Adaptive Mean-Shift Tracking with Auxiliary Particles

Junqiu Wang, Yasushi Yagi, *Member, IEEE,*

***Abstract*— We present a new approach for robust and efficient tracking by incorporating the efficiency of the mean-shift algorithm with the multi-hypothesis characteristics of particle filtering in an adaptive manner. The aim of the proposed algorithm is to cope with problems brought about by sudden motions and distractions. The mean-shift tracking algorithm is robust and effective when the representation of a target is sufficiently discriminative, the target does not jump beyond the bandwidth, and no serious distractions exist. We propose a novel two-stage motion estimation method that is efficient and reliable. If a sudden motion is detected by the motion estimator, some particle filtering-based trackers can be used to outperform the mean-shift algorithm at the expense of using a large particle set. In our approach, the mean-shift algorithm is used as long as it provides reasonable performance. Auxiliary particles are introduced to cope with distractions and sudden motions when such threats are detected. Moreover, discriminative features are selected according to the separation of the foreground and background distributions when threats do not exist. This strategy is important since it is dangerous to update the target model when the tracking is in an unsteady state. We demonstrate the performance of our approach by comparing it with other trackers in tracking several challenging image sequences.**

I. Introduction and Related Work

VISUAL tracking is a crucial task in many applications including robotics, surveillance [7], [3], [38], and human computer interfaces [5]. Tracking objects through image sequences is still difficult despite intensive investigations over recent decades. Among the obstacles leading to difficulties, distractions and sudden motions have not been addressed in an integrated and effective way. This work aims at alleviating the problems brought by distractions and sudden motions by incorporating the efficiency of the mean-shift algorithm with the robustness of particle filtering in an adaptive manner. Specifically, we use Markov random fields (MRF) [12] motion priors to deal with distractions via modeling the interactions between a target and its distraction.

The mean-shift algorithm [5], [9] is a robust non-parametric probability density estimation method. It is a deterministic approach and focuses on target representation and localization. Comaniciu *et al.* [9] defined a spatially smooth similarity function and reduced the state estimation problem to a search of the basin of attraction of this function. Since the similarity function is smooth, a gradient optimization method is applied, which leads to fast localization. Despite its efficiency and robustness, the mean-shift algorithm is not good at coping with quick motions. In addition, distractions in the neighborhood of the target are threats to successful mean-shift-based tracking. The basic mean-shift algorithm assumes the target representation is sufficiently discriminated against the background. This assumption is not always true, especially when tracking is carried out on a dynamic background such is the case for surveillance with a moving camera. We introduce particles to deal with the first two problems because they provide multiple hypotheses. Adaptive tracking is one possible solution for alleviating the third problem [7], [36].

Particle filtering [13], [16] is a probabilistic method developed on the basis of filtering and data association. Particle filtering stands out from other filtering-based techniques because it represents multi-modal probability distributions using a weighted sample set $S = \{(s^{(n)}, \pi^{(n)})|n = 1, \ldots, N\}$ that maintains multiple hypotheses of the states of the targets [13], [16]. When the tracking is performed in a cluttered environment where multiple objects similar to the target may be present, particle filters find the target by validation and association of the measurements. However, since the number of particles can be large, a potential drawback of particle filtering is the high computational cost. Moreover, the particle set can degenerate and diffuse in a long sequence. Only a few particles with high weights are useful after the tracking in certain frames. Accurate models of shape and motion learned from examples have been used to deal with these problems [16]. Nevertheless, a drawback of this method is that the construction of explicit models is sometimes hardly achievable because of viewpoint changes.

The mean-shift tracking algorithm outperforms the particle filter when the representation of a target is sufficiently discriminative, the target does not jump beyond the bandwidth, and no serious distractions exist. Although it seems these conditions are too strict, we observed they can be met in a large percentage of real image sequences captured for surveillance or other applications.

In this work, the mean-shift algorithm is adopted as the main tracker as long as these conditions are met. In other words, only one particle driven by the mean-shift searching is used to estimate the state of the target. Auxiliary particles are introduced when sudden motion or distractions are detected. When sudden motion happens, we initialize one set of particles to model the highly dynamic properties of the target. When distractions approach the target, we initialize an auxiliary particle set for the target and another set for the distraction. The particles are distributed according to the distance between the target and distraction. The interaction between the target

The authors are with the Institute of Scientific and Industrial Research, Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, 560-0047, Japan, e-mail: *jerywangjq@gmail.com*, *yagi@am.sanken.osaka-u.ac.jp*.

Manuscript received April 27, 2008; revised October 19, 2008; accepted March 31, 2009.

and distraction is modeled by MRF motion priors. An MRF (or Markov networks), is a model of the joint probability distribution of a set of random variables having the Markov property. To simplify the problem, we use a pairwise MRF to obtain a more realistic motion model. The joint particle filtering is performed until the threat of distraction disappears. The use of the mean-shift algorithm then resumes.

We update the target model according to the separation of the foreground and background distributions. However, it is important to choose appropriate instances for model updating. As an approach different to the approaches used by other adaptive trackers [36][35][41], we propose to update the target model only when there are no threats because it is dangerous to update the target model when the tracking is unsteady. We compute log likelihood ratios of class-conditional sample densities of the target and its background. These ratios are applied in feature selection and distraction detection. The target model is updated according to feature selection results. Sudden motions are estimated using efficient motion filters [34].

Surveillance is an important application of visual tracking [38]. The proposed tracker is able to track targets in dynamical environments, which was been previously difficult [3]. The proposed method can deal with distractions and sudden motions, which have been the major obstacles in multiple object tracking.

Compared with the mean-shift and particle filtering algorithms, and several combinations of the two used in previous works, the proposed approach offers several advantages. It achieves high efficiency when the target moves smoothly. When sudden motions or distractions are detected, auxiliary particles are initialized to support the mean-shift tracker. Specifically, two auxiliary particle sets are distributed to deal with the threats of distractions. The interaction between a target and distractions is handled by MRF motion priors. Particle filtering partially solves the problems resulting from sudden motions or distractions.

The remainder of the paper is organized as follows. We introduce related works in Section II. Section III briefly presents the target modeling method. Section IV describes the feature selection and model updating methods. Section V introduces motion estimation and distraction detection. Section VII presents the application of auxiliary particles. The performance of the proposed method is evaluated in Section VIII and conclusions are given in Section IX.

II. Related Work

Particle filtering has been improved by combining different levels of information. Blake *et al.* [17] proposed the ICONDENSATION algorithm in which high and low levels of information are combined using importance sampling. However, modeling the dynamic characteristics accurately in an uncontrolled environment is difficult. One of the distinguishing characteristics of particle filtering techniques is that it represents multiple hypotheses about an object state in the form of a multi-modal state density. The state estimation is performed by calculating simple moments of the state density. This approach is not valid when distractions approach the target, which leads to several peaks in the state density. Multiple hypotheses can increase the variance of the distribution and shift the means of the particle set from the object position.

Particle filtering has been extended by Pitt and Shephard [27], who replaced the standard resampling schemes with a more sophisticated algorithm. They explicitly included an auxiliary indexing variable by considering a proposal over the entire path of the process up to the current time. Their approach is effective in finding particles that are more likely to survive at the next time step. However, it depends on two assumptions that cannot be met in many cases. First, they assume the observation density is locally smooth. In fact, the distribution of particles contains multiple peaks when there is distraction near the target. Second, it is assumed the distribution for a specifically given base sample, the motion probability distribution, is much narrower than the overall prediction distribution. This assumption can be incorrect since distractions might lead to large bias. Owing to the above limitations, their approach can meet difficulties in dealing with distractions.

