Object Tracking using Particle Swarm Optimization and Earth Mover's Distance

Gongyi Xia and Simone A. Ludwig
North Dakota State University
Fargo, ND, USA
{gongyi.xia,simone.ludwig}@ndsu.edu

***Abstract*—Visual object tracking is an active research field in the area of computer vision. The tracking process usually includes the construction of an object appearance model and the object localization. This paper investigates the use of Particle Swarm Optimization (PSO) as the object localization method based on the Bayesian tracking framework. The widely adopted particle filter tracking technique, however, suffers from high computational cost due to the approximation requirement of the distribution of particles. Thus, PSO is applied since it can adaptively adjust the computational expenditure according to each frame in the video. Furthermore, a new appearance model based on Earth mover's distance is proposed. The experimental results show that the proposed approach enhances the accuracy of the tracking algorithm significantly compared to the basic particle filter tracking method. Furthermore, the proposed appearance model is more robust than other Earth mover's distance based tracking algorithms.**

I. Introduction

Object tracking plays an important role in many different applications that are related to vision such as motion analysis, activity recognition, video surveillance and traffic monitoring. Even though a lot of progress has been made in the past decade, however, given the nature of the object tracking task with the many challenging parts, it is still a very active research area with incremental improvements being made. The challenging parts that are involved in object tracking are due to the changes in the appearance model when objects are being tracked such as camera motion, occlusion, illumination, pose variation, and shape deformation.

The tracking process involves the following major steps. First, an appearance model is used to represent the object with the appropriate features using an object representation, which is done in a fixed frame. Then, in successive frames, a tracking method is applied to estimate the likely state of an object. Possible tracking methods are Kalman filter [1], and particle filter [2], [3]. As has been pointed out by many researchers, the appearance model of the object has the most influence on the accuracy of the tracking algorithm. There are many different factors that need to be considered for an effective appearance model. For example, an object can be represented by different features such as texture [4], Haar-like features [5], [6], [7], intensity [8], color [2], or superpixels [9]. Furthermore, the representation scheme can be based on local histograms [10] or holistic templates [1].

978-1-5090-4601-0/17/$31.00

Earth mover's distance (EMD) was first used for image retrieval. Due to its extraordinary capability of measuring distance between distributions, researchers started applying this distance measure to the field of visual object tracking. In a typical visual object tracking application using EMD, two histograms or distributions are usually extracted from the target and candidate, respectively. Then the earth mover's distance between two histograms or distributions is calculated, which in turn is used as input to locate the tracked object.

In this paper, we investigate the use of Particle Swarm Optimization (PSO) as the object localization method based on the Bayesian tracking framework together with an EMD based appearance model. The widely adopted particle filter tracking technique suffers from high computational cost due to the approximation requirement of the distribution of particles in fact most of particles located in low likelihood area contribute little towards particles. In this paper, particles are tailored and the PSO algorithm is applied whereby the computational expenditure according to each frame in the video can be adaptively adjusted. This way the tracker is more robust in cases of recovering from a losing target because of PSO's ability to explore a wider search area. Our previous work [11] used PSO and the Bayesian tracking framework as well, however, it was focused on the sampling mechanism of particles.

II. Related Work

Particle filters were first introduced to visual tracking in [12]. Since then it has become a popular method for object tracking since it has shown very good performance in particular for nonlinear target motion as well as it is flexible to different object representations [13]. Particle filters, also known as sequential Monte Carlo models, uses particles that are sampled in order to construct the target representation. Particle filters are more likely to perform well in cluttered and noisy environments. One of the shortcomings, however, is the high computational cost of the particle filter trackers, which increases linearly with the number of particles. Thus, several methods have been proposed to overcome this limitation in the past. The following describes object trackers that are based on the particle filter technique.

For example, in [2] a color-based probabilistic tracking algorithm is proposed. The algorithm is based on color histogram distance using the particle filter tracking technique. The particle filter enables to handle the color clutter in the

background and also addresses complete occlusion of the tracked objects over a few frames. The algorithm uses a multi-part color modeling scheme to capture the spatial layout that is ignored by global histograms. Furthermore, the algorithm incorporates a background color model when needed, and extents the algorithm to track multiple objects.

Another particle filter based algorithm is described in [8]. This approach uses an incremental learning method that learns a low-dimensional subspace representation, and efficiently adapts to changes in the appearance of the target. The model update is based on principal component analysis and includes two features. The first feature is a method to update the sample mean, and the second is to include a forgetting factor to ensure less modeling power is expended to fit older observations. These introduced features improve the overall tracking performance significantly.

Another approach based on the particle filter framework is given in [14]. The approach formulates the tracking task as a structured multi-task sparse learning problem. In the model, the particles are represented as linear combinations of dictionary templates that are updated dynamically, and the learning of the representation of each particle is considered as a single task in the multi-task tracking method. The authors introduce a popular sparsity-inducing L mixed norm to regularize the representation problem enforcing joint sparsity to learn the particle representation.

As an effective optimization method, PSO has been used under the Bayesian framework, mostly together with the particle filter, to achieve the best estimation. The authors in [15] use an improved PSO-Gaussian swarm merged into particle filter to estimate the nonlinear system state. The PSO is used to move the sampled particles to high likelihood regions, where the fitness function is defined based on the discrepancy between the new observation and the predicted observation. The authors claim that the particle impoverishment problem is thus avoided.

A similar idea was adopted in [16], where after the generation of the prior samples, these samples are moved to the desired region so that both the likelihood and the prior density of the particles is significant. The problem is solved as a multi-objective optimization problem, where a conventional weighted aggregation (CWA) approach is employed to combine the two objectives. The resulting maximization problem is then solved using PSO. Experimental results suggest the proposed method as a promising alternative in cases of high system and measurement noise levels and small sample sizes.

The so-called multi-layer importance sampling proposed in [17] improves the performance of the particle filter by moving particles to the region with high likelihood using PSO. Since the posterior of the particles depends on the likelihood of observation, thus the best estimation is obtained using the Maximum a Posterior (MAP). The authors claim that this method is better than the particle filter as well as the unscented particle filter. In addition, the method also works more effectively for dynamic optimization problems than the sequential Monte Carlo methods.

The authors in [18] implemented a particle filter where the motion model is implicitly canceled out by the special importance density. The PSO is only used to locate the high likelihood region instead of sampling the particles as is the case in other papers. Then, particles are sampled according to both the global best and the individual best of the swarm. The appearance model used in their paper is a combination of histogram of color (HOC) and histogram of gradient (HOG). The experiments show that the proposed method achieves higher accuracy.

A locally orderless tracking approach is outlined in [19]. The algorithm automatically estimates the amount of local disorder in the object by specializing in both rigid and deformable objects. A probabilistic model of object variations is used that is based on the EMD controlling the cost of the moving pixel as well as the changing of their color. These costs are adjusted online during the tracking task to address the amount of local disorder in the object.

Authors in [20] proposed an algorithm which uses Gaussian mixtures as appearance model and EMD as similarity measurement. In [21], EMD was used to match images. EMD was employed to measure the similarity between target and candidate, each is represented by a set of signatures generated by the super pixel technique based on color distributions. A signature is described by its center and size space that are given by the super pixel algorithm. Its center is a point in the 5-D space (3-D for colors and 2-D for the geometric position).

Object tracking algorithms in [22] use EMD to compare a group of patches to the target template, each patch is sampled from a small area within the potential target and expressed as a 1-D histogram.

Authors in [23] developed a differential EMD method to enable gradient decent search by obtaining the derivative of EMD with respect to the location. Similar to [22] EMD in this paper is used as similarity measure between the signatures based on the color distributions.

The most critical problem with EMD is its high computational cost. As many papers suggest, the complexity of EMD is $O(n^3)$. In order to mitigate this issue, practical applications of EMD devote large efforts to reduce the data size. As proposed in [21] and adopted in [19], [22], [23], a set of signatures based on color distributions instead of pixels could reduce the data size by several folds while preserving the essential color information.

The adoption of signatures based on color distribution successfully mitigates the excessive computational complexity of EMD. However, there are several drawbacks that present challenges to future improvement. First, the high dimensions of the space where the signature center resides makes efforts of extending on the dimension difficult since this will increase the data size dramatically. Second, this signature is designed to work on color images/videos. By including 3 color dimension, the higher dimension increases distances between the signature sets thus helps to differentiate them better. However, color information is not universally available. For example, sometimes color images/videos are not available. Sometimes

even with color cameras, saturation of the captured video may be very weak in case of adverse light environments. Hence, it is desirable to develop an appearance model that effectively measures the EMD distance between the image patches and also works on grayscale videos.

III. Proposed Approach

In order to broaden the applicable scenarios, especially in case of grayscale videos, a new appearance is proposed which works together with PSO enabled Bayesian tracking to provide improved performance.

A. Appearance Model

In this section, we construct an appearance model that describes the target and measures the similarity between the target and candidates. This includes constructing histograms from frames for target and candidates and distances between histograms, and also other adjustments as needed.

1) 2-D Histogram: Because of the high complexity of EMD, researchers were forced to use smaller data size to control the running time within an acceptable timeframe. The information is compressed considerable during the tracking process. For example, the pixels are grouped together by the super pixel algorithm. Thus, their geometric information is blurred.

In this paper, we are interested in keeping each pixel as independent and investigate how their geometric information can be used during the process. Specifically, affine transformation is employed to resize each target or candidates to an array of 32×32 pixels. Each pixel only holds intensity information. Color videos will be transformed into grayscale first. By doing so we obtain a 2-D histogram where the bins are the exact geometric position of each pixel and the value of each bin is the intensity of its corresponding pixel as shown in Figure 1.

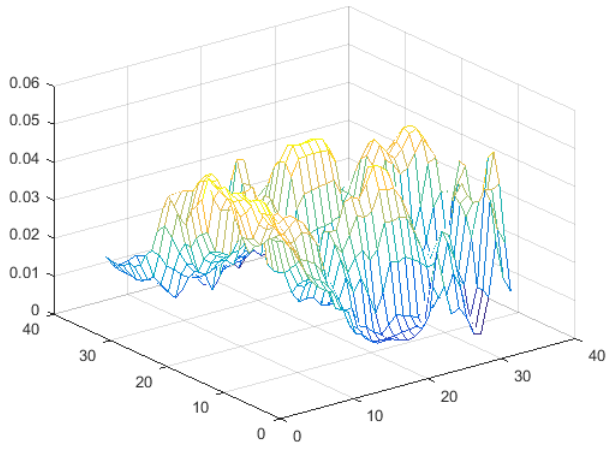

Fig. 1. 2-D Histogram

The dimension of the histogram is drastically reduced by using grayscale frames and treating the intensity of the pixels as the value for each bin instead of another dimension. However, to measure the EMD distance between two 32×32 2-D histograms is still too large according to our experiments. There are certain approximations we can choose from. In this paper, we use EMD-L1 proposed in [24] as an approximation of EMD between two histograms. We denote the distance between two histograms measured by EMD-L1 as D_{EMD}.

2) Partitioning: In order to differentiate candidates from the background better, we partition the target area into smaller overlapping patches as shown in Figure 2. We then compare each patch with its corresponding counterpart, then the distances of each patch are summed. Because of the spatial arrangement of these patches, the spatial information is naturally used.

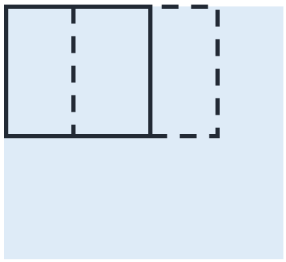

Fig. 2. Target partitioning

3) Contrast Enhancement: Intensity information is susceptible to illumination variation, which presents a big challenge to intensity based appearance models. Usually a normalization is performed to mitigate the effect of illumination variation. However, it is not enough when EMD is used. The normalization enlarges or shrinks the values of the entire data set at the same time, so all bins are adjusted towards the same direction. It does not change the relative difference between bins significantly. For example, for a target turning dark because of the shadow of a surrounding building, all of its pixels are pushed to lower intensity values. The texture of the target becomes obscured due to the lower contrast. After normalization, the difference between pixels, i.e. contrast, still remains small. Contrast enhancement is employed to create a histogram of intensity with enough diversity. In this paper, a technique called contrast-limited adaptive histogram equalization (CLAHE) [25] is used to enhance the contrast of video frames.

B. PSO based Bayesian Tracking

1) Bayesian Tracking: In the context of Bayesian inference, the visual object-tracking problem can be formulated by two models: system model and measurement model. The system model is given in Eq. (1), which represents the underlying system state and its dynamics. It is usually inaccessible due to persistent noise during the measurement process; what can be observed is a measurement of the state that is obtained through the measurement process, which is given in Eq. (2):

$$x_k = f_k(x_{k-1}, v_{k-1}) \tag{1}$$

$$z_k = h_k(x_k, n_k) \tag{2}$$

where the sequence $\{x_k, k \in N\}$ denotes the target state, the sequence $\{z_k, k \in N\}$ denotes the measurement of the state sequence, and v_k and n_k are system noise and measurement noise, respectively.

The goal is to obtain an estimation of the current state, which can be formulated into derivation of a conditional density $p(x_k|z_{1:k})$, which is performed in a recursive manner consisting of a prediction and an update operation. The conditions include prior knowledge about the system and the measurement process as well as all measurements $z_{1:k} = \{z_i, i = 1, 2, \ldots, k\}$ up to time k. By the criteria of Maximum-a-Posterior (MAP), the optimal estimation is the state where the posterior probability density is the highest. By Bayes' rule, the posterior density of $p(x_k|z_{1:k})$ is derived as in Eq. (3) according to the Bayes' rule:

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \tag{3}$$

where $p(z_k|x_k)$ denotes the likelihood of z_k being a real measurement of x_k and $p(x_k|z_{1:k-1})$ denotes the prior density of x_k before time k. The normalizing denominator $p(z_k|z_{1:k-1}$ is required per Bayes' rule and remains constant across all potential x_k.

The prior density of x_k is predicted based on all the available measurements up to time $k-1$ as given in Eq. (4).

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\mathrm{d}x_{k-1} \tag{4}$$

where $p(x_k|x_{k-1})$ is the motion model, and $p(x_{k-1}|z_{1:k-1})$ is the posterior density of state x_{k-1}.

It is very rare that an exact expression of equations above can be found for real problems. This problem is circumvented by adoption of density approximation by particles, where each particle represents a sample from the approximated density and is drawn based on an approximated density. In such way, the distribution of the particles describes the approximated density, and the denser the particles are, the higher the probability is.

The particle filter, which is widely used in object tracking, serves as a direct implementation of the Bayesian inference. In a particle filter, the posterior density is approximated by weighted particles. These particles are propagated forward between time steps in the prediction stage based on the previous particle set by an importance sampling mechanism as shown in Eq. (5), where $q(\cdot)$ is the importance density. Afterwards, the weights of these particles are updated by a new measurement in the update stage as described in Eq. (6).

$$x_k \sim q(x_k|x_{0:k-1}, z_{1:k}) \tag{5}$$

$$w_k^i \propto w_{k-1}^i \frac{p(z_k|x_k)p(x_k|x_{k-1})}{q(x_k|x_{0:k-1}, z_{1:k})} \tag{6}$$

2) Particle Swarm Optimization: PSO was first introduced in [25] as a population based stochastic optimization method. Similar to other evolutionary algorithms, PSO was inspired by some natural phenomenon. PSO mimics the social behavior that is often observed in bird flocking by adopting the concept of a swarm in which individuals can communicate with their peers. In a swarm (bird flock), each individual (bird) represents a solution (a position) and each solution corresponds to a fitness value (distance to the food). An individual's movement is influenced by both its own experience and its peers' knowledge. Eventually, the whole swarm is likely to converge at an optimal position.

Formally, the social behavior within PSO can be represented as follows:

$$v_{id}^{t+1} = wv_{id}^t + c_1p_1(p_{id}^t - x_{id}^t) + c_2p_2(p_{gd}^t - x_{id}^t) \tag{7}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \tag{8}$$

where x_{id}^t represents the status of an individual, i.e. particle, v_{id}^t represents the velocity of individuals, p_{id}^t represents the best known position an individual has explored, p_{gd}^t represents the global best known position exchanged between the peers, w is the inertia coefficient, and c_1, c_2 are random weights.

In object tracking applications, the task of searching for the best suitable candidate can be formulated as an optimization problem, and PSO is employed as the optimizer to locate the global optimum. During such PSO process, each candidate of the target can be represented by a particle, and the multiple dimensional search space is constructed on the basis of the candidate position, size, and rotation, etc. The fitness value is often defined as the similarity between the candidate and the target or the confidence level of the candidate being the real target. Once initialized, the PSO process does not require any input of the candidate, it compares the fitness values of the current candidates and generates a new set of candidates according to the PSO update equations.

3) Particle Swarm Optimization in Bayesian Tracking Framwork: As discussed in Section III-B1, in most real applications, the Bayesian inference can only be solved in a non-linear manner. Thus, this makes the particle filter a good choice when performing Bayesian inference for visual object tracking. However, the particle filter itself suffers from the following problems. The first issue is degeneracy [26], [27], where a large portion of the particles will have negligible weight after a few iterations. Thus, these particles contribute very little towards the distribution approximation because of their small weights. To cope with the problem of degeneracy, Sampling Importance Resampling (SIR) was adopted in [26] where particles of small weights have a smaller chance to be forwarded to the next iteration. While this alleviates the degeneracy problem to some extent, however, the resampling may lead to the impoverishment problem, where the diversity of particles is reduced especially in the case of small process noise.

Extensive research efforts have been spent on dealing with the degeneracy problem without giving rise to the impoverishment problem. The regularized particle filter in [28] resamples particles from a continuous posterior density created by applying the kernel smoothing to the discrete posterior density in order to distribute the particles more evenly. The auxiliary particle filter in [29] avoids resampling from particles that have low likelihood by constructing an importance density that has

high conditional likelihood. The kernel particle filter proposed in [30] employs Mean-Shift to migrate particles towards higher probability locations on the posterior probability landscape.

Recently, due to its broad searching capability, PSO was investigated to be integrated with the particle filter to overcome the degeneracy problem mentioned above. As discussed previously, the PSO approach adopted in [31], [32], [17] moves particles towards regions with higher likelihood, thus, alleviating the problem of degeneracy. In addition, in [33] an additional sampling operation is performed following the PSO process. To reduce the degeneracy, the particles have to be moved or sampled as directed by the PSO process instead of being propagated from the previous time step through the importance density. The denominator, i.e., importance density term in Eq. (6), is set to $p(x_k|x_{k-1})$ to make the equation even. By doing this, the system motion model $p(x_k|x_{k-1})$ is canceled out during the particle filter iterations. This means the method discussed above completely neglects the system model. And the particles are not sampled from importance density, instead they are generated by the PSO process. It is worth to note that the optimal importance density is actually the system motion model $p(x_k|x_{k-1})$, and it will be canceled out in Eq. (6) [34]. However, the particles have to be drawn by the motion model first.

It can be observed that both problems of degeneracy and impoverishment are inherent in the fact that densities in the particle filter are approximated by a limited number of particles, and the particles are sampled from the posterior density before it is even estimated. Although larger number of particles will certainly mitigate these problems, it is difficult to determine what number will be adequate. Increasing the number of particles unboundedly is both computationally expensive and practically prohibitive.

In this paper, we try to develop a tracking scheme which does not suffer from the degeneracy problem as in particle filters and conforms to the Bayesian inference framework. We do this by formulating the Bayesian tracking problem into a PSO optimization problem. Let us rewrite Eq. (3) and Eq. (4) as such:

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \tag{9}$$

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\mathrm{d}x_{k-1} \tag{10}$$

In alignment with the Maximum a Posterior (MAP), the optimal estimation of the system state at time $\hat{x}_{k-1}$ can be obtained by:

$$\hat{x}_{k-1} = \arg\max_{x_{k-1}}(p(x_{k-1}|z_{1:k-1})) \tag{11}$$

Here, if $\hat{x}_{k-1}$ is significant enough, we can use the optimal estimation $\hat{x}_{k-1}$ instead of the entire distribution $p(x_{k-1}|z_{1:k-1})$, thus Eq. (10) in turn becomes:

$$p(x_k|z_{1:k-1}) = p(x_k|\hat{x}_{k-1})p(\hat{x}_{k-1}|z_{1:k-1}) \propto p(x_k|\hat{x}_{k-1}) \tag{12}$$

and Eq. (9) becomes:

$$p(x_k|z_{1:k}) \propto p(z_k|x_k)p(x_k|\hat{x}_{k-1}) \tag{13}$$

Now the Bayesian tracking problem is formulated as an optimization problem, and can be optimized using PSO in the space of the current system state at time k. The fitness function is given in Eq. (13), and $p(x_k|\hat{x}_{k-1})$ is the system motion model. The key in this PSO optimization process is the significance of $\hat{x}_{k-1}$ compared to the entire distribution $p(x_{k-1}|z_{1:k-1})$. It largely depends on the chosen appearance model. Usually, if the appearance model is discriminative enough, then its likelihood function $p(z_{k-1}|x_{k-1})$ would peak significantly at the real target position.

Another benefit by formulating the tracking problem as a PSO optimization task includes the adaptive size of the particle swarm. Different from the particle filter, which uses a fixed particle swarm size throughout the entire tracking process, the required iterations of PSO can be adjusted according to the fitness value or some other criterion. For the frames in which the target is distinct, the PSO process can use fewer iterations to locate the target. Otherwise, more iterations can be used to search a larger scope.

IV. Experiments and Results

In this section, we evaluate our tracking algorithm on a video data set comprised of a wide range of videos. We first examine the validity of the fitness function. Then, the performance of our object trackers is evaluated against other trackers that use EMD as the distance measure.

A. Experimental Setup

The experiments are implemented based on the object tracking framework developed in [35]. The video set is also provided by this framework, which includes 51 videos of different scenes. All of the code is run in Matlab R2015 on a Windows machine equipped with Core i5@2.8GHz*4 Cores, 16GB RAM.

1) System Model: For the observation model, we use the appearance model described in Section III-A. The likelihood is defined in Eq. (14):

$$p(x_k|z_k) \propto D_{EMD} \tag{14}$$

where D_{EMD} is the earth mover's distance between target and candidates.

The state variable x_k is modeled by affine transformation parameters $(x_k, y_k, s_k, \theta_k, \alpha_k, \phi_k)$, where $\{x_k, y_k\}$ are the coordinates of the target center, s_k denotes the scale, θ_k denotes the rotation, α_k denotes the aspect ratio, and ϕ_k denotes the skew direction at time k. A set of affine transformation parameters specify an affine transformation from unit cube at the origin of the image to the state x_k.

The system motion model is assumed to be of Gaussian distribution $p(x_k|x_{k-1}) \sim N(x_{k-1}, \Sigma)$, where Σ is the diagonal covariance matrix whose diagonal elements $(\sigma_x^2, \sigma_y^2, \sigma_s^2, \sigma_\theta^2, \sigma_\alpha^2, \sigma_\phi^2)$ denote the variance for each dimension. In this paper, we set $\sigma_x^2, \sigma_y^2 = 18^2$, σ_s^2, σ_α^2, σ_θ^2 and σ_ϕ^2 to 0.

2) PSO Setup: The PSO process in our tracker is set as follows. The population size of the particle swarm is set to 30. The stopping criteria used are:

1) after 30 iterations are elapsed, or
2) if the overlap ratio of the best bounding box of the previous 6 frames as compared to the current frame is at least 99%.

In this paper, we only vary the target location, scale and aspect while leaving the other two parameters unchanged. Therefore, the search space is 4-dimensional. The swarm of the particles are initialized uniformly for each dimension, where x, y is distributed evenly within ± 30 pixel on both the horizontal and vertical axis around the previous target position, s is distributed between ± 0.06, and α is distributed between ± 0.009.

B. Results

We first verify the applicability of using the optimal estimation instead of the entire distribution as explained in Section III-B3. For the sake of simplicity, the posterior density $p(x_{k-1}|z_{1:k-1})$ at time $k-1$ is approximated by $p(x_{k-1}|z_{k-1})$. This posterior density serves as the prior density for the calculation of $p(x_k|z_{1:k})$ at time k and consists of 1,681 particles covering ± 20 pixels in both horizontal and vertical directions centered at the optimal estimation of the target at the previous time step $k-1$. In order to better visualize the distribution, we only vary the candidate location while keeping the other parameters unchanged.

We compute the posterior density $p(x_k|z_{1:k})$ at time k with the prior density $p(x_{k-1}|z_{k-1})$ represented by 1,681 particles as shown in Eq. (9) and with the prior density $p(x_{k-1}|z_{k-1})$ represented only by $\hat{x}_{k-1}$, Eq. (13), respectively. Figures 3 and 4 show that in case of a high discriminative appearance model, using the optimal estimation $\hat{x}_{k-1}$ only instead of whole set of particles does impact the shape of the resulting posterior density $p(x_k|z_{1:k})$ noticeably.

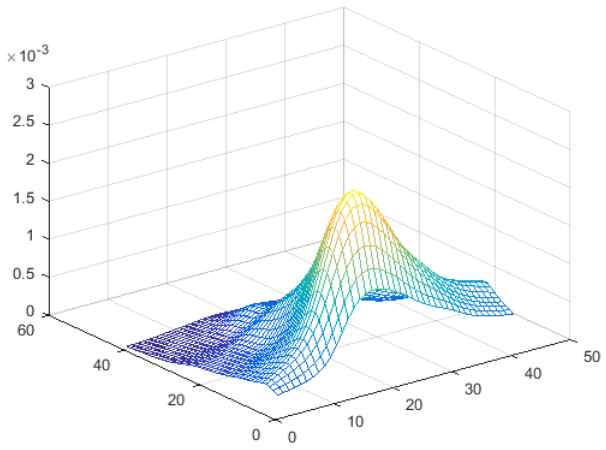

Fig. 3. Posterior density with prior density containing all particles

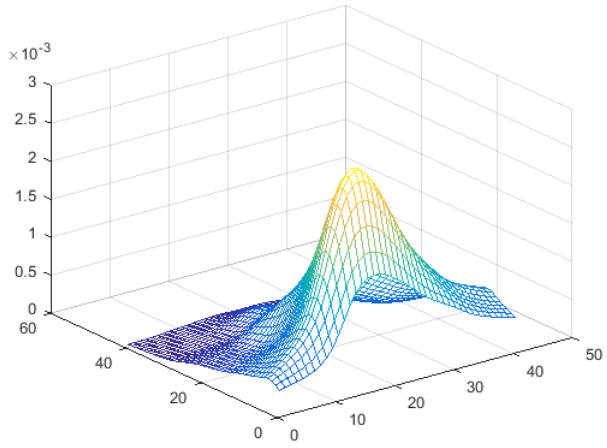

Fig. 4. Posterior density with prior density containing optimal estimation

1) Evaluation Method: In this work, we use the precision and success rate for the quantitative analysis as in [35] to evaluate the tracking ability of our approach. Both precision rate and success rate are measured on all 51 video sequences to eliminate the dependency on the specific videos and alleviate potential resulting variance. In detail, the precision rate measures the accuracy of the tracking in terms of the distance between the tracking result and the ground truth. Specifically, the distance is derived by calculating the Euclidean distance between the centers of the tracking result and the ground truth. Given $\{x,y\}_r$ is the center of the tracking results, and $\{x,y\}_g$ is the center of the ground truth, then the precision rate is obtained based on Eq. (15):

$$\text{precision rate} = \frac{N_{\|\{x,y\}_r - \{x,y\}_g\| \leq \text{threshold}}}{N_{total}} \tag{15}$$

where N_{total} is the number of total frames for which the tracking is performed, $N_{\|\{x,y\}_r - \{x,y\}_g\| \leq threshold}$ is the number of frames which is within a given distance of the threshold from the ground truth, and $\|\cdot\|$ represents the Euclidean distance.

The success rate on the other hand measures the robustness of the tracking in terms of the overlap between the tracking result and the ground truth. Given P_r is the set of pixels within the tracking result bounding boxes, and P_g is the set of pixels within the ground truth bounding boxes, the success rate is given by Eq. (16):

$$\text{success rate} = \frac{N_{S \geq \text{threshold}}}{N_{total}} \tag{16}$$

where N_{total} is the total number of frames for which the tracking is performed, $N_{S \leq threshold}$ is the number of frames whose overlap score S is equal or greater than the given threshold, $S = \frac{|P_r \cap P_g|}{|P_r \cup P_g|}$ is the overlap score between the tracking result and the ground truth, and $|\cdot|$ measures the number of pixels in the pixel set.

2) Performance Evaluation: In this section, we compared our PSO based Bayesian Tracking Framework (B-PSO) with the proposed EMD based appearance model with several other

trackers including the EMD based tracker (LOT) [19], regular bootstrap particle filter [36] with Sampling Importance Resampling (SIR) [26] connected with our EMD based appearance model (PF-EMD), and Particle Filter incorporating PSO [17] with our EMD based appearance model (PSO-PF). All these trackers share the same PSO parameters.

As shown in Figures 5 and 6, our PSO based Bayesian Tracking Framework (B-PSO) tracker performs best. Considering the fact that B-PSO requires only the intensity information while LOT uses color information, our B-PSO is promising for further improvements. The x-axis in both figures are thresholds in pixel as defined in Eq. (15) and Eq. (16). Compared to the regular particle filter, B-PSO with the PSO optimization process can search a broader scope without causing much additional cost. The embedded system motion model helps to reduce the chances of premature convergence.

We also compared regular Particle Filter (PF-EMD) with our B-PSO using the same appearance model. In our experiment, it employed 500 particles to approximate the posterior density. It uses a system motion model to propagate the particles by resampling particles according to their weights. However, particle filter does not solve the problems of degeneracy and impoverishment. Hence it performs much worse than B-PSO.

PSO-PF is closely related to our PSO based Bayesian algorithm. The PSO process locates the high likelihood region and drives particles towards such regions, whereby it reduces the problem of degeneracy significantly. However, since the motion model is omitted in this tracking algorithm, the particles are driven by the PSO process almost unconstrainedly except for its inherent velocity limits and space boundary. These parameters have to be set with conservative values to avoid premature convergence which limits its search capability in particular for fast motion scenarios.

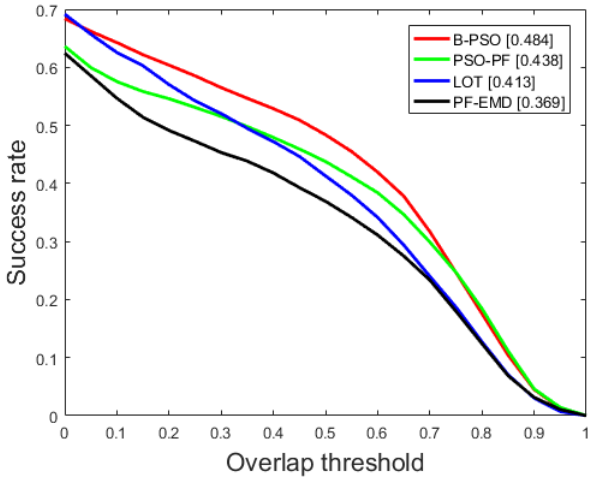

Fig. 5. Success rate versus overlap threshold

Table I shows the comparison of the running time and the average particle number. It can be observed that the running time is directly proportional to the particle number employed in each frame. B-PSO runs slower than PF-PSO because it takes more iterations to converge partially due to the different

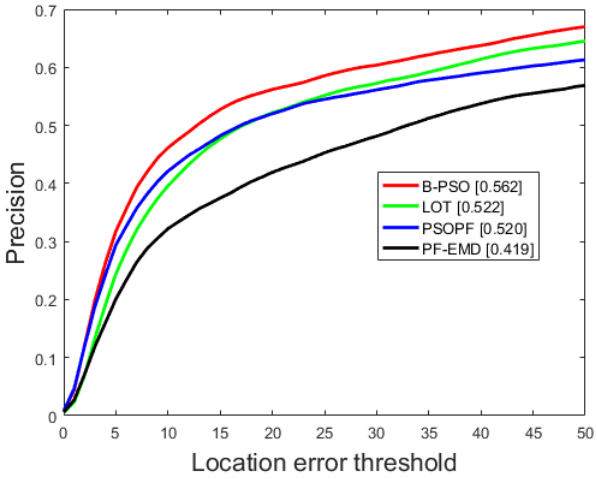

Fig. 6. Precision rate versus location error threshold

TABLE I
RUNNING TIME AND AVERAGE PARTICLE NUMBER PER FRAME

Trackers	fps	Particle number
B-PSO	1.14	380.02
PSO-PF	0.94	435.49
LOT	1.25	-
PF-EMD	0.89	500

initialization strategies. However, on the other hand the use of more particles pays back with better performance. LOT runs slightly faster than B-PSO partially due to the compressed data size by using the super pixel process. PF-EMD is slower than B-PSO and PSO-PF since more particles are utilized.

V. Conclusion

In this paper, we reviewed several variations of the particle filter in the field of object tracking as well as PSO based particle filter approaches. We found that when incorporating PSO into the particle filter, the system motion model of the particle filter is canceled out, though, the omission of the motion model does not necessarily cause performance degradation since an accurate system motion model is not always available. However, a correct motion model could guide the search process, and therefore improve the tracking success rate and precision.

In this paper, we simplified the Bayesian tracking framework by substituting the posterior density by its optimal estimation in order to formulate the Bayesian tracking task as an optimization problem. With an appropriate fitness function, we employed PSO to perform the optimization task. We also developed an EMD based appearance model, which does not rely on color information, thus, is universal applicable compared to other EMD based appearance models.

The experiments showed that the proposed EMD based appearance model together with the proposed PSO based Bayesian tracking framework works better compared to other EMD based appearance model based tracker. The embedding

of the system motion model into the PSO process helps to reduce the chances of premature convergence whereby allowing a broader search. The experiments also showed that the proposed approach is more computational effective thanks to the flexibility of the PSO optimization process enabling an early termination mechanism.

In the future, the simplification of the Bayesian tracking framework by substituting the posterior density by its optimal estimation could be further improved to reduce the information loss. Also, in this paper the system motion model is simply based on a multi-variant Gaussian distribution, which is a very coarse assumption. Applying other motion estimation or motion modeling techniques would likely improve the tracking performance.

REFERENCES

[1] Comaniciu, D., Ramesh, V., Meer, P. (2003). Kernel-based object tracking. IEEE Transactions on pattern analysis and machine intelligence, 25(5), 564-577.

[2] Prez, P., Hue, C., Vermaak, J., Gangnet, M. (2002, May). Color-based probabilistic tracking. In European Conference on Computer Vision (pp. 661-675). Springer Berlin Heidelberg.

[3] Li, Y., Ai, H., Yamashita, T., Lao, S., Kawade, M. (2008). Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(10), 1728-1740.

[4] Avidan, S. