Tracking of a Non-Rigid Object via Patch-based Dynamic Appearance Modeling and Adaptive Basin Hopping Monte Carlo Sampling

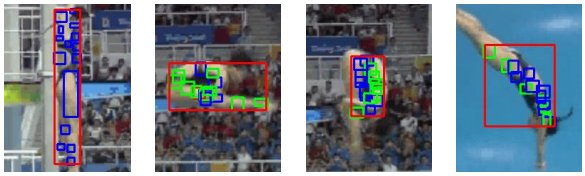
Junseok Kwon and Kyoung Mu Lee
Department of EECS, ASRI, Seoul National University, 151-742, Seoul, Korea
{paradis0, kyoungmu}@snu.ac.kr

Abstract

We propose a novel tracking algorithm for the target of which geometric appearance changes drastically over time. To track it, we present a local patch-based appearance model and provide an efficient scheme to evolve the topology between local patches by on-line update. In the process of on-line update, the robustness of each patch in the model is estimated by a new method of measurement which analyzes the landscape of local mode of the patch. This patch can be moved, deleted or newly added, which gives more flexibility to the model. Additionally, we introduce the Basin Hopping Monte Carlo (BHMC) sampling method to our tracking problem to reduce the computational complexity and deal with the problem of getting trapped in local minima. The BHMC method makes it possible for our appearance model to consist of enough numbers of patches. Since BHMC uses the same local optimizer that is used in the appearance modeling, it can be efficiently integrated into our tracking framework. Experimental results show that our approach tracks the object whose geometric appearance is drastically changing, accurately and robustly.

1. Introduction

Object tracking is well-known in the computer vision community. Recently, it has been addressed in real-world scenarios rather than a lab environment by many researchers [18]. In real-world settings, objects are typically complex and difficult to track. We consider these scenarios especially in movies and sports games which usually contain large amount of *extreme geometric appearance changes* in target objects. To deal with this problem, the tracking algorithms have to adapt for the appearance changes of the target objects by on-line update. Many on-line learning algorithms tackle the *photometric appearance changes* of target objects and successfully track them [4, 6, 17]. There are, however, few studies on geometric appearance changes in target objects. In this paper, we address the problem of tracking non-rigid objects whose geometric appearances are drastically changing as time goes on. Fig.1 shows some tracking examples of such objects by our method.



(a) Frame #1 (b) Frame #117 (c) Frame #125 (d) Frame #212

Figure 1. **Example of tracking results** in *diving* seq. Our tracking algorithm successfully tracks a target even when the geometric appearance of the target is drastically changing. The blue and green squares represent patches of our appearance model.

The philosophy of our method is to make the best of both histogram-based appearance model [2, 12] and pixel-wise one [1, 4]. The histogram-based appearance model covers the geometric variations to some degree but loses spatial information of target objects. On the other hands, the pixel-wise one preserves all of the spatial information but typically fails to capture the extreme geometric changes of target objects. To both cover geometric changes and preserve spatial information of target objects, we propose a local patch-based appearance model as in [3, 9] and present a new strategy for the on-line construction of the appearance model. In our model, the topology between local patches evolves as the appearance of the target changes geometrically. Simultaneously, the position relationship between local patches provides the spatial information of the target, which makes our method more robust.

The first contribution of this paper is to present an on-line updating scheme to evolve a local patch-based appearance model. Our appearance model needs *no* specific model for the target and *no* training phase for learning the appearance or behavior of the target. This appearance model is successfully applied to track non-rigid objects. The second contribution is the proposal of a new method of measurement which measures the robustness of patches by land-

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

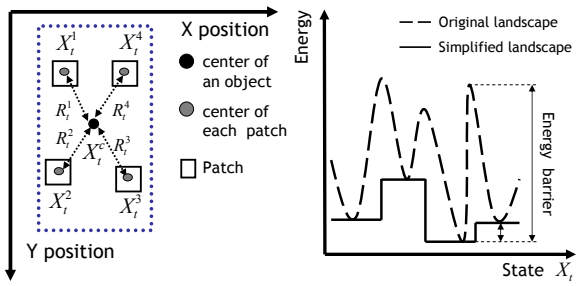
scape analysis. The robustness of a patch is determined by the smoothness and steepness properties of its local mode which is obtained by [10]. The last contribution is that, to the best of our knowledge, we are the first to introduce the Basin Hopping Monte Carlo (BHMC) sampling method [19] to the tracking problem. BHMC simplifies the landscape of a solution space by combining the Monte Carlo method with deterministic local optimizer [10]. In our tracking problem, it gives an efficient way of reaching the global optimum with a small number of samples, even in a huge solution space, owing to the increase of the number of local patches. We extend it to the adaptive BHMC by adding an adaptive proposal density, which further improves the sampling efficiency.