Sullivan and Rittscher [30] first noticed the advantages of the mean-shift and particle filter algorithms. They proposed particle filtering-based tracking guided by deterministic searching based on a sum-of-squared differences (SSD) type cost function. The size of the particle set is adjusted according to the difficulty of the problem at hand, which is associated with motion. A deterministic search using the mean-shift has also been applied in a hand tracking algorithm by embedding the mean-shift optimization into particle filtering to move particles to local peaks in the likelihood, which improves the sampling efficiency [29]. The integration brings uncertainty to the deterministic method so that the statistical property can improve the robustness of their systems. However, directly biasing sampled particles from the previous proposal distribution changes the overall posterior distribution. This makes updating the weights of the particles without bias extremely difficult, and no method suitable for updating the particle weights after correcting for the mean-shift bias has been given. Moreover, no solutions have been provided for solving the problems brought by distractions, which is the main topic of this paper.

Maggio and Cavallaro [25] combine mean-shift and particle filtering with an adaptive state transition model. The transition model, which is calculated using the average state velocity in previous certain frames, is a simplified version of Zhou *et al.*'s work [41]. The transition model might be helpful in handling sudden motions. However, the likelihood guiding the particles can be misleading since no feature selection or foreground/background separation are taken into account. Dore *et al.* [10] treat target localization as a two-step process. First, mean-shift is adopted to find approximate locations and then particle filtering refines the results. The major drawback of their approach is that errors in the first step can lead to tracking failures. Moreover, none of the above works has effective sudden motion or distraction detection methods, which are important for difficult tracking problems.

In the above works, the variational searching algorithm can be attracted to a distraction approaching the target. As a result, the trackers cannot find the target even after the distraction has

passed. Rather than using a combination of these algorithms, we introduce MRF motion priors. We initialize an auxiliary particle set for each of the target and distraction. The particles are distributed according to the distance between the target and distraction. Cai *et al.* [6] embedded the mean-shift algorithm into the particle filter framework to stabilize the trajectories of the targets. It is necessary to learn classifiers for the targets in their work, and this is not always possible in tracking applications. Although the mean-shift and particle filters have been combined in various ways in previous works, none of the combinations dealt with occlusions and distractions explicitly.

Sudden motions have been dealt with by density estimation or incorporation of high level knowledge. Yuan *et al.* [39] dealt with the problems brought by low frame rates using recognition techniques. Such techniques require a reliable a priori object appearance model, which has to be learnt before tracking. Our work aims to deal with tracking tasks in the absence of an a priori object appearance model. We update a target model on the fly during tracking. The updating is necessary because the object appearance can change. By doing so, we do not need a separate model for the object appearance model since this can be determined from the object motion. Kwon and Lee [20] introduced the Wang-Landau algorithm to estimate the density of states terms, which is helpful in dealing with sudden motions. They achieved excellent results for a few sequences with sudden motion. However, their approach can meet problems when there are other objects with similar appearance to the target in the sequence. The density estimation can mislead the tracking.

Zhou *et al.* [41] proposed a particle filtering-based approach that focuses on adaptive target appearance modeling. The target modeling was conducted by updating a mixture appearance model. Their work also changes the number of particles, which bears certain similarity to our work. However, they did not consider the discriminative power of different features based on the contrast between a target and its background, which is very important for effective target representation. In addition, their method cannot deal with distractions, with there being no clear solution to this problem in their work.

III. Target Modeling

The target model should be as discriminative as possible to distinguish between the complex target and background. In addition, it should be able to adapt to appearance changes due to illumination or viewpoint variations. Color is a powerful feature for tracking deformable objects in image sequences with complex backgrounds. It has been widely used in mean-shift [9], [37] and particle filtering [17], [26]. However, the performance of tracking using color features is not good when an object and its background have similar colors. A similarity measure of color cues is not sufficiently discriminative because the appearance of the target is transformed into a global histogram. Moreover, it is computationally expensive to compute histograms for a large sample set for particle filtering.

Multi-cues have been widely used in tracking and detection systems [4], [17]. We use an adaptive tracking algorithm that represents the target using reliable features selected from color and shape-texture cues [36]. It has been demonstrated that using multi-cues can result in better tracking performance against distractions [36].

Color cues are computed in three color spaces: RGB, HSV and normalized rg. There are seven color features (R, G, B, H, S, r, g) in the candidate feature set. These color channels are each quantized into 12 bins. A color histogram is calculated using a weighting scheme in which the Epanechnikov kernel is applied [9]:

$$k(\mathbf{x}) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-\|\mathbf{x}\|^2), & \text{if } \|\mathbf{x}\|^2 \leq 1; \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where c_d is the volume of the unit d-dimensional sphere. Thus we increase the reliability of the color distribution when these boundary pixels belong to the background or get occluded.

The color distribution $h_f = \{p_f^{(b_{in})}\}_{b_{in}=1\ldots m}$ of the target is given by

$$p_f^{(b_{in})} = C_f \sum_{\mathbf{x}_i \in R_f} k(\|\mathbf{x}_i\|)\delta[h(\mathbf{x}_i) - b_{in}], \tag{2}$$

where δ is the Kronecker delta function and $h(\mathbf{x}_i)$ assigns one of the m-bins ($m = 12$) of the histogram to a given color at location $\mathbf{x}_i$. C_f is a normalization constant. It is calculated as

$$C_f = \frac{1}{\sum_{\mathbf{x}_i \in R_f} k(\|\mathbf{x}_i\|^2)}. \tag{3}$$

A shape-texture cue is described by an orientation histogram, which is computed based on image derivatives. The derivatives are obtained by running Scharr masks [18]. Scharr masks can give more precise results than Sobel masks do [18]. The orientations are also quantized into 12 bins. Each orientation is weighted and assigned to one of two adjacent bins according to its distance from the bin centers.

The similarity between the model and its candidates is determined using the Bhattacharya coefficient [9]. The Bhattacharya coefficient is an approximate metric measuring the amount of overlap between two samples. It can be interpreted as the cosine angle between two vectors in the viewpoint of geometry.

IV. Feature Selection and Model Updating

A. *Log-Likelihood Ratio Images*

To determine the descriptive abilities of different features, we compute log-likelihood ratio images [7], [31] based on the histograms of the target and its background. Log-likelihood ratio images are also employed in detecting possible threats to the target.

The likelihood ratio can be used to produce a function that maps feature values associated with the target to positive values and those associated with the background to negative values. The frequencies of the pixels that appear in a histogram bin ($p^{(b_{in})}$) are calculated as $\zeta_f^{(b_{in})} = p_f^{(b_{in})}/n_f$ and $\zeta_b^{(b_{in})} = p_b^{(b_{in})}/n_b$, where n_f is the pixel number of the target region and n_b the pixel number of the background.

The log-likelihood ratio of a feature value is given by

$$L^{(b_{in})} = \max(-1, \min(1, \log\frac{\max(\zeta_f^{(b_{in})}, \delta_L)}{\max(\zeta_b^{(b_{in})}, \delta_L)})), \quad (4)$$

where δ_L is a very small number (δ_L is set to 0.001 in this work). The likelihood image for each feature is created by back-projecting the ratio for each pixel in the image.

B. *Discriminative Feature Selection*

The discriminative abilities for different features used in separating the target from the background may differ. Those features with high discriminative abilities probably provide reliable tracking. Feature selection can help minimize the tracking error and maximize the descriptive ability of the feature set. Given m_d features available for tracking, we wish to find the subset of features of size m_m ($m_m < m_d$) with high discriminative ability.

The goodness of a feature can be evaluated using different criteria such as mutual information or the variance ratio. The mutual information theory is used first to select features that discriminate the target and background. The mutual information is the relative entropy between the joint distribution of the target and background. It is defined as

$$M(F, B) = H(F) + H(B) - H_j(F, B), \quad (5)$$

where $H(F)$ and $H(B)$ are the entropies of the foreground (target) and background, respectively, and H_j is the joint entropy of the foreground and background. The mutual information is minimized when the feature is the most discriminative. Zaffalon and Hutter [15] derived an exact analytical expression for the mean of mutual information and an analytical approximation of the variance. Leung and Gong [21] adopted these in their Haar feature selection. We implemented feature selection using mutual information and the results were not good owing to only the first order approximation being used.

