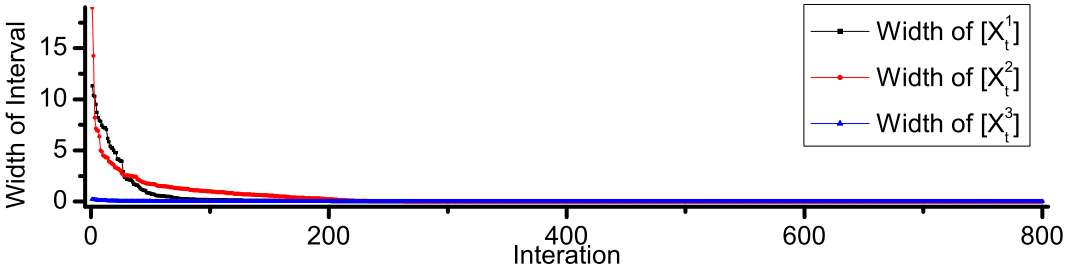


Figure 4.15: **The width of interval coverges as iteration goes on.**

Convergence: Our method has a real solution. The IMCMC [88] algorithm makes our method to converge, although the method fuses four estimation methods and combines two posterior distributions constructed by two trackers. In addition, MUP enables our method to produce a meaningful solution by decreasing the width of interval during the tracking process, although the method starts from the interval. This is also why the best solution can be represented by a single state in (4.61) instead of interval. Figure 4.15 empirically demonstrates that the width of interval decreases and usually converges into a single state as time goes on.

4.3.5.2 Comparison with State-of-the-Art Methods

Tables 4.11 and 4.12 demonstrates that our method is the best in terms of tracking accuracy. For this experiment, several state-of-the-art tracking methods were compared using challenging benchmark datasets. Other tracking methods showed good tracking performance. However, when the target appearance and state were highly ambiguous in either the initial step or during the tracking process, those methods failed to accurately track the targets. Notably, the success rate results in Table 4.12 were consistent with the center locations results in Table 4.11. High center location error but low success rate produced by some tracking methods means that they are weak to deal with severe scale changes.

Table 4.11: **Comparison of tracking results using the center location error.** The numbers indicate average center location errors in pixels. Red is the best result and blue is the second-best result. IT* denote our method combined by VTD, which produces the most accurate tracking result among the methods in Table 4.8. *singer1** and *skating1** are the modified version of *singer1* and *skating1* to have partially low frame rate.

	MS	MC	FRAGT	IVT	MIL	L1T	MTT	VTD	VTS	IT*
car4	37	142	52	**3**	50	**3**	**3**	35	123	6
coke	31	40	63	30	21	29	**5**	43	34	**18**
david	88	49	46	**4**	23	19	7	12	7	**5**
girl	35	19	27	24	32	23	**5**	16	16	**9**
mission	199	261	227	201	171	192	229	201	**164**	**11**
penguin	152	210	94	54	249	68	**16**	129	95	**11**
rhinoceros	**101**	209	214	214	238	156	210	208	224	**3**
shaking	241	97	61	95	38	66	9	**5**	**5**	**4**
*singer1**	51	149	59	**8**	29	**5**	45	11	25	10
*skating1**	77	163	85	160	64	78	63	**8**	9	**8**
soccer	97	47	82	151	41	40	**17**	21	**15**	21
sylv	13	13	11	48	11	**5**	**5**	21	15	**3**
terminator	200	158	307	236	328	140	**104**	318	308	**10**
tiger1	93	27	40	65	15	23	28	13	**6**	**4**
tiger2	30	33	38	47	**17**	26	23	45	26	**6**

Table 4.12: **Comparison of tracking results using the success rate.** The numbers indicate the amount of successfully tracked frames (score > 0.5), where the score is defined by the overlap ratio between the predicted bounding box B_p and ground truth bounding box B_{gt}: $\frac{area(B_p \bigcap B_{gt})}{area(B_p \bigcup B_{gt})}$.

	MS	MC	FRAGT	IVT	MIL	L1T	MTT	VTD	VTS	IT*
car4	39	20	43	**95**	34	85	**95**	39	38	77
coke	31	16	25	31	37	31	**90**	17	17	**39**
david	15	17	21	**98**	29	31	70	35	55	**97**
girl	39	48	41	43	41	43	**75**	50	50	**51**
mission	18	13	12	20	19	19	22	18	**19**	**98**
penguin	17	14	20	45	16	33	**65**	12	12	**88**
rhinoceros	**21**	12	12	14	12	14	16	15	15	**98**
shaking	13	23	25	22	82	22	87	90	**98**	**99**
*singer1**	22	21	22	**91**	32	**98**	33	76	33	69
*skating1**	26	22	28	22	37	26	33	**87**	82	**84**
soccer	22	24	22	20	26	26	**34**	32	**35**	33
sylv	67	66	86	75	87	**96**	**96**	70	80	**99**
terminator	14	13	12	12	**17**	12	13	13	13	**93**
tiger1	27	34	29	30	65	37	34	69	**80**	**97**
tiger2	15	16	27	23	**70**	30	40	23	33	**80**

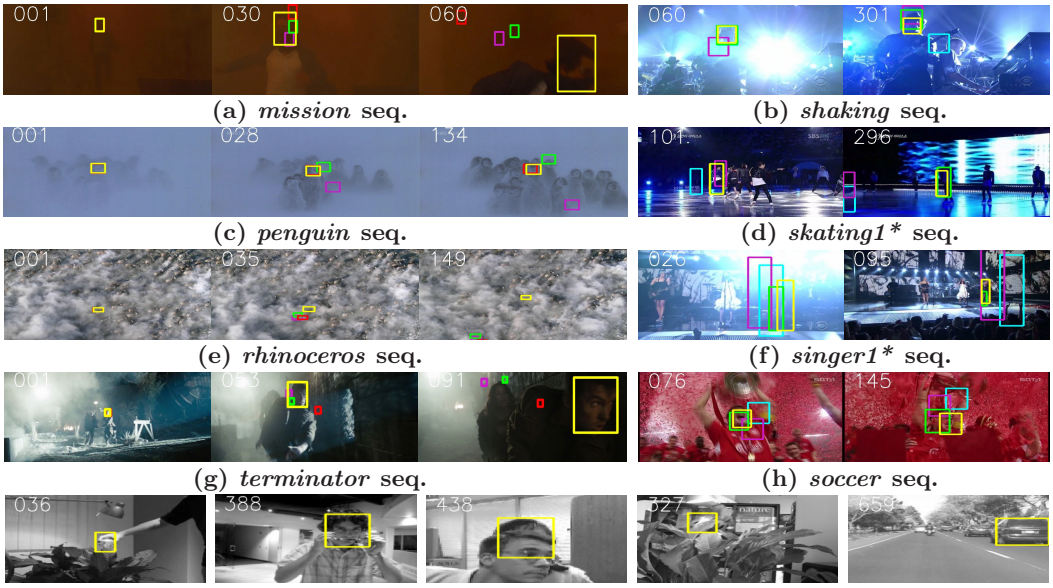


Figure 4.16: **Qualitative comparison of the tracking results using other methods.** The yellow, red, green, pink, and blue boxes represent the tracking results of IT*, MTT, VTS, MIL, and FRAGT, respectively.