2. Related Work

Tracking methods for non-rigid objects: Schindler et al. [15] represent an object as the constellations of parts to accurately track a bee with the Rao-Blackwellized Particle Filter. This method, however, fixes the topology of the constellation whereas our method evolves it via on-line update. Ramanan et al. [13] propose a tracking method operated by detecting models of the target whose appearances should be built first. This method shows good results in tracking an articulated person. By making shape models of humans, Zhao et al. [20] successfully track humans in crowded environments where occlusion persistently occurs. All of these tracking methods, however, basically assume that specific models of the targets are given. In contrast, our method utilizes *no* prior knowledge of the specific model for the target and *no* off-line training phase.

Tracking methods with online appearance learning: By approximately estimating the pixel-wise color density in a sequential manner, Han et al. [4] successfully track an object where lighting conditions, pose, scale, and view-point are changing over time. Ross et al. [14] present an adaptive tracking method which utilizes the incremental principal component analysis and shows robustness to large changes in pose, scale, and illumination. These two methods, however, do not consider extreme geometric changes of an object. Our method explicitly tackles these changes with a local patch-based on-line appearance model.

Sampling based tracking methods: In tracking problems, the particle filter [5] has shown efficiency in handling non-gaussianity and multi-modality. The Markov Chain Monte Carlo (MCMC) method is well applied to the multi-object tracking problems by reducing computational costs [8, 16]. As the dimension of a solution space increases, however, these methods still suffer from the problem of getting trapped in deep local minima and handling a vast number of samples. Our method based on BHMC sampling solves these problems by transforming the landscape of the solution space into a simple one.



(a) Example of X_t (b) Landscape of X_t

Figure 2. **Example of basin-hopping landscape transformation.** (a) shows an example of state X_t. As shown in (b), BHMC simplifies the landscape, which consists of local minima only.

3. BHMC Based Tracking Method

3.1. Bayesian Object Tracking Formulation

In our tracking method, an object is represented by a local patch-based dynamic graph model as shown in Fig. 2(a). Then, the object state $\mathbf{X}_t$ at time t is defined by $\mathbf{X}_t = (\mathbf{X}_t^c, \mathbf{X}_t^1, \cdots, \mathbf{X}_t^m, \mathbf{R}_t^1, \cdots, \mathbf{R}_t^m)$ where $\mathbf{X}_t^c$ denotes the center position of an object, $\mathbf{X}_t^i$ indicates the center position of the ith local patch, $\mathbf{R}_t^i$ represents the relative position between $\mathbf{X}_t^c$ and $\mathbf{X}_t^i$, and m is the total number of local patches. 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 which 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 model and the transition model, we obtain the Maximum a Posteriori (MAP) estimate over the N number of samples at each time t.

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

where $\mathbf{X}_t^{(l)} = (\mathbf{X}_t^{c(l)}, \mathbf{X}_t^{1(l)}, \cdots, \mathbf{X}_t^{m(l)}, \mathbf{R}_t^1, \cdots, \mathbf{R}_t^m)$ represents the lth sample of the object state $\mathbf{X}_t$ and $\hat{\mathbf{X}}_t = (\hat{\mathbf{X}}_t^c, \hat{\mathbf{X}}_t^1, \cdots, \hat{\mathbf{X}}_t^m, \mathbf{R}_t^1, \cdots, \mathbf{R}_t^m)$ denotes the best configuration which can explain the current state with the given observation. Note that $\mathbf{R}_t^1, \cdots, \mathbf{R}_t^m$ is fixed in the sampling process.

3.2. Adaptive Basin Hopping Monte Carlo

Since the solution space generally becomes large as the number of local patches m in the state $\mathbf{X}_t$ increases, the

conventional MCMC method is not an efficient way to compute the integration in (1). Therefore we introduce the BHMC method [19] to our tracking problem which provides better performance in those high dimensional solution spaces. The BHMC method typically transforms the rough landscape of the original solution space into a simpler one by using robust local optimization techniques in the sampling process, as depicted in Fig. 2(b). In a new transformed energy landscape, energy barriers are lowered and energy maxima of an original function are no longer of concern in the sampling process. This means that we have more chance of reaching the global optimum with a smaller number of samples. Actually, in all of our experiments, *20* samples are sufficient to obtain the MAP estimate.

