Non-Rigid Object Tracking with Elastic Structure of Local Patches and Hierarchical Sampling

Kwang Moo Yi, Soo Wan Kim, Hawook Jeong, Songhwai Oh, Jin Young Choi

Department of Electrical Engineering and Computer Science, Seoul National University, ASRI, PIRC
Email: kmyi@snu.ac.kr,swkim@neuro.snu.ac.kr,{hwjeong,songhwai,jychoi}@snu.ac.kr

Abstract

To solve the problem of real-time object tracking under partial occlusions and non-rigid deformations, we propose a tracking method based on sequential Bayesian inference. The proposed method is mainly consisted of two parts: (1) modeling the target object using elastic structure of local patches for robust performance; and (2) efficient hierarchical sampling method to obtain an acceptable solution in real-time. The elastic structure of local patches allows the proposed method to handle partial occlusions and non-rigid deformations through the relationship among neighboring patches. The proposed hierarchical sampling method generate samples from the region where the posterior is concentrated to reduce computation time. The method is extensively tested on a number of challenging image sequences with occlusion and non-rigid deformation, demonstrating its real-time capability and robustness under different situations.

Keywords: Visual Tracking, Local Patches, Markov Random Field, Particle Filter, Sampling

1 Introduction

Object tracking is an important problem in the field of computer vision. Tracking is used in many applications, such as, robot vision, visual surveillance, video analysis, home automation, and behavior recognition. These applications require accurate tracking results in order for the whole system to work correctly. Conventionally available tracking algorithms, such as the kernel-based tracking algorithm [1], particle filtering [2], and subspace based trackers [3], have proved to be successful for these applications. However, the performance of these algorithms are somewhat limited to "lab environments" and do not show satisfying results when tested on real-world scenarios.

The reason for such limitation of conventional methods is that they have strong assumptions about the input video sequence, such as, constant movement, and consistent views. Also, in real-world scenarios, lots of occlusion can occur with non-rigid deformations, limiting the applicability of conventional methods even more. For instance, when a group of people are walking together, a common case, people tend to occlude one another with non-rigid movements, making it extremely difficult for conventional methods to successfully track the target person throughout the image sequence.

Kernel-based tracking [1] is a widely used method, because of its computational efficiency and robust performance. This method is very efficient and suitable for real-time purposes. However, since the method uses local optimization, it is easy for the tracker to fall into local maxima (or minima). Also, the method gets easily distracted by the background and has no clear way of adapting to scale and illumination changes. Particle filtering based tracking methods have become popular in recent years, since its first introduction in [2]. These methods try to solve the tracking problem using Monte Carlo (MC) simulation. Unlike kernel-based trackers, particle filtering considers not only translational movements but also affine motions as well. Many variations have been proposed with different measurement models, e.g., [3], [4], and [5]. Among these, subspace-based measurements [3], inspired by the work of Black et.al. [6], have proven to be successful. The subspace-based methods are able to adapt to various changes such as change in views and illumination and changes within the model. However, these methods assume slow changes of the target object and do not consider occlusions during learning, causing the drift problem.

Recently, methods considering the tracking problem as a Maximum A Posterior - Markov Random Field (MAP-MRF) problem [7], and modeling the target object as a collection of local patches [8] have drawn attention. Under this consideration, the target is modeled using MRF, and the tracking problem becomes finding the MAP solution of this MRF. In [7], tracking is performed through feature matching, as a problem of finding the transforma-

978-1-4244-9630-3/10/$26.00 ©2010 IEEE

tion which maximizes the posterior energy in the MRF formulation. Also, the rigidity and adaptiveness of the model is automatically determined from the tracking results, which enables the tracker to track various types of movements. Since this algorithm is based on features, it shows robust results to background clutter and occlusion problems. However, for relatively small objects or cases with noises from the motion of a camera, feature points are unreliable and its tracking performance may degrade. Also, the method only considers the relationship between two subsequent frames without exploiting the dynamic properties of a target. In [8], a star model of local patches is used as a MRF configuration. The target object is described with local patches, and Adaptive Basin Hopping Monte Carlo (A-BHMC) sampling is used to minimize the posterior energy. The patches describing the model are constantly updated through a heuristic scheme which enables the tracker to be able to adapt to drastic changes in the appearance and shape of the target. The A-BHMC reduces the number of particles required for tracking and keeps the computation time tractable. However, the heuristic update scheme is not mathematically well defined and does not have a complete problem formulation, which can cause tracking failures for situations which are not taken into account beforehand (e.g. partial occlusions are not considered in their method). Also, the star model of this method does not preserve the local dependency of each patch, making each patch only dependent on the global position of the target object. This makes the algorithm weak against occlusions since the occluded patches will not be tracked and removed from the update scheme. Moreover, even with A-BHMC, this algorithm still requires a large amount of computation as demonstrated in Section 6, which makes it difficult to run in real-time.

