Highly Non-Rigid Object Tracking via Patch-based Dynamic Appearance Modeling

Junseok Kwon, *Student Member, IEEE,* and Kyoung Mu Lee, *Member, IEEE*

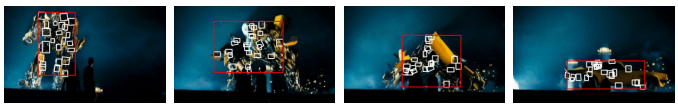
Abstract—A novel tracking algorithm is proposed for targets with drastically changing geometric appearances over time. To track such objects, we develop a local patch-based appearance model and provide an efficient online updating scheme that adaptively changes the topology between patches. In the online 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.

Index Terms—Object Tracking, Non-Rigid Object, Local Patch-based Appearance Model, Basin Hopping Sampling, Markov Chain Monte Carlo, Likelihood Landscape Analysis.

1 Introduction

Object tracking is one of the most important problems in computer vision. Practically, it can be used in a large number of different applications including surveillance, intelligent robots, augmented reality, medical imaging, and so on. Recent research trends address challenging real-world tracking problems more than experiments in a simple lab-like environment [1]. In real-world settings, objects are typically complex and difficult to track. 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 [2][3][4][5][6][7][8][9][10][11][12]. 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 1 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 [13][14] and pixel-wise models [15][6]. 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 [16][17][18] 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.



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

Fig. 1: **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.

The present work has the following main contributions:

• A new local patch-based appearance model and its online update scheme for highly non-rigid objects are proposed. The appearance model comprises multiple

• *The authors are with the Department of Electrical Engineering and Computer Science, Automation and Systems Research Institute, Seoul National University, 599 Gwanak-ro, Gwanak-gu, Seoul 151-744, Korea. E-mail: s98parad@gmail.com, kyoungmu@snu.ac.kr.*

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 adaptive Basin Hopping Monte Carlo (ABHMC)-based tracking method [9] 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 [19]. In our tracking problem, 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. This method is extended to an adaptive version of ABHMC by adding an adaptive proposal density, which further improves the sampling efficiency.
- The ABHMC-based tracking method is further extended and ABHMC-F- and ABHMC-FS-based tracking methods are proposed. The ABHMC-based tracking method 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-based tracking method 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-based tracking method is developed. Using the segmentation results, the redesigned likelihood function covers severe scale changes in the target. These extended works are described in detail in Sections 4.2 and 5.3.

2 Related Work

The related works are grouped into five categories.

Tracking methods with online appearance learning: By approximately estimating the pixel-wise color density in a sequential manner, Han et al. [6] successfully track an object where lighting conditions, pose, scale, and view-point change over time. Ross et al. [10] present 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 proposed method explicitly tackles these changes using a local patch-based online appearance model.

Tracking methods for non-rigid objects: Schindler et al. [20] represent 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. [21] propose 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. [22] successfully track humans in crowded environments where occlusion occurs persistently. All these tracking methods, however, basically assume that specific models of the targets are given. By contrast, the proposed method utilizes *no* prior knowledge of the specific model for the target and *no* off-line training phase. Cehovin et al. [23] combines local appearance with global appearance of the target using a novel coupled-layer visual model. Godec et al. [24] employs 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. The proposed ABHMC-FS tracker also uses global appearance obtained from a rough segmentation. However, in contrast with aforementioned two methods, the ABHMC 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 ABHMC-FS tracker shows better tracking performance under the challenging tracking environments including illumination changes as well as severe deformation of the targets.

Tracking methods using multiple patches: Adam et al. [25] present 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. [26] model 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. [27] 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 proposed local patch-based appearance model is more flexible, because

the patch may be removed, newly added, and moved by affine transformation and transition. Thus, the proposed method is able to track more severe non-rigid objects.

Tracking methods using segmentation results: Chockalingam et al. [28] 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 [29] 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 proposed method covers the temporally changing geometric structure of the targets. Hence, the method robustly tracks the targets, of which geometric appearance severely change over time.

Sampling-based tracking methods: In tracking problems, the particle filter [30] has shown efficiency in handling non-Gaussianity and multi-modality. The Markov Chain Monte Carlo (MCMC) method can be well applied to multi-object tracking problems because of its reduction of computational costs [31][32]. 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 proposed method, 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, our method can find a solution using smaller number of samples even in a very high-dimensional solution space.

3 Bayesian Object Tracking Approach

The tracking problem can be interpreted as Bayesian filtering. Given the state at time t, $\mathbf{X}_t$ and the observation up to time t, $\mathbf{Y}_{1:t}$, the Bayesian filter updates the posteriori probability $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ with the following rule:

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

where $p(\mathbf{Y}_t|\mathbf{X}_t)$ is the observation model that measures the similarity between the observation at the estimated state and the given model, and; $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ is the transition model that predicts the next state $\mathbf{X}_t$ based on the previous state $\mathbf{X}_{t-1}$. With the posteriori probability $p(\mathbf{X}_t|\mathbf{Y}_{1:t})$ computed by the observation and transition models, the Maximum a Posteriori (MAP) estimate over the N number of samples at each time t is obtained.

$$\hat{\mathbf{X}}_t = \underset{\mathbf{X}_t^{(l)}}{\arg\max}\, p(\mathbf{X}_t^{(l)}|\mathbf{Y}_{1:t}) \quad for\ l = 1, \ldots, N, \quad (2)$$

where $\mathbf{X}_t^{(l)}$ represents the l-th sample of the object state $\mathbf{X}_t$, and $\hat{\mathbf{X}}_t$ denotes the best sample of the object state, which explains the object configuration mostly well given the observations.

3.1 MCMC-based Tracking Method

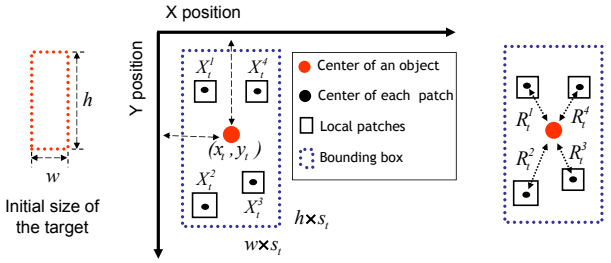
The integration in (1) is not feasible in a large state space. To solve this problem, we utilize the Metropolis Hastings (MH) algorithm [33], a popular MCMC sampling method. The MH algorithm defines a single Markov chain and acquires samples over the chain. To obtain samples, two main steps are performed, namely, the proposal and acceptance steps. These two steps are iteratively executed until the number of iterations reaches a predefined value.

• **Proposal Step:** The proposal step proposes a new state given the previous state, based on some prior knowledge of the motion. The most commonly used prior knowledge of the motion is that the transition is governed by the Gaussian distribution. Thus, the proposal density is designed by: $Q(\mathbf{X}_t^{(l+1)}; \mathbf{X}_t^{(l)}) = G(\mathbf{X}_t^{(l)}, \sigma^2)$, where $\mathbf{X}_t^{(l)}$ is the previous state, $\mathbf{X}_t^{(l+1)}$ is the new state, and G denotes the Gaussian function with mean $\mathbf{X}_t^{(l)}$ and variance σ^2.