Instead of directly ranking the features using the mutual information, we find the features with the largest corresponding variances of the mutual information. Although variance ratios cannot be calculated directly based on foreground/background histograms, the likelihood images make the calculation possible by transforming histograms into unimodal distributions. We find the features with the largest corresponding variances. Following the method used by [7], on the basis of the equality $\text{var}(x) = E[x^2] - (E[x])^2$, the variance of Equation(4) is calculated as

$$\text{var}(L; p) = E[(L^{b_{in}})^2] - (E[L^{b_{in}}])^2.$$

The variance ratio of the likelihood function is defined as [7]

$$\text{VR} = \frac{\text{var}(B \cup F)}{\text{var}(F) + \text{var}(B)} = \frac{\text{var}(L; (p_f + p_b)/2)}{\text{var}(L; p_f) + \text{var}(L; p_b)}. \quad (6)$$

C. *Weighting of Different Cues*

We choose a number of cues because they are needed for describing a wide range of targets. For successful tracking, there are only a few cues that are helpful for foreground/background discrimination. Blindly using cues having little discriminative ability only misleads the tracking and may even hide useful information. We use the first two features selected using the method described in the previous subsection. These features are combined for the mean-shift tracking. Since the first two features have different descriptive abilities, different weights should be assigned to them. It may seem the weights should be based on the variance ratios calculated using Eq. 6. However, variance ratios represent the discriminative abilities of different features, whereas the weights given to different features should reflect their descriptive abilities. In this work, we use an alternative to compute the weights for the appropriate integration of the selected features.

To compute the descriptive ability of an input cue for a given target, it is assumed the probability distribution functions of the target region are represented by the log-likelihood images obtained using feature i. We calculate the probabilities of a pixel $\mathbf{x}$ to be assigned to the foreground (P_f^i) and background (P_b^i) based on the selected feature i:

$$P_f^i(\mathbf{x}) = \log(\max(\zeta_f^i(\mathbf{x}), \delta_L)), \quad (7)$$

and

$$P_b^i(\mathbf{x}) = \log(\max(\zeta_b^i(\mathbf{x}), \delta_L)). \quad (8)$$

Since the labeling of the pixels is provided in the first frame, we calculate the wrong label assignment on the basis of back-projection using feature i for pixel $\mathbf{x}$ with Bayesian formulation. It is calculated as the sum of probabilities of a foreground pixel having a background label assigned and a background pixel having a foreground label assigned:

$$\begin{aligned} P^i(\mathbf{x}) &= P(\mathbf{x} \in f)P(\mathbf{x} \to b | \mathbf{x} \in f) \\ &\quad + P(\mathbf{x} \in b)P(\mathbf{x} \to f | \mathbf{x} \in b). \end{aligned}$$

Summing the probabilities, we obtain

$$\begin{aligned} P_i(\mathbf{x}) &= \frac{1}{2}\Big(\int_{\{\mathbf{x}: p_f^i(\mathbf{x}) > p_b^i(\mathbf{x})\}} p_b^i(\mathbf{x}) d\mathbf{x} \\ &\quad + \int_{\{\mathbf{x}: p_b^i(\mathbf{x}) > p_f^i(\mathbf{x})\}} p_f^i(\mathbf{x}) d\mathbf{x}\Big). \end{aligned}$$

Based on the above equation, we calculate the probability of feature i as

$$P_i = \frac{1}{2}\int \min(p_f^i(\mathbf{x}), p_b^i(\mathbf{x})) d\mathbf{x} \quad (9)$$

The weight of feature i is then

$$w_i = \frac{(P_i)^{-1}}{\sum_{k=1}^{m_m}(P_k)^{-1}}, \quad (10)$$

where m_m is the number of the selected features.

D. *Updating the Target Model*

It is necessary to update the target model because the appearance of a target tends to change during tracking. Unfortunately, updating the target model adaptively may lead to tracking drift because of the imperfect classification of the target and background. We found the feature adaptation needs to be carried out very carefully. The updated model may drift

from the actual model because the new model cannot describe the target perfectly. In our method, the adaptation is employed when the new feature is better than the previous one. Only one feature is updated in each adaptation to maintain smooth tracking.

To reliably update the target model, we propose an approach based on similarities between the initial and current appearances of the target. Similarity θ is measured by simple correlation-based template matching performed between the initial and current frames. The updating is conducted according to the similarity θ:

$$H_m = (1-\theta)H_i + \theta H_c, \tag{11}$$

where the H_i is the histogram computed for the initial target, H_c is the histogram of the current appearance of the target, and H_m the updated histogram of the target.

Template matching is performed between the initial model and the current candidates. Since we do not use the search window that is necessary in template matching-based tracking, the matching process is efficient and contributes little computational cost to our algorithm.

In an unstable tracking period (when sudden motions or distractions are detected), the classification of the target and background is not reliable. It is difficult to update the target model reliably at these moments. Thus the model is updated when the tracker is in a stable state.

V. Sudden Motion Estimation

Discriminative mean-shift tracking is sufficient for determining the position of a target when it moves smoothly and slowly. Particles are necessary for estimating the correct position of the target when it moves quickly. The number of particles is adjusted according to motion information for the target.

Both the target and its background can have sudden motions, which threaten successful tracking. We propose a novel sudden motion estimation method that is a two-step process. Efficient local motion filtering is carried out for each frame. If local motion is sufficiently large, we estimate the global motion by fitting a motion model. We found that large local motion can be detected no matter the sudden motions of the target or background. This two-step process is very efficient.

A. *Local Motion Filters*

We revised the motion filters that have been applied in pedestrian detection [34]. We estimate the motion of foreground and background regions simultaneously and partially solve the problem of having a dynamic background.

Five motion filters are applied to image pairs:

$$\Delta_i = \frac{1}{n_{Rg}} \int_{\mathbf{x}\in Rg} |I_t(\mathbf{x}) - I_{t+1}^{\tau_i}(\mathbf{x})| d\mathbf{x}, \tag{12}$$

where I_t and I_{t+1} are consecutive images, n_{Rg} is the number of pixels in a specific region, and $\tau_i \in \{\diamond, \leftarrow, \rightarrow, \uparrow, \downarrow\}$ is the image shift operator denoting no shift and a left shift, right shift, upward shift, and downward shift of one pixel, respectively.

The motion filters are applied to the target and its background region. The results for the four motion filters ($\Delta_i, i \in \{1,2,3,4\}$) are compared with the absolute differences Δ_0:

$$M_i^f = |\Delta_i^f - \Delta_0^f|, M_i^b = |\Delta_i^b - \Delta_0^b| \tag{13}$$

where M_i represents the likelihood that a particular region is moving in a given direction.

We compute the maximum motion likelihood to determine the number of particles for the tracking:

$$M_{max} = \max(|M_i^f - M_i^b|)_{i=1,2,3,4.}. \tag{14}$$

Given the high efficiency of the estimation method, it is performed for each frame before tracking is carried out.

B. *Background Motion Estimation*

When large local motions are detected, we estimate background motion. Background motion is estimated using frame differencing in image sequences captured by a stationary camera. However, frame differencing is not directly applicable when the background of the target is dynamic.

We fit a parametric background motion model using local features detected in consecutive frames. The Harris features [14], scale-invariant features (SIFT) [22] or other more reliable features can be used in the motion estimation. We choose Kanade-Lucas-Tomasi (KLT) [23], [2] features because they are computationally cheaper than other features. Error may exist in feature correspondences found using normalized cross correlation. We filter out the outliers out using a variation [32] of the random sample consensus (RANSAC) algorithm [11]. The largest set of inliers obtained from the RANSAC is used to fit an affine motion model. The subregion in the current frame is warped on the basis of the background motion model.

VI. Distraction Detection

Distractions in the neighborhood of the target have a similar appearance to the target and are possible threats to successful tracking. When the similarity between the target model and its candidate is less than a certain value (ρ^T), distraction detection is performed using spatial reasoning [7] to find peaks besides the target in the log-likelihood ratio images. Note that the log-likelihood ratio images here are back-projection results of the conditional distributions based on selected features.