The BHMC method consists of two main steps similar to the conventional MCMC method; the proposal step and the acceptance step. In the proposal step, we use two different proposal densities; q_1 and q_2. The proposal density q_1 is used in each frame whereas q_2 is used once at the start of each frame to connect the current frame to the previous one. In q_1, we only propose the new sample of the object center $\mathbf{X}_t^{c(l+1)}$ given the lth sample $\mathbf{X}_t^{c(l)}$ via a Gaussian perturbation.

$$q_1(\mathbf{X}_t^{c(l+1)}; \mathbf{X}_t^{c(l)}) = G(\mathbf{X}_t^{c(l)}, \sigma^2), \tag{3}$$

where G denotes the Gaussian distribution with mean $\mathbf{X}_t^{c(l)}$ and variance σ^2. The new center position of the ith local patch $\mathbf{X}_t^{i(l+1)}$ is automatically determined by the following rule [1].

$$\begin{aligned} \mathbf{X}_t^{1(l+1)} &= \mathbf{X}_t^{c(l+1)} + \mathbf{R}_t^1, \\ &\vdots \\ \mathbf{X}_t^{m(l+1)} &= \mathbf{X}_t^{c(l+1)} + \mathbf{R}_t^m. \end{aligned} \tag{4}$$

In q_2, we assume that the center of an object has to be near the centroid of the local patches. With this assumption, we propose the *Adaptive BHMC* (A-BHMC) whose proposal density is adaptively changing according to the relative positions between the object center and each local patch. Therefore our adaptive proposal is

$$q_2(\mathbf{X}_t^{c(1)}; \hat{\mathbf{X}}_{t-1}^c) = G(\hat{\mathbf{X}}_{t-1}^c, \sigma) + p\delta, \tag{5}$$

where $q_2(\mathbf{X}_t^{c(1)}; \hat{\mathbf{X}}_{t-1}^c)$ means that the first sample $\mathbf{X}_t^{c(1)}$ at the current frame is proposed based on the MAP estimate $\hat{\mathbf{X}}_{t-1}^c$ of the previous frame. In (5), δ denotes the adapting constant that is set to 0.3 and, p represents the adapting parameter that is defined by

$$p = \begin{cases} p+1 & \text{if } \hat{\mathbf{X}}_{t-1}^c \ll \frac{1}{m}\sum_{i=1}^m \hat{\mathbf{X}}_{t-1}^i \\ p-1 & \text{if } \hat{\mathbf{X}}_{t-1}^c \gg \frac{1}{m}\sum_{i=1}^m \hat{\mathbf{X}}_{t-1}^i \\ p & \text{otherwise .} \end{cases} \tag{6}$$

[1]We contain X_t^i and R_t^i in the global state simply for representing the likelihood in (8) as X_t^i and R_t^i components.

The adapting parameter initially has the value zero and ranges from -5 to 5. Two vectors $\hat{\mathbf{X}}_{t-1}^c$ and $\hat{\mathbf{X}}_{t-1}^i$ in (6) are compared in component-wise manner for each x and y position.

Most performance in A-BHMC comes from the new acceptance step. In this step, the acceptance ratio is calculated by the likelihood ratio at the *local mode (optimum)* of each patch. The local mode is determined by the mode-seeking method f_o such as the Lucas-Kanade image registration method [10]. Then, our acceptance ratio is defined by

$$\begin{aligned} a = min[1, &\frac{p(\mathbf{Y}_t|f_o(\mathbf{X}_t^{(l+1)}))p(f_o(\mathbf{X}_t^{(l+1)})|\mathbf{X}_t^{(l+1)})}{p(\mathbf{Y}_t|f_o(\mathbf{X}_t^{(l)}))p(f_o(\mathbf{X}_t^{(l)})|\mathbf{X}_t^{(l)})} \\ &\times \frac{q(\mathbf{X}_t^{(l)}; \mathbf{X}_t^{(l+1)})}{q(\mathbf{X}_t^{(l+1)}; \mathbf{X}_t^{(l)})}], \end{aligned} \tag{7}$$

where $p(\mathbf{Y}_t|f_o(\mathbf{X}_t^{(l)}))$ denotes the product of likelihood at the local mode of each patch, and $q(\mathbf{X}_t^{(l+1)}; \mathbf{X}_t^{(l)})$ represents the proposal density in (3). The detailed procedure estimating the likelihood is described in the next section.

4. On-Line Appearance Model Construction

4.1. Appearance Model

Our appearance model adopts the philosophy of representing objects as an assembly of parts [3, 9]. We assume that each local patch is only dependent on the center of an object, which assumption is similar to those of the star model [3] and the implicit shape model [9]. With this assumption, the likelihood in (7) is represented by

$$\begin{aligned} &p(\mathbf{Y}_t|f_o(\mathbf{X}_t^{(l)}))p(f_o(\mathbf{X}_t^{(l)})|\mathbf{X}_t^{(l)}) \\ &\approx \prod_{i=1}^m p_a(\mathbf{Y}_t|f_o(\mathbf{X}_t^{i(l)}))p_s(f_o(\mathbf{X}_t^{i(l)})|\mathbf{X}_t^{c(l)}, \mathbf{R}_t^i), \end{aligned} \tag{8}$$

where p_a denotes the photometric likelihood and p_s indicates the geometric likelihood.

