The Fifteenth International Symposium on Artificial Life and Robotics 2010 (AROB 15th '10),
B-Con Plaza, Beppu,Oita, Japan, February 4-6, 2010

A color-based particle filter for multiple objects tracking in outdoor environment

Budi Sugandi, Hyoungseop Kim, Joo Kooi Tan and Seiji Ishikawa
Graduate School of Engineering, Kyushu Institute of Technology
1-1 Sensui, Tobata, Kitakyushu, Fukuoka, Japan
Tel : +81-93-884-3185; Fax :+ 81-93-861-1159
budi@kimlab.cntl.kyutech.ac.jp

Abstract: Tracking of multiple objects is more challenging than tracking a single object. Some problems arise in multiple object tracking that do not exist in single object tracking such as objects occlusion, appearing new object and disappearing already existed object, updating the occluded object, etc. In this paper, we present an approach to handle multiple objects tracking in the presence of occlusions, background clutter and appearance change. The occlusion is handled considering the predicted trajectories of the objects based on a dynamic model and likelihood measures. We propose also a target-model update conditions, ensuring the proper tracking multiple objects. The proposed method is implemented in the probabilistic framework such as particle filter in conjunction with a color feature. The particle filter has proven very successful for non-linear and non-Gaussian estimation problems. It approximates a posterior probability density of the state such as the object position by using samples or particles, where each state is denoted as the hypothetical state of the tracked object and its weight. The observation likelihood of the objects is modeled based on color histogram. The sample weight is measured based on Bhattacharya coefficient that measures the similarity between each sample's histogram with a specified target model. The algorithm can track the multiple objects in the present of occlusion and noise successfully. Some experimental results show the effectiveness of our method to track the multiple objects.

Keywords: particle filter, color feature, multiple objects, occlusion.

I. INTRODUCTION

The increasing interest in visual tracking is motivated by a huge number of promising applications that can now be tackled in real-time applications. These applications include performance analysis, surveillance, video-indexing, smart interfaces, teleconferencing and video compression. However, tracking objects can be extremely complex and time-consuming especially when it is done in outdoor environments. Here, we can mention some problems of object tracking in outdoor environments such as fake-motion background, illumination changes, shadows and presence of clutter. Some problems are also taken into account concerning whether a single or multiple objects should be tracked. Because the multiple objects tracking has more challenging problem due to the presence of objects occlusion, objects appearing and disappearing or appearance changes.

Traditionally, the tracking problem is formulated by Ristic et al. [1] as a sequential recursive estimation that is having an estimate of the probability distribution of the target in the previous frame and estimates the target distribution in the new frame using all available prior knowledge and the new information brought by the new frame. The state-space formalism where the current tracked object properties are described in an unknown state vector updated by noisy measurements is very well adapted to model the tracking. One of the methods of the sequential estimation is Kalman filter [2] which is dealing with target tracking in the probabilistic framework. But it cannot resolve the tracking problem when the model is nonlinear and non-Gaussian. The extended Kalman filter [2] can deal to this problem, but still has a problem when the nonlinearity and non-Gaussian cannot be approximated accurately.

Recently, particle filter, a numerical method that allows to find an approximate solution to the sequential estimation [3] has been proven very successful for nonlinear and non-Gaussian estimation problems [4-6]. It approximates a posterior probability density of the state such as the object position by using samples which are called particles. As for one of the particle filters, the Condensation algorithm was introduced by M. Isard et al. [4]. This algorithm has been typically used for tracking problems of moving object contours. For another particle filter, Monte Carlo filter was introduced by Kitagawa [5] and Bayesian bootstrap filter was introduced by Gordon et al. [6].

Although particle filters have been widely used in recent years, they have important drawbacks [7]. One of them is sampling impoverishment that is samples are spread around several modes pointing out the different hypotheses in the state space, but most of them may be spurious. Especially in the multiple objects tracking, the objects with higher likelihood may monopolize the samples set and objects whose samples exhibit lower likelihood have higher probability of being lost. Besides that, the multiple target tracking has another problem with occlusion, appearance change and background clutter.