Assuming the region R_T actually contains the target and the region R_D is a possible distraction, we want to find the region that is the greatest threat to the target. That is, we want to find the region where the sum of the log-likelihood ratios is closest to that in the target region:

$$\min(|\sum_{R_D} L^{(b_{in})} - \sum_{R_X} L^{(b_{in})}|), \tag{15}$$

where R_X is a region in the neighborhood of the target.

A simple approach for detecting distractions is an exhaustive search. This process requires the generation of histograms for the regions centered at every possible point. It is too expensive to compare the sums of log-likelihood ratios in all possible regions with that in the target region. The searching process can be accelerated using two-step processing based on an integral likelihood image [28].

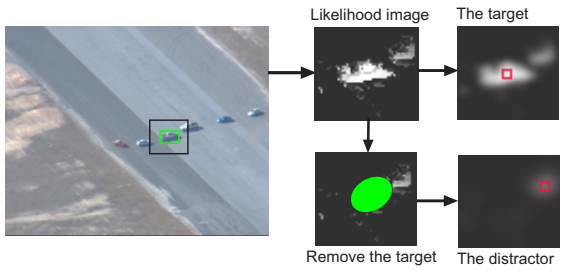

Fig. 1. Distraction detection. The target is found and then the region corresponding to the target is removed. The region with the maximum sum of likelihood ratios is detected.

Fig. 2. Fast distraction detection using integral log-likelihood ratio images. The sum of the log-likelihood ratios in the rectangle $(x', y'; x", y")$ is defined by its upper left (x', y') and lower right $(x", y")$ corners in the image. The sum of any rectangle in the searching region can be computed in constant time.

A. *Fast Distraction Detection Based on the Integral Likelihood Image*

Viola and Jones [33] found it is possible to calculate the sum of the values within rectangular regions in linear time without repeating the summation operation for each possible region. A constant number of operations for each rectangular sum is needed to compute such sums over distinct rectangles many times. A cumulative image function is defined such that each element of this function holds the sum of all values to the left and above the pixel including the value of the pixel itself. Starting from the top left corner and traversing to the right and then downward, the value at the current pixel of the cumulative image is obtained by the addition of the top-left and the bottom-right values, then subtraction of the top-right and bottom-left values.

We calculate the integral likelihood image on the basis of on an idea in [33]. The similarity value for each pixel in the image is computed using addition and subtraction, which are computationally efficient. The peak D_T representing the target region can be found in the similarity value image. The target region in the log-likelihood image is removed and a second integral log-likelihood image is computed from the current image. The most dangerous distraction is detected by searching for the peak D_D in the convolved image.

The difference between the two peaks represents the threat strength of the distraction:

$$\rho = |D_D - D_T|. \tag{16}$$

The distraction may attract the mean-shift tracker to an incorrect position if it is strong enough. We initialize an auxiliary particle set to track the distraction region when ρ is less than the given threshold ρ^T.

VII. Auxiliary Particle Filtering

Particle filters, also known as sequential Monte Carlo methods, are powerful model estimation techniques based on particle approximation. We give a brief description of particle filtering techniques in the following subsection. Particle filtering for sudden motions and distractions is then described.

A. *Particle Filtering*

Particle filtering is a sequential non-linear Bayesian state estimation method that uses various approximations. In particular, particle filtering represents the probability distribution of the state vector as a weighted particle set, which is updated and propagated by the algorithm recursively. Particle filtering aims to estimate the distribution $P(x_t|z_{0:t})$, where x_t is the unobserved state at time t and $z_{0:t}$ is the sequence of observations from time 0 to time t. Particle filtering does not have the restrictions of Gaussian assumptions regarding the transition and noise models, which are required by Kalman filters. Therefore, particle filtering is capable of evaluating a wide range of distributions.

Letting the weighted particle set $\{s_t^{(n)}, \pi_t^{(n)}\}_{n\in\{1,\cdots,J\}}$ be the representation of $P(x_t|z_{0:t})$, where $\pi_t^{(n)}$ is the weight of particle $s_t^{(n)}$, $P(x_t|z_{0:t})$ is described by

$$P(x_t|z_{0:t}) \approx \sum_n \pi_{t-1}^{(n)} \delta(x_t - x_{t-1}^n).$$

The particle filtering estimates the states of a target in a recursive manner:

$$P(x_t|z_{0:t}) = \alpha P(z_t|x_t) \int P(x_{t-1}|z_{0:t-1}) P(x_t|x_{t-1} dx_{t-1}).$$

Given the distribution $P(x_{t-1}|z_{0:t-1})$, particle filtering calculates the filtered distribution $P(x_t|z_{0:t})$. Embedding the weighted particle set into the above equation, we obtain the approximation of $P(x_t|z_{0:t})$:

$$P(x_t|z_{0:t}) \approx \alpha P(z_t|x_t) \sum_n \pi_{t-1}^{(n)} P(x_t|x_{t-1}^{(n)}).$$

It is important to compute samples in the weighted particle set. Sequential importance re-sampling has been widely used for this purpose [1]. Using this technique, particle filtering works in the following way. First, J samples $x_t^{(n)}$ are drawn from the proposal distribution $q(x_t)$:

$$x_t^{(n)} \sim q(x_t) = \sum_n \pi_{t-1}^{(n)} P(x_t|x_{t-1}^{(n)}),$$

by selecting a random number r uniformly from [0,1], choosing the corresponding particle i, and then sampling from $P(x_t|x_{t-1}^{(n)})$. This transition model can be any model from which samples are easily drawn. Second, we set the weight $\pi_t^{(n)}$ as the likelihood:

$$\pi_t^{(n)} = P(z_t|x_t^{(n)}).$$

The samples $x^{(n)}$ are fair samples from $P(x_t|z_{0:t-1})$. Reweighting them in this fashion accounts for the observation z_t. Third, we normalize the weights $\pi^{(n)}$ using

$$\pi_t^{(n)} \propto \frac{p(z_t|x_t^{(n)})p(x_t^{(n)}|x_{t-1}^{(n)})}{\pi(x_t^{(n)}|x_{0:t-1}^{(n)}, z_{0:t})},$$

where $\sum_{n=1}^{J} \pi_t^{(n)} = 1$.

B. *Particle Filtering for Sudden Motion*

We handle sudden motions by introducing particle filtering and incorporating motion information estimated. The number of particles (N_p) is determined from the motion computed:

$$J_S = \max(\min(J_0 M_{max}, J_{max}), J_{min}), \tag{17}$$

where J_0 is a coefficient and J_{min} is the smallest number of particles and J_{max} the largest number of particles required to maintain reasonable number of particles.

The motion model is a normal density function centered on the previous pose with a constant shift vector:

$$\mathbf{x}_t^j = \mathbf{x}_{t-1} + \mathbf{x}^c + \mathbf{u}_t^j, \tag{18}$$

where $\mathbf{u}_t^j$ is a standard normal random vector and $\mathbf{x}^c$ a constant shift vector from the previous position according to the motion estimation results.

We assume a standard linear Gaussian model. The initial joint state is Gaussian:

$$P(\mathbf{x}_0) = \mathcal{N}(X_0; \mathbf{m}_0, \mathbf{v}_0), \tag{19}$$

where $\mathbf{m}_0$ is the mean and $\mathbf{v}_0$ is the corresponding covariance matrix. Gaussian noise is added to the linear motion model for the model to deal with:

$$P(\mathbf{x}_t|P_{t-1}) = \mathcal{N}(X_t; A\mathbf{x}_{t-1}, \Gamma), \tag{20}$$

where Γ is the prediction covariance and A is a linear prediction matrix.

The state estimation results of the mean-shift are passed to particle filters. The motion prior is given on the basis of the motion estimated. When the motion is large, more particles are distributed.

C. *Particle Filtering for Distractions*

After distractions are detected, a joint particle filter with an MRF motion model is initialized [19]. The motion interaction between the target and distraction $\psi(X_{it}, X_{jt})$ is described by the Gibbs distribution $\psi(X_{it}, X_{jt}) \propto exp(-g(X_{it}, X_{jt}))$, where $g(X_{it}, X_{jt})$ is a penalty function approximated by the distance between the target and distraction.