Actually, $f_o(\mathbf{X}_t^{i(l)})$ returns the image registration result of the ith local patch centered on $\mathbf{X}_t^{i(l)}$. In the image registration process, the ith local patch is warped to the patch at local mode, which is the best match to ith template image T_t^i at time t. Then, the photometric likelihood is defined by

$$p_a(\mathbf{Y}_t|f_o(\mathbf{X}_t^{i(l)})) = exp^{-\lambda_a NSSD(f_o(\mathbf{X}_t^{i(l)}), T_t^i)}, \tag{9}$$

where $NSSD$ function returns the normalized sum of squared differences between the patch at local mode and its template image, and λ_a denotes the weighting parameter that is set to 30. The geometric likelihood is calculated by

$$p_s(f_o(\mathbf{X}_t^{i(l)})|\mathbf{X}_t^{c(l)}, \mathbf{R}_t^i)) = exp^{-\lambda_s DIST(f_o(\mathbf{X}_t^{i(l)}) - \mathbf{X}_t^{c(l)}, \mathbf{R}_t^i)}, \tag{10}$$

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

Figure 3. **Example of patch initialization** in *diving* seq. (b) displays 50 points which have small K and (c) illustrates fifteen initialized local patches.

where $DIST$ function returns the difference between the relative position of $f_o(\mathbf{X}_t^{i(l)})$ with respect to $\mathbf{X}_t^{c(l)}$ and $\mathbf{R}_t^i$, and λ_s denotes the weighting parameter that is set to 1.

4.2. On-line Updating

Our appearance model evolves photometric and geometric appearance of an object via on-line update. In the process of on-line update, the local patches of our appearance model can be newly added, deleted or moved to a different position. On-line update occurs once at the end of each frame.

4.2.1 Initializing the Patches

The initial position of patches has to be chosen so as to be good for the image alignment. For this, we utilize the condition number K of the Hessian Matrix H.

$$K = \frac{\sigma_{max}(H)}{\sigma_{min}(H)}, \tag{11}$$

where $\sigma_{max}(H)$ and $\sigma_{min}(H)$ denote maximal and minimal singular values of H respectively. In (11), small K means that the matrix is numerically stable .

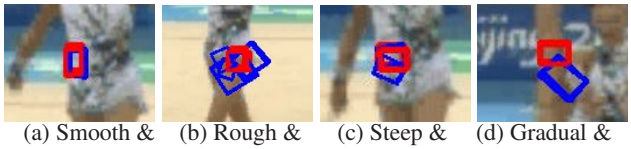
To initialize the patches, we manually draw a bounding box around the target and choose the center of the first local patch within the bounding box as the point that has the least K value. The size of the patch is randomly determined. Then, the second one is chosen as the point that has the next least K value so as not to overlap with existing local patches. The procedure is repeated and terminated when there is no space to make local patches or the number of local patches reaches a predefined value. Fig. 3 shows the process of patch initialization.

4.2.2 Examining the Patches by Landscape Analysis

When the *landscape of local mode (LLM)* of each patch has good properties, our appearance model reliably estimates the likelihood in (8), which is important for the success of tracking. As the measure of good LLM, we use smoothness and steepness. Smooth LLM represents that local modes are gathered in a narrow region of a solution space while steep LLM means that these local modes have a steep shape. Both smooth and steep LLM guarantee that there is a very strong local mode for the patch.

	status
$M_{sm}^{-1} \leq \theta_{sm}$	The landscape of local modes is smooth.
$M_{sm}^{-1} > \theta_{sm}$	The landscape of local modes is rough.
$M_{st}^{-1} \leq \theta_{st}$	The shape of local modes is steep.
$M_{st}^{-1} > \theta_{st}$	The shape of local modes is gradual.

Table 1. **The status of local modes.** In all experiments, we set θ_{sm} as 1.0 and θ_{st} as 4.0.



(a) Smooth & steep (b) Rough & gradual (c) Steep & rough (d) Gradual & smooth

Figure 4. **Example of local modes for a patch** in *gymnastics* seq. Red squares denote samples of a local patch. Blue ones represent local modes of these samples.

	Case (a)	Case (b)	Case (c)	Case (d)
M_{sm}^{-1}	0.030000	10.115001	1.502500	0.000000
M_{st}^{-1}	0.582418	4.431388	0.370085	7.647641

Table 2. **Quantitative analysis** of Fig. 4. Case (a): There are strong local modes because they are both smooth and steep. Case (b): The landscape is very rough. Case (c): The shape of local modes is steep but the landscape is rough. Case (d): Conversely, the landscape is very smooth but the shape of local modes is gradual.