In recent years, many improvements have been introduced, but there is still much ground to cover. Different approaches have been taken in order to overcome these facts. Nummiaro et al. [8] used a particle filter based on color histograms features. Histograms are robust to partial occlusions and rotations but no shape analysis is taken into account. Moreover, no multiple-target tracking is considered and complete occlusions are not handled. Perez et al. [9] proposed also a particle filter based on color histogram. They introduced interesting extension in multiple-part modeling, incorporation of background information and multiple targets tracking. Nevertheless, it requires an extremely large number of samples, since one sample contains information about the state of all targets, dramatically increasing the state dimensionality. Further, no appearance model updating is performed, what usually leads to target loss in dynamic scenes. Comaniciu et al. [10] proposed to use mean shift in order to track non-rigid object. His work has real time capabilities. However, the problem for object tracking with color occurs when the region around the object is cluttered and illumination is change. In this way, a color feature based tracking does not provide reliable performance because it fails to fully model the target especially when occlusion occurs. In another work, Comaniciu et al. [11] approach relied on gradient-based optimization and color-based histograms. In this case, no dynamic model is used therefore no occlusion can be predicted. Deutscher et al. [12] presented an interesting approach called annealing particle filter which aims to reduce the required number of samples, however, it could be inappropriate in a cluttered environment. They combine edge and intensity measures but they focused on motion analysis, and thus, no occlusion handling is explored. Some effort have been done in contour tracking [13] although it may be inappropriate, if used as the only cue, in crowded scenarios because of multiple occlusions.

Another issue of the multiple target tracking is the management of multiple tracks caused by newly appearing targets and the disappearance of already existing targets. Some of them rely on hybrid sequential state estimation. In [14], the state vector denoting all the existing targets is augmented by a discrete random variable which represents the number of existing objects in a video sequence. The particle filter developed in [15] has multiple models for the object motion, and comprises an additional discrete state component, denoting which of the motion models is active. The Bayesian Multiple-Blob Tracker (BraMBLe) [16] presented a multiple persons tracking system based on statistical appearance models. The multiple blob tracking is managed by incorporating the number of objects present in the state vector and state vector is augmented as in [14] when a new object enters the scene.

In this paper, we propose to solve some tracking problems related to the difficulties described above, such as multiple objects tracking with unknown dynamics in presence of background clutter and strong noise and in the presence of occlusion.

The remaining of this paper is organized as following. In section 2, we describe a probabilistic tracking framework. Section 3 describes the multiple objects tracking. Section 4 presents some experimental results and finally section 5 is the conclusion of the paper.

II. PROBABILISTIC TRACKING FRAMEWORK

Since the tracking problem is formulated by sequential recursive estimation, a probabilistic framework is commonly used. The computation of the expected state $\mathbf{x}_k$ given all observations to data $\mathbf{z}_{1:k}$ is called filtering. Under certain assumptions, the posterior probability distribution function (pdf) $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ can be calculated through recursive estimation,

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \propto \underbrace{\underbrace{p(\mathbf{z}_k|\mathbf{x}_k)}_{\text{likelihood}}}_{\text{updating}} \underbrace{\int \underbrace{p(\mathbf{x}_k|\mathbf{x}_{k-1})}_{\text{trans. model}} \underbrace{p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})}_{\text{prev. post}} dx_{k-1}}_{\text{prediction}} . \quad (1)$$

The pdf is projected forward according to the transition model to make a prediction. Then, it is updated in agreement with the new observation $\mathbf{z}_k$. When no assumptions are taken about the involved distributions, this problem is overcome by simulating N random samples from the posterior pdf, $\{\mathbf{x}_k^i; i = 1:N\}$. This approach is called particle filters [4-6].