The posterior for the joint state X_t is approximated as a set of J weighted samples:

$$\begin{aligned} P(X_t|Z^t) \quad &\approx \quad kP(Z_t|X_t) \prod_{ij \in E} \psi(X_{it}, X_{jt}) \\ &\sum_J \pi_{t-1}^{(J)} \prod_i P(X_{it}|X_{i(t-1)}^{(J)}), \end{aligned}$$

where the samples are drawn from the joint proposal distribution, k is a normalizing constant that does not depend on the state variables, E is the edges in the MRF model, and the samples are weighted according to the factored likelihood function:

$$\pi_t^{(s)} = \prod_i^2 P(Z_{it}|X_{it}^{(s)}) \prod_{ij \in E} \psi(X_{it}^{(s)}, X_{jt}^{(s)}),$$

where Z_{it} are measurement nodes.

Based on this form, the predictive motion models of the target and its distraction are kept. In addition, the MRF interaction potentials afford us the possibility of easily specifying domain knowledge governing the joint behavior of the interaction.

The interaction requires a particle filter in the joint configuration space. We use a mixture of particle filters in a method know as probabilistic exclusion [24], for which each pixel measured belongs to only one target.

D. *Algorithm Summary*

The detailed steps of the proposed tracking algorithm are as follows.

Algorithm: *Adaptive Mean-Shift Tracking with Auxiliary Particles.*

Input: Initial target region given in the first frame I_1
t video frames $I_1, \ldots, I_t$;
Output target regions in $I_2, \ldots, I_t$

Initialization in I_1
1. Save the initial target appearance for model updating;
2. Compute the similarity (θ_1) between the target model and the candidate.

FOR each new frame I_j:
Estimate the motion (M_j) in the consequential frames;
IF $M_j > M_T$
THEN initialize particles according to the motion estimated.
ELSE
IF the similarity is less than a given threshold ($\theta_{j-1} < \theta^T$)
THEN detect distractions in the neighborhood of the target
IF Distraction is detected ($\rho < \rho^T$)
Initialize MRF particles;
ELSE
Update the target model.
END IF
END IF
END IF
Estimate the position of the target.
Compute the similarity S_j for the next frame.
END FOR

We give implementation details that are important in the tracking.

Switching from mean-shift to particle filtering When sudden motions or distractions are detected during the tracking, it

Fig. 3. Tracking results for Egtest05. The proposed tracker can deal with occlusions in the sequence.

is necessary to adapt to particle filtering to deal with tracking problems. The particle filtering steps have been introduced in the previous subsections.

Switching from particle filtering to mean-shift When threats such as sudden motions or distractions disappear, the variance of the particle set becomes very low. The mean of the particle set is calculated and re-initialized as the starting point of the mean-shift algorithm. The mean-shift algorithm runs until another threats are detected and particle filtering resumes.

Occlusions The tracking switches to particle filtering when the observation likelihood of the target is below a certain threshold, which indicates a possible occlusion. During the occlusion period, the updating of the target is stopped to avoid drifts of the target model. Since the camera and targets are not assumed to be stationary, zero-motion is assumed with different uncertainties calculated based on motion detection results.

VIII. Experimental Results

The proposed tracker was implemented and its performance tested for a wide variety of challenging image sequences in different environments and for different applications. In some sequences, both sudden motions and distractions exist, which makes the tracking challenging. We have carried out qualitative and quantitative tests on these sequences. The advantage of the proposed tracker is demonstrated by comparing its performance with the performances of other trackers.

A. *Qualitative Performance Evaluation*

Face tracking is crucial for the new generation of human and computer interfaces. A video sequence[1] containing faces is tested using our method (Fig.4). A man's face moves from left to right very quickly and then back. The illumination of the face varies in the sequence. There are also clutter and ambiguous colors in the background. The results for the proposed tracker are compared with the results for the mean-shift and particle filtering algorithms. Note that rg color features are used in the mean-shift and particle filtering algorithms to deal with the varying illumination. The basic mean-shift algorithm fails at the 5^{th} frame. The particle filtering algorithm does a better job than the mean-shift algorithm does. However, the tracking accuracy is not high owing to the low discriminative ability of the feature. The proposed tracker selects good features and tracks the face successfully for the whole sequence.

The second tracking example (Fig.5) demonstrates the usefulness of the auxiliary particles and the model updating strategy. The target's appearance changes greatly in the sequence. Furthermore, the target has large motion. It is difficult to track the target through the sequence. The mean-shift tracker fails when the girl turns her face away from the camera. The particle filtering tracker was also unable to cope with the girl turning away. In contrast, the proposed tracker updates the target model adaptively when the appearance changes and auxiliary particles are introduced to deal with sudden motions. Thus it was able to track the target successfully thorough the sequence.

The third example (Fig.6) uses a sequence captured using a moving camera. The background in the sequence is dynamic. Thus a background subtraction method is useless in this case. There are objects (cars) in the sequence that have appearances similar to that of the target, the target appearance changes greatly and there are sudden motions of the camera in the sequence, all of which make tracking difficult. The target was tracked by five trackers: the proposed tracker, the basic mean-shift tracker, the variance ratio tracker [7], the peak difference tracker [7] and the fore/background ratio tracker [5] trackers. We only show the tracking results for the proposed tracker in Fig.6 because all other trackers failed to track the target in the sequence. (The quantitative performance is given in the next subsection.) The basic mean-shift algorithm and the variance ratio algorithm failed when the target met other objects with similar appearances. Other trackers failed owing to the car turning or sudden motion of the camera. In contrast, the proposed algorithm detects the threats of objects with similar appearances and sudden motions. The tracker uses an appropriate strategy to deal with these difficulties and tracks the target successfully throughout the sequence. This example demonstrates the importance of the sudden motion estimation and threat detection.

The above tests show that the proposed tracker can detect threats such as sudden motion and distraction. Choosing an appropriate strategy to deal with these problems makes the tracking robust. Moreover, the tracking is efficient thanks to the use of the mean-shift and particle filtering.

B. *Quantitative Performance Evaluation*

We quantitatively evaluate our approach using a public CMU dataset with ground truth [8]. The dataset comprises six sequences: EgTest01(1820 frames), EgTest02 (1300 frames), EgTest03 (2570 frames), EgTest04 (1832 frames), EgTest05 (1763 frames) and Redteam (1917 frames). There are several factors that make the tracking challenging: different

[1]The human face image sequence can be obtained from http://vision.stanford.edu/∼birch/headtracker/.

Fig. 4. Face tracking results using the basic mean-shift tracker (first row), the particle filtering tracker (second row), and the proposed method (third row). The face in this sequence moves very quickly.

