Kernel-Based Object Tracking via Particle Filter and Mean Shift Algorithm

Y.S. Chia W.Y. Kow W.L. Khong A. Kiring K.T.K. Teo
Modelling, Simulation and Computing Laboratory
School of Engineering and Information Technology
Universiti Malaysia Sabah
Kota Kinabalu, Malaysia
msclab@ums.edu.my ktkteo@ieee.org

Abstract — **One of the critical tasks in object tracking is the tracking of fast-moving object in random motion, especially in the field of machine vision applications. An approach towards the hybrid of particle filter (PF) and mean shift (MS) algorithm in visual tracking is proposed. In this proposed system, complete occlusion and random movement of object can be handled due to its ability in predicting the object location with adaptive motion model. In addition, the PF is capable to maintain multiple hypotheses to handle clutters in background and temporary failure. However PF requires a large number of particles to approximate the true posterior of the target dynamics. Therefore, MS algorithm is applied to the sampling process of the PF to move these particles in gradient ascent direction. Consequently a small sample size will be sufficient to represent the system dynamics accurately. The proposed approach is aimed to track the moving object in random directions under varying conditions with acceptable computational time.**

Keywords – kernel-based; object tracking; particle filter; mean shift

I. INTRODUCTION

In general, object tracking is a challenging problem due to the abrupt object motion, varying appearance of the object and background, complete occlusions, scene illumination changes, and camera motion. There are two major categories in a typical object tracker: Target estimation and localization, and the filtering and data association. The first category is a bottom-up process dealing with the appearance changes of the target; the latter is a top-down process dealing with the dynamics of the tracked object [1].

Reference [2] reported that there are several categories of object tracking, such as point tracking, kernel tracking and silhouette tracking. Kernel tracking is performed by computing the motion of target object with primitive object region representation. Due to the computational cost of brute force search is high, more efficient approaches have been proposed to limit the object search within the neighbourhood of its previous position [3].

Reference [4] proposed the mean shift (MS) applications. The MS tracker maximizes the appearance similarity iteratively by comparing the histograms of the target candidate and the hypothesized object model. Histogram similarity is defined in terms of the Bhattacharya coefficient. The coefficient is increased at each iteration in the computation of MS vector until convergence is achieved [5]. However, kernel tracking is limited to the condition that at least some parts of the object reside in the region of the previous position of the target. Reference [6] suggested that Kalman filter (KF) or particle filter (PF) approach is used in conjunction with MS tracker to enhance the prediction of the target object location. Reference [7] proposed a nonlinear dynamic model where PF was employed for state estimation.

PF approach is able to deal with nonlinear system and it is not restricted to the Gaussian model as required in the conventional KF approach. However PF requires a large number of particles to represent the true posterior density accurately [8]. The common problem in PF is the degeneracy problem. It is necessary to resample the particles according to their associated weights after each iteration [9]. This step discards particles that have negligible weights and concentrates on the particles with larger weight. Nevertheless, the resampling process launches another problem namely sample impoverishment. Therefore MS is integrated into the PF sampling step, to move these particles in gradient ascent direction until they converge to their neighboring local maxima [10]. The hybrid of MS and PF tracker is expected to solve these problems with smaller number of particles than conventional PF approach.

II. PARTICLE FILTER TRACKING FRAMEWORK

A. Particle Filter Object Tracking

PF is a hypothesis tracker based on the recursive Bayesian filter with Monte Carlo sampling. The prediction step and update step are required to estimate the state of dynamics from observations. The implementation is accomplished by two fundamental models. The first one is a model predicting the evolution of the state over time (state transition model) and the latter is a model measuring the observations and updating the measurement to the current state (observation model).

B. State Transition Model

The state transition model is associated with a process noise to propagate the particles from the previous frame into the current frame. It is implemented with an adaptive model of the average state velocity in the previous successfully tracked *k* frames, as illustrated in (1). It is capable to deal with the rapid changes of the target direction and speed in a

978-1-4577-2152-6/11/$26.00 ©2011 IEEE

better way as compared to the fixed velocity state transition model.

$$V_t = \frac{1}{k}\sum_{m=t-k}^{t}[x_m - x_{m-1}] \tag{1}$$

The state transition model is defined in (2),

$$x_t = x_{t-1} + V_t + u_t \tag{2}$$

where u_t is a zero mean Gaussian process noise.