Figure 4.16 shows qualitative tracking results of several state-of-the-art tracking methods. In Figure 4.16(a),(c),(e) and (g), the initial target models at the frame 1 were severely corrupted by occlusions and illumination changes. Nevertheless, our method (yellow boxes) robustly tracked the targets in the following frames. Other methods, MTT, VTS, and MIL failed to track the targets in the following frames due to this ambiguous initialization. In Figure 4.16(b),(d),(f) and (h), our method more accurately tracked the targets rather than other methods, even though the sequences include real-world tracking scenarios such as illumination changes, abrupt motions, and occlusions. In Figure 4.16(i)-(m), our method successfully tracked the targets on the widely used benchmark datasets.

4.3.6 Appendix

Lemma 6. If $\dot{X}_t^i = \mathbb{E}_{p(\mathbf{Y}_t|[X_t^i])}[X_t^i]$, the likelihood at the state $\dot{X}_t^i$ is equal to expectation of $[p](\mathbf{Y}_t|[X_t^i])$ with respect to $p(\mathbf{Y}_t|[X_t^i])$:

$$p(\mathbf{Y}_t|\dot{\mathbf{X}}_t) = \mathbb{E}_{p(\mathbf{Y}_t|[\mathbf{X}_t])}[p](\mathbf{Y}_t|[\mathbf{X}_t]). \tag{4.62}$$

Proof.

$$\begin{aligned}
&\mathbb{E}_{p(\mathbf{Y}_t|[\mathbf{X}_t])}[p](\mathbf{Y}_t|[\mathbf{X}_t]) = \int_A [p](\mathbf{Y}_t|[\mathbf{X}_t])p(\mathbf{Y}_t|[\mathbf{X}_t])\ dp(\mathbf{Y}_t|[\mathbf{X}_t]) \\
&= p(\mathbf{Y}_t|\dot{\mathbf{X}}_t)\int_A p(\mathbf{Y}_t|[\mathbf{X}_t])\ dp(\mathbf{Y}_t|[\mathbf{X}_t]) + \sum_{i=1}^{3}\lambda_i \otimes \int_A ([X_t^i] \ominus \dot{X}_t^i)p(\mathbf{Y}_t|[\mathbf{X}_t])\ dp(\mathbf{Y}_t|[\mathbf{X}_t]) \\
&= p(\mathbf{Y}_t|\dot{\mathbf{X}}_t) + \sum_{i=1}^{3}\lambda_i \otimes \underbrace{\int_A [X_t^i]p(\mathbf{Y}_t|[\mathbf{X}_t])\ dp(\mathbf{Y}_t|[\mathbf{X}_t])}_{\dot{X}_t^i = \mathbb{E}_{p(\mathbf{Y}_t|[\mathbf{X}_t])}[X_t^i]} - \sum_{i=1}^{3}\lambda_i \otimes \dot{X}_t^i \underbrace{\int_A p(\mathbf{Y}_t|[\mathbf{X}_t])\ dp(\mathbf{Y}_t|[\mathbf{X}_t])}_{1} \\
&= p(\mathbf{Y}_t|\dot{\mathbf{X}}_t) + \sum_{i=1}^{3}\lambda_i \otimes (\dot{X}_t^i - \dot{X}_t^i) = p(\mathbf{Y}_t|\dot{\mathbf{X}}_t).
\end{aligned} \tag{4.63}$$

Lemma 7. $\dot{X}_t^i$ is the MMSE estimate of X_t^i over $[X_t^i]$:

$$\dot{X}_t^i = \hat{X}_{\text{mmse}} = \arg\min_{\hat{X}} \mathbb{E}_{p(\mathbf{Y}_t|[X_t^i])}\|[X_t^i] - \hat{X}\|^2. \tag{4.64}$$

Proof. To find the MMSE state $\hat{X}_{\text{mmse}}$ over $[X_t^i]$, the derivative of $\mathbb{E}\Big(\|X_t^i - \hat{X}\|^2\Big)$ over $[X_t^i]$ is taken with respect to $\hat{X}$ to zero.

$$\begin{aligned}
&\frac{d}{d\hat{X}}\mathbb{E}\Big(\|[X_t^i] - \hat{X}\|^2\Big) = 0, \\
&\frac{d}{d\hat{X}}\mathbb{E}\|[X_t^i]\|^2 - 2\frac{d}{d\hat{X}}\hat{X}^T\mathbb{E}[X_t^i] + \frac{d}{d\hat{X}}\hat{X}^T\hat{X} = 0, \\
&\hat{X} = \mathbb{E}[X_t^i],
\end{aligned} \tag{4.65}$$

where $\hat{X}_{\text{mmse}} = \hat{X}$. Since $\dot{X}_t^i = \mathbb{E}[X_t^i]$, $\dot{X}_t^i = \hat{X}_{\text{mmse}}$.

Chapter 5

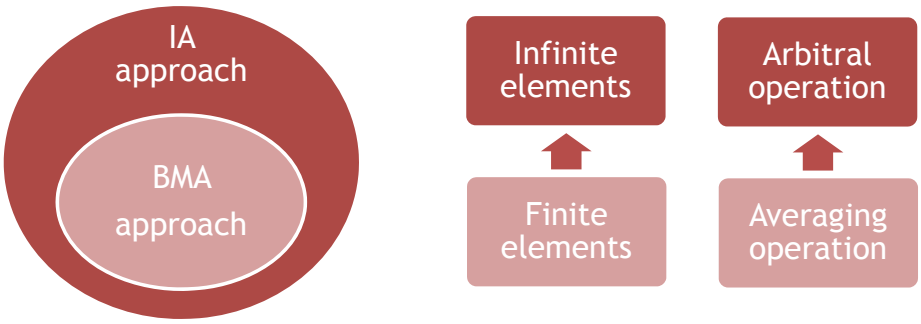
Conclusion and Future Work

Summary and contributions of the dissertation are presented in section 5.1. In section 5.2. future direction is briefly introduced.

5.1 Summary and Contributions of the Dissertation

To solve ambiguities in probabilistic models, two approaches, IA and BMA, were presented. In two approaches, IA is the superset of BMA, as described in Figure 5.1(a). IA has advanced two properties in Figure 5.1(b). First, IA utilizes an infinite number of candidates in interval, while BMA only utilizes a finite number of probabilistic model candidates. Second, IA can perform any arbitral operations using candidates, while BMA only averages probabilistic model candidates.

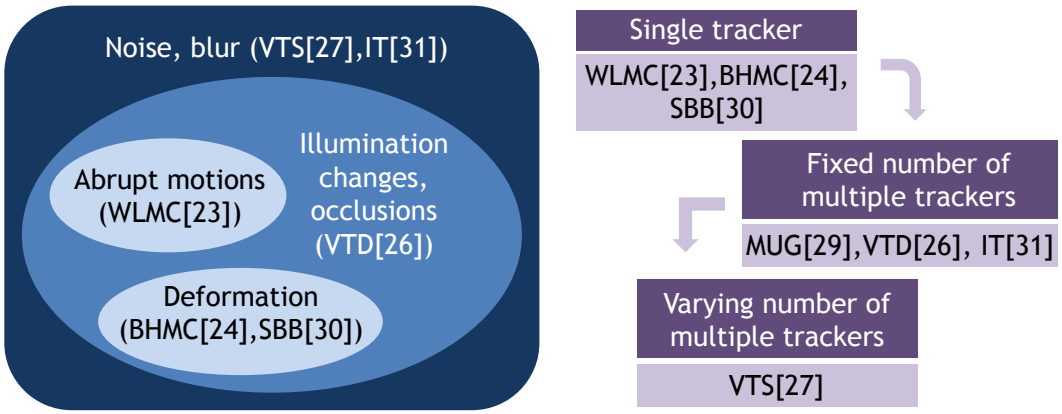
Using the IA and BMA approaches, the tracking methods have been developed toward tracking the targets under more complex and combinatorial tracking environment, as depicted in Figure 5.2(a), and toward using varying number of multiple trackers, as depicted in Figure 5.2(b).