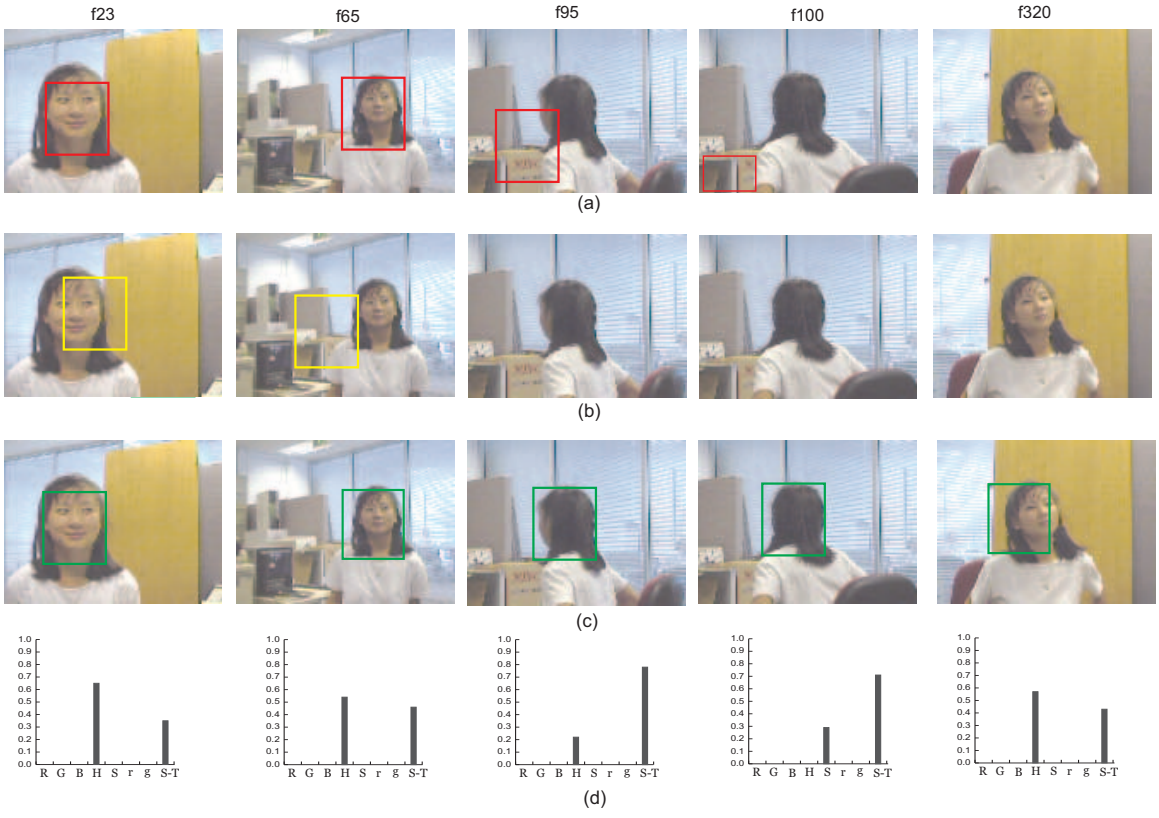

Fig. 5. Tracking results for the woman's face sequence using (a) the basic mean-shift tracker, (b) the particle filtering tracker, and (c) the proposed method. The features selected by the proposed tracker and their weights are shown in (d). During the initialization, the hue and shape-texture features are selected for representing the target. Hue has a larger weight than the shape-texture feature does because the face can be described well by hue. When the woman rotates her head, the weight of hue decreases. The hue feature is replaced by the shape-texture feature when the woman's head has turned around. The hue feature is selected again when her head rotates back. Note the initial target representation has never been discarded totally. It is used to anchor the current representation.

Fig. 6. Tracking results for the EgTest02 sequence using the proposed tracker. Cars similar to the target are threats to robust tracking in the sequence. Both the target and the background have large variations in appearance.

viewpoints (the sequences are captured by moving cameras), similar objects nearby, sudden motions, illumination changes, reflectance variations of the targets, and partial occlusions.

The tracking results are compared with those of basic mean-shift and particle filtering trackers. Since the proposed tracker updates the target model on the basis of feature selection, it is reasonable to compare it with those adaptive trackers. The variance ratio, peak difference [7] and the adaptive multi-feature [36] trackers are included for this purpose. We have implemented the mean-shift-embedded particle filtering tracker [29] using the same parameter settings as those in the work [29]. The performance of that tracker is also compared because it is a representative variation of the combination of the mean-shift algorithm and particle filtering. For the particle filtering tracker, the target model is represented by $12\times12\times12$-bin RGB histograms. There are 100 samples in the sample set. RGB histograms are also adopted in the basic mean-shift algorithm. The similarity measure is the Bhattacharya distance between the model and its candidate.

The most important criterion for the comparison is the fraction of the dataset tracked, which is ratio of the number of tracked frames and the total number of frames. The tracking is considered to be lost if the bounding box does not overlap the ground truth. The tracking is then stopped then even if the tracker accidentally catches the target again. The tracking success rates achieved by each tracker are compared and the results are shown in Table I. The numbers in parenthesis are the ranks of the trackers. The proposed tracker gives the best results (or results no worse than any other trackers) in all test sequences. The basic mean-shift algorithm has poor performances for several sequences (EgTest01, EgTest02, EgTest04, and EgTest05). This is reasonable because the basic mean-shift does not include an adaptive strategy and does not detect threats of sudden motions and distractions, which are common in these sequences. The variance ratio tracker [7] has better performance than the basic mean-shift tracker does for EgTest01 and Redteam sequences. However, it is worse than the basic mean-shift tracker for other sequences because only color features are used. The peak difference tracker has similar performance to that of the variance ratio tracker. All these mean-shift-based trackers have difficulties in dealing with sudden motions and distractions. The particle filtering-based tracker and the mean-shift-embedded particle filtering tracker [29] perform well for EgTest01, EgTest04, and Redteam sequences. One reason for this is that the particle filtering framework is good at dealing with sudden motions appeared in these sequences. We noticed that the proposed tracker has a performance similar to or better than the performances of other particle filtering-based trackers. The particle filtering-based trackers have relatively low performances for EgTest02 and EgTest03 sequences because distractions in the two sequences threaten the tracking. The proposed tracker overcomes the threats owing to its strategy of modeling the interaction between the targets and distractions on the basis of two sets of auxiliary particles with MRF motion priors.

We evaluate the tracking accuracy of the proposed tracker as another important criterion. Our tracker has better tracking accuracy than that in [36] has. We evaluate the tracking accuracy in the tracked frames using two criteria: average overlap between bounding boxes (OL-BB), which is the percentage of the overlap between the tracking bounding box and the box identified by ground truth files; average overlap between bitmaps within the overlapping bounding box area (OL-BM), which is computed in the area of the intersection between the user bounding box and the ground truth bounding box. The comparison results (including tracking success rates

and tracking accuracy) are given in TableI. Compared with the performance of the tracker used by [36], the proposed approach gives better tracking accuracy for four sequences. The proposed tracker has similar tracker accuracy for the other two sequences. We believe that the performance gain is achieved because we use a weighted histogram to represent a target, which is different from the case for the tracker used by [36] that employs a fixed two-dimensinoal histogram. The localization error of the proposed tracking is lower since the back-projection is carried out on the basis of dense histograms. Moreover, the tracking accuracy for the difficult frames is improves.

The comparisons demonstrate the proposed tracking algorithm has a performance superior to the performances of the other trackers. The tracking results for EgTest02 are shown in Fig. 6. Despite the distractions and sudden motions in the sequence, the proposed tracker completes the tracking successfully. Fig. 6(d) illustrates how the appearance of the target changes over time.

There are sudden motions and image blur in the EgTest04 sequence, which lead to the failure of the basic mean-shift tracker. The proposed tracker detects the motion successfully and initializes auxiliary particles. These particles enable the proposed tracker to cope with the sudden motion.

The proposed method has relatively poor performance in EgTest03, EgTest04, and EgTest05 sequences. The main reason leading to failure for EgTest03 and EgTest04 is the congestion of several similar objects. This shows the proposed method is not perfect in handling the distraction problem. There are long-duration occlusions in the EgTest05 sequence, which lead to tracking failure. The proposed tracker can deal with short-term occlusions, but it is still limited in dealing with long-duration occlusions. Despite these limitations, the proposed tracker has shown its potential in the pursuit of efficiency and robustness.

C. *Computational Complexity*

The running time of the proposed tracker depends on the difficulty of the image sequence in which a target is being tracked. If there are frequent sudden motions or distractions happen frequently, the efficiency of the proposed tracker is relatively low. Otherwise it has high efficiency because the mean-shift algorithm is adopted in most cases. The current implementation runs on an Intel Centrino 1.6GHz laptop with 1Gb RAM.

The performance of computational complexity is given in TableI. The proposed tracker is more efficient than other particle filtering-based trackers in all the test sequences because it adopts particles only if necessary. Although mean-shift-based trackers are sometimes faster than our algorithm does, they cannot achieve stable tracking and have lower tracking success rates than our algorithm. The proposed tracker uses less time than the approach in [36] in EgTest01 and RedTeam because it updates the appearance model at necessary occasions. The two sequences are not very difficult although a few sudden motions exist, which lead to the failure of the other trackers in EgTest01. The distractions and sudden motions in EgTest02 and EgTest03 make the proposed tracker slower than for EgTest01 and RedTeam. The occlusion handling strategy also makes the tracking for EgTest05 less efficient. These results reflect our idea that: complicated approaches should be used for those difficult frames and a simple and efficient strategy should be used for those easy frames.