Our method is targeted to solve the problem of partial occlusion and non-rigid deformation, in real-time. The proposed algorithm models the target object through elastic structure of local patches with spring-type connections, formulated under the MAP-MRF framework. Each local patch is considered to be connected with its neighbors, and therefore, the local structures of the target object is embedded into the MRF structure. Since the patches that are not occluded will enforce the positions of the neighboring occluded patches through the relationship between patches, our method becomes robust to partial occlusions. Non-rigid deformations are also well described since the tracking result is described as a collection of local motions of each patch. This is a distinctive feature of our method against other methods using MRF. Also, the method we proposed does not match the detected features in subsequent frames [7], or employ a heuristic update scheme using feature detection for patches [8]. Rather, the proposed method embeds the MRF structure inside the sequential Bayesian inference. The proposed method therefore considers not only subsequent frames but also the whole image observed before the current observation. In addition, since we use lots of local patches, the search space is very large, and a hierarchical sampling approach is proposed to achieve real-time computation under such condition.

To demonstrate the effectiveness of our method, we have experimented with a number of challenging image sequences. The experimental results show that our method is robust against partial occlusions and non-rigid deformations, compared with other methods. Especially, our method runs in real-time, whereas the methods proposed in [3] and [8] runs only a few frames per second.

2 Sequential Bayesian Inference Framework

For our work, we treat the problem of object tracking as a sequential Bayesian inference problem. Let $\mathbf{X}_t = \left(\mathbf{X}_t^1, \mathbf{X}_t^2, \cdots, \mathbf{X}_t^m\right)$ denote the object state at time t, where $\mathbf{X}_t^k$ denotes the state of the k^{th} local patch of the object (in our method, just the position of the patch), and m is the total number of local patches. Then, if we denote the observations (input images) up to time t as $\mathbf{Y}_{1:t}$, the problem of object tracking can be defined as the following:

$$\hat{\mathbf{X}}_t = \underset{\mathbf{X}_t}{\arg\max} P(\mathbf{X}_t|\mathbf{Y}_{1:t}). \tag{1}$$

For sequential Bayesian inference, the posterior probability $P(\mathbf{X}_t|\mathbf{Y}_{1:t})$ can be sequentially updated as the following:

$$P(\mathbf{X}_t|\mathbf{Y}_{1:t}) \propto \\ P(\mathbf{Y}_t|\mathbf{X}_t) \times \int P(\mathbf{X}_t|\mathbf{X}_{t-1}) P(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1}). \tag{2}$$

Here, $P(\mathbf{Y}_t|\mathbf{X}_t)$ is the likelihood between the current state $\mathbf{X}_t$ and the current observation $\mathbf{Y}_t$, $P(\mathbf{X}_t|\mathbf{X}_{t-1})$ is the transition probability from $\mathbf{X}_{t-1}$ to $\mathbf{X}_t$. Typically, for object tracking, since we consider many types of movements (e.g. translation, rotation, scale, and affine motions), searching the whole solution space is intractable. Especially, in our formulation, the dimension of the solution space increases as the number of local patches increases. Therefore as in [2], particle filtering (also known as sequential Monte Carlo sampling) is used to solve the problem.

Our method differs from the traditional particle filtering methods in the fact that the likelihood

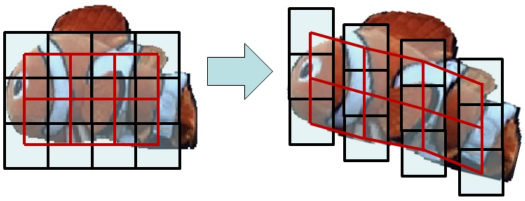


Figure 1: Example of elastic structure of local patches used to describe the target object.

$P(\mathbf{Y}_t|\mathbf{X}_t)$ is obtained through an MRF-style manner. Therefore, object tracking is performed to maximize both the individual likelihood of each patches and the relationship among them. This is similar to the method proposed in [8], but they used a star model whereas our method uses an MRF model which holds the local structures of each patch. Therefore, the structured local patches, which will be explained in 3, has an advantage that the resultant posterior distribution considers the local structures. To allow the proposed method to obtain meaningful solution within the real-time constraint, unlike conventional particle filter based algorithms, our method does not use a simple random walk model. Instead, we adopt a hierarchical sampling scheme, which focuses on sampling from the region of the solution space where the posterior is concentrated. The details of the proposed method are described in the following sections.