C. Observation Representation Weighted by Kernel

An appropriate kernel is used to introduce a spatially-smooth function to reduce the search on small neighborhood region. The convex and monotonically decreasing Epanechnikov kernel has been selected to mask the target's density estimate spatially, as defined by (3),

$$K_E(x) = \begin{cases} \frac{1}{2C_d}(d+2)(1-\|x\|^2) & if \|x\| < 1 \\ 0 \quad otherwise \end{cases} \tag{3}$$

where x is the normalized pixel coordinates within the target region, C_d is the volume of the unit d–dimensional sphere and $\|x\|^2$ is the squared Euclidean distance of each pixel from the center of the target region.

Since it is dealing with a 2-dimensional image space for video object tracking purpose, $d = 2$ and $C_d = \pi$, the kernel function in (3) can be expressed as in (4).

$$K_E(x) = \frac{2}{\pi}(1-\|x\|^2) \tag{4}$$

The rationale of using the kernel as a weighted mask is to assign smaller weights to the pixels farther away from the center of the target, since those pixels are often affected by occlusion or interference from the background.

D. The Color Feature Model

The Hue-Saturation-Value (HSV) color space is selected because it is less dependent to illumination. Color histogram is used as the representation of the discrete densities of the objects' color. Using the Epanechnikov kernel $K_E(x)$ in (4) and assuming that the kernel bandwidth as 1, the influence of different target dimensions is eliminated. The probability of color bin u in the model is defined by (5),

$$q_u^{color} = C\sum_{i=1}^{n} k(\|x_i\|^2)\delta(b(x_i)-u) \tag{5}$$

On the other hand, the probability of color bin u in the target candidate is shown in (6),

$$p_u^{color}(x) = C\sum_{i=1}^{n} k\left(\left\|\frac{x-x_i}{h}\right\|\right)^2 \delta(b(x_i)-u) \tag{6}$$

where x_i denotes the center of model or target candidate with $i = 1 \ldots n$, $b(x_i)$ denotes the color bin in the quantized color space at x_i, C is the normalization constant and δ is the Kronecker delta function.

E. The Edge Orientation Histogram

The color feature is generally reliable in most of the tracking tasks. However the color feature performs poorly when the background is similar to target object's color. Hence, edge feature is added to describe the characteristics of the local contour [11].

The edge orientation histogram (EOH) is used due to its simplicity. It is built after all the edge points of the object are detected by Sobel edge detection algorithm. In Sobel, the edges are detected by calculating the horizontal and vertical derivative approximations, which can be obtained by convolving the grayscale intensity image I with two 3×3 kernels, as shown in (7).

$$G_x(x,y) = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & +2 & -1 \end{bmatrix} * I(x,y)\ ,$$

$$G_y(x,y) = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I(x,y) \tag{7}$$

The strength and the orientation of the edges are given by (8) and (9) respectively. The EOH within the tracking window region is constructed by quantizing the angles of the edges into separated intervals.

$$S(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{8}$$

$$\theta = \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \tag{9}$$

By using (4), the edge probability density function (PDF) of bin u in the edge model is defined as in (10).

$$q_u^{edge} = C\sum_{i=1}^{n} k\left(\|x_i\|^2\right)\delta\left(b(x_i)-u\right) \tag{10}$$

On the other hand, the edge PDF of bin u for the target candidate is given by (11),

$$p_u^{edge}(x) = C\sum_{i=1}^{n} k\left(\left\|\frac{x - x_i}{h}\right\|^2\right)\delta(b(x_i) - u) \qquad (11)$$

where x_i denotes the center of model or target candidate with $i = 1 \ldots n$, $b(x_i)$ denotes the edge orientation bin in the quantized edge orientation space at x_i, C is the normalization constant, and δ is the Kronecker delta function.

III. Mean Shift Tracking Framework

A. Mean Shift Object Tracking

The MS algorithm is introduced to optimize the smooth similarity function and obtain the direction of the target's movement. The derivative of (4) is shown in (12).

$$g(x) = -K_E'(x) = \begin{cases} \dfrac{2}{\pi} & if\|x\| < 1 \\ 0 & otherwise \end{cases} \qquad (12)$$

Epanechnikov profile is selected as the kernel mask instead of other kernel profiles such as Gaussian due to its derivative is a constant as shown in (10), which simplifies the use of the gradient optimization method. The similarity distance measured between the target and current distribution is the indication to find the target object in the current frame.