To measure these properties quantitatively, we design a new method of measurement inspired by [4]. The degree of smoothness is approximately estimated by

$$M_{sm}(i)^{-1} = Var(f_o(\mathbf{X}_t^{i(l)})|_{l=1,\ldots,N}), \tag{12}$$

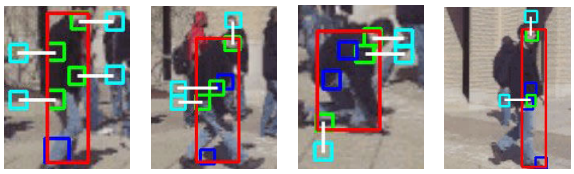
where f_o finds local modes for the N number of samples of the ith local patch and Var returns the variance on the positions of these local modes. The degree of steepness is measured as the mean of the distance between the positions of samples and of local modes:

$$M_{st}(i)^{-1} = Mean(DIST(f_o(\mathbf{X}_t^{i(l)}), \mathbf{X}_t^{i(l)})|_{l=1,\ldots,N}). \tag{13}$$

In Table 1, the status of local modes is summarized. Fig. 4 and Table 2 describe the experimental results of measuring the degree of smoothness and steepness.

4.2.3 Modifying the Patches

According to the landscape analysis, we can identify the bad patches as those with rough landscapes or gradual local



(a) Frame #2 (b) Frame #9 (c) Frame #19 (d) Frame #81

Figure 5. **Example of background patches** in *pedestrian* seq. Blue squares denote unmodified local patches whereas green denote modified ones. The red rectangle represents the bounding box. To construct a background patch (sky-blue), we firstly choose the nearest bounding line to each modified local patch. Then, we select a center position of the background patch outside this bounding line in the perpendicular direction of the line so that the patch is ten pixels away from the bounding box. The size of the background patch is equal to that of the modified local patch.

modes. These bad patches are modified on-line. In the modifying process, we provide two criteria for the modification such that

> **Criterion 1:** A modified local patch has to be similar to the foreground and not to the background.
>
> **Criterion 2:** A modified local patch should not be in the high density regions of other local patches.

The first criterion typically prevents local patches from drifting away from an object and into a background. On the other hand, the second criterion generally makes local patches escape from the center of an object. The first criterion is formulated by $\frac{likelihood\ over\ foreground}{likelihood\ over\ background} \geq \theta_{C1}$. The foreground model is constructed by the average of color histograms of unmodified local patches and the background model is made by the color histogram of *one* background local patch for *each* modified local patch. Fig. 5 displays the construction of background patches. The merit of background patches is that they preserve the spatial information of the background for each modified local patch. The foreground and background model utilize the Bhattacharyya coefficient as a similarity measure [12]. The second criterion checks that no patches exist within the radius θ_{C2} of the modified local patch.

When the above two criteria are satisfied, modifications are performed by adding new patches, deleting or moving bad patches. First, our algorithm tries to move a bad patch via the Gaussian perturbation centered on the current position. The size of a moved patch is equal to that of the original one. If, until the predefined number of iterations, the algorithm cannot find the patch which satisfies the above criteria, it deletes the patch. The adding process occurs only when the algorithm finds the patch within the half number of predefined iterations. In this case, it chooses a new position of the patch utilizing the condition number explained in section 4.2.1.

Algorithm 1 A-BHMC tracker with local patch-based dynamic appearance modeling

Input: $\mathbf{X}_{t-1} = (\mathbf{X}_{t-1}^c, \mathbf{X}_{t-1}^1, \cdots, \mathbf{X}_{t-1}^m, \mathbf{R}_{t-1}^1, \cdots, \mathbf{R}_{t-1}^m)$
Output: $\hat{\mathbf{X}}_t = (\hat{\mathbf{X}}_t^c, \hat{\mathbf{X}}_t^1, \cdots, \hat{\mathbf{X}}_t^m, \mathbf{R}_t^1, \cdots, \mathbf{R}_t^m)$

Transition phase
  Propose $\mathbf{X}_t^{c(1)}$ using (5).
End
Sampling phase
  for $l = 1$ to $N - 1$ **do**
    1.Propose $\mathbf{X}_t^{c(l+1)}$ using (3).
    2.Determine $\mathbf{X}_t^{i(l+1)}$ using (4).
    3.Calculate the likelihood score using (8).
    4.Accept $\mathbf{X}_t^{(l+1)}$ with probability (7).
  end for
  Estimate the MAP state $\hat{\mathbf{X}}_t$ using (2).
End
Updating phase
  1.Initialize patches using (11) in an initial frame.
  2.Select patches to be modified using (12) and (13).
  3.Modify the patches using the criterion 1 and 2.
  4.Update the appearance model using (14) and (15).
End