3 Elastic Structure of Local Patches

The target object is described by $n \times n$ size local patches and the number of patches are automatically computed according to the size of the target object, as in Fig. 1 (denoted by black boxes). The patch size n is a pre-defined parameter. The local patches are connected in an spring-type elastic structure to describe the target object. Since each local patch is connected with its neighbors forming an MRF, the local structure of the target object can be preserved. Therefore, if some of the patches are occluded, other un-occluded patches will direct the occluded patches to the correct position, causing our model to be robust against partial occlusions. Also, since we describe the target object using local patches, we are able to well describe non-rigid deformations.

We use the posterior probability of this structured local patches model as the likelihood $P(\mathbf{Y}_t|\mathbf{X}_t)$ in (2). Therefore, $P(\mathbf{Y}_t|\mathbf{X}_t)$ is defined as

$$P(\mathbf{Y}_t|\mathbf{X}_t) \propto \prod_{k=1}^{m} P(\mathbf{Y}_t|\mathbf{X}_t^k) \prod_{j \in N_k} P(\mathbf{X}_t^k|\mathbf{X}_t^j), \quad (3)$$

where, $P(\mathbf{Y}_t|\mathbf{X}_t^k)$ is the likelihood of a single patch, $P(\mathbf{X}_t^k|\mathbf{X}_t^j)$ is the prior probability describing the relationship among neighboring patches, and N_k denotes the neighbors of the k^{th} patch. In the energy form, if we denote the total energy of the configuration as $E(\mathbf{Y}_t;\mathbf{X}_t)$, the energy of a single patch as $E(\mathbf{Y}_t;\mathbf{X}_t^k)$, energy from the relationship between patches as $E(\mathbf{X}_t^k,\mathbf{X}_t^j)$, we can write the total energy of the MRF model (which is simply the sum of the observation and neighborhood energy of all patches) as

$$E(\mathbf{Y}_t;\mathbf{X}_t) \equiv Z + \sum_{k=1}^{m} \left[E(\mathbf{Y}_t;\mathbf{X}_t^k) + \sum_{j \in N_k} E(\mathbf{X}_t^k,\mathbf{X}_t^j) \right], \quad (4)$$

where Z is a normalizing constant. Here, the relationship between (3) and (4) is that $Probability \propto exp(-Energy)$, assuming the Gibbs distribution. Now we can easily define the probabilities in terms of energy.

The local patches are described by 12-D features. The first nine dimensions are HOG (Histogram of Oriented Gradients) features. Nine bins denote eight directions of the image gradients and a bin for denoting no gradient. When counting the edges, only edges with intensity change larger than 5 are considered to increase robustness. The remaining three dimensions are the average RGB values of the local patch. This feature is similar to the one used in [9], but one feature is assigned to a single patch not a single pixel as in [9]. The $E(\mathbf{Y}_t;\mathbf{X}_t^k)$ in (4) is defined as the distance between the model of the patch and the observation in this feature space. This 12 dimension feature can be obtained through integral images, greatly reducing the computation time. The distance in the feature space is obtained using a truncated L-2 norm with different weights for different feature dimensions. The reason why we use a truncated L-2 norm is to refrain one patch from having too large energy dominating all others. In other words, it is used to reduce the effect of outliers. For the k^{th} local patch, if we denote the 9-D HOG of the observation as $f_{HOG}^{o,k}$, 9-D HOG of the model as $f_{HOG}^{m,k}$, 3-D RGB features of the observation as $f_{RGB}^{o,k}$, and 3-D RGB features of the model as $f_{HOG}^{m,k}$, then $E(\mathbf{Y}_t;\mathbf{X}_t^k)$ can be defined as

$$E(\mathbf{Y}_t;\mathbf{X}_t^k) = \sqrt{\mu \left\langle f_{HOG}^{o,k}, f_{HOG}^{m,k} \right\rangle_{HOG} + \nu \left\langle f_{RGB}^{o,k}, f_{RGB}^{m,k} \right\rangle_{RGB}}, \quad (5)$$

where $< a,b >_{HOG}$ and $< a,b >_{RGB}$ are truncated L-2 norms. The μ and ν here are parameters which control the influence of HOG and RGB.