• **Acceptance Step:** The acceptance step determines whether the new proposed state $\mathbf{X}_t^{(l+1)}$ is accepted or not. This step can be calculated simply by the likelihood ratio a between the previous and new states: $a = min\left[1, \frac{p(\mathbf{Y}_t|\mathbf{X}_t^{(l+1)})Q(\mathbf{X}_t^{(l)};\mathbf{X}_t^{(l+1)})}{p(\mathbf{Y}_t|\mathbf{X}_t^{(l)})Q(\mathbf{X}_t^{(l+1)};\mathbf{X}_t^{(l)})}\right]$, where $p(\mathbf{Y}_t|\mathbf{X}_t^{(l)})$ denotes the likelihood term over the state $\mathbf{X}_t^{(l)}$, and $Q(\mathbf{X}_t^{(l+1)}; \mathbf{X}_t^{(l)})$ represents the proposal density.

In the proposal and acceptance steps, the design of the state, $\mathbf{X}_t$ is crucial to the success of MCMC-based tracking approaches because it significantly affects the performance of the MCMC sampling method. Ordinarily, the state at time t is represented as a three-dimensional vector, $\mathbf{X}_t = (x_t, y_t, s_t)$, where x_t, y_t, and s_t indicate the xy center positions 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 4.

However, the local patch-based appearance model generates serious problems in 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 3.1 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



(a) Example of $\mathbf{X}_t = (x_t, y_t, s_t, \mathbf{X}_t^1, \mathbf{X}_t^2, \mathbf{X}_t^3, \mathbf{X}_t^4)$ (b) Example of $\mathbf{R}_t = (\mathbf{R}_t^1, \mathbf{R}_t^2, \mathbf{R}_t^3, \mathbf{R}_t^4)$

Fig. 2: **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 $\mathbf{R}_t$.

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 6.

4 Design of the Appearance Model

An object is represented by a local patch-based dynamic graph model as shown in Figure 2. In the proposed model, the object state $\mathbf{X}_t$ at time t is defined by $\mathbf{X}_t = (x_t, y_t, s_t, \mathbf{X}_t^1, \cdots, \mathbf{X}_t^i, \cdots, \mathbf{X}_t^m)$ where x_t, y_t, and s_t denotes the xy center positions and scale of an object, respectively, $\mathbf{X}_t^i = (x_t^i, y_t^i)$ 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, such like the star model [17][18]. The star model is frequently used because of its efficiency, which only considers the relation between center and each patch. As reported in [17], 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 increases. The k-fan model [16] has advantages of both the star and fully connected models. Although this model showed good performance for the recognition problem, the model did not produce good results for our tracking problem. The recognition problem typically considers a static scenario, in which the relations between patches undergo relatively small changes. On the other hand, the tracking problem should cover severe changes in relations between patches over time, especially if the target is highly non-rigid. To cover topology changes in the model both accurately and efficiently, the model should be designed with as simple relations between patches as possible, such like the star model, as addressed in [17][18]. In our tracking problem, the star model produced more accurate results with lower complexity compared with the k-fan 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 (x_t, y_t) 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 2(b). The objective of our tracking method is then finding the best sample of the object state, $\hat{\mathbf{X}}_t = (\hat{x}_t, \hat{y}_t, \hat{s}_t, \hat{\mathbf{X}}_t^1, \cdots, \hat{\mathbf{X}}_t^i, \cdots, \hat{\mathbf{X}}_t^m)$ using the MAP estimation in (2), where the l-th state sample is represented by $\mathbf{X}_t^{(l)} = (x_t^{(l)}, y_t^{(l)}, s_t^{(l)}, \mathbf{X}_t^{1(l)}, \cdots, \mathbf{X}_t^{i(l)}, \cdots, \mathbf{X}_t^{m(l)})$.

4.1 Photometric and Geometric Likelihoods

The likelihood is designed as:

$$p\left(\mathbf{Y}_t | O(\mathbf{X}_t^{(l)})\right) \approx \prod_{i=1}^{m} \left[p_p\left(\mathbf{Y}_t | O(\mathbf{X}_t^{i(l)})\right) p_g\left(O(\mathbf{X}_t^{i(l)}) | x_t^{(l)}, y_t^{(l)}, \mathbf{R}_t^i\right)\right], \quad (3)$$

where p_p denotes the photometric likelihood and p_g indicates the geometric likelihood. In (3), $O(\mathbf{X}_t^{i(l)})$ 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 [19]. 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:

$$p_p\left(\mathbf{Y}_t | O(\mathbf{X}_t^{i(l)})\right) = exp^{-\lambda_p F_1\left(I\left[O(\mathbf{X}_t^{i(l)})\right], \mathbf{M}_t^i\right)}, \quad (4)$$

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 (4) is updated by

$$\mathbf{M}_{t+1}^i = (1-\omega)\mathbf{M}^{i(ref)} + \omega \mathbf{M}_t^{i(dyn)}, \quad (5)$$

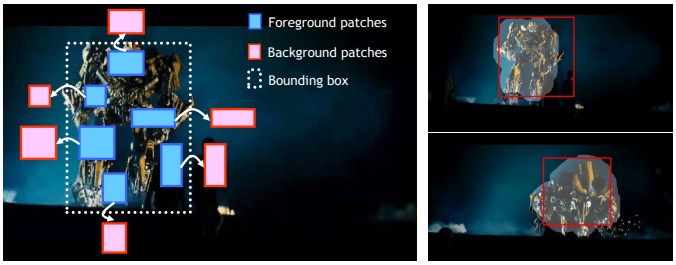
where $\mathbf{M}^{i(ref)}$ indicates the i-th reference local patch model in the initial frame and $\mathbf{M}_t^{i(dyn)}$ represents the model image obtained in the region of $O(\hat{\mathbf{X}}_t^i)$ at time t. The local patch model in the initial frame, $\mathbf{M}^{i(ref)}$, prevents from learning drastic appearance changes [34].

The geometric likelihood is defined by

$$p_g\left(O(\mathbf{X}_t^{i(l)}) | x_t^{(l)}, y_t^{(l)}, \mathbf{R}_t^i\right) = exp^{-\lambda_g \left\| \left[O(\mathbf{X}_t^{i(l)}) - (x_t^{(l)}, y_t^{(l)})\right] - \mathbf{R}_t^i \right\|_2}, \quad (6)$$

where $\|[O(\mathbf{X}_t^{i(l)}) - (x_t^{(l)}, y_t^{(l)})] - \mathbf{R}_t^i\|_2$ returns the 2-norm distance between two vectors $[O(\mathbf{X}_t^{i(l)}) - (x_t^{(l)}, y_t^{(l)})]$ and $\mathbf{R}_t^i$, and λ_g denotes the weighting parameter set to 1. In (6), $[O(\mathbf{X}_t^{i(l)}) - (x_t^{(l)}, y_t^{(l)})]$ 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

$$\mathbf{R}_{t+1}^i = (1-\omega)\mathbf{R}_t^i + \omega\left(O(\hat{\mathbf{X}}_t^i) - (\hat{x}_t, \hat{y}_t)\right), \quad (7)$$



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

Fig. 3: **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.

where $\hat{\mathbf{X}}_t^i$, $\hat{x}_t$, and $\hat{y}_t$ denote the MAP states for $\mathbf{X}_t^i$, x_t, and y_t, 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 5.4.