(a) IA ⊃ BMA. (b) Advanced properties of IA

Figure 5.1: **Development direction to solve ambiguities in probabilistic models**.

To track both smooth and abrupt motions, the WLMC based trackers are proposed. To the best of our knowledge, it is the first to introduce the Wang-Landau sampling method to address the tracking problem. In the framework, the method provides an efficient sampling schedule by employing the DOS information. It encourages sampling from less-visited subregions of the state space while spending more time to explore subregions where the target might be. With the scheduling, the method provides a statistical way to reach the global maximum. The Wang-Landau sampling method is modified into an annealed version and present the AWLMC tracker. When the Wang-Landau sampling method obtains diverse samples in a whole state space, it needs a vast number of samples. To enhance the efficiency of the sampling process, the method sequentially reduces the state space into a smaller one, which contains the target states compactly. The performance of the WLMC and AWLMC trackers depends on the accuracy of the DOS estimate. However, given a fixed number of samples, the accuracy substantially decreases as the dimension of the state space increases. To preserve good performance in the high-dimensional state space, the N-Fold way algorithm is adapted, which can estimate the DOS with



(a) Tracking environment (b) Tracking strategy

Figure 5.2: **Development direction of the tracking methods**.

a very small number of samples. Moreover, the NFWL tracker is a rejection-free algorithm. It always accepts the proposed states using its efficient proposal density. This property also enables the method to need a smaller number of samples. With the efficient sampling in the high-dimensional state space, the NFWL-based tracking method successfully tracks the target even when there are drastic changes in scale as well as position.

To track severely deformable targets, the BHMC based trackers are proposed. For this, A new local patch-based appearance model and its online update scheme for highly non-rigid objects are proposed. The appearance model comprises multiple local patches and the topology between those patches, which covers geometric changes while preserving spatial information of the target. The proposed model needs *no* specific object model and *no* training phase for learning the appearance or behavior of the object. Instead, the model evolves automatically through a novel update scheme, reflecting the photometric and geometric appearance changes of the target. A novel likelihood landscape analysis (LLA) is proposed for the update scheme and

employed to measure the robustness of each patch. Using LLA, the robustness of a patch is measured by evaluating the degree of smoothness and steepness of the likelihood landscape of the patch. The ABHMC tracker is proposed to reduce the complexity in the proposed appearance model. The BH sampling method simplifies the landscape of a solution space by combining the Monte Carlo method with a deterministic local optimizer. The method gives an efficient way to reach the global optimum using a small number of samples, even in a huge solution space, that is associated with a large number of local patches. The ABHMC tracker is extended and the ABHMC-F and ABHMC-FS trackers are proposed. The ABHMC tracker is designed to deal with geometric appearance changes in particular. This method is extended to cope with severe illumination conditions and scale changes as well. To accomplish the extension of the method, the appearance model is enhanced with multiple features, and the ABHMC-F tracker is developed. In the enhanced appearance model, the local patches are constructed using different features, and a good feature for each patch is selected automatically. With adaptive feature selection, the method deals with local appearance changes of the target and tracks it robustly during local changes in the illumination conditions.The likelihood function is also improved using the rough segmentation results, and the ABHMC-FS tracker is developed. Using the segmentation results, the re-designed likelihood function covers severe scale changes in the target.

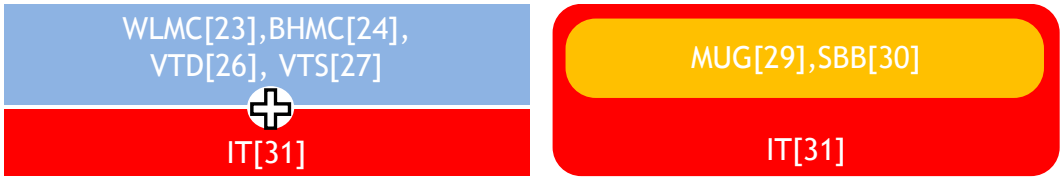
To track severely deformable targets, the SBB tracker is also proposed, which employs a new bounding box representation. A highly non-rigid object cannot be adequately described by any single bounding box. To solve the ambiguity in the conventional bounding box representation, the target is represented as a range of the bounding box called the SBB. Based on the theory of evidence, it is proved

that the inner bounding box and the outer bounding box form the lower bound and the upper bound of the bounding box, respectively. In addition, it is proved that the optimal inner bounding box is obtained by maximizing the similarity between the region inside of the bounding box and the target model; and the optimal outer bounding box is obtained by maximizing the dissimilarity between the region outside of the bounding box and the target model. Finally, an efficient tracking system using the SBB representation and a new Constrained Markov Chain Monte Carlo (CMCMC) sampling method are presented. The SBB representation explores the complementary connection between the inner and the outer bounding boxes to track highly non-rigid targets accurately. In practice, the inner bounding box is robust to the deformation of the target but sensitive to noise because of its small size. The outer bounding box is resistant to noise but imprecise on the deformation of the target because of its large size. Hence, these two representations complement each other to be insensitive to both deformation and noise. The CMCMC sampling method efficiently determines the best states of the inner and outer bounding boxes while addressing the constraint in which the outer bounding box must include the inner bounding box.

To deal with severe appearance changes, the MUG tracker is proposed. The MUG tracker utilizes the *MUG* instead of the MAP estimation. By finding the state that minimizes the gap between the likelihood bounds, the tracker overcomes the drift problem caused by a noisy target model in conventional MAP estimation and successfully tracks the target when there are severe illumination changes, occlusions, and pose variations. For this, the lower and upper bounds of the likelihood is optimally derived. An efficient strategy to obtain the state that has the minimum uncertainty gap is proposed. The MUG tracker constructs two chains and inferences

the best state on the chains using the IMCMC method. In the first chain, the proposed method finds the state that maximizes the average bound of the likelihood. In the second chain, the method searches for the state that minimizes the gap between the likelihood bounds. These chains communicate with each other to obtain the best state that maximizes the average bound and minimizes the gap between bounds at the same time.