In Bayesian sequential estimation, the filtering distribution can be computed according to the two step recursions, prediction and update step. The prediction step follows from marginalization and updated step is obtained through observation model. This recursion requires the specification of a dynamic model describing the state evolution, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and a model that gives the likelihood of any state in the light of the current observation, $p(\mathbf{z}_k|\mathbf{x}_k)$. The recursion is initialized with some distribution for the initial state $p(x_0)$.

The basic idea behind the particle filter is very simple. Starting with a weighted set of samples at k-1 $\{\mathbf{x}_{k-1}^i, \pi_{k-1}^i; i = 1:N\}$ approximately distributed according to $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$, new samples are generated from a suitable proposal distribution, which may depend on the previous state and the new measurements. To maintain a consistent sample, the new importance weights are set to

$$\pi_k^i = \pi_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)\, p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k|\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k})} . \quad (2)$$

The new particle set is re-sampled using normalized weights $\bar{\pi}_k^i$ as probabilities. This sample set represents the posterior at time k, $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. Then, the expectations can be approximated as

$$E\,p(\mathbf{x}_k|\mathbf{z}_{1:k}) \cong \sum_{i=1}^{N} \bar{\pi}_k^i \hat{\mathbf{x}}_k^i \ . \tag{3}$$

This implementation of the approach corresponds to the bootstrap filter as proposed in [6].

III. MULTIPLE OBJECTS TRACKING

In this paper, we proposed an algorithm based on particle filtering in conjunction with color feature. The motion of the central point of a bounding box is modeled using first-order dynamics model.

The state of the object is defined as $\mathbf{s}_k = (\mathbf{x}_k, \dot{\mathbf{x}}_k, \mathbf{w}_k, \dot{\mathbf{w}}_k, \mathbf{A}_k)$ where the components are position, speed, bounding-box size, bounding-box scale and pixel appearance, respectively. Each object associates one specific appearance model to the corresponding samples, allowing multiple objects tracking. The observations $\mathbf{z}_k$ is given by input images $\mathbf{I}_k$.

3.1. System Flow Diagram

In this paper, we want to apply a particle filter in a color model-based framework to track multiple objects in outdoor environment. Initially, the samples are drawn randomly for the first frame. The sample prediction is performed based on a system model. The weight calculation is performed based on histogram distance computed using Bhattacharya coefficient. The estimate is performed based on the sample's weight. Then resampling is performed for the next sample iteration. Target model is updated to ensure the proper tracking multiple objects. The overall working flow diagram is shown in Figure 1.

3.2 Tracking Model

We consider the motion of an object as the discrete time 2-dimensional (2D) motion with constant velocity.

The state vector at a time step k is denoted by $\mathbf{s}_k$, including position, speed, size and bounding box scale of each sample and is predicted according to

$$\begin{aligned} \hat{\mathbf{x}}_k &= \mathbf{x}_{k-1} + \dot{\mathbf{x}}_{k-1}\Delta t + \xi_{\mathbf{x}} , \\ \hat{\dot{\mathbf{x}}}_k &= \dot{\mathbf{x}}_{k-1} + \xi_{\dot{\mathbf{x}}} , \\ \hat{\mathbf{w}}_k &= \mathbf{w}_{k-1} + \dot{\mathbf{w}}_{k-i}\Delta t + \xi_{\mathbf{w}} , \\ \hat{\dot{\mathbf{w}}}_k &= \dot{\mathbf{w}}_{k-1} + \xi_{\dot{\mathbf{w}}} . \end{aligned} \tag{4}$$

The random vectors $\xi_{\mathbf{x}}, \xi_{\dot{\mathbf{x}}}, \xi_{\mathbf{w}}, \xi_{\dot{\mathbf{w}}}$ provide the system with a diversity of hypotheses.

3.3. Likelihood Function

The likelihood function computes the pdf of image features given the state. The target appearance can be represented by means of color histograms. Histograms are broadly used to represented human appearance, since they are claimed to be less sensitive than color templates to rotations in depth, the camera point of view, non-rigid targets, and partial occlusions.