B. Prediction of Target Movement using Mean Shift Calculation

The similarity between the current target distribution and the next frame candidate distribution need to be maximized to predict the direction of the target's movement. The distance evaluation metric between the two discrete distributions is defined in (13).

$$d(y) = \sqrt{1 - \rho[p(y), q]} \qquad (13)$$

The Bhattacharyya coefficient is selected to determine the amount of relative similarity, which is shown in (14),

$$\rho(p(y), q) = \sum_{u=1}^{m} \sqrt{p_u(y) q_u} \qquad (14)$$

where ρ is the cosine of the angle between m-dimensional unit vectors. Using Taylor expansion around the value $p_u(x_0)$, the linear approximation of the Bhattacharyya coefficient can be obtained as shown in (15).

$$\rho(p(y), q) \approx \frac{1}{2}\sum_{u=1}^{m} \sqrt{p_u(x_0) q_u} + \frac{1}{2}\sum_{u=1}^{m} \sqrt{\frac{q_u}{p_u(x_0)}} \qquad (15)$$

By substituting with (6), (15) can be expressed as in (16),

$$\rho(p(y), q) \approx \frac{1}{2}\sum_{u=1}^{m} \sqrt{p_u(x_0) q_u} + \frac{1}{2}\sum_{i=1}^{n} w_i k\left(\left\|\frac{x - x_i}{h}\right\|^2\right) \qquad (16)$$

where, $w_i = \sum_{u=1}^{m} \sqrt{\dfrac{q_u}{p_u(x_o)}}\delta(b(x_i) - u)$ (17)

The terms q_u and p_u in (17) are the values of the target and candidate histograms corresponding to pixel x_i of the candidate object. This mapping is to visualize the changes of the target object over a period and the corresponding distribution of weights for it.

From (16), the first term is independent of x, in order to minimize the distance in (13), the second term has to be maximized. By employing the MS calculation, the center of the target candidate is successively shifted according to (18),

$$x_{new} = \frac{\sum_{i=1}^{n} x_i w_i g\left(\left\|\frac{x_o - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} w_i g\left(\left\|\frac{x_o - x_i}{h}\right\|^2\right)} \qquad (18)$$

where x_0 is the current location of the candidate and $g(x)$ is the derivative function $k(x)$. According to (12), the derivative of Epanechnikov kernel is a constant and hence (18) can be reduced to a weighted distance average as shown in (19).

$$x_{new} = \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} \qquad (19)$$

IV. Proposed Mean Shift Embedded Particle Filter Tracking Approach

A. Design of Mean Shift Embedded Particle Filter Object Tracking System

The task of tracking objects which are abruptly changing their moving direction is difficult to be accomplished by MS alone. Although PF can overcome this circumstance, its computational cost is high since the dynamic model need to take the worst scenario into consideration. Therefore the combination of both PF and MS approaches to improve the object tracking system is proposed. MS is incorporated into the sampling step of the PF to reduce the number of particles required to track an object.

In the proposed system, the MS optimization is applied to each particle after all the particles are propagated by the state transition model $p(x_t|x_{t-1})$. MS can move each of the samples in the gradient ascent direction according to the observation likelihood until each of the particles converge to their neighboring local maxima. The new resulting particle set represents the modes of the distribution, results in fewer particles are needed to maintain the multi-modal distribution.

The observation distribution model can be characterized by Gaussian distribution as shown in (20), with d as the Bhattacharyya similarity distance in (13),

$$L(z|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{d^2}{2\sigma^2}\right) \tag{20}$$

where σ^2 is the variance of Gaussian distribution.

By assuming that the observation likelihood from the features of color and edge are statistically independent from each other, the entire multi-cue likelihood given the state x_t could be combined as shown in (21).

$$L(z|x) = L_{color}\left(Z_{color}|x_t\right).L_{edge}\left(Z_{edge}|x_t\right) \tag{21}$$