To deal with abrupt motions, deformation, illumination changes, and occlusions at the same time, the VTS tracker is proposed. The VTS tracker was tested using unconstrained videos obtained from broadcast networks, such as music concerts, sports events, or documentaries. In these videos, the tracker obtained more accurate and reliable tracking results compared with state-of-the-art tracking algorithms. To design the tracker, four important ingredients of the Bayesian tracking approach is fully considered, which are the appearance model, motion model, state representation type, and observation type, thereby making the tracker robust against a wider range of variations, including occlusions, illumination changes, abrupt motions, severe noise, and motion blur. Using these components, the VTS tracker construct multiple basic trackers and integrates them into one robust compound tracker while interactively improving the performance of all basic trackers. If severe appearance or motion changes occur, the tracker increases the number of trackers and spends more resources to track the target. If not, the tracker decreases trackers and saves resources. This process can be achieved because the proposed framework allows the addition or removal of a tracker itself. By doing this, the VTS tracker reduces computational costs compared with conventional methods, which always utilize a fixed number of trackers or samples. Additionally, the VTS tracker evolves toward reflecting the target characteristic over time. The components and parameters comprising

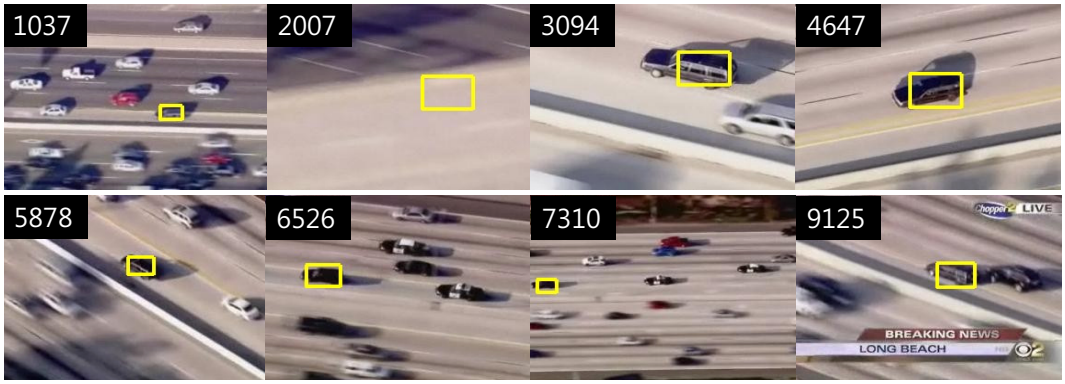


(a) IT vs. WLMC, BHMC, VTD and VTS (b) IT vs. MUG and SBB

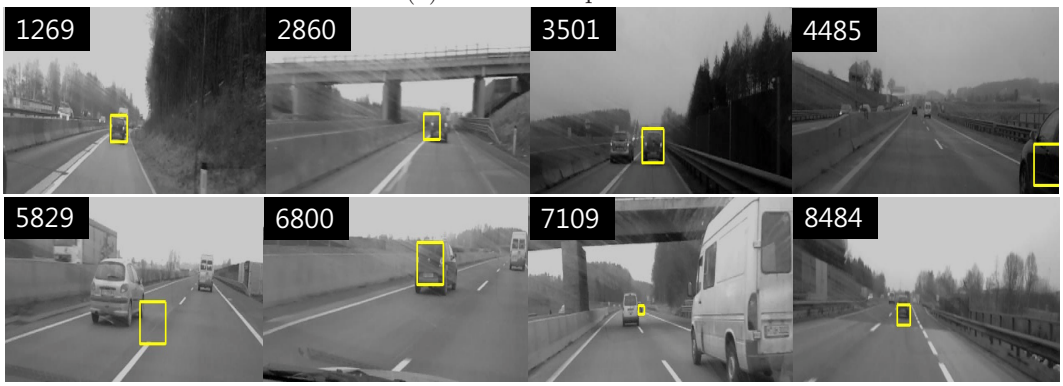
Figure 5.3: **IT vs. other tracking methods**.

the trackers adaptively change during the tracking process by learning multiple cues in the video, thereby improving tracking accuracy significantly in the real-world tracking environment.

Finally, the IT-based tracking method is proposed, which includes all aforementioned trackers. IT can be combined with WLMC, BHMC, VTD and VTS, as shown in Figure 5.3(a). Then, IT combined with WLMC, BHMC, VTD and VTS improve the tracking performance of original WLMC, BHMC, VTD and VTS. In addition, IT integrates MUG and SBB because IT utilizes the intervals of both the state in SBB and appearance models in MUG, as shown in Figure 5.3(b). IT presents a novel M4 estimation for visual tracking and improves the MAP estimation in the following three aspects. First, the M4 estimation can find the MAP state, which is robust to outliers by the help of the MMSE estimation. Second, the M4 estimation can find the MAP state, of which the estimation is made to be very confident by the MUP estimation. Third, the M4 estimation can find the MAP state, which is also good with respect to the ML estimation. interval linearization is introduced, which efficiently decomposes a posterior with interval into the mean posterior without interval and the uncertainty of the posterior estimation. The mean posterior without interval is similar to conventional representation of the posterior. The uncertainty of the posterior estimation, however, cannot be mathematically obtained by conven-



(a) *carchase* sequence



(b) *volkswagen* sequence

Figure 5.4: **The tracking results of IT in long video sequences**.

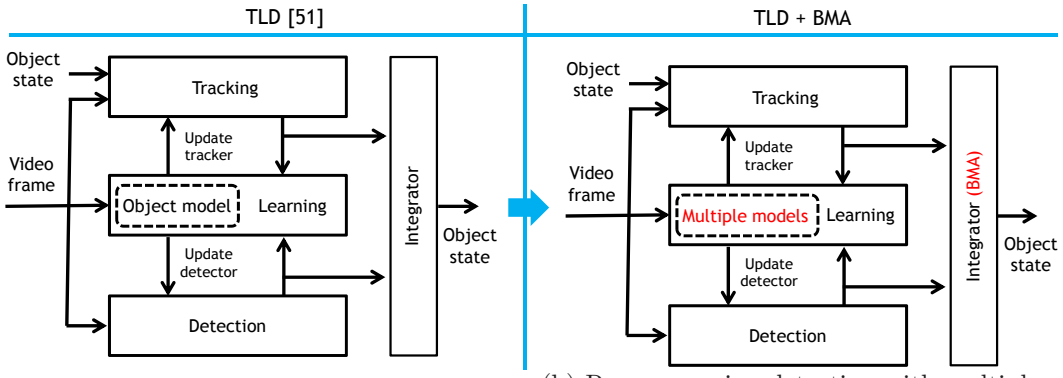
tional tracking approaches. Highly applicable tracking system is presented. IT is not sensitive to initialization of target positions and appearance models. It is robust even when information on target positions and appearance models is insufficient. In addition, IT can be easily integrated into existing tracking algorithms and can greatly improve their tracking performance.

However, the conventional trackers, no matter how complex and well designed, will fail eventually in long video sequences. In many practical applications it is quite rare that a single object is in view over extended period of time without

(a) After full occlusions (b) Recovery using interval

Figure 5.5: **Advantage of IT in long video sequences**.

full occlusion. Full occlusion makes most trackers to fail to track the targets. To solve this problem, the trackers require a way to recover from failure. In long video sequences, our trackers also fail due to full occlusion because our trackers do not have any recovery mechanism. Nevertheless, IT successfully tracked the targets in two long video sequences, namely *carchase* and *volkswagen*. This is because the interval in IT increased chances to recover the tracker after full occlusions. The *carchase* and *volkswagen* sequences include 9928 and 8576 frames, respectively. In these sequences, full occlusions and drastic appearance changes frequently occur. Figure 5.4 shows the tracking results of IT in long video sequences. When there were full occlusions at the frame 2007 and the frame 5829 in the *carchase* and *volkswagen* sequences, respectively, IT missed the targets. However, IT recovered the target states after full occlusions. If full occlusions occur, uncertainty in the target states increases, which makes a large state interval. And this large state interval of IT may include the true target states with high probability rather than a conventional single state, as demonstrated in Figure 5.5.