To achieve robustness against mixed color, rotation and variant illumination condition, we focus on weighted color histograms to represent the target model. In this paper, the color histogram is used as the discretized color distribution. The histograms are calculated from the function $h(\boldsymbol{x}_i)$ that assign the color at location $\boldsymbol{x}_i$ to the corresponding bin. Following [8-10], we do not use the entire image as a measurement, but rather we compute the color histogram inside the image region that is specified by the state vector.

To increase the reliability of the target model, smaller weight are assigned to the pixels that are further away from region center by employing a weighting function

$$g(r) = \begin{cases} 1 - r^2 & r < 1 \\ 0 & otherwise \end{cases} , \tag{5}$$

here, r is the distance from the center of the region.

The color histogram $p_{\boldsymbol{y}} = \{p_{\boldsymbol{y}}^{(u)}\}u = 1,...,m$ at location $\boldsymbol{y}$ is calculated as

$$p_{\boldsymbol{y}}^{(u)} = f \sum_{j=1}^{I} g\left(\frac{\|\boldsymbol{y} - \boldsymbol{x}_j\|}{a}\right)\delta\left[h(\boldsymbol{x}_j) - u\right], \tag{6}$$

here respectively, I is the number of pixels in the region, $\boldsymbol{x}_j$ is the position of pixels in the region, δ is the Kronecker delta function, a is the normalization factor, and f is the scaling factor defined as

$$f = \frac{1}{\sum_{i=1}^{I} g\left(\frac{\|\boldsymbol{y} - \boldsymbol{x}_i\|}{a}\right)} , \tag{7}$$

to ensures that $\sum_{u=1}^{m} p_{\boldsymbol{y}}^{(u)} = 1$.

The similarity between two color histograms $\boldsymbol{p} = \left\{p^{(u)}\right\} u = 1,...,m$ and $\boldsymbol{q} = \left\{q^{(u)}\right\} u = 1,...,m$ is measured using Bhattacharyya distance defined as

$$d = \sqrt{1 - \rho[\boldsymbol{p}, \boldsymbol{q}]} , \tag{8}$$

where

$$\rho[\boldsymbol{p}, \boldsymbol{q}] = \sum_{u=1}^{m} \sqrt{p^{(u)} q^{(u)}} \ . \tag{9}$$

From this equation, the larger ρ shows the more similar the distributions. For two identical histograms we obtain $\rho = 1$, indicating a perfect match.

The weight $\pi^{(i)}$ of i-th state $\boldsymbol{x}^{(i)}$ is calculated as

$$\begin{aligned} \pi^{(i)} &= \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{d^2}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{\left(1 - \rho\left[\boldsymbol{p}\left(\boldsymbol{x}^{(i)}\right), \boldsymbol{q}\right]\right)}{2\sigma^2}\right) \end{aligned} . \tag{10}$$

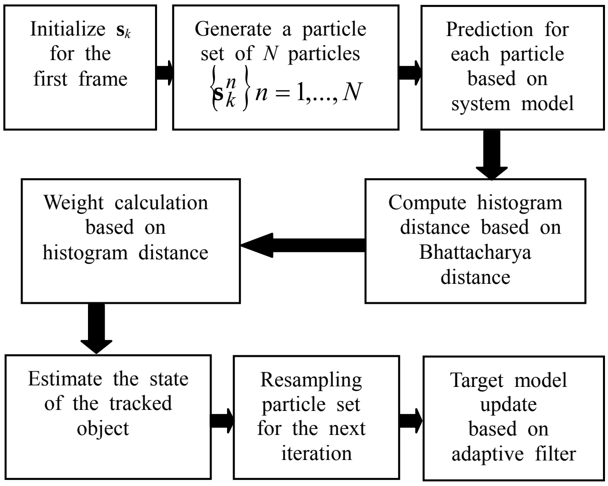

Fig. 1. Particle filter implementation flow

Where $\boldsymbol{p}(\boldsymbol{x}^{(i)})$ and $\boldsymbol{q}$ are the color histogram of sample and target, respectively. From this equation, we can see that small Bhattacharya distance corresponds to large weight. During resample step of particle filter, samples with a high weight may be chosen several times leading to identical copies, while others with relatively low weights may be ignored.