4.2 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(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.

With the foreground and background patches, the proposed method segments the target using the random-walk algorithm in [35], as described in Figure 3(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, $p_s(\mathbf{Y}_t|S[O(\mathbf{X}_t^{(l)})])$:

$$p\left(\mathbf{Y}_t|O(\mathbf{X}_t^{(l)})\right) \approx p_s\left(\mathbf{Y}_t|S\left[O(\mathbf{X}_t^{(l)})\right]\right) \times \prod_{i=1}^{m}\left[p_p\left(\mathbf{Y}_t|O(\mathbf{X}_t^{i(l)})\right) p_s\left(O(\mathbf{X}_t^{i(l)})|x_t^{(l)}, y_t^{(l)}, \mathbf{R}_t^i\right)\right], \quad (8)$$

$$p_s\left(\mathbf{Y}_t|S\left[O(\mathbf{X}_t^{(l)})\right]\right) = exp^{-\lambda_s F_2\left(I\left(S\left[O(\mathbf{X}_t^{i(l)})\right]\right),\mathbf{M}_t\right)},$$

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

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

Fig. 4: **Example of patch initialization** in *diving* seq. (b) displays 50 points that have small K and (c) illustrates 15 initialized local patches.

the image patch inside the bounding box B, which is defined by

$$\begin{aligned} B_w &= \max\left[\{\hat{x}_{t-1}^i\}_{i=1}^m\right] - \min\left[\{\hat{x}_{t-1}^i\}_{i=1}^m\right], \\ B_h &= \max\left[\{\hat{y}_{t-1}^i\}_{i=1}^m\right] - \min\left[\{\hat{y}_{t-1}^i\}_{i=1}^m\right], \\ B_c &= \left(\min\left[\{\hat{x}_{t-1}^i\}_{i=1}^m\right] + \frac{B^w}{2}, \min\left[\{\hat{y}_{t-1}^i\}_{i=1}^m\right] + \frac{B^h}{2}\right), \end{aligned} \quad (9)$$

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; $\hat{x}_{t-1}^i$ and $\hat{y}_{t-1}^i$ indicate the x and y positions of the MAP state of the i-th local patch at time $t-1$, respectively. In (8), our method uses the segmented region to compute the histogram. The function F_2 returns Bhattacharyya similarity [14] between HSV histogram of the segmented region $S[O(\mathbf{X}_t^{(l)})]$ and $\mathbf{M}_t$. Using (8), our method could cover severe scale changes in the target. Notably, the segmentation method 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 due to severe scale changes.

5 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.

5.1 Initialization of Patches

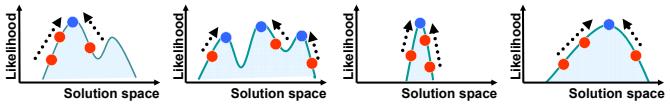
The initial positions of patches have to be chosen carefully for image alignment. Thus, the condition number K of the Hessian Matrix H is used for measuring the goodness of a patch.

$$K = \frac{\sigma_{max}(H)}{\sigma_{min}(H)}, \quad (10)$$

where $\sigma_{max}(H)$ and $\sigma_{min}(H)$ denote the maximal and minimal singular values of H respectively. In (10), the small K means that the matrix is numerically stable [1].

To initialize the patches, a bounding box around the target is drawn manually, and the center of the first

1. 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 [19]. 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

Fig. 5: **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.

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 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 4 shows the patch initialization process.

5.2 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 (8), which is important for the success of tracking. Smoothness and steepness are used to characterize 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 5 shows examples of different LLM types.

To measure these properties quantitatively, a new method of measurement inspired by [6] 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:

$$D_{sm}(i) = \frac{1}{\left\| \frac{1}{N}\sum_{l=1}^{N}\left[O(\mathbf{X}_t^{i(l)}) - \frac{1}{N}\sum_{l'=1}^{N} O(\mathbf{X}_t^{i(l')})\right]^2 \right\|_2}, \tag{11}$$

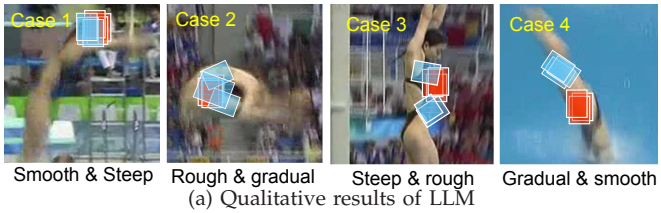
where $O(\cdot)$ finds local modes for the N number of samples of the i-th local patch. The degree of steepness D_{st} of the i-th local patch is measured by the mean distance between the positions of samples and of local modes:

$$D_{st}(i) = \frac{1}{\frac{1}{N}\sum_{l=1}^{N} \|O(\mathbf{X}_t^{i(l)}) - (x_t^{i(l)}, y_t^{i(l)})\|_2}. \tag{12}$$

With the new methods of measurement in (11) and (12), the status of local modes, such as that in Table 1, could be determined. Figures 6(a) and 6(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 is no ambiguous region 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 5.4.

TABLE 1: **The status of local modes.**

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

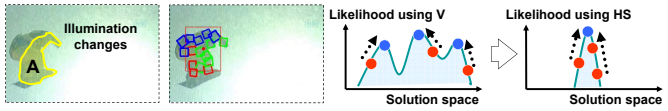


(a) Qualitative results of LLM

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

Fig. 6: **Experimental results of the likelihood landscape analysis** Red squares denote samples of a local patch. Blue squares represent the local modes of these samples.



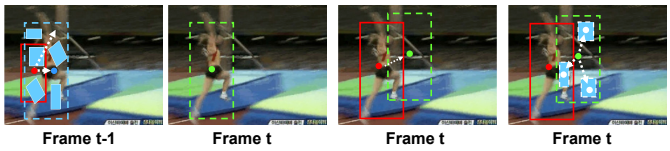
(a) Illumination changes (b) Feature selection results (c) Advantage of the feature selection

Fig. 7: **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 (**HS**) features are better than the value (**V**) feature because they make the likelihood landscape smooth and steep.

5.3 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 7(a). To track the target robustly in this environment, the proposed method uses locally different features to describe the target, as shown in Figure 7(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 LLM is rough and gradual, as illustrated in Figure 7(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 $D_{sm}(i)$



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

Fig. 8: **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 (13). 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 (15). (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 (16).

in (11) and $D_{st}(i)$ in (12) return the degree of smoothness and steepness of the i-th local patch, respectively. In Figure 7, S_{LLM} increases from 7.14 to 61.96 by adaptively selecting the *Hue*, *Saturation*, and *Value* features, compared with the *Value* feature only.

5.4 Online Modification of Patches

According to the results of the likelihood landscape analysis in Sections 5.2 and 5.3, 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(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 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

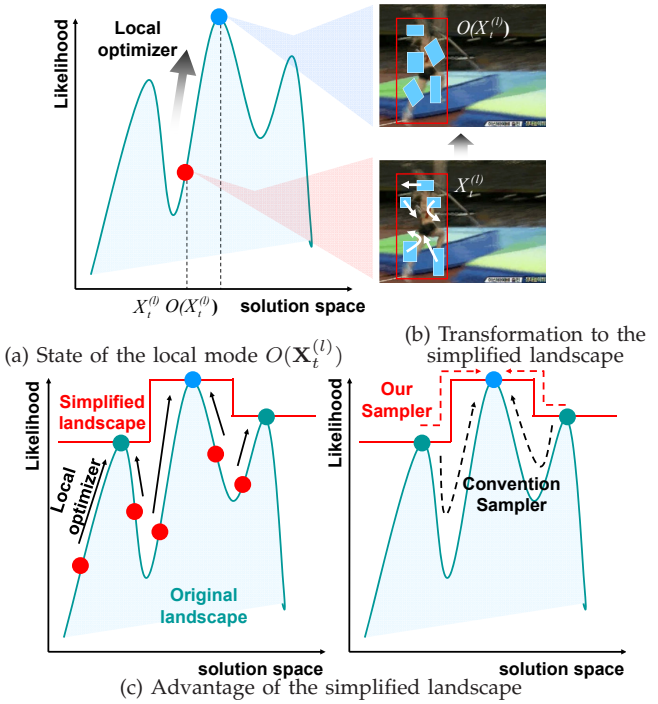


Fig. 9: **Process of the acceptance step** in *high-jump* seq. (a) A state (red circle) is moved to the state of the local mode (blue circle) using a local optimizer. The states of the local modes represent the states of the local patches changed via affine transformation (blue rectangles). (b) After all states (red circles) proposed by Q in (13),(15), and (16) 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.

for the patch using the condition number explained in Section 5.1. If the new patch satisfies the aforementioned criteria within half the number of predefined iterations, the new patch is added.

6 Inference via the ABHMC Sampling

The solution space generally becomes larger as the number of local patches in our appearance model increases. Thus, the conventional MCMC method is inefficient for computing the integration in (1). Therefore the BH sampling method [36] 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 8. 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

$$\begin{aligned} Q_1(x_t^{(1)};\hat{x}_{t-1}) &= G(\hat{x}_{t-1},\sigma_x^2)+\gamma_x\delta_x, \\ Q_1(y_t^{(1)};\hat{y}_{t-1}) &= G(\hat{y}_{t-1},\sigma_y^2)+\gamma_y\delta_y, \\ Q_1(s_t^{(1)}\hat{s}_{t-1}) &= G(\hat{s}_{t-1},\sigma_s^2)+\gamma_s\delta_s, \end{aligned} \tag{13}$$

where $Q_1(x_t^{(1)};\hat{x}_{t-1})$ and $Q_1(y_t^{(1)};\hat{y}_{t-1})$ indicate that the first sample of the object center $(x_t^{(1)},y_t^{(1)})$ at the current frame is proposed based on the MAP state of the object center $(\hat{x}_{t-1},\hat{y}_{t-1})$ at the previous frame. In (13), 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

$$\gamma_x = \begin{cases} \gamma_x+1 & \text{if } \hat{x}_{t-1} \ll \frac{1}{m}\sum_{i=1}^{m}\hat{x}_{t-1}^i \\ \gamma_x-1 & \text{if } \hat{x}_{t-1} \gg \frac{1}{m}\sum_{i=1}^{m}\hat{x}_{t-1}^i \\ \gamma_x & \text{otherwise .} \end{cases},$$

$$\gamma_y = \begin{cases} \gamma_y+1 & \text{if } \hat{y}_{t-1} \ll \frac{1}{m}\sum_{i=1}^{m}\hat{y}_{t-1}^i \\ \gamma_y-1 & \text{if } \hat{y}_{t-1} \gg \frac{1}{m}\sum_{i=1}^{m}\hat{y}_{t-1}^i \\ \gamma_y & \text{otherwise .} \end{cases}, \tag{14}$$

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

where B_w and B_h respectively denote the width and height of the bounding box B defined by (9) and B_0 denotes the initial diagonal size of the bounding box of the target. In (14), 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 $(x_t^{(l+1)},y_t^{(l+1)})$ and scale $s_t^{(l+1)}$, given the previous sample $(x_t^{(l)},y_t^{(l)})$ and $s_t^{(l)}$, respectively, via Gaussian perturbation:

$$\begin{aligned} Q_2(x_t^{(l+1)};x_t^{(l)}) &= G(x_t^{(l)},\sigma_x^2), \\ Q_2(y_t^{(l+1)};y_t^{(l)}) &= G(y_t^{(l)},\sigma_y^2), \\ Q_2(s_t^{(l+1)};s_t^{(l)}) &= G(s_t^{(l)},\sigma_s^2), \end{aligned} \tag{15}$$

where G denotes the Gaussian distribution with mean $x_t^{(l)}$,$y_t^{(l)}$, and $s_t^{(l)}$ and; variance σ_x^2,σ_y^2, and σ_s^2 to propose the new center and scale, respectively.

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

$$Q_3(\mathbf{X}_t^{i(l+1)};\mathbf{R}_t^i) = (x_t^{(l+1)},y_t^{(l+1)})+\mathbf{R}_t^i, \quad for\ i=1,\ldots,m, \tag{16}$$

where $\mathbf{X}_t^{i(l+1)}$ is a new sample of the center position for the i-th local patch, $(x_t^{(l+1)},y_t^{(l+1)})$ is a new sample of the object center obtained by (15), and $\mathbf{R}_t^i$ is the parameter estimated by (7).

• **Acceptance Step:** Most performance in the BH sampling method comes from the novel acceptance step. The primary difference between the conventional acceptance ratio in Section 3.1 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

$$a = min\left[1, \frac{p\left(\mathbf{Y}_t|O(\mathbf{X}_t^{(l+1)})\right)Q(\mathbf{X}_t^{(l)};\mathbf{X}_t^{(l+1)})}{p\left(\mathbf{Y}_t|O(\mathbf{X}_t^{(l)})\right)Q(\mathbf{X}_t^{(l+1)};\mathbf{X}_t^{(l)})}\right], \tag{17}$$

where $Q(\mathbf{X}_t^{(l+1)};\mathbf{X}_t^{(l)})$ represents the proposal density defined in (13)(15)(16), and $p(\mathbf{Y}_t|O(\mathbf{X}_t^{(l)}))$ in (8) 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 as the Lucas-Kanade image registration method [19], as shown in Figure 9(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 9(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 9(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 1 illustrates the whole process of our tracking method, including local patch-based dynamic appearance modeling and adaptive Basin Hopping Monte Carlo sampling.