(a) Recovery using detection

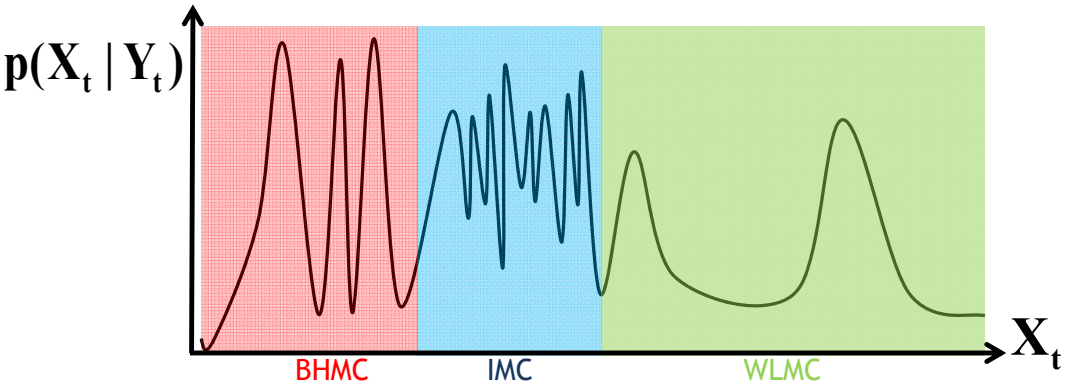
(b) Recovery using detection with multiple models

Figure 5.6: **Combination of TLD and BMA**.

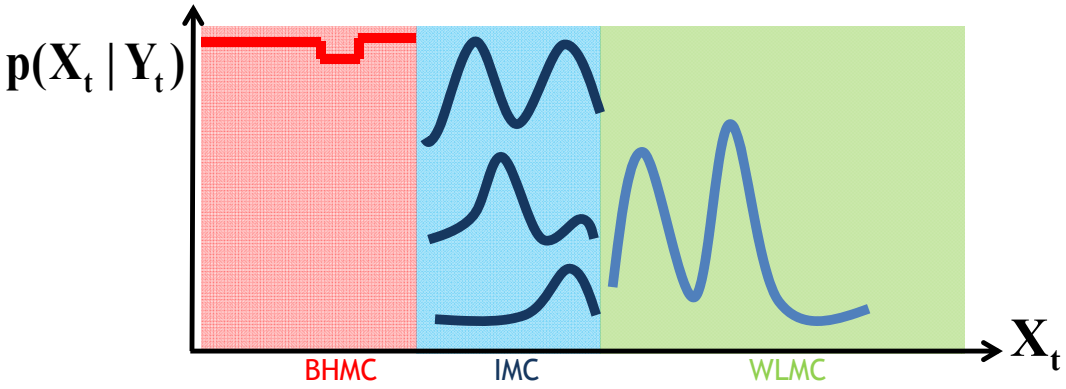
5.2 Future Directions

5.2.1 Explicit Recovery using Detection and Multiple models

To robustly handle full occlusions, the tracking method should explicitly decompose the long-term tracking task into tracking, learning and detection, as illustrated in Figure 5.6(a). The tracker follows the object from frame to frame. The detector localizes all appearances that have been observed so far and corrects the tracker if necessary. The learning estimates detector's errors and updates it to avoid these errors in the future. This idea is realized by the TLD framework in [51]. Then, this TLD framework can be easily enhanced by our BMA framework, which uses multiple appearance models, as illustrated in Figure 5.6(b). In the TLD+BMA framework, multiple detectors localize multiple appearances and correct multiple trackers. The learning estimates multiple detectors errors and updates them to avoid these errors. Using the TLD+BMA framework, we can reduce uncertainty in appearance models of trackers and detectors, and, thus, more robustly recover the trackers when they



(a) Rough posterior landscape



(b) Simple posterior landscape after adopting multiple inference algorithms

Figure 5.7: **Advantage of multiple inference algorithms**.

missed the targets after full occlusions.

5.2.2 Multiple Inference Algorithms

The fifth important ingredients of a visual tracker is an inference algorithm together with appearance, motion, state, and observation models. The inference algorithm should depend on the tracking environment. For example, Basin Hopping Monte Carlo based inference algorithm should be adopted to prevent the tracker from getting stuck in local optima if the tracking environment includes background clutters.

(a) Multiple appearance functions

(b) Multiple motion functions

Figure 5.8: **Multiple functions**.

Interacting Monte Carlo based inference algorithm describe a rough posterior using the combination of simple posteriors. If the state transition is very abrupt, Wang Landau Monte Carlo based inference algorithm should be adopted to alleviate the constraint of smooth state transitions. Then, after adopting multiple inference algorithms, a rough posterior landscape is transformed into a simple posterior landscape, which makes trackers to easily find the global optimum state, as shown in Figure 5.7. They all have unique strengths and weaknesses in different environments, but in general, none seems sufficient on its own. The preliminary work [109] introduces a hybrid sampling strategy that adaptively combines multiple sampling strategies.

5.2.3 Combination of Multiple Functions

To makes multiple appearance models, we used different target models $\theta_i, i = 1, \ldots, m$ while we used a same likelihood function f. To makes multiple motion models, we used different variances $\sigma_i, i = 1, \ldots, n$ with a same Gaussian function f. However, to exploit advantages of using multiple models, the function f should be also constructed differently, as shown in Figure 5.8. For example, the multiple functions $f_i, i = 1, \ldots, o$ can be constructed by using multiple similarity measures like diffusion distance [69] and Bhattacharyya similarity coefficient [33]. The pre-

liminary work [49] introduces four different similarity measures using L_{21}, L_{11}, L_{11}, and L_1 and exploits advantages of using multiple functions. L_{21} and L_{11} outperform L_{11} and L_1. That is because L_1 and L_11 trackers represent particles independently, while L_{21} and L_{11} capitalize on the dependencies among different particles to obtain a more robust joint representation [49].

5.2.4 Relations between Multiple Models

Whereas most studies present schemes to extract the time-invariant characteristics of the target and adaptively update the appearance model, the probabilistic dependency between sequential target appearances should be modeled. To actualize this interest, a new Bayesian tracking framework can be formulated under the autoregressive Hidden Markov Model (AR-HMM), where the probabilistic dependency between sequential target appearances is implied. During the learning phase at each time step, the tracker separates formerly seen target samples into several clusters based on their visual similarity, and learns cluster specific classifiers as multiple appearance models, each of which represents a certain type of the target appearance. Then the dependency between these appearance models is learned. During the searching phase, the target state is estimated by inferring the most probable appearance model under the consideration of its dependency on formerly utilized appearance models. The preliminary work [110] introduces the probabilistic dependency between appearances models is implied.

5.2.5 Adaptive Dimension of Multiple Models

To handle the dynamic tracking environment, the appearance model of the target object should be composed of proper subset of the candidate features. While numer-

ous criteria to evaluate generality and discriminability of the features in the subset have been proposed, the importance of determining the proper amount of it has been overlooked. However, since the size of the candidate feature pool which can be evaluated during the tracking process is strictly limited, and reliability of candidate features varies over the type of the target object, time, and degree of background clutter, pre-fixing the amount may cause the over-fitting or under-fitting of the model. To overcome, a probabilistic method to select the proper subset of the candidate features should be proposed, where the size of the subset is adaptively decided dependent on the dynamic tracking environment observed. The preliminary work in [111] introduces the probabilistic method to adaptively determine the proper dimension of the appearance model for visual tracking.

Bibliography

[1] D. A. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *Int'l J. Computer Vision*, vol. 77, no. 1, pp. 125–141, 2008.

[2] R. M. Neal, "Annealed importance sampling," *Technical Report No. 9805, Dept. of Statistics, University of Toronto*, 1998.

[3] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int'l J. Computer Vision*, vol. 88, no. 2, pp. 303–308, 2009.

[4] H. X. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: a review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, 2012.

[5] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, 2006.

[6] B. Babenko, M. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," *Proc. Conf. Computer Vision and Pattern Recognition*, 2009.

[7] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 10, pp. 1631–1643, 2005.

[8] C. L. H. Grabner and H. Bischof, "Semi-supervised on-line boosting for robust tracking," *Proc. European Conf. Computer Vision*, 2008.

[9] M. G. H. Grabner and H. Bischof., "Real-time tracking via on-line boosting," *Proc. British Machine Vision Conference*, 2006.

[10] B. Han and L. Davis, "On-line density-based appearance modeling for object tracking," *Proc. Int'l Conf. Computer Vision*, 2005.

[11] A. D. Jepson, D. J. Fleet, and T. F. E. Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 10, pp. 1296–1311, 2003.

[12] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool, "Coupled object detection and tracking from static cameras and moving vehicles," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 30, no. 10, pp. 1683–1698, 2008.

[13] X. Mei and H. Ling, "Robust visual tracking using l1 minimization," *Proc. Int'l Conf. Computer Vision*, 2009.

[14] S. Stalder, H. Grabner, and L. V. Gool, "Cascaded confidence filtering for improved tracking-by-detection," *Proc. European Conf. Computer Vision*, 2010.

[15] V. Philomin, R. Duraiswami, and L. Davis, "Quasi-random sampling for condensation," *Proc. European Conf. Computer Vision*, 2000.

[16] 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 lifespans," *Proc. Conf. Computer Vision and Pattern Recognition*, 2007.

[17] X. Zhou and Y. Lu, "Abrupt motion tracking via adaptive stochastic approximation monte carlo sampling," *Proc. Conf. Computer Vision and Pattern Recognition*, 2010.

[18] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 11, pp. 1805–1918, 2005.

[19] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *Proc. Conf. Computer Vision and Pattern Recognition*, 2000.

[20] W. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.

[21] A. Benavoli, M. Zaffalon, and E. Miranda, "Robust filtering through coherent lower previsions," *IEEE Trans. Automat. Contr.*, vol. 56, no. 7, pp. 1567–1581, 2011.

[22] J. Kwon and K. M. Lee, "Tracking of abrupt motion using wang-landau monte carlo estimation," *Proc. European Conf. Computer Vision*, 2008.

[23] ——, "Wang-landau monte carlo-based tracking methods for abrupt motions," *IEEE Trans. Pattern Anal. Machine Intell., Accepted to Publication*, 2012.

[24] ——, "Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling," *Proc. Conf. Computer Vision and Pattern Recognition*, 2009.

[25] ——, "Highly non-rigid object tracking via patch-based dynamic appearance modeling," *IEEE Trans. Pattern Anal. Machine Intell., In Revision*, 2012.

[26] ——, "Visual tracking decomposition," *Proc. Conf. Computer Vision and Pattern Recognition*, 2010.

[27] ——, "Tracking by sampling trackers," *Proc. Int'l Conf. Computer Vision*, 2011.

[28] ——, "Tracking by sampling and integrating multiple trackers," *IEEE Trans. Pattern Anal. Machine Intell., In Revision*, 2012.

[29] ——, "Minimum uncertainty gap for robust visual tracking," *Proc. Conf. Computer Vision and Pattern Recognition, Submitted*, 2013.

[30] J. Kwon, J. Roh, and K. M. Lee, "Visual tracking with soft bounding box representation," *Proc. Conf. Computer Vision and Pattern Recognition, Submitted*, 2013.

[31] J. Kwon and K. M. Lee, "Robust visual tracking with m4 estimation via interval analysis," *Proc. Conf. Computer Vision and Pattern Recognition, Submitted*, 2013.

[32] M. Isard and A. Blake, "Icondensation: Unifying low-level and high-level tracking in a stochastic framework," *Proc. European Conf. Computer Vision*, 1998.

[33] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," *Proc. European Conf. Computer Vision*, 2002.

[34] M. Isard and J. MacCormick, "Bramble: A bayesian multi-blob tracker," *Proc. Int'l Conf. Computer Vision*, 2001.

[35] J. MacCormick and A. Blake, "Probabilistic exclusion principle for tracking multiple objects," *Proc. Int'l Conf. Computer Vision*, 1999.

[36] K. Smith, D. Gatica-Perez, and J.-M. Odobez, "Using particles to track varying numbers of interacting people," *Proc. Conf. Computer Vision and Pattern Recognition*, 2005.

[37] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," *Proc. Conf. Computer Vision and Pattern Recognition*, 2004.

[38] G. Hua and Y. Wu, "Multi-scale visual tracking by sequential belief propagation," *Proc. Conf. Computer Vision and Pattern Recognition*, 2004.

[39] G. Roberts and J. Rosenthal, "Examples of adaptive MCMC," *J. Comput. Graph. Stat.*, vol. 18, no. 2, pp. 349–367, 2009.

[40] G. Roberts, A. Gelman, and W. Gilks, "Weak convergence and optimal scaling of random walk metropolis algorithms," *Ann. Appl. Prob*, vol. 7, no. 1, pp. 110–120, 1997.

[41] W. Li, X. Zhang, and W. Hu, "Contour tracking with abrupt motion," *Proc. Int'l Conf. Image Processing*, 2009.

[42] G. Schindler and F. Dellaert, "A rao-blackwellized parts-constellation tracker," *Proc. Int'l Conf. Computer Vision Workshop*, 2005.

[43] D. Ramanan, D. Forsyth, and A. Zisserman, "Tracking people by learning their appearance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 1, pp. 65–81, 2007.

[44] L. Cehovin, M. Kristan, and A. Leonardis, "An adaptive coupled-layer visual model for robust visual tracking," *Proc. Int'l Conf. Computer Vision*, 2011.

[45] M. Godec, P. M. Roth, , and H. Bischof, "Hough-based tracking of non-rigid objects," *Proc. Int'l Conf. Computer Vision*, 2011.

[46] S. M. S. Nejhum, J. Ho, and M.-H. Yang, "Visual tracking with histograms and articulating blocks," *Proc. Conf. Computer Vision and Pattern Recognition*, 2008.

[47] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," *Proc. Conf. Computer Vision and Pattern Recognition*, 2012.

[48] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 33, no. 11, pp. 2259–2272, 2011.

[49] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," *Proc. Conf. Computer Vision and Pattern Recognition*, 2012.

[50] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," *Proc. Int'l Conf. Computer Vision*, 2011.

[51] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 34, no. 7, pp. 1409–1422, 2012.

[52] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "Prost: Parallel robust online simple tracking," *Proc. Conf. Computer Vision and Pattern Recognition*, 2010.

[53] B. Han, S. Joo, and L. S. Davis, "Probabilistic fusion tracking using mixture kernel-based bayesian filtering," *Proc. Int'l Conf. Computer Vision*, 2007.