3.4 Weight Normalization

In a multiple targets tracking scenario, those targets whose samples exhibit lower likelihood are more likely to be lost, since the probability of propagating one mode is proportional to the cumulative weights of its samples. In order to avoid one target absorbing other target samples, we proposed to normalize the weights as

$$\bar{\pi}_k^{i,l} = \frac{\pi_k^{i,l}}{\sum_{i=1,j=l}^{N} \pi_k^{i,j}} \frac{1}{L}, \tag{11}$$

here L is the number of tracked targets. Each weight is normalized according to the total weight of the target's samples. Thus, all targets have the same probability of being propagated, since the addition of the weights of each target samples sums $1/L$.

3.5 State Estimation

The k-target estimates are computed according to

$$\begin{aligned}
\mathbf{x}_k &= (1-\alpha_{\mathbf{x}})(\mathbf{x}_{k-1} + \dot{\mathbf{x}}_{k-1}\Delta t) + \alpha_{\mathbf{x}}\left(L\sum_{i=1}^{N} \bar{\pi}_k^i \hat{\mathbf{x}}_k^i\right) \\
\dot{\mathbf{x}}_k &= (1-\alpha_{\dot{\mathbf{x}}})\dot{\mathbf{x}}_{k-1} + \alpha_{\dot{\mathbf{x}}}\left(\frac{\mathbf{x}_k - \mathbf{x}_{k-1}}{\Delta t}\right) \\
\mathbf{w}_k &= (1-\alpha_{\mathbf{w}})\mathbf{w}_{k-1} + \alpha_{\mathbf{w}}\left(L\sum_{i=1}^{N} \bar{\pi}_k^i \hat{\mathbf{w}}_k^i\right) \\
\dot{\mathbf{w}}_k &= (1-\alpha_{\dot{\mathbf{w}}})\dot{\mathbf{w}}_{k-1} + \alpha_{\dot{\mathbf{w}}}\left(\frac{\mathbf{w}_k - w_{k-1}}{\Delta t}\right)
\end{aligned} \tag{12}$$

where $\alpha_{\mathbf{x}}, \alpha_{\dot{\mathbf{x}}}, \alpha_{\mathbf{w}}, \alpha_{\dot{\mathbf{w}}} \in [0,1]$ denote the adaptation rates.

3.6 Target update

The target appearance must also be updated. However, this is a sensitive task. The target models are only updated when two conditions hold: (i) the target is not occluded and (ii) the likelihood of the estimated target's state suggests that the estimate is sufficiently reliable. In this case, they are updated using an adaptive filter

$$\mathbf{q}_k = (1-\alpha_{\mathbf{q}})\mathbf{q}_{k-1} + \alpha_{\mathbf{q}}\mathbf{p}_{est}, \tag{13}$$

where $\alpha_{\mathbf{q}} \in [0,\ 1]$ is the learning rate that contribute to the updated histogram and $\mathbf{p}_{est}$ is histogram of the estimated state, respectively. In order to determine when the estimate is reliable, the likelihood of the current estimate is computed, π_{est}. The appearance is then updated when this value is higher than an indicator of the expected likelihood value and is calculated following an adaptive rule

$$\lambda_k = (1-\alpha_u)\lambda_{k-1} + \alpha_u \pi_{est}, \tag{14}$$

here λ_k is expected likelihood, $\alpha_u \in [0,\ 1]$ is the learning rate and π_{est} is estimated likelihood, respectively. This value indicates that the object has to be well matched to the model histogram before the update is applied.