The enhancement of object tracking in PF by MS algorithm is implemented as follows:

```
1. In the first frame, the region of target is initialized.
2. In the second frame, the sample set of N particles is randomly generated
   based on the uniform distribution in the neighborhood of the target model
   location obtained from the previous frame.
3. In the third frame onwards, given the sample set $\{x_{t-1}^i, w_{t-1}^i\}$.
   (a) Sampling:
       for $i = 1, \ldots, N$
           Each sample $x_{t+1}^i$ is propagated by the state transition model $p\left(x_k^i | x_{k-1}^i\right)$.
       end for
   (b) MS:
       for $i = 1, \ldots, N$
           Generate modified samples $x_{t+1}^{'(i)}$ by applying MS independently on each sample.
       end for
   (c) Weights calculation:
       for $i = 1, \ldots, N$
         Calculate the weights according to (21).
         Normalize the weights as $\hat{w}_{t+1}^i = \frac{w_{t+1}^i}{\sum_{j=1}^{N} w_{t+1}^j}$
       end for
   (d) Output:
       Compute the posterior mean or weighted average $E\left[x_{t+1}|z_{t+1}\right]$ so that the target position in current
       frame, $\hat{x}_{t+1} = E\left[x_{t+1}|z_{t+1}\right] = \sum_{i=1}^{N} \hat{w}_{t+1}^i x_{t+1}^{'(i)}$.
   (e) Resampling
       Obtain N new random samples approximately by importance resampling.
```

B. Occlusion Detection and Recovery Scheme

In most of the object tracking situations, the target object is frequently occluded by other objects. Therefore the occlusion detection and recovery scheme is introduced with adaptive number of particles and adaptive state transition model. Firstly, the object occlusion can be detected through the weight normalization factor, which is the sum of the weights of the corresponding particles set, as defined in (22).

$$\hat{w}_t = \sum_{i=1}^{N} w_t^i \tag{22}$$

When the target model is not appearing in the state space, the total weights for all the particles would be insignificant. Hence occlusion can be detected when the normalization factor is smaller than the experimentally determined threshold value.

Initially the normal state transition model is defined as (2), when occlusion is detected, normal state is switched to occlusion state transition model to propagate the particles according to a uniform distribution around x_{t-1}, as shown in (23).

$$x_t = x_{t-1} + U_t \tag{23}$$

where U_t denotes the uniform distributed random integer around x_{t-1} and the interval of the distribution is determined empirically. This uniform distribution covers larger searching area to predict the location of the target when it reappears. Besides the switching of the transition model, the number of the particles is increased to include different possibilities especially when the target reappears after the occlusion is relieved.

V. Experimental Results

The performance of the MS embedded PF tracker is evaluated on selected videos to exemplify different conditions. The color histogram is built based on 4×4×2 bins, with less sensitivity to value (V) channel, to make the tracking algorithm less sensitive to lighting condition. Due to the angles of the edges is in the range of [–90, 90], the EOH is built in 8-bins, with 22.5° of interval. Experiments are conducted to test for its ability to handle abrupt direction and velocity changes, complete occlusion as well as overlapping with like-colored background.

A. First Experiment on Balls Juggling Video

This experiment aims to test and evaluate the performance of MS embedded PF tracker in tracking the target red ball which continuously experiences abrupt direction and velocity changes, as shown in Fig. 1. This result demonstrated successive tracking of the juggled balls by MS embedded PF tracker. The success of this tracker in dealing with abrupt direction and velocity changes of fast moving object supports the utilization of the tracker in tracking random motion object.

The effect of the number of particles on the performance of the PF-based tracker is illustrated in Fig. 2. Higher Bhattacharyya coefficient indicates that the estimated target distribution is matched with the target model distribution with higher similarity. For fair comparison, the PF tracker employs the same state transition model and observation model as the MS embedded PF tracker. It can be observed that the PF tracker alone can only achieve sufficiently high coefficient when the number of particles is around 150, as compared to the MS embedded PF tracker where its sample size is only 20. Therefore, the result shows that MS applied on PF approach can greatly reduce the number of particles required for robust tracking compared to the conventional PF approach.

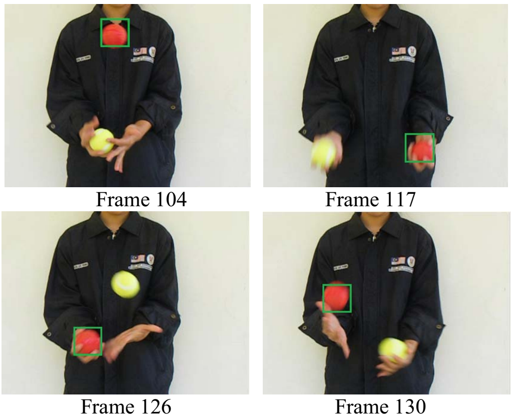

Frame 104
Frame 117
Frame 126
Frame 130

Figure 1. The tracking result of MS embedded PF tracker in the two balls juggling video.