4.2.4 Updating the Appearance Model

In the case of the photometric appearance, the template T_t^i in (9) is naively updated by

$$T_{t+1}^i = 0.5T^{i(ref)} + 0.5T_t^{i(dyn)}, \tag{14}$$

where $T^{i(ref)}$ indicates the reference template in an initial frame and $T_t^{i(dyn)}$ represents the template image obtained in the region of $f_o(\hat{\mathbf{X}}_t^i)$ at time t. Various methods for the template update are discussed in [11]. In the case of the geometric appearance, our method updates R_t^i in (10), which is the relative position of the ith local patch with respect to the center of an object.

$$R_{t+1}^i = 0.5R_t^i + 0.5DIST(f_o(\hat{\mathbf{X}}_t^i), \hat{\mathbf{X}}_t^c), \tag{15}$$

where $\hat{\mathbf{X}}_t^i$ and $\hat{\mathbf{X}}_t^c$ are the MAP estimates of $\mathbf{X}_t^i$ and $\mathbf{X}_t^c$, repectively. Note that this updating process is for unmodified local patches. For modified local patches, T_{t+1}^i and R_{t+1}^i are already determined in the modification process explained in section 4.2.3. Algorithm 1 illustrates the whole process of our tracking method.

5. Experimental Results

We compared the proposed algorithm with three different tracking methods: on-line appearance learning method of Ross [14], standard MCMC method based on [8], which uses an HSV color histogram as the appearance model in

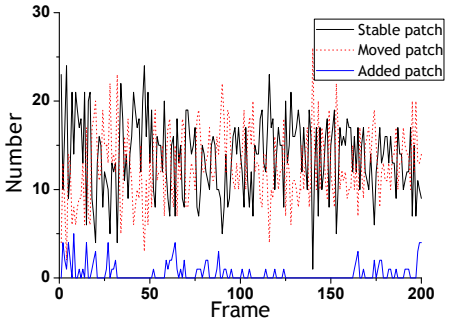
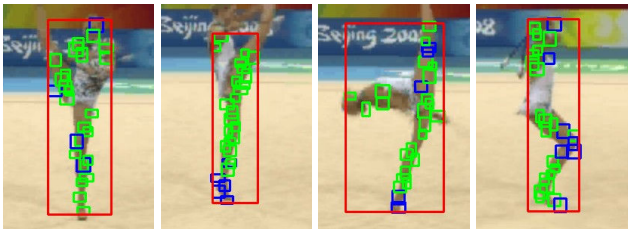


Figure 6. **Number of stable, moved and newly added patches** in *gymnastics* seq.



(a) Frame #32 (b) Frame #90 (c) Frame #153 (d) Frame #195

Figure 7. **Geometric appearance of the target when the number of moved patches peak** in Figure 6. Among 27 local patches, 23 patches are moved at (a), 22 at (b), 22 at (c) and, 20 at (d), where green squares denote moved patches and blue squares denote stable ones.

[12] while dividing an object into upper and lower body, and the Mean Shift method based on the implemented function in OpenCV. Note that we used the software of Ross for the method [14].[2]

5.1. Quantitative Results

We tested our method by evaluating the performance of local patch-based appearance model and A-BHMC sampling method, individually.

The performance of the appearance model: To evaluate the performance of our appearance model, we first checked the number of modified and unmodified local patches in each frame. As illustrated in Fig. 6, our appearance model actively moves, deletes or adds patches based on the landscape analysis at each frame. This means that the topology between local patches in the model evolves as time goes on. Fig. 7 shows that our appearance model adaptively modifies the position and number of patches, particularly when geometric appearance of the target is drastically changing. Our method successfully captures the movements of head, legs and arms without the specific model of the target.

The target is considered as correctly tracked in that frame

[2]The videos of tracking results and original datasets are available at $http://cv.snu.ac.kr/paradiso$.