3.7 Occlusion Handling

Occlusions can cause a failure in the tracking multiple objects. They may cause inaccurate in estimation and updating of the position and size of the tracked object. Thus, the target's estimated position would be shifted and its size should be adapted to the area that the target can be seen or not occluded. Moreover, the appearance model may be updated with completely wrong values which would cause target loss in few frames. The situation during complete occlusion could be even worst since the likelihood of the occluded target would be meaningless, the resampling phase would propagate the wrong random samples and quickly cause the loosing of the object. Therefore, a proper handling of occlusions is crucial.

In this paper, occlusions are predicted according to the dynamic models by using the predicted distance between the objects such as,

$$(\mathbf{x}_{m,k}^i - \mathbf{x}_{n,k}^i)^2 + (\mathbf{y}_{m,k}^i - \mathbf{y}_{n,k}^i)^2 < R^2 \tag{15}$$

here x_m, y_m, x_n and y_n are the sample position of each object and R is a threshold, respectively. When the predicted distance exceeds a certain threshold, the object is pointed out as occluded object. Subsequently, by exploring the maximum sample likelihoods and comparing them with recent historical values, we can conclude which object is being occluded. When the occlusion is detected, the object status turns into occluded object. This status involves several changes in the normal development of the process. First of all, the adaptation rates $\alpha_{\mathbf{x}}, \alpha_{\dot{\mathbf{x}}}, \alpha_{\mathbf{w}}, \alpha_{\dot{\mathbf{w}}}$ are set to zero and the target estimated speed is kept constant and the position is updated only according to its speed. In addition, no size or appearance adaptation is performed. Finally, those samples belonging to the occluded target are not resampled according to their weights since they are meaningless but they are just propagated. As a result samples spread around the target, because of the uncertainty predictions terms. The other object samples are normally resampled but they cannot be assigned to the occluded target. When the occlusion is no longer predicted or sample likelihood exceeds the value of previous likelihood of the occlusion object, the object status turns into not occluded, which immediately implies the samples to be resampled again. In addition, position and speed are again updated.

IV. EXPERIMENTAL RESULT

In order to evaluate our proposed method, we have done the experiments to track the multiple objects in the presence of occlusion and background clutter in outdoor environment. The experiments are implemented on Pentium IV with 2.53 GHz CPU and 512 MB RAM. The resolution of each frame is 320×240 pixels image. The color histogram is calculated in RGB space with 8×8×8 bins.

Figure 2 and 3 show the experimental results of the tracking objects. The small circle shows the estimated position of the tracking objects and the square shows the region of the tracked object used in color histogram.

On the first experiment, we try to track the objects when they move on the opposite direction and merge in the middle of the scene. The tracking performance is shown in Figure 2. Firstly, the first object appears on the left side of scene and the second object appears afterward on the right side. After several frames, the first object occludes the second object in the middle of the scene. Occlusion is correctly detected by avoiding resampling of samples of the occluded object and appearance models updating. The occluded object is successfully tracked as shown in frame #123, #127 and #133. The object is also tracked successfully although the full occlusion is occurred (frame #127). The system successfully recovers the object from occlusion (frame #137). After the occlusion is no longer occurred, each object is normally resampled and the appearance is update again. On both conditions (occlusion and not occlusion), we successfully update the appearance models when reliable measures are obtained.

On the second experiment, the objects have same moving direction and merge in the middle of the scene. The second object moves faster than the first one. On this experiment the occlusion status takes longer time (from frame #190 to frame #300). The occlusion is detected in the frame #190. At that time, the samples of the occluded object are not resampled but they are just propagated. The first complete occlusion is occurred on frame #220. The second object is still in the status of occlusion after several frames (frame #240 ~ frame #265). The second complete occlusion then occurs on frame #285 as the second object change the motion direction to the left side of scene. The system successfully recovers the second object from occlusion (frame #315). After the occlusion is no longer occurred, each object is normally resampled and the appearance is update again. On both conditions (occlusion and not occlusion), we successfully update the appearance models when reliable measures are obtained. Figure 3 shows the tracking performance of our method.