The relationship between neighboring patches are modeled to behave as if the neighboring patches are connected by springs. As in Fig. 1 (left), nearest-neighbors are connected (denoted by red lines). The energy between connected local patches $E(\mathbf{X}_t^k, \mathbf{X}_t^j)$ in (4) is defined in the form of potential energy of a spring. Mathematically, if we denote the neighborhood strength by β, the energy is defined as

$$E(\mathbf{X}_t^k, \mathbf{X}_t^j) = \beta \frac{\|l_c(j,k) - l_m(j,k)\|_2^2}{\|l_m(j,k)\|_2^2}, \qquad (6)$$

where $l_c(j,k)$ and $l_m(j,k)$ denotes the distances between the j^{th} and k^{th} patches of the current (sample) configuration and the model, respectively. Here, in order to deal with global scale changes, relative change of the distance between local patches. Using this relative change, same energy can be given for the same changes at different scales. The model update scheme is just an weighted average with weight 0.01 to the new input and 0.99 to the current model as follows

4 Hierarchical Sampling

When sampling using particle filtering to solve the tracking problem, a random-walk model is typically used at the diffusion step, i.e., estimating $P(\mathbf{X}_t|\mathbf{X}_{t-1})$. However, in our model, the dimension of the solution space is too large to simply apply random-walk in all state dimensions. For example, if we consider the case when only one patch is used for tracking, and if this case requires 100 particles to track the movement, we would need 100^m particles if we use m patches. This surely requires intractable amount of calculation, which makes the method impossible to run in real-time. Therefore, simple diffusion using random-walk is not feasible.

To use a small number of samples, we sample from the region where the actual solution exists with high probability. Under the assumption that the deformation of the target object is not large between subsequent frames, which holds in most situations, we diffuse particles hierarchically in two steps: globally for the motion and locally for the deformation. We diffuse all local patches equally according to the Gaussian distribution with a relatively large variance, then diffuse each patch separately according to the Gaussian distribution with a relatively small variance. This is illustrated in Fig. 2. In the global step, the samples are diffused so that the relative positions of local patches in a sample are preserved. Then, in the local step, each local patches are diffused independently. Mathematically the proposed hierarchical sampling method can be described as

$$\mathbf{X}_t^{k,(l)} = \mathbf{X}_{t-1}^{k,(l)} + \Delta_{(l)} + \delta_{k,(l)}, \qquad (7)$$

where, $\mathbf{X}_t^{k,(l)}$ denotes the state of the k^{th} local patch of the l^{th} sample at time t, $\Delta_{(l)}$ denotes the 3-dimension global diffusion (translation in x, y direction, and scale change for the whole object) for the l^{th} sample, and $\delta_{k,(l)}$ denotes the 2-dimension local diffusion (translation in x, y direction for a local patch) for the k^{th} local patch of the l^{th} sample. Here, $\Delta_{(l)} \sim N(0, \sigma_G^2)$ and $\delta_{k,(l)} \sim N(0, \sigma_L^2)$, where $N(0, \sigma^2)$ denotes a Gaussian distribution with 0 mean and standard deviation of σ. σ_G and σ_L are parameters for the diffusion. The proposed sampling produces an acceptable solution with a relatively less number of particles than the simple random walk approach.

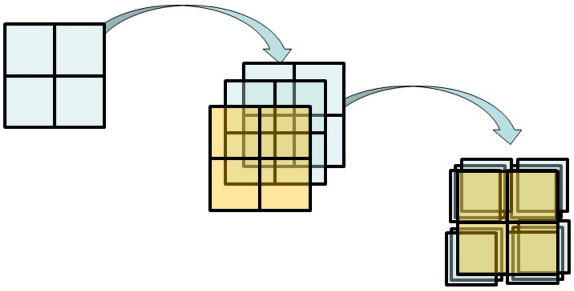
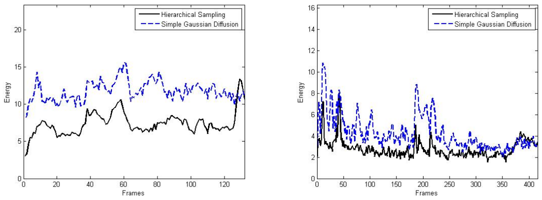


Figure 2: Sampling to increase meaningful samples. Global movement of the total configuration of patches is sampled first, then local movement of individual patches are sampled.



(a) **caviar**

(b) **pedestrian**

Figure 3: Minimum energy obtained for each frame of the (a) **caviar** sequence and the (b) **pedestrian** sequence (1000 samples are used). Solid line denotes the minimum energy obtained using hierarchical sampling and dashed line denotes the minimum energy obtained using simple Gaussian diffusion for each local patch. For simple Gaussian diffusion, variance of 5 was used for both x and y directions.

The effectiveness of the proposed hierarchical sampling method is demonstrated in Fig. 3. As more samples are "focussed" where the solution is likely to exist, $E(\mathbf{Y}_t; \mathbf{X}_t)$ of the MAP solution obtained by particle filtering is better than simple random-walk (Gaussian diffusion for each local patch). In Fig. 3 we can see some points where the minimum energy of the proposed sampling method is larger than simple Gaussian diffusion (around frame 130 in (a), and around 50 in (b)). These cases are when the tracking result gets stuck in some region due to tracking failures, and the object model learns some wrong object. When we do not use hierarchical sampling, the number of samples needed for successful tracking becomes too large to be run in real time.