Algorithm 1 ABHMC-FS

Input: $\mathbf{X}_{t-1}=(x_{t-1},y_{t-1},s_{t-1},\mathbf{X}_{t-1}^1,\cdots,\mathbf{X}_{t-1}^i,\cdots,\mathbf{X}_{t-1}^m)$
Output: $\hat{\mathbf{X}}_t=(\hat{x}_t,\hat{y}_t,\hat{s}_t,\hat{\mathbf{X}}_t^1,\cdots,\hat{\mathbf{X}}_t^i,\cdots,\hat{\mathbf{X}}_t^m)$
Initialize patches using (10) in an initial frame.
Propose $x_t^{(1)}$,$y_t^{(1)}$ and $s_t^{(1)}$ using Q_1 in (13).
for $l=1$ to $N-1$ **do**
  Propose $x_t^{(l+1)}$, $y_t^{(l+1)}$, and $s_t^{(l+1)}$ using Q_2 in (15).
  Determine $\mathbf{X}_t^{i(l+1)}$ for all i using Q_3 in (16).
  Obtain $S(\mathbf{X}_t^{(l+1)})$ with the process described in Section 4.2.
  Calculate the likelihood score using (8).
  Accept $\mathbf{X}_t^{(l+1)}$ with probability (17).
end for
Estimate the MAP state $\hat{\mathbf{X}}_t$ using (2).
Select patches to be modified using (11) and (12).
Choose features of the patches using the method described in Section 5.3.
Modify the patches using the criterion 1 and 2 introduced in Section 5.4.
Obtain updated parameters $\mathbf{M}_{t+1}^i$ and $\mathbf{R}_{t+1}^i$ using (5) and (7).

7 Experimental Results

For the experiments, 17 video sequences[2] were tested, namely, *snowboard*, *diving*, *high-jump*, *transformer*, and *gymnastic* sequences in [9]; *femaleskater*, *maleskater*, *indiandancer*, and *dancer* sequences in [26]; *dinosaur* and *hand2* sequences in [23]; *motocross2* and *skiing* sequences in [24]; *coke*, *girl*, *tiger1*, and *tiger2*, sequences in [3][37][38]. The proposed algorithms (ABHMC,

2. The source code, test datasets, and result videos are available at http://cv.snu.ac.kr/research/abhmcfs/.

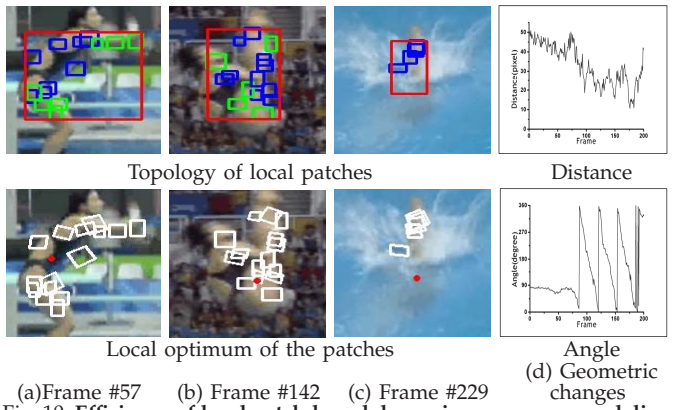
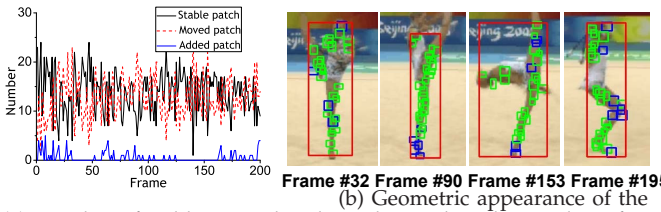


Fig. 10: **Efficiency of local patch-based dynamic appearance modeling** in *diving* seq.



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

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

Fig. 11: **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.

TABLE 2: **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 5.3. 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

ABHMC-F, ABHMC-FS) are compared with 8 different state-of-the-art tracking methods, namely, Mean-Shift tracker (MS) based on [13][39], Standard MCMC (MCMC) based on [31], Incremental learning for Visual Tracking (IVT) in [10], Fragment-based tracker (FRAGT) in [25], Block Histogram-based Tracker (BHT) in [26], Multiple Instance Learning tracker (MIL) in [3], Local Global Tracker (LGT) in [23], and Hough-based Tracker (HT) in [24].

ABHMC denotes our original method in [9]. ABHMC-F denotes the improved version of the ABHMC with adaptive feature selection. ABHMC-FS is the final version, which utilizes rough segmentation results. **All parameters of the proposed method are fixed for all experiments.** λ_p in (4), λ_g in (6), ω in (5)(7), λ_s in (8), θ_{sm}, θ_{st} in Table 1, θ_{C1}, θ_{C1} in Section 5.4, 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 methods were not sensitive to all these parameters.

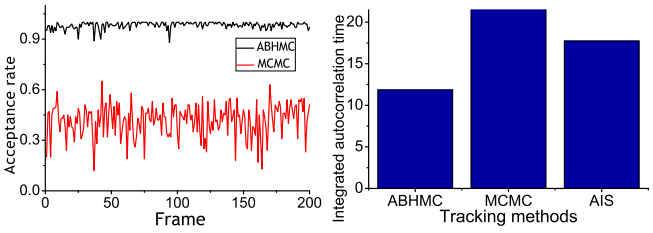
MCMC uses an HSV color histogram for the appearance model as in [14], while dividing an object into the upper and lower body. Proposal variances of the MCMC are set to 8 and 4 for the x and y directions, respectively. The parameters of other trackers are adjusted to produce the best tracking performances. Note we used the codes of the other methods provided by the authors in [3][10][23][24][25][26]. We obtained the ground truths of the publicly available datasets from the corresponding authors, and constructed those of our datasets manually.

7.1 Efficiency of the Proposed Appearance Model

• **Qualitative Performance:** Figure 10 presents the qualitative performance of the dynamic appearance modeling scheme in *diving* seq. This sequence includes severe geometric changes of the target appearance as shown in Figure 10(d), which depicts the angle and distance changes of the head position with respect to the center position of the target object over time. Even under severe geometric changes of the object appearance, our method successfully tracked the object. The first row of Figure 10 illustrates our constructed appearance models where blue squares denote unmodified local patches and green ones denote modified ones. The second row of Figures 10 represents the local modes of the patches as white squares and the estimated center of the object as a red point. Figure 10 shows the robustness of our on-line appearance model. In Figures 10(a), some local patches in the background region are removed from the next frame because their local modes had very rough landscapes and did not satisfy the two modification criteria in Section 5.4. Additionally, the proposed algorithms dealt with the occlusion of the object by deleting the occluded patches adaptively. As shown in Figures 10(c), the algorithm reduced the number of local patches from 15 to 4.