IX. Conclusions and Future Work

We have described an adaptive mean-shift tracking algorithm with auxiliary particles in the pursuit of robust and efficient tracking. The arrangement of the particle filtering and the mean-shift algorithm is based on the difficulty of the tracking associated with sudden motions and distractions. The strategy of updating the model by our tracker effectively deals with changes in the appearance of targets. The proposed approach provides better performance than do the mean-shift, particle filtering, and other related trackers.

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Trans. on Signal Processing*, 50(2):174–188, Feb. 2002.

[2] S. Baker and I. Matthews. "Lucas-Kanade 20 Years On: A Unifying Framework". *Int'l. Journal of Computer Vision,* 56(3): 221–255, March, 2004.

[3] A. Bakhtari, B. Benhabib. "An active vision system for multitarget surveillance in dynamic environments," *IEEE Trans. on System, Man, and Cybernetics, Part B*, 37(1): 190–198, 2007.

[4] S. Birchfield, "Elliptical Head Tracking using Intensity Gradients and Color Histograms", in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 232–237, 1998.

[5] G.R. Bradski, "Computer Vision Face Tracking as a Component of a Perceptual User Interface," in *Proc. of the IEEE Workshop Applications of Computer Vision*, pp. 214–219, 1998.

[6] Y. Cai, N. Freitas, and J. Little, "Robust Visual Tracking for Multiple Targets," in *Proc. of 6th European Conf. on Computer Vision*, pp. 893–908, 2006.

[7] R. T. Collins, Y. Liu and M. Leordeanu, "On-line Selection of Discriminative Tracking Features." *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10): 1631–1643, 2005.

[8] R. T. Collins, X. Zhou, and S. K. Teh, "An Open Source Tracking Testbed and Evaluation Web Site", in *IEEE Int'l Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005)*.

[9] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based Object Tracking," *IEEE Trans. Pattern Analysis Machine Intelligence*, 25(5): 564–577, 2003.

[10] A. Dore, A. Beoldo and C. Regazzoni. "Multiple cue adaptive tracking of deformable objects with particle filter". in *Proc. of Int. Conf. on Image Processing*, pp. 237–240, 2008.

[11] M. Fischler and R. Bolles, "Random Sample Consensus," *Communications of the ACM*, 24(6):381–395, 1981.

[12] S. Geman, D. Geman. "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images", *Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[13] N. Gordon, D. Salmond, and A. Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Proceedings of the IEEE*, 140(2): 107–113, 1993.

[14] C. Harris and M.J. Stephens. "A combined corner and edge detector", In *Proc. of Alvey Vision Conference*, pp. 147–152, 1988.

[15] M. Hutter. "Distribution of Mutual Information", *Advances in Neural Information Processing Systems*, 14:399–406, 2002.

[16] M. Isard and A. Blake, "Condensation - conditional density propagation for tracking," *Int'l Journal of Computer Vision*, 29(1): 2–28, 1998.

[17] M. Isard and A. Blake, "ICONDENSATION: unifying low-level and high-level tracking in a stochastic framework", in *Proc. of 5th European Conf. on Computer Vision*, Vol. I, pp. 893–908, 1998.

[18] B. Jhne, H. Scharr, and S. Krkel, "Principles of filter design", in *Handbook of Computer Vision and Applications*, B. Jhne, H. Hauecker, and P. Geiler, Eds. New York: Academic, 1999, vol. 2, pp. 125–151.

TABLE I

QUANTITATIVE PERFORMANCE EVALUATION OF DIFFERENT TRACKERS: TRACKING SUCCESS RATES, TRACKING TIME, AND TRACKING ACCURACY.

(a) EgTest01.

Tracker	Mean-shift	Particle filtering	Variance ratio	Peak diff	MS-embedded	AdaptiveMF	TheProposed
Success rate (%)	$17.58^{(7)}$	$99.4^{(5)}$	$29.12^{(6)}$	$100^{(1)}$	$99.45^{(4)}$	$99.56^{(3)}$	$100^{(1)}$
Time/frame (ms)	$31^{(1)}$	$96^{(7)}$	$43^{(4)}$	$54^{(5)}$	$52^{(6)}$	$45^{(2)}$	$39^{(3)}$
OL-BB (%)	$65.50^{(7)}$	$66.76^{(4)}$	$76.87^{(1)}$	$61.76^{(4)}$	$66.62^{(3)}$	$61.62^{(6)}$	$71.38^{(2)}$
OL-BM (%)	$66.26^{(3)}$	$53.67^{(7)}$	$61.30^{(5)}$	$57.76^{(6)}$	$66.18^{(4)}$	$68.38^{(2)}$	$69.62^{(1)}$

(b) EgTest02.

Tracker	Mean-shift	Particle filtering	Variance ratio	Peak diff	MS-embedded	AdaptiveMF	TheProposed
Success rate (%)	$39.23^{(3)}$	$21.01^{(7)}$	$27.69^{(5)}$	$30.77^{(4)}$	$25.32^{(6)}$	$100^{(1)}$	$100^{(1)}$
Time/frame (ms)	$32^{(1)}$	$89^{(7)}$	$46^{(4)}$	$61^{(6)}$	$49^{(5)}$	$43^{(3)}$	$42^{(2)}$
OL-BB (%)	$91.09^{(2)}$	$86.22^{(6)}$	$85.19^{(7)}$	$90.54^{(4)}$	$90.11^{(5)}$	$93.32^{(1)}$	$90.86^{(3)}$
OL-BM (%)	$74.69^{(1)}$	$73.57^{(3)}$	$73.32^{(4)}$	$65.91^{(7)}$	$69.56^{(6)}$	$72.70^{(5)}$	$73.89^{(2)}$

(c) EgTest03.

Tracker	Mean-shift	Particle filtering	Variance ratio	Peak diff	MS-embedded	AdaptiveMF	TheProposed
Success rate (%)	$20.62^{(3)}$	$11.9^{(7)}$	$12.06^{(4)}$	$12.06^{(4)}$	$12.06^{(4)}$	$23.23^{(2)}$	$24.12^{(1)}$
Time/frame (ms)	$29^{(1)}$	$92^{(7)}$	$43^{(4)}$	$52^{(6)}$	$46^{(5)}$	$39^{(2)}$	$41^{(3)}$
OL-BB (%)	$86.96^{(6)}$	$85.43^{(7)}$	$93.74^{(1)}$	$92.27^{(2)}$	$90.75^{(4)}$	$88.66^{(5)}$	$91.74^{(3)}$
OL-BM (%)	$66.65^{(7)}$	$67.04^{(6)}$	$70.79^{(1)}$	$67.20^{(5)}$	$70.17^{(2)}$	$69.37^{(3)}$	$68.96^{(4)}$

(d) EgTest04.

Tracker	Mean-shift	Particle filtering	Variance ratio	Peak diff	MS-embedded	AdaptiveMF	TheProposed
Success rate (%)	$9.84^{(5)}$	$26.01^{(4)}$	$9.84^{(5)}$	$3.83^{(7)}$	$27.62^{(3)}$	$40.1^{(2)}$	$42.62^{(1)}$
Time/frame (ms)	$32^{(1)}$	$91^{(7)}$	$44^{(4)}$	$48^{(6)}$	$41^{(5)}$	$43^{(2)}$	$43^{(2)}$
OL-BB (%)	$66.78^{(4)}$	$66.56^{(5)}$	$66.03^{(6)}$	$63.60^{(7)}$	$68.43^{(3)}$	$69.52^{(2)}$	$69.98^{(1)}$
OL-BM (%)	$59.70^{(6)}$	$63.68^{(4)}$	$66.74^{(1)}$	$66.42^{(2)}$	$60.17^{(5)}$	$56.34^{(7)}$	$64.33^{(3)}$

(e) EgTest05.