5 Summary of the Proposed Method

The proposed method uses particle filtering to get a MAP solution for the object tracking problem. For the proposed method, the initial model of the target needs to be given manually. The update of the model is performed as an weighted mean between the current observation and the model. Given the initial model $\left\{f_{HOG}^{m,k}, f_{RGB}^{m,k}, l_m(j,k); \forall k\right\}$, the proposed method can be summarized as **Algorithm 1**. As described in the preceding sections, the evaluation through the structured local patches model gives robust results, and the hierarchical sampling allows the proposed method to run in real-time.

6 Experiments

Evaluation of the proposed method was performed through several image sequences. Each image sequence consists of different types of situations (occlusion, outer-plane motion, non-rigid deformation, etc.) To let particle filtering focus more on samples with lower energies, concentration parameter α for the weights of particle filter was set to be 2. The truncation was done at $\sqrt{8 \times 0.5^2}$ for HOG and $\sqrt{3 \times 0.26^2}$ for RGB (RGB values were scale to range between 0 and 1), the update weight was set to 0.01, and the number of particles was set to 1000. The parameter β determining the strength of the neighborhood was set to 0.4, and except for the **robot** sequence $\sigma_G = (5.0, 5.0, 0.03)$ and $\sigma_L = (2.0, 2.0)$. The parameters μ and ν which control the influence of HOG and RGB were set around 1 and 8, respectively, according to the quality of the image sequence. The implementation was done in C++. All experiments were held on a 3.0GHz desktop and ran comfortably about 20-60 frames per second depending on the number of patches. In all result images, the thick red box denotes the overall tracking result and the thin blue boxes denote the tracking result for the local patches.

Fig. 4 is the result of the proposed method for the **robot** sequence. Since the sequence contains abrupt horizontal motion, σ_G in Section 4 were set to $\sigma_G = (8.0, 5.0, 0.03)$. As shown in the figure, even with large occlusion in (e) and (g), or non-rigid deformation in (d) and (h) (the robot is bent and lay down), the proposed method was able to track the target object correctly. For comparison, we have tested with the Mean-Shift tracker [1] and the methods proposed in [3] and [8]. As in Fig. 4 (i), the Mean-Shift tracker is not able to describe complex movements and the tracking results are not accurate compared to the proposed method. In Fig. 4 (j), the method of [3] cannot describe the non-rigid deformation and tracks only a part of the object, where our method successfully tracks the object describing the non-rigid deformation as a group of motions of local patches. Eventually, after learning the wrong part of the object, their method fails to track object. As in Fig. 4 (k), the method proposed in [8] fails to track objects under occlusions, where our method successfully tracks the object. Also, our method ran over 60 frames per second, whereas the method of [3] ran 1-2 frames per second with 600 particles and the method of [8] ran only about 1 frame per second (using the implementation authors uploaded in [11]), which both are not in real-time.

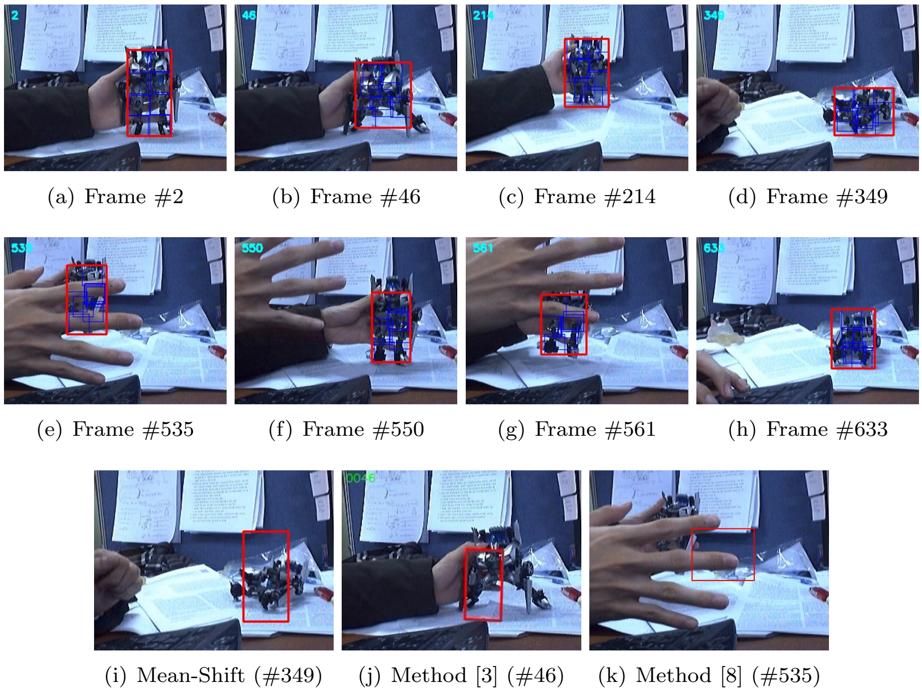
Fig. 5 is the tracking results for the **caviar** sequence. This sequence is available at [10]. As in (a)-(d), our method successfully tracks the object throughout the sequence. The Mean-Shift algorithm (e)-(h) fails from heavy occlusion and tracks a wrong person. In (i)-(l), since the method of [3] does not consider occlusions, as soon as occlusions occur, the tracking result shrinks to a small box and fails to give any meaningful result. The method of [8] (m)-(p) does not lose track of the object. However, as occlusion occurs, only the parts that are not occluded are followed. Also, their method tends to avoid tracking lower body parts, which moves consistently.