[54] W. Du and J. Piater, "A probabilistic approach to integrating multiple cues in visual tracking," *Proc. European Conf. Computer Vision*, 2008.

[55] B. Stenger, T. Woodley, and R. Cipolla, "Learning to track with multiple observers," *Proc. Conf. Computer Vision and Pattern Recognition*, 2009.

[56] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," *Proc. Conf. Computer Vision and Pattern Recognition*, 2006.

[57] V. Badrinarayanan, P. Perez, F. L. Clerc, and L. Oisel, "Probabilistic color and adaptive multi-feature tracking with dynamically switched priority between cues," *Proc. Int'l Conf. Computer Vision*, 2007.

[58] P. Chockalingam, N. Pradeep, and S. Birchfield, "Adaptive fragments-based tracking of non-rigid objects using level sets," *Proc. Int'l Conf. Computer Vision*, 2009.

[59] L. Lu and G. D. Hager, "A nonparametric treatment for location/segmentation based visual tracking," *Proc. Conf. Computer Vision and Pattern Recognition*, 2007.

[60] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient l1 tracker with occlusion detection," *Proc. Conf. Computer Vision and Pattern Recognition*, 2011.

[61] F. Faux and F. Luthon, "Robust face tracking using colour dempster-shafer fusion and particle filter," *FUSION*, 2006.

[62] R. Munoz-Salinas, R. Medina-Carnicer, F. Madrid-Cuevas, and A. Carmona-Poyato, "Multi-camera people tracking using evidential filters," *Ann. Math. Statist.*, vol. 50, no. 2009, pp. 732–749, 2009.

[63] S. Avidan., "Ensemble tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 2, pp. 261–271, 2007.

[64] M. Yang and Y. Wu, "Tracking non-stationary appearances and dynamic feature selection," *Proc. Conf. Computer Vision and Pattern Recognition*, 2005.

[65] J. Kwon, K. M. Lee, and F. C. Park, "Visual tracking via geometric particle filtering on the affine group with optimal importance functions," *CVPR*, 2009.

[66] F. Wang, S. Yua, and J. Yanga, "Robust and efficient fragments-based tracking using mean shift," *Int. J. Electron. Commun.*, vol. 64, no. 7, pp. 614–623, 2010.

[67] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artificial Intelligence Res*, vol. 4, no. 1, pp. 237–285, 1996.

[68] N. d. F. Yizheng Cai and J. J. Little, "Robust visual tracking for multiple targets," *Proc. European Conf. Computer Vision*, 2006.

[69] H. Ling and K. Okada, "Diffusion distance for histogram comparison," *Proc. Conf. Computer Vision and Pattern Recognition*, 2006.

[70] B. Schulz, K. Binder, and M. Mueller, "Flat histogram method of wang-landau and n-fold way," *Int. J. Mod. Phys. C*, vol. 13, no. 4, pp. 477–494, 2002.

[71] R. Belardinelli and V. Pereyra, "Wang-landau algorithm: A theoretical analysis of the saturation of the error," *J. Chem. Phys.*, vol. 127, no. 18, p. 184105, 2007.

[72] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," *Proc. Int'l Conf. Computer Vision*, 2011.

[73] C. Bibby and I. Reid, "Robust real-time visual tracking using pixel-wise posteriors," *Proc. European Conf. Computer Vision*, 2008.

[74] D. Crandall, P. Felzenszwal, and D. Huttenlocher, "Spatial priors for part-based recognition using statistical models," *Proc. Conf. Computer Vision and Pattern Recognition*, 2005.

[75] R. Fergus, P. Perona, and A. Zisserman, "A sparse object category model for efficient learning and exhaustive recognition," *Proc. Conf. Computer Vision and Pattern Recognition*, 2005.

[76] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. Int'l Joint Conf. Artificial Intelligence*, 1981.

[77] L. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 6, pp. 810–815, 2004.

[78] T. H. Kim, K. M. Lee, and S. U. Lee, "Generative image segmentation using random walks with restart," *Proc. European Conf. Computer Vision*, 2008.

[79] L. Zhan, B. Piwowar, W. K. Liu, P. J. Hsu, S. K. Lai, and J. Z. Y. Chen, "Multicanonical basin hopping: A new global optimization method for complex systems," *J. Chem. Phys.*, vol. 120, no. 12, pp. 5536–5542, 2004.

[80] A. Saffari, M. Godec, T. Pock, C. Leistner, and H. Bischof, "Online multi-class lpboost," *Proc. Conf. Computer Vision and Pattern Recognition*, 2010.

[81] R. Collins, "Mean-shift blob tracking through scale space," *Proc. Conf. Computer Vision and Pattern Recognition*, 2003.

[82] B. Berg, "Introduction to markov chain monte carlo simulations and their statistical analysis," *Phys .Stat. Mech*, 2004.

[83] K. Hamacher and W. Wenzel, "The scaling behaviour of stochastic minimization algorithms in a perfect funnel landscape," *Phys. Rev. E*, vol. 59, no. 1, pp. 938–941, 1999.

[84] T. Herges, A. Schug, H. Merlitz, and W. Wenzel, "Stochastic optimization methods for structure prediction of biomolecular nanoscale systems," *Nanotechnology*, vol. 14, pp. 1161–1167, 2003.

[85] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," *Proc. Conf. Computer Vision and Pattern Recognition*, 1998.

[86] K. Toyama and E. Horvitz, "Bayesian modality fusion: Probabilistic integration of multiple vision algorithms for head tracking," *Proc. Asian Conf. Computer Vision*, 2000.

[87] C. Vondrick and D. Ramanan., "Video annotation and tracking with active learning," *Proc. Advances in Neural Information Processing Systems*, 2011.

[88] J. Corander, M. Ekdahl, and T. Koski, "Parallell interacting MCMC for learning of topologies of graphical models," *Data Min. Knowl. Discov.*, vol. 17, no. 3, 2008.

[89] A. d'Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet, "A direct formulation for sparse PCA using semidefinite programming," *SIAM Rev.*, vol. 49, no. 3, 2007.

[90] B. Zhang, M. Hsu, and U. Dayal, "K-harmonic means - a data clustering algorithm," *HP Technical Report*, 1999.

[91] J. Sullivan, A. Blake, M. Isard, and J. MacCormick, "Bayesian object localisation in images," *Int'l J. Computer Vision*, vol. 44, no. 2, pp. 111–135, 2001.

[92] I. J. Good, "Rational decisions," *J. Roy. Statistical Society*, vol. Ser. B., no. 14, pp. 107–114, 1952.

[93] A. E. Raftery and Y. Zheng, "Discussion: Performance of bayesian model averaging," *J. Amer. Statistical Assoc.*, vol. 98, 2003.

[94] J. Corander, M. Gyllenberg, and T. Koski, "Bayesian model learning based on a parallel MCMC strategy," *Stat. Comput.*, vol. 16, no. 4, pp. 355–362, 2006.

[95] T. B. Dinh, N. Vo, and G. G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," *Proc. Conf. Computer Vision and Pattern Recognition*, 2011.

[96] P. Kovesi, "Image features from phase congruency," *Videre: J. Computer Vision Research*, vol. 1, no. 3, 1999.

[97] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," *Proc. Conf. Computer Vision and Pattern Recognition*, 2012.

[98] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," *Proc. Int'l Conf. Computer Vision*, 2011.

[99] A. C. Courville, N. D. Daw, G. J. Gordon, and D. S. Touretzky, "Model uncertainty in classical conditioning," *Proc. Advances in Neural Information Processing Systems*, 2003.

[100] C. Jia, H. Shenb, and M. Westc, "Bounded approximations for marginal likelihoods," *Technical report*, 2010.

[101] C. I. Byrnes and A. Lindquist, "A convex optimization approach to generalized moment problems," *Control and Modeling of Complex Systems, Springer*, 2003.

[102] F. Liang, C. Liu, and R. J. Carroll, "Stochastic approximation in monte carlo computation," *J. Amer. Statist.*, vol. 102, no. 477, pp. 305–320, 2007.

[103] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *Ann. Math. Statist.*, vol. 38, no. 2, pp. 325–339, 1967.

[104] S. Glenn, "A mathematical theory of evidence," *Princeton University Press*, 1976.

[105] "http://en.wikipedia.org/wiki/dempster-shafer_theory."

[106] J. Lim, D. Ross, R.-S. Lin, and M.-H. Yang, "Incremental learning for visual tracking," *Proc. Advances in Neural Information Processing Systems*, 2005.

[107] G. Trombettoni, I. Araya, B. Neveu, and G. Chabert, "Inner regions and interval linearizations for global optimization," *Proc. AAAI Conf. Artificial Intelligence*, 2011.

[108] D. J. C. Mackay, "Introduction to monte carlo methods," *In Learning in Graphical Models, M. I. Jordan, Ed. NATO Science Series. Kluwer Academic Press*, 1998.

[109] D. Hsu and Z. Sun, "Adaptively combining multiple sampling strategies for probabilistic roadmap planning," *Proc. Int'l Conf. Robotics, Automation and Mechatronlcs*, 2004.

[110] D. W. Park, J. Kwon, and K. M. Lee, "Robust visual tracking using autoregressive hidden markov model," *Proc. Conf. Computer Vision and Pattern Recognition*, 2012.

[111] ——, "Visual tracking using adaptive dimensional appearance model," *Proc. Conf. Computer Vision and Pattern Recognition, Submitted*, 2013.

한글 초록

본 학위 논문에서는 실제 환경에서 강인하게 동작하는 여러 개의 물체 추적 기법들을 제안한다. 물체 추적기는 외양 모델, 움직임 모델, 상태 모델, 그리고 관측 모델의 4개의 중요한 요소로 이루워져 있다. 이 요소들은 확률 분포들로 나타내어진다. 이 확률 분포들이 모여 하나의 사후 확률 분포가 만들어지고 물체 추적의 목적은 관측값이 주어졌을 때 사후 확률을 최대로 하는 물체의 상태를 찾는 것이다. 그렇지만 실제로 많은 경우에 위 요소들에 대한 정보 부족으로 인하여 위의 분포들을 정확하게 디자인할 수 없기 때문에 하나로 만들어진 사후 확률 분포는 필연적으로 오류를 포함할 수 밖에 없다. 따라서 우리는 사후 확률에 대한 모호성이 있다는 것을 문제로 삼고 매우 어려운 물체 추적 환경에서도 정확하게 물체를 추적하는 새로운 기법들을 제안한다. 사후 확률에 대한 모호성을 해결하기 위해 두가지 방법 즉 베이시안 모델 평균화 기법과 구간 분석 기법을 이용한다. 두가지 방법의 핵심은 사후 확률의 모호성 때문에 사후 확률 분포가 하나로 유일하게 정해지지 않으며 사후 확률 분포을 표현하기 위해서는 여러개의 사후 확률 후보 분포들로 표현되어야 한다는 것이다. 모호성을 해결하는 방법과 모호성을 고려한 요소에 따라 WLMC, BHMC, VTS, MUG, SBB, 그리고 IT로 불리는 6개의 서로 다른 물체 추적 기법들을 제안한다.

먼저 베이시안 모델 평균화 기법을 이용하여 WLMC 물체 추적기는 느린 움직임과 빠른 움직임을 동시에 추적할 수 있도록 움직임 모델을 평균화하여 모호성을 줄인다. BHMC 물체 추적기는 여러개의 상태 모델을 평균화 한다. 여러개의 상태

모델을 이용하면 시간에 따라 물체의 기하학적 외양이 심하게 변할때에도 물체를 잘 묘사할 수 있다. VTS 물체 추적기는 여러개의 외양 모델와 움직임 모델과 더불어 여러개의 관측 모델과 상태 모델도 평균화 한다. 각각의 외양 모델와 움직임 모델은 물체가 가지는 서로 다른 종류의 외양과 움직임을 표현할 수 있다. 각각의 관측 모델과 상태 모델은 노이즈는 움직임 블러 현상이 일어나는 경우에서 물체 추적기가 강인하게 동작할 수 있도록 도와준다. 구간 분석 기법을 이용하여 MUG 물체 추적기는 우도의 최저값과 최고값 사이의 구간을 얻고 이 구간을 최소화 시킴으로써 우도 예측 오류를 줄인다. SBB 물체 추적기는 상태의 최조값과 최고값 사이의 구간을 얻고 이 구간을 이용하여 비강체 물체를 효율적으로 표현하다. IT 물체 추적기는 우도의 최저값과 최고값 사이의 구간과 상태의 최저값과 최고값 사이의 구간을 모두 얻고 M4 예측 기법을 통해 사후 확률을 최대로 하면서 사후 확률의 예측 오류를 줄이는 물체의 최적 상태를 얻는다.

베이시안 모델 평균화 기법은 다음과 같은 구간 분석 기법의 두가지 이점 때문에 구간 분석 기법에 포함된다고 할 수 있다. 첫번째 이점은 베이시안 모델 평균화 기법이 유한한 갯수의 모델 후보들을 사용함에 반해 구간 분석 기법은 무한한 갯수의 모델 후보를 사용한다는 점이다. 두번째 이점은 베이시안 모델 평균화 기법이 여러개의 모델에 대해 평균화만 수행할 수 있음에 반해 구간 분석 기법은 여러개의 모델에 대해 어떤 작업도 할 수 있다는 점이다. WLMC, BHMC, VTS, MUG, SBB, 그리고 IT 물체 추적기들은 보다 더 어려운 물체 추적환경에서 물체 추적을 강인하게 할 수 있도록 개발되었다. 이때 IT 추적기는 WLMC, BHMC, 그리고 VTS 추적기들과 결합될 수 있는데 이 경우 기존 물체 추적기의 물체 추적 성능은 개선된다. 그리고 IT 추적기는 MUG 추적기가 사용하는 우도 구간과 SBB 추적기가 사용하는 상태 구간을 모두 고려하기 때문에 이들 추적기의 상위 추적기라고 할 수 있다.

실험 결과에서 제안하는 물체 추적 기법들은 앞서 제기한 사후 확률의 모호성을 효과적으로 해결하고 있다. 여러개의 실제 비디오에서 제안하는 물체 추적 기

법들은 물체 추적 환경이 시간이 지남에 따라 매우 심하게 변함에도 불구하고 물체를 정확하고 강인하게 추적한다. 그리고 제안하는 기법들은 최근의 아주 좋은 물체 추적 기법들 보다 더 좋은 성능을 보인다.

주요어: 물체 추적, 확률 모델 모호성 분석, 베이시안 모델 평균화 기법, 구간 분석 기법, 마코브 체인 몬테 카를로

학번: 2008-30208