• **Quantitative Performance:** 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 11(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 11(b) shows that the proposed appearance model adaptively modifies the position and number of patches, particularly when geometric appearance of the object is drastically changing. The proposed methods successfully capture 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 2 demonstrates how our dynamic appearance modeling scheme enhances the LLMs of the patches in the appearance model. As aforementioned in Section 5.3, 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



(a) Acceptance rate (b) Integrated autocorrelation time

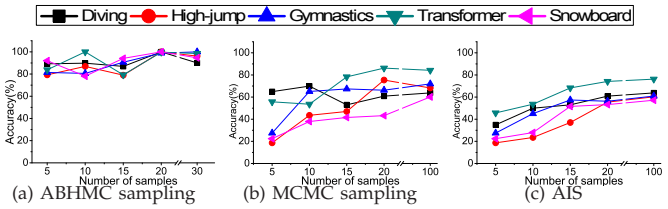
Fig. 12: **Property of the adaptive Basin Hopping Monte Carlo sampling** in *car4* seq. in [10] (a) Acceptance rate is defined by the number of accepted samples over the total number of samples in each frame. (b) To derive τ_{int}, each method used 500 samples. AIS denotes the Annealed Importance Sampling method in [40].

steeper LLMs. To obtain good local patches, our dynamic appearance modeling scheme adopts the online update step in Section 5.4 and the feature selection step in Section 5.3. After the online update and feature selection steps, the LLMs 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 trackers tracked the object robustly despite severe photometric and geometric appearance changes. Instead of color features, other features such as Gabor filters could be utilized in the C step. However, the Gabor filters did not provide good LLMs, rather making them very rough and gradual as demonstrated by the C* step in Table 2. This means that the sampling method using Gabor filters need very 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.

7.2 Efficiency of the Proposed ABHMC Sampling

• **Property of Sampling Strategy:** To evaluate the performance of the ABHMC sampling more analytically and qualitatively, it was compared with the standard MCMC-based tracking algorithm [31]. In this experiment, the test video only contained a rigid object for fair comparison. An equal number of samples, same appearance model, and same transition model for both methods were used. Note that one particularly good property of the ABHMC 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 12(a), our tracking algorithm had higher acceptance rates than the standard MCMC method. This means that the proposed methods easily escape from the local optima and obtains more diverse samples.

Autocorrelation time measures the degree of statistical independence between samples [41]. 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 12(b) illustrates the integrated autocorrelation time τ_{int}, where τ_{int} of the



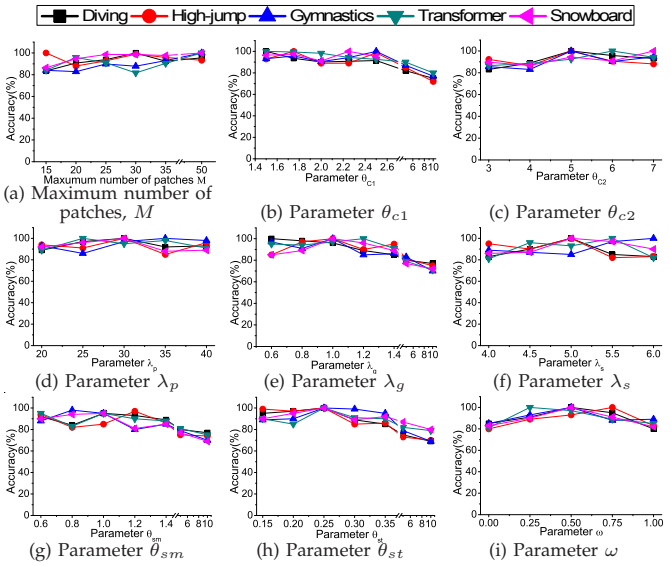
(a) ABHMC sampling (b) MCMC sampling (c) AIS

Fig. 13: **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 [40].

ABHMC sampling was smaller than that of the MCMC sampling and the AIS (Annealed Importance Sampling) [40]. This finding suggests that the ABHMC sampling method produces highly uncorrelated samples, which sufficiently minimizes the statistical error of the MAP estimate in (2).

• **Efficiency of Sampling Strategy:** The appearance model generally consists of 20 to 50 local patches, indicating a very large solution space. The proposed methods, however, use a very small number of samples, 20, in all experiments for tracking an object. This performance typically benefits from the ABHMC sampling. Figure 13 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 several local minima and thus needs only a 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 a good solution because they only employ the stochastic process.

In the theoretical aspect, the simulated annealing method and its variants such as the AIS method suffers from the notorious "freezing" problem, as reported in [42]. 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 problem in simulated annealing, as reported in [43].



(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

Fig. 14: **Experiments on parameter settings of ABHMC-FS.** In the experiments, the tracking accuracy was obtained by $\frac{\text{Pascal score using current parameters}}{\text{Best Pascal score}} * 100$.

TABLE 3: **Quantitative analysis of individual component within the proposed methods.** The numbers indicate tracking accuracy, which are evaluated by the Pascal score [44]. 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**, Section 3.1), **B step**: Local patch-based appearance modeling step after A step (Section 4.1), **C step**: Online updating step after B step (**ABHMC**, Section 5.1, 5.2, and 5.4), **D step**: Feature selecting step after C step (**ABHMC-F**, Section 5.3), **E step**: Segmentation step after D step (**ABHMC-FS**, Section 4.2)

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

7.3 Accuracy of the Proposed Tracking Methods

• **Effect of Parameter Settings:** Tracking accuracy could be dependent on several parameters. Thus the robustness of ABHMC-FS is evaluated to the variation of the parameters. As shown in Figures 14(a)-(i), our method was not severely sensitive to all these parameters, although the tracking accuracy slightly decreased if the parameter values extremely increase. ABHMC-FS made our original ABHMC less sensitive to the parameter settings by employing the global appearance model. Figure 14(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 also increases. Figure 14(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 5.4. 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 14(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 5.4. If θ_{C2} increases, the ability to explore a missed object region is improved. However, the ability of exploiting the fine object region is decreased. Figures 14(d)-(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 the weights of photometric, geometric, and segmentation factors in the likelihood function, respectively. The likelihood models have the different weights by assigning different values to λ_p, λ_g, and λ_s in (3) and (8). Although these weights can be made adaptive to the tracking environment, we set them fixed to simplify our algorithm. This is feasible because the weights does not affect much the tracking performance, as demonstrated in Figures 14(d)-(f). Figure 14(g) and 14(h) illustrate the tracking accuracy as the LLM parameters, θ_{sm} and θ_{st} increase. The parameter θ_{sm} and θ_{st} are the threshold values to determine the smoothness and the steepness of LLM, respectively. If θ_{sm} and θ_{st} increase, local patches are more frequently updated because LLMs of local patches are considered as bad (rough and gradual) ones. Hence, our method can cover drastic appearance changes of the targets. However, the method has difficulty in handling gradual appearance changes of the target. Figure 14(i) describes the tracking accuracy as the appearance updating parameter ω increases. The parameter ω adjusts weights of old and new appearances of the target to construct the appearance model. If ω increases, the appearance model is constructed by reflecting the current appearance more rather than the old one. Hence, the tracking method accurately tracks the target although the appearance frequently changes. However, the constructed appearance model can be easily corrupted because the current appearance may contain erroneous tracking results.