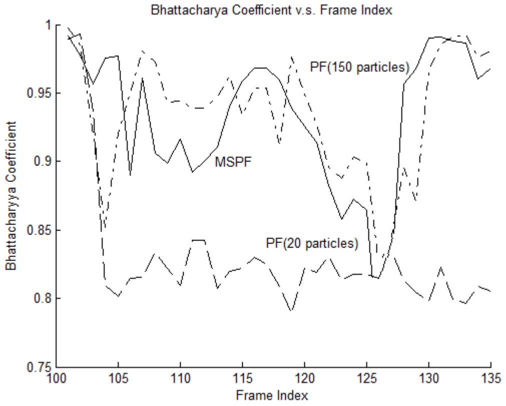


Figure 2. Comparison of different number of particles on PF tracker with the MS embedded PF tracker.

B. Second Experiment on Occlusion

This experiment aims to test and evaluate the MS embedded PF tracker performance when it experiences complete occlusion, as shown in Fig. 3.

Fig. 3 shows that the target object can be kept in track even it undergoes complete occlusion. Before occlusion occurs, the tracking system is in normal state with nominal number of particles as shown in frame 231. When the complete occlusion occurred in frame 241, the tracking system is switched to occlusion state and the number of particles is increased with dispersion at larger searching area. Once the object reappeared in frame 251, it is successfully recaptured. The recovery of occlusion is indicated by the normalization factor and the tracking system switches back to the normal state with nominal number of particles in frame 256.

C. Third Experiment on Like-colored Background Video

The experiment is conducted by using a video where part of the background is like-colored with the target model. The tracking results of MS embedded PF tracker without employing edge feature is shown in Fig. 4. On the other hand, Fig. 5 shows the tracking results of the MS embedded PF tracker with edge feature.

From Fig. 4, the target is lost track when it passes through the background that has similar color with the target.

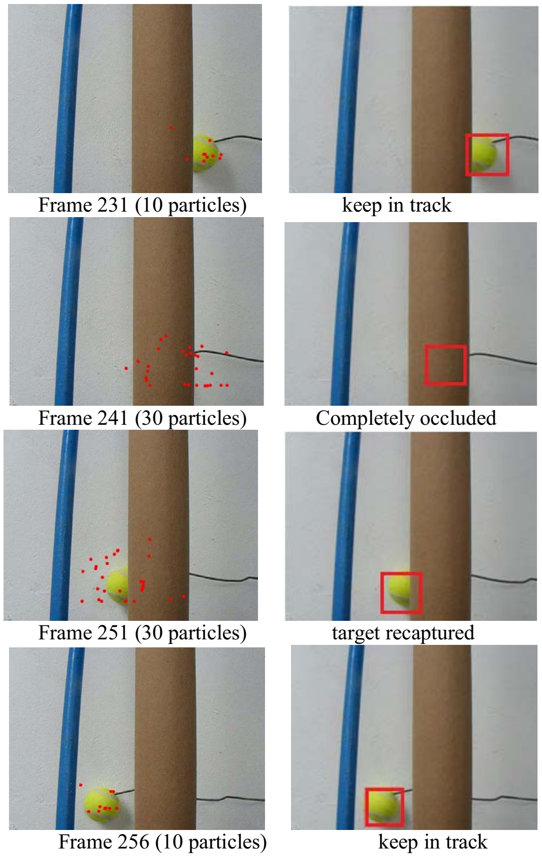
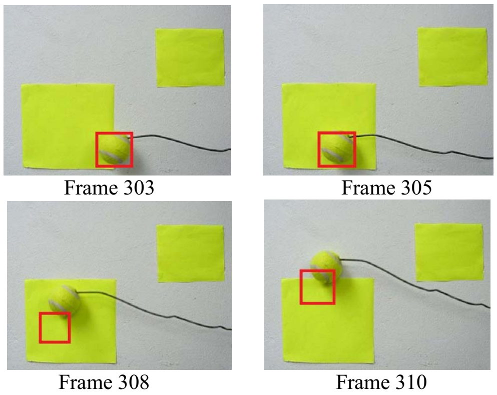

Frame 231 (10 particles)
keep in track
Frame 241 (30 particles)
Completely occluded
Frame 251 (30 particles)
target recaptured
Frame 256 (10 particles)
keep in track