Tracker	Mean-shift	Particle filtering	Variance ratio	Peak diff	MS-embedded	AdaptiveMF	TheProposed
Success rate (%)	$13.64^{(4)}$	$15.64^{(3)}$	$13.64^{(4)}$	$13.64^{(4)}$	$13.64^{(4)}$	$15.81^{(2)}$	$17.96^{(1)}$
Time/frame (ms)	$31^{(1)}$	$90^{(7)}$	$46^{(2)}$	$48^{(5)}$	$49^{(6)}$	$47^{(3)}$	$56^{(4)}$
OL-BB (%)	$94.58^{(1)}$	$85.34^{(6)}$	$86.46^{(5)}$	$86.98^{(4)}$	$89.46^{(3)}$	$72.65^{(7)}$	$92.62^{(2)}$
OL-BM (%)	$84.02^{(2)}$	$79.31^{(4)}$	$85.12^{(1)}$	$69.90^{(5)}$	$64.32^{(7)}$	$64.45^{(6)}$	$80.13^{(3)}$

(f) Redteam.

Tracker	Mean-shift	Particle filtering	Variance ratio	Peak diff	MS-embedded	AdaptiveMF	TheProposed
Success rate (%)	$84.29^{(7)}$	$100^{(1)}$	$100^{(1)}$	$98.43^{(6)}$	$100^{(1)}$	$100^{(1)}$	$100^{(1)}$
Time/frame (ms)	$26^{(1)}$	$89^{(7)}$	$37^{(4)}$	$43^{(6)}$	$40^{(5)}$	$37^{(3)}$	$31^{(2)}$
OL-BB (%)	$68.37^{(7)}$	$72.56^{(4)}$	$73.24^{(2)}$	$72.38^{(5)}$	$70^{(6)}$	$72.61^{(3)}$	$73.28^{(1)}$
OL-BM (%)	$75.05^{(4)}$	$58.22^{(6)}$	$75.30^{(3)}$	$78.54^{(1)}$	$54.65^{(7)}$	$55.58^{(5)}$	$75.62^{(2)}$

[19] Z. Khan, T. Balch, and F. Dellaert. "An MCMC-based particle filter for tracking multiple interacting targets". in *Proc. of 5th European Conf. on Computer Vision*, Vol. I, pp. 893–908, 2004.

[20] J. Kwon and K.Lee. "Tracking of Abrupt Motion Using Wang-Landau Monte Carlo Estimation". in *Proc. of 7th European Conf. on Computer Vision*, pp. 387–400, 2008.

[21] A.P. Leung and S. Gong, "Online Feature Selection Using Mutual Information for Real-Time Multi-View Object Tracking," in *ICCV Workshop on AMFG*, 2005.

[22] D. G. Lowe, "Object recognition from local scale-invariant features", In *Proc. of the IEEE Int. Conf. on Computer Vision*, pp. 1150–1157, 1999.

[23] B. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision." In *Proc. of the Int'l. Joint Conf. on Artificial Intelligence.* pp. 674–679, 1981.

[24] J. MacCormick and A. Blake. "A probabilistic exclusion principle for tracking multiple objects." In *Proc. IEEE Int'l Conf. on Computer Vision*, pp. 572–578, 1999.

[25] E. Maggio and A. Cavallaro. "Hybrid Particle Filter and Mean Shift tracker with adaptive transition model." in *Proc. of Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 221–224, 2005.

[26] P. Perez, C. Hue, J. Vermaak, M. Gangnet, "Color-Based Probabilistic Tracking," in *Proc. of of 4th European Conf. on Computer Vision*, Vol. I, pp. 661–675, 2002.

[27] M.K. Pitt and N. Shephard. "Filtering via simulation: Auxiliary particle filters", *Journal of the American Statistical Association*, 94(446): 590–599, 1999.

[28] F. Porikli. "Integral histogram: a fast way to extract histograms in cartesian spaces," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 1, pp. 829–836, June 2005.

[29] C. Shan, T. Tan, and Y. Wei, "Real-time hand tracking using a mean shift embedded particle filter", *Pattern Recognition*, 40(7): 1958–1970, 2007.

[30] J. Sullivan and J. Rittscher, "Guiding Random Particles by Deterministic Search," *Proc. of Eighth IEEE Int'l Conf. on Computer Vision*, vol. I, pp. 323–330, 2001.

[31] M. Swain and D. Ballard, "Color Indexing," *Int'l. Journal of Computer Vision*, 7(1): 11–32, 1991.

[32] P.H.S. Torr and A. Zisserman, "MLESAC: a new robust estimator with application to estimating image geometry," *Comput. Vision and Image Understand*, 78(1): 138–156, 2000.

[33] P. A. Viola, M. J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Vol.I, pp. 511–518, 2001.

[34] P. Viola, M.J. Jones, and D. Snow. "Detecting pedestrians using patterns of motion and appearance," *Int'l. Journal of Computer Vision*, 63(2): 153–161, 2005.

[35] J. Wang and Y. Yagi, "Discriminative mean shift tracking with auxiliary particles", *Proc. 8th Asian Conference on Computer Vision*, pp. 576–585, 2007.

[36] J. Wang and Y. Yagi, "Integrating color and shape-texture features for adaptive real-time tracking", *IEEE Trans. on Image Processing*, 17(2): 235–240, 2008.

[37] J. Wang and Y. Yagi, "Switching local and covariance matching for efficient object tracking", *Proc. of the 19th IEEE Int'l Conf. on Pattern Recognition*, pp. 1–4, 2008.

[38] J. Xue, N. Zheng, J. Geng, and X. Zhong. "Tracking multiple visual targets via particle-based belief propagation", *IEEE Trans. on System, Man, and Cybernetics, Part B*, 38(1): 196–209, 2008.

[39] L.Yuan, H. Ai, T. Yamashita, S. Lao, and M. Kawade. "Tracking in

Low Frame Rate Video: A Cascade Particle Filter with Discriminative Observers of Different Life spans", *IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[40] X. Zhang, W. Hu, S. Maybank, X. Li, and M. Zhu. "Sequential Particle Swarm Optimization for Visual Tracking", *IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[41] S. Zhou, R. Chellappa, B. Moghaddam, "Visual Tracking and Recognition Using Appearance-adaptive Models in Particles Filters", *IEEE Trans. on Image Processing*, 13(11): 1434–1456, 2004.

Junqiu Wang received the B.E. and M.E. from Beijing Institute of Technology, China, in 1992 and 1995 respectively. He was awarded the Ph.D degree by Peking University in January 2006, China. He joined the Institute of Scientific and Industrial Research, Osaka University, Japan. His research interests are in image processing, computer vision and robotics, including visual tracking, content-based image retrieval, segmentation, active vision, and vision-based localization.

Yasushi Yagi is a Professor at the Institute of Scientific and Industrial Research, Osaka university, Ibaraki, Japan. He received B. E. and M. E. degrees in control engineering, 1983 and 1985, respectively and the Ph. D. degree in 1991, from Osaka University. In 1985, he jointed the Product Development Laboratory, the Mitsubishi Electric Corporation, where he was working on robotics and inspections. In 1990, he was Research Associate of Information and Computer Science, Faculty of Engineering Science, Osaka University. From 1993 to 1996, he was Lecturer of Systems Engineering, Faculty of Engineering Science, Osaka University. From 1996 to 2003, he was an Associate Professor of Systems and Human Science, Graduate School of Engineering Science, Osaka University. From June 1995 to March 1996, he was an academic visitor of Department of Engineering Science, University of Oxford. Computer vision and robotics are his research subjects. and a visiting associate professor at University of Picardie Jules Verne in 2002. He is an associate editor-in-chief of IPSJ Transactions on Computer Vision and Image Media. He is a program chair of Asian Conference on Computer Vision 2007 and a general chair of Asian Conference on Computer Vision 2009. His interests are in the fields of computer vision, image processing, mobile robot and medial imaging.