• **Comparison among the proposed methods:** Using multiple features (ABHMC-F) and additional segmentation results (ABHMC-FS), we enhanced the performance of ABHMC, as shown in Table 3. ABHMC-FS always outperformed the ABHMC and ABHMC-F. 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, 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 proposed in [9] is robust because it includes the online updating step and

TABLE 4: **Quantitative comparison with other methods.** The numbers indicate mean and standard deviation of tracking accuracy, which are evaluated by the Pascal score [44]. 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 (0.07)	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 (0.10)	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 (0.10)	**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 (0.11)	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 (0.17)	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** (0.13)	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** (0.11)	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** (0.10)	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 (0.09)	**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 (0.11)	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 (0.15)	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** (0.13)	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 (0.13)	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 (0.12)	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 (0.06)	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 (0.08)	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 (0.10)	0.13 (0.10)	**0.65** (0.10)	0.22 (0.09)	0.35 (0.09)	**0.62** (0.09)

our ABHMC-FS 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 ABHMC. And the appearance model should include both the local and global appearance of the target like ABHMC-FS.

Table 3 also shows the computation time of each individual component in ABHMC-FS. The tracker approximately takes 2 seconds per frame using the Pentium 4 quad core 2.4GHz CPU. It is notable that our ABHMC without the segmentation procedure (before the E step in Table 3) is real time. It approximately takes 0.1 seconds per frame. In spite of such computational overhead, the segmentation procedure in ABHMC-FS is very useful because it greatly improves the tracking accuracy.

• **Comparison with other tracking methods:** Table 4 summarizes the tracking results of nine different test sequences that include objects whose geometric appearances are changing drastically over time. ABHMC-FS 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 were more severe. Compared with these methods, ABHMC-FS produced accurate tracking results even though there were illumination changes and deformation of targets at the same time. With the online-feature selection process, ABHMC-FS constructed a good appearance model using the selected local patches, which was robust to both the illumination 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 algorithms like the MS and MCMC trackers, ABHMC-FS produced better tracking results since the non-rigid object severely changed its color appearance caused by deformation, while the MS and MCMC trackers suffered from drastic color changes. ABHMC-FS 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*. Note that ABHMC-FS showed small standard deviation, which means that the method produced stable tracking results. The main reason of small standard deviation is that ABHMC-FS partly utilizes the deterministic method, which produces zero standard deviation.

In Figures 15(a) to 15(c), videos that include background clutter similar to the object were tested. In the case of other methods, trajectories were 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 15(d) and 15(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 were reversed. Figure 15(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 15(g)-(l). The test video used in Figures 15(m)-(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 15(p)-(r) show the comparison among the proposed methods (ABHMC, ABHMC-

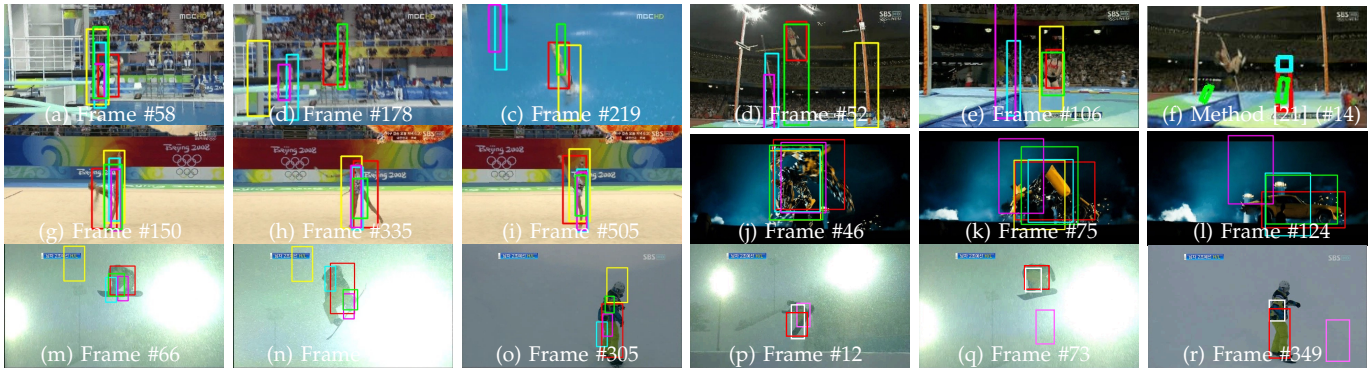


Fig. 15: **Qualitative comparison with other methods using color sequences.** In (a) to (o), the green, magenta, cyan, yellow and red rectangles denote the bounding boxes of MCMC, IVT, FRAGT, BHT, and ABHMC-FS, respectively. In (p) to (r), the pink, white, and red rectangles denote the bounding boxes of ABHMC, ABHMC-F, and ABHMC-FS, respectively.

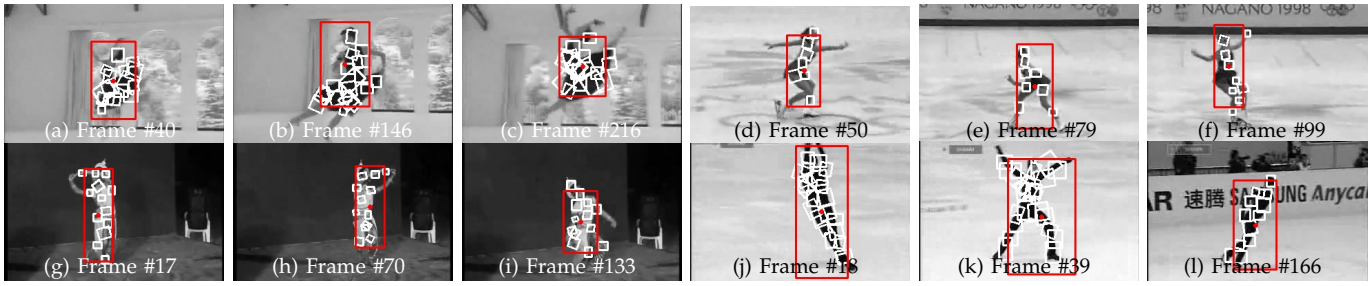


Fig. 16: **Tracking results of the proposed method using gray sequences.** The red rectangles denote the bounding boxes of ABHMC-FS. White squares describe the local modes of patches in our appearance model.

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 16(a)-(l) show the tracking results of the proposed ABHMC-FS when only intensity values of gray sequences are available. In this case, the method neither 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 17 and 18 demonstrates that ABHMC-FS 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.