V. CONCLUSION

This paper presented a new method to track multiple objects employing color-based particle filter. The robust color likelihood is used to properly evaluate samples associated to target which present high appearance variability. We rely on Bhattacharya coefficient between target and sample histogram to perform this task. Model updating is carried to update the object in the presence of appearance change. The multiple objects tracking cause several problems such as occlusion. These problems can be tackled by redefining the weight normalization, prediction based on the dynamical model and likelihood measures. The performance of the tracking algorithm was tested by experiments. From the results, the algorithm can successfully track multiple objects moving in the presence of occlusion, background clutter and appearance change.

It should point out that although the experimental data used in this paper only contain close to linear motion model, there is no inherent difficulty for the proposed method to handle nonlinear motion model. This is because that the particle filter framework is generally not constrained to linear motion model. Furthermore, the performance of the multiple objects tracking can be improved in several ways such as adding the background modeling information [9] in the calculation of likelihood, detection of appearing objects and disappearing already existed object using hybrid particle filter [13] and so on. Taking them into consideration could lead to some improvement. These are remaining for our future works.

REFERENCES

[1] Ristic B, Arulampalam S and Gordon N (2004), Beyond the Kalman filter: particle filters for tracking applications, Artech House.

[2] Arulampalam S, Maskell S, Gordon N and Clapp T (2002), A tutorial on particle Filters for on-Line nonlinear/non-Gaussian Bayesian tracking", IEEE Trans. on Signal Processing, 2002, pp. 174-188.

[3] Doucet A, Freitas J, Gordon N (2001), Sequential Monte Carlo methods in practice, Springer, New York.

[4] Isard M and Blake A (1998), Condensation: Conditional density propagation for visual tracking, Int J of Computer Vision, 29(1): 5-28.

[5] Kitagawa G (1996), Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, J of Comput and Graph Stat 5(1):1-25.

[6] Gordon N, Salmon D and Smith A (1993), Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEE Proceeding on Radar and Signal Processing, 140(2):107-113.

[7] King O and Forsyth D (2000), How does Condensation behave with a finite number of samples, Proceedings of 6th European Conference on Computer Vision, Dublin, Ireland, pp 695-709.

[8] Nummiaro K, Koller E, Gool L (2002), An adaptive color-based particle filter, Image and Vision Comput, 25(1): 99-110.

[9] Perez P, Hue C, Vermaak J and Gangnet M (2002), "Color-based probabilistic tracking", Proceedings of the 7th European Conference on Computer Vision, London, pp. 661-675.

[10] Comaniciu D, Ramesh V, and Meer P (2000), Real-time tracking of non-rigid objects using mean shift, Proceedings of the IEEE Conference on Comp Vision and Pattern Recog, pp 142–149.

[11] Comaniciu D, Ramesh V, and Meer P (2003), Kernel-based object tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence, 25(5):564-577.

[12] Deutscher J and Reid I (2005). Articulated body motion capture by stochastic search, Int J of Computer Vision, 61(2):185-205.

[13] MacCormick J and Blake A (1999). A probabilistic exclusion principle for tracking multiple objects, Proceedings of the Int Conference on Computer Vision, pp 572-578.

[14] Czyz J, Ristic B, and Macq B (2007), A particle filter for joint detection and tracking of color objects, Image and Vision Comput, 25(8): 1271-1281.

[15] Isard M and. Blake A (1998), A mixed-state condensation tracker with automatic model-switching, Proceedings of Int. Conference on Comp Vision, pp. 107-112.

[16] Isard M and MacCormick J (2001), Bramble: a Bayesian multiple blob tracker, Proceedings of Int. Conference on Comp Vision, pp. 34-41.