Fig. 6 is the tracking results for the **highjump** sequence provided in [11]. This scene consists of lots of non-rigid deformation as the person almost performs a back-flip. For this sequence, the Mean-Shift tracker and the method of [3] fails to track the target person. As in the figure, the tracking results are similar. However, the method of [8] runs only about one frame per second according to the authors C++ implementation on their web site [11]. This is certainly not real-time, whereas our algorithm runs comfortably over 60 frames per second, assuring real-time performance.

Fig. 7 is the tracking results for the **pedestrian** sequence. The scene contains lots of motion blurs and the scene itself is also blurry. Even in these situations, the proposed method successfully tracks the target object very accurately. However, the method of [8] gives some inaccurate results, due to the blurriness of the scene.

Algorithm 1 Tracking with Local Patches (for each frame)

```
For each sample, compute
    $E(\mathbf{Y}_t;\mathbf{X}_t) \equiv Z + \sum_{k=1}^{m}\left[E(\mathbf{Y}_t;\mathbf{X}_t^k) + \sum_{j\in N_k} E(\mathbf{X}_t^k,\mathbf{X}_t^j)\right]$ (4)
    where,
    $E(\mathbf{Y}_t;\mathbf{X}_t^k) = \sqrt{\mu\left\langle f_{HOG}^{o,k}, f_{HOG}^{m,k}\right\rangle_{HOG} + \nu\left\langle f_{RGB}^{o,k}, f_{RGB}^{m,k}\right\rangle_{RGB}}$ (5)
    $E(\mathbf{X}_t^k,\mathbf{X}_t^j) = \beta\frac{\|l_c(j,k)-l_m(j,k)\|_2^2}{\|l_m(j,k)\|_2^2}$ (6)
Assign sample weights $w$ according to the likelihood
    $w \propto P(\mathbf{Y}_t|\mathbf{X}_t) \propto \exp(-\alpha \times E(\mathbf{Y}_t;\mathbf{X}_t))$
Find MAP solution
    $\hat{\mathbf{X}}_t = \arg\max_{\mathbf{X}_t} P(\mathbf{X}_t|\mathbf{Y}_{1:t})$ (1), (2)
Update object model
Re-sample according to weights
Diffuse samples
    $\mathbf{X}_{t+1}^{k,(l)} = \mathbf{X}_t^{k,(l)} + \Delta_{(l)} + \delta_{k,(l)}$ (7)
```