8 Conclusion

In this paper, we have proposed an novel tracking algorithm evolving a local patch-based appearance model by the analysis of LLM (Landscape of the Local Mode). With the model, the algorithm robustly tracked the object whose geometric appearance is drastically changing over time, while efficiently finding the best state of the object with the BH (Basin-Hopping) sampling. By selecting robust features and using segmentation results, the tracking performance was further enhanced. The experimental results demonstrated that the proposed method outperformed conventional tracking algorithms in severe tracking environments. Future work aims to extend the proposed method to deal with severe occlusions and multi objects.

References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, 2006.
[2] S. Avidan., "Ensemble tracking," *PAMI*, vol. 29, no. 2, pp. 261–271, 2007.
[3] B. Babenko, M. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," *CVPR*, 2009.
[4] R. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *PAMI*, vol. 27, no. 10, pp. 1631–1643, 2005.
[5] H. Grabner and H. Bischof, "On-line boosting and vision," *CVPR*, 2006.
[6] B. Han and L. Davis, "On-line density-based appearance modeling for object tracking," *ICCV*, 2005.
[7] A. D. Jepson, D. J. Fleet, and T. F. E. Maraghi, "Robust online appearance models for visual tracking," *PAMI*, vol. 25, no. 10, pp. 1296–1311, 2003.
[8] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *PAMI*, vol. 34, no. 7, pp. 1409–1422, 2012.
[9] J. Kwon and K. M. Lee, "Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling," *CVPR*, 2009.
[10] D. A. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1-3, pp. 125–141, 2008.
[11] M. Yang and Y. Wu, "Tracking non-stationary appearances and dynamic feature selection," *CVPR*, 2005.
[12] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," *ICCV*, 2011.
[13] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *CVPR*, 2000.
[14] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," *ECCV*, 2002.
[15] C. Bibby and I. Reid, "Robust real-time visual tracking using pixel-wise posteriors," *ECCV*, 2008.
[16] D. Crandall, P. Felzenszwal, and D. Huttenlocher, "Spatial priors for part-based recognition using statistical models," *CVPR*, 2005.
[17] R. Fergus, P. Perona, and A. Zisserman, "A sparse object category model for efficient learning and exhaustive recognition," *CVPR*, 2005.
[18] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool, "Coupled object detection and tracking from static cameras and moving vehicles," *PAMI*, vol. 30, no. 10, pp. 1683–1698, 2008.

Fig. 17: **Tracking results of the proposed ABHMC-FS 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.

Fig. 18: **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.

[19] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *IJCAI*, 1981.
[20] G. Schindler and F. Dellaert, "A rao-blackwellized parts-constellation tracker," *ICCV Workshop*, 2005.
[21] D. Ramanan, D. Forsyth, and A. Zisserman, "Tracking people by learning their appearance," *PAMI*, vol. 29, no. 1, pp. 65–81, 2007.
[22] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," *CVPR*, 2004.
[23] L. Cehovin, M. Kristan, and A. Leonardis, "An adaptive coupled-layer visual model for robust visual tracking," *ICCV*, 2011.
[24] M. Godec, P. M. Roth, , and H. Bischof, "Hough-based tracking of non-rigid objects," *ICCV*, 2011.
[25] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," *CVPR*, 2006.
[26] S. M. S. Nejhum, J. Ho, and M.-H. Yang, "Visual tracking with histograms and articulating blocks," *CVPR*, 2008.
[27] M. Yang, J. Yuan, and Y. Wu, "Spatial selection for attentional visual tracking," *CVPR*, 2007.
[28] P. Chockalingam, N. Pradeep, and S. Birchfield, "Adaptive fragments-based tracking of non-rigid objects using level sets," *ICCV*, 2009.
[29] L. Lu and G. D. Hager, "A nonparametric treatment for location/segmentation based visual tracking," *CVPR*, 2007.
[30] M. Isard and A. Blake, "Icondensation: Unifying low-level and high-level tracking in a stochastic framework," *ECCV*, 1998.
[31] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *PAMI*, vol. 27, no. 11, pp. 1805–1918, 2005.
[32] K. Smith, D. G. Perez, and J. Odobez, "Using particles to track varying numbers of interacting people," *CVPR*, 2005.
[33] W. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.
[34] L. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *PAMI*, vol. 26, no. 6, pp. 810–815, 2004.
[35] T. H. Kim, K. M. Lee, and S. U. Lee, "Generative image segmentation using random walks with restart," *ECCV*, 2008.
[36] 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.
[37] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," *ICCV*, 2011.
[38] A. Saffari, M. Godec, T. Pock, C. Leistner, and H. Bischof, "Online multi-class lpboost," *CVPR*, 2010.
[39] R. Collins, "Mean-shift blob tracking through scale space," *CVPR*, 2003.
[40] R. M. Neal, "Annealed importance sampling," *Technical Report No. 9805, Dept. of Statistics, University of Toronto*, 1998.
[41] B. Berg, "Introduction to markov chain monte carlo simulations and their statistical analysis," *Phys .Stat. Mech*, 2004.
[42] 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.
[43] 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.
[44] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, vol. 88, no. 2, pp. 303–308, 2009.

Junseok Kwon received the BS degree in electrical engineering and the MS degree in electrical engineering and computer science from Seoul National University(SNU), Seoul, Korea, in 2006 and 2008, respectively. He is currently working toward the PhD degree in electrical engineering and computer science at Seoul National University. His research interests include visual tracking, event detection, and surveillance. He is a student member of the IEEE.

Kyoung Mu Lee received the B.S. and M.S. degrees in Control and Instrumentation Engineering from Seoul National University (SNU), Seoul, Korea in 1984 and 1986, respectively, and Ph. D. degree in Electrical Engineering from the USC (University of Southern California), Los Angeles, California in 1993. He has been awarded the Korean Government Overseas Scholarship during his Ph. D. courses. From 1993 to 1994 he was a research associate in the SIPI (Signal and Image Processing Institute) at USC. He was with the Samsung Electronics Co. Ltd. in Korea as a senior researcher from 1994 to 1995. On August 1995, he joined the department of Electronics and Electrical Eng. of the Hong-Ik University, and worked as an assistant and associate professor. From September 2003, he is with the Department of Electrical Engineering and Computer Science at Seoul National University as a professor, and leads the Computer Vision Laboratory. His primary research is focused on statistical methods in computer vision that can be applied to various applications including object recognition, segmentation, tracking and 3D reconstruction. Prof. Lee has received several awards, in particular, the Most Influential Paper over the Decade Award by the IAPR Machine Vision Application in 2009, the ACCV Honorable Mention Award in 2007, the Okawa Foundation Research Grant Award in 2006, and the Outstanding Research Award by the College of Engineering of SNU in 2010. He served as an Editorial Board member of the EURASIP Journal of Applied Signal Processing, and is an associate editor of the Machine Vision Application Journal, the IPSJ Transactions on Computer Vision and Applications, and the Journal of Information Hiding and Multimedia Signal Processing. He has (co)authored more than 100 publications in refereed journals and conferences including PAMI, IJCV, CVPR, ICCV and ECCV.