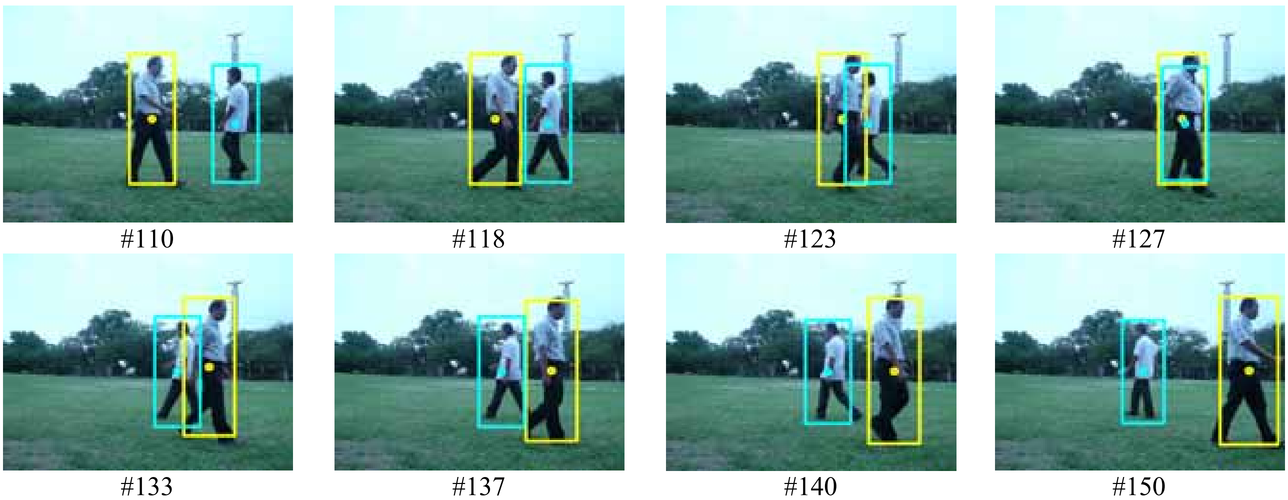

Fig.2. Two objects move in opposite direction and occlude in the middle of scene

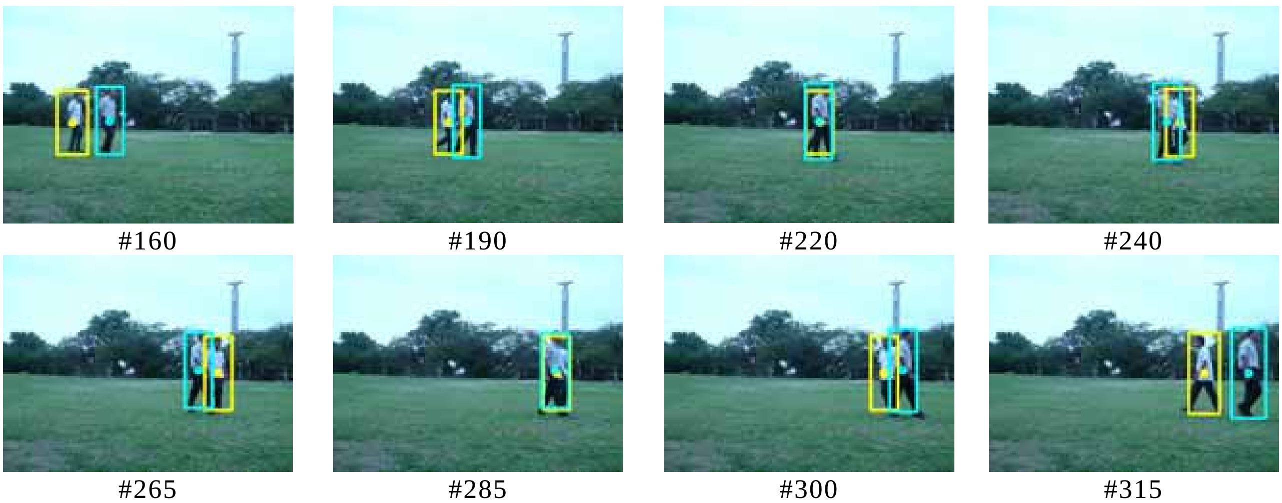

Fig.3. Two objects move in the same direction and occlude in the middle of scene