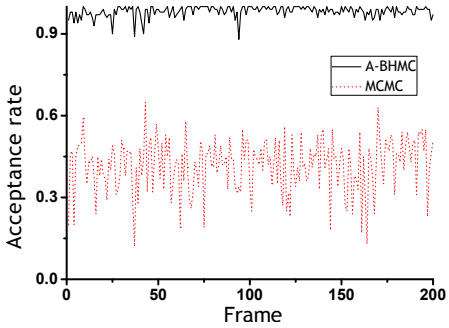


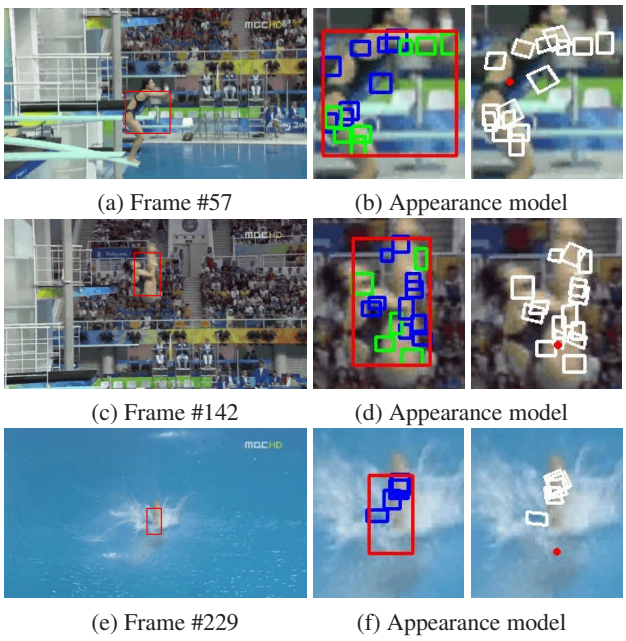
Figure 8. **Acceptance rate** of *car4* seq in [14]. The acceptance rate is defined by the number of accepted samples over the total number of samples in each frame.

Method \ Seq.	*Pedestrian*	*Diving*	*High-jump*
Our method	succeed	succeed	succeed
Method [14]	succeed	fail/#56	fail/#13
MCMC(1000)	fail/#53	fail/#43	fail/#20
MCMC(3000)	fail/#56	fail/#50	fail/#20
MCMC(5000)	fail/#62	fail/#61	fail/#22
Mean Shift	fail/#3	fail/#123	fail/#13

Table 3. **Comparison of tracking results.** The index indicates that the tracking algorithm fails to track a target from that index of a frame. We utilized 1000,3000 and 5000 samples for evaluating the MCMC method.

if the root mean square error of an object center is smaller than 30. For this experiment, we manually draw the center of the target object as ground truth and tested different tracking methods. Table 3 summarizes the tracking results of three different test sequences that include objects whose geometric appearances are changing drastically over time. In all of the test sequences, our method successfully tracked the targets. Other methods, however, missed the targets at the frames where the targets changed their geometric appearances. Note that the LLM analysis in our appearance model is a critical factor for good tracking. Without LLM analysis, our method cannot measure badness of patches and cannot modify those patches. In the experiment, tracker without LLM analysis failed to track the objects because bad patches survive and drift away from them.

The performance of A-BHMC: The appearance model generally consists of 5 to 30 local patches, which indicates that the solution space is very large. Our tracking method, however, used a very small number of samples, 20 in all experiments for tracking an object. This performance typically benefits from A-BHMC. To evaluate the performance of A-BHMC in our tracking algorithm more analytically and qualitatively, we compared it with the standard MCMC-based tracking algorithm [8]. In this experiment, the test video only contained a rigid object for fair comparison. We utilized an equal number of samples, equal appear-



(a) Frame #57 (b) Appearance model

(c) Frame #142 (d) Appearance model

(e) Frame #229 (f) Appearance model

Figure 9. **Tracking results** of *diving* seq.

ance model and equal transition model for testing. One of the good properties of A-BHMC is that it can easily jump over the energy barriers by transforming an energy landscape into a simpler one. Therefore, the A-BHMC method frequently accepts proposed samples since high energy barriers are lowered. As shown in Fig. 8, our tracking algorithm had higher acceptance rates than the standard MCMC method. This means that our method easily escapes from local minima and obtains more diverse samples.

5.2. Qualitative Results

Fig. 9 presents the tracking results in *diving* seq. Under the severe geometric changes of a target appearance, our method successfully tracked the target. The left parts of Fig. 9(b)(d)(f) illustrate our constructed appearance models where blue squares denote unmodified local patches and green ones denote modified ones. The right parts of Fig. 9(b)(d)(f) represent the local modes of the patches as white squares and the estimated center of an object as a red point. Fig. 9(b)(f) show the robustness of our on-line appearance model. In Fig. 9(b), our method removed some local patches in the background region in the next frame since local modes of those patches had very rough landscapes and they did not satisfy two modification criteria in the section 4.2.3. Additionally, our algorithm dealt with the occlusion of a target by adaptively deleting patches which are occluded. As described in Fig. 9(f), the algorithm reduced the number of local patches from 15 to 4.

In Fig. 10(a)-(d), we tested the video that includes background clutter and pedestrians whose appearance is similar to that of a target. In the case of the conventional MCMC method, a trajectory was hijacked by other pedestrians wearing similar colors of clothing to the target when the target changes its geometric appearance. On the other hand, our tracking method robustly tracked the target in spite of background clutter and geometric appearance changes. Fig. 10(e)-(l) demonstrate how the proposed method outperformed conventional tracking algorithms in drastic geometric appearance changes. The conventional tracking algorithms failed to track the target when the positions of head and legs are reversed. The result of the method [13] shows that the specific model of the target occasionally cannot capture the drastic geometric changes of the target. The test video used in Fig. 10(e)-(l) includes the scale change of a target. In this environment, for tracking the target which became smaller over time, our method adaptively shortened the range between the center of a target and each local patch, and successfully tracked it. Our method also tracked the target which transforms its appearance from a robot to a car by evolving an appearance model as shown in Fig. 10(m)-(p).

6. Conclusion

In this paper, we have proposed an effective tracking algorithm evolving a local patch-based appearance model by the analysis of landscape of local modes. With A-BHMC sampling, the algorithm efficiently addresses tracking of a target whose geometric appearance is drastically changing over time. Experimental results demonstrated that the proposed method outperformed conventional tracking algorithms in severe tracking environments. Our future work is to extend our method to deal with severe occlusions and multi objects.

Acknowledgement

This research was supported in part by the IT R&D program of MKE/IITA (2008-F-030-01), and in part by the ITRC program of MKE/IITA through 3DRC (IITA-2008-C1090-0801-0018), Korea

References

[1] C. Bibby and I. Reid. Robust real-time visual tracking using pixel-wise posteriors. *ECCV*, 2008.

[2] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. *CVPR*, 2000.

[3] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. *CVPR*, 2005.

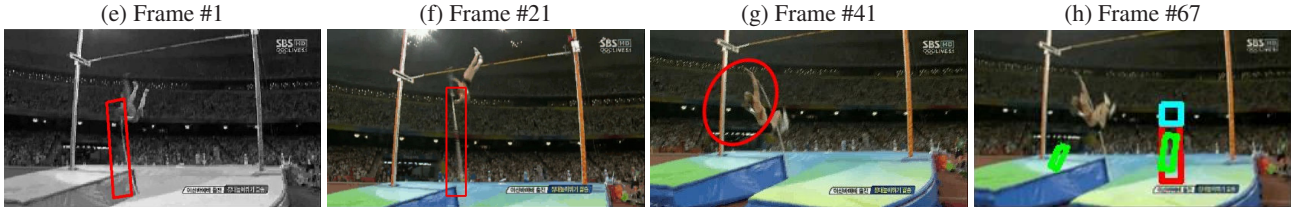
[4] B. Han and L. Davis. On-line density-based appearance modeling for object tracking. *ICCV*, 2005.

[5] M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. *ECCV*, 1998.

(a) Frame #1 (b) Frame #29 (c) Frame #53 (d) Frame #85

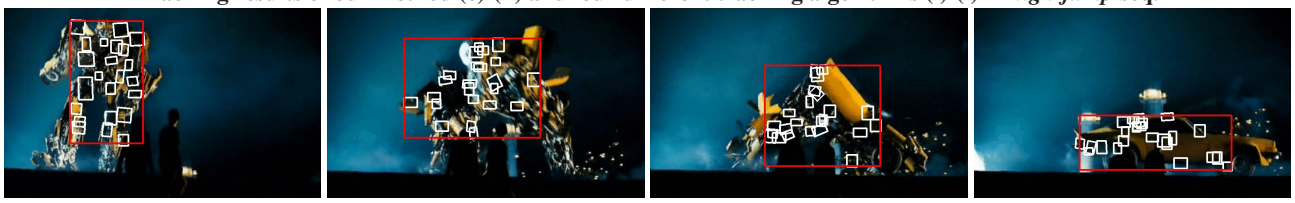
Tracking results of our method in *pedestrian* seq in [7]

(e) Frame #1 (f) Frame #21 (g) Frame #41 (h) Frame #67



(i) Method [14] (#20) (j) Standard MCMC (#31) (k) Mean shift (#13) (l) Method [13] (#14)

Tracking results of our method (e)-(h) and four different tracking algorithms (i)-(l) in *high-jump* seq.



(m) Frame #6 (n) Frame #54 (o) Frame #77 (p) Frame #123

Tracking results of our method in *transformer* seq.

Figure 10. **Tracking results in other test sequences.** White squares denote the local modes of patches in our appearance model and red ones indicate the bounding box of a target object.

[6] A. Jepson, D. Fleet, and T. E. Maraghi. Robust online appearance models for visual tracking. *PAMI*, 25(10):1296–1311, 2003.

[7] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. *ICCV*, 2007.

[8] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *PAMI*, 27(11):1805–1918, 2005.

[9] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *PAMI*, 30(10):1683–1698, 2008.

[10] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI*, 1981.

[11] L. Matthews, T. Ishikawa, and S. Baker. The template update problem. *PAMI*, 26(6):810–815, 2004.

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

[13] D. Ramanan, D. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *PAMI*, 29(1):65–81, 2007.

[14] D. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008.

[15] G. Schindler and F. Dellaert. A rao-blackwellized parts-constellation tracker. *ICCV Workshop*, 2005.

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

[17] M. Yang and Y. Wu. Tracking non-stationary appearances and dynamic feature selection. *CVPR*, 2005.

[18] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006.

[19] 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*, 120(12):5536–5542, 2004.

[20] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. *CVPR*, 2004.