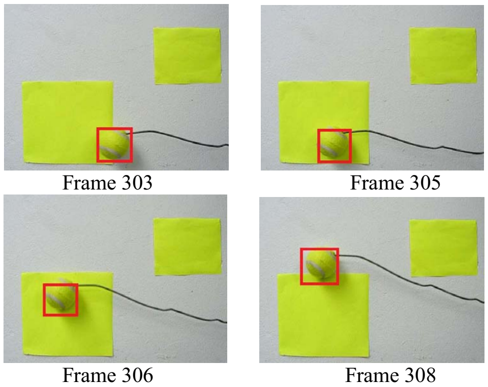
Figure 3. Occlusion detection and recovery process by MS embedded PF tracker in tennis ball occlusion video.



Frame 303 Frame 305

Frame 308 Frame 310

Figure 4. The tracking result of MS embedded PF tracker without edge feature in the background overlapping video.



Frame 303 Frame 305

Frame 306 Frame 308

Figure 5. The tracking result of MS embedded PF tracker with edge feature in the background overlapping video.

Due to the color pixels of the background region are grouped in the same bins as the target model, the color information alone is not distinct enough to represent the target feature. From Fig. 5, the tennis ball does not lost track when it resides within the like-colored background region. This is due to the edge information which is used in conjunction with the color information in the algorithm can provide better discriminative description of the target model compared to the background.

VI. Conclusion

The developed MS embedded PF tracker produces satisfactory tracking performance and being computationally inexpensive. The implementation of the MS embedded PF object tracker is demonstrated with the use of color and edge information as the feature representations of the target. This tracker is capable to track abrupt direction and velocity changing object. The ability of maintaining multiple hypotheses on PF combined with the gradient optimization property of MS enable the tracker able to deal with occlusion and clutter situations. Furthermore, experiments show that MS applied on PF algorithm can greatly reduce the number of particles required for robust tracking compared to conventional PF approach.

Currently, the tracking performance in the developed tracking system could be affected if the target accelerates at the moment it changes the moving direction as the motion behaviour would deviate from the average velocity model. Although the current first order state transition model is simpler, it could be improved with a second order auto-regressive dynamic model. The improvement is due to the capability to immediately deal with the extreme motion behaviour of the target at the moment it changes the direction.

Acknowledgment

The authors would like to acknowledge the financial assistance from the University Postgraduate Research Scholarship Scheme (PGD) awarded by Ministry of Science, Technology and Innovation of Malaysia (MOSTI).

References

[1] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 142-149, 2000.

[2] A. Yilmaz, O. Javed, and S. Mubarak, “Object tracking: A survey,” ACM Computing Surveys, vol. 38, No. 4, Article 13, 2006.

[3] H. Schweitzer, J.W. Bell, and F. Wu, “Very fast template matching,” European Conference on Computer Vision, pp. 358-372, 2002.

[4] D. Comaniciu, and P. Meer, “Mean shift analysis and applications,” IEEE International Conference on Computer Vision, vol. 2, pp. 1197-1203, 1999.

[5] D. Comaniciu, “Bayesian kernel tracking,” Annual Conference of the German Society for Pattern Recognition, pp. 438-445, 2002.

[6] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” IEEE Trans. Pattern Analysis Machine Intell., vol. 25, pp. 564–575, 2003.

[7] M. Isard, and A. Blake, “Condensation-conditional density propagation for visual tracking,” Int. J. Comput. Vision, vol. 29, pp. 5-28, 1998.

[8] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.J. Nordlund, “Particle filers for positioning, navigation, and tracking,” Signal Processing IEEE, vol. 50, pp. 425-437, 2002.

[9] W.L. Khong, W.Y. Kow, F. Wong, I. Saad, and K.T.K. Teo, “Enhancement of particle filter approach for vehicle tracking via adaptive resampling algorithm,” Computational Intelligence, Communication Systems and Networks, pp. 259-263, 2011.

[10] K.J. Bai, and W.M. Liu, “Improved object tracking with particle filter and mean shift,” IEEE International Conference on Automation and Logistics, pp. 431-435, 2007.

[11] W.Y. Kow, W.L. Khong, F. Wong, I. Saad, and K.T.K. Teo, “Adaptive tracking of overlapping vehicles via markov chain monte carlo with CUSUM path plot algorithm,” Computational Intelligence, Communication Systems and Networks, pp. 253-258, 2011.