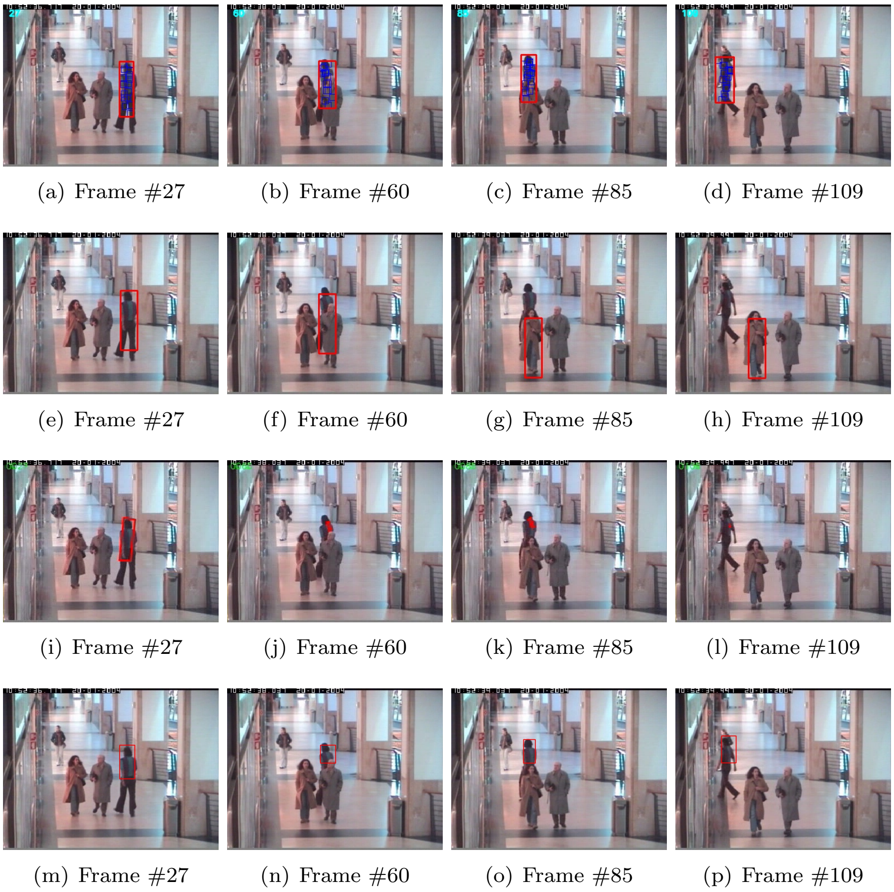


(a) Frame #2 (b) Frame #46 (c) Frame #214 (d) Frame #349

(e) Frame #535 (f) Frame #550 (g) Frame #561 (h) Frame #633

(i) Mean-Shift (#349) (j) Method [3] (#46) (k) Method [8] (#535)

Figure 4: Tracking results for the **robot** sequence. (a)-(h) The proposed method, (i) Mean-Shift [1], (j) the method of [3], and (k) the method of [8]



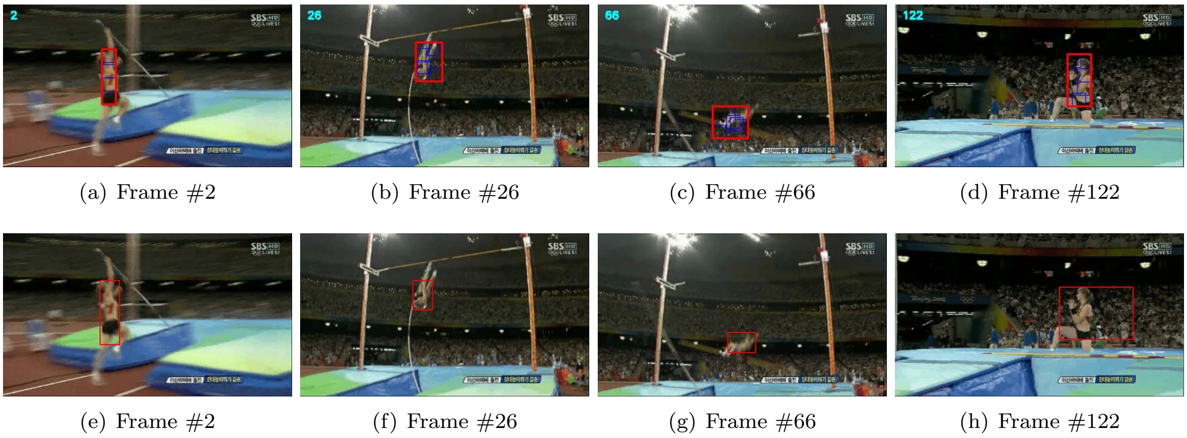
(a) Frame #27 (b) Frame #60 (c) Frame #85 (d) Frame #109

(e) Frame #27 (f) Frame #60 (g) Frame #85 (h) Frame #109

(i) Frame #27 (j) Frame #60 (k) Frame #85 (l) Frame #109

(m) Frame #27 (n) Frame #60 (o) Frame #85 (p) Frame #109

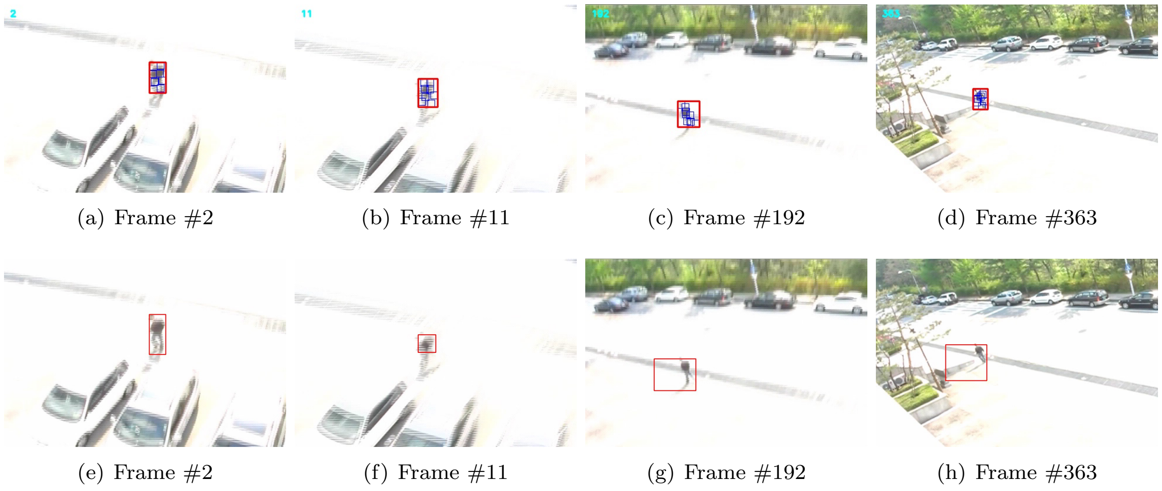
Figure 5: Tracking results for the **caviar** sequence. (a)-(d) The proposed method, (e)-(h) Mean-Shift, (i)-(l) the method of [3], and (m)-(p) the method of [8]



(a) Frame #2 (b) Frame #26 (c) Frame #66 (d) Frame #122

(e) Frame #2 (f) Frame #26 (g) Frame #66 (h) Frame #122

Figure 6: Tracking results for the **highjump** sequence. (a)-(d) The proposed method and (e)-(h) the method of [8].



(a) Frame #2 (b) Frame #11 (c) Frame #192 (d) Frame #363

(e) Frame #2 (f) Frame #11 (g) Frame #192 (h) Frame #363

Figure 7: Tracking results for the **pedestrian** sequence. (a)-(d) The proposed method and (e)-(h) the method of [8].

7 Conclusion

A new tracking method based on sequential Bayesian inference is proposed. The proposed method tackles the occlusion and non-rigid deformation problems of object tracking by modeling the target object with an elastic structure of local patches, and by performing hierarchical sampling in the solution space. By modeling the target object with an elastic structure of local patches, the proposed method was able to track objects with partial occlusions and non-rigid deformations. Also, through hierarchical sampling, an acceptable solution was obtainable in real-time. The method was evaluated through several image sequences with large occlusions and non-rigid deformations. The experimental results show the robustness of the proposed method against occlusions and non-rigid deformations compared to existing methods.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking”, IEEE Trans. on PAMI, vol. 25, pp 564–575, 2003.

[2] M. Isard and A. Blake, “CONDENSATION - Conditional density propagation for visual tracking”, Int’l J. on Computer vision, 29, 1, pp 5–281, 1998.

[3] J. Lim, D. Ross, R. S. Lin, and M. H. Yang, “Incremental learning for visual tracking”, in Advances in Neural Information Processing Systems, 2004.

[4] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, Color-based probabilistic tracking, in Proc. European Conf. on Computer Vision, vol. 1, pp 661–675, 2002.

[5] J. Kwon, K. M. Lee, F. C. Park, Visual Tracking via Geometric Particle Filtering on the Affine Group with Optimal Importance Functions, in Proc. IEEE Conf. on CVPR, 2009.

[6] M. J. Black , A. D. Jepson , EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation, Int’l J. on Computer vision, vol. 26, pp 63–84, 1998.

[7] M. Yang, Y. Wu, Granularity and Elasticity Adaptation in Visual Tracking, in Proc. IEEE Conf. on CVPR, 2008.

[8] J. Kwon, K. M. Lee, Tracking of Non-Rigid Object via Patch-based Dynamic Appearance Modeling and Adaptive Basin Hopping Monte Carlo Sampling, in Proc. IEEE Conf. on CVPR, 2009.

[9] S. Avidan, Ensemble Tracking, IEEE Trans. on PAMI, vol. 29, pp 261–271, 2007.

[10] Caviar project page, “Caviar Test Case Scenarios”, http://homepages.inf.ed.ac.uk/rbf/ CAVIARDATA1/, visited on 9/15/2010

[11] Tracking of a Non-Rigid Object via Patch-based Dynamic Appearance Modeling and Adaptive Basin Hopping Monte Carlo Sampling, “Home Page”, http://cv.snu.ac.kr/research/ bhmctracker/ index.html, visited on 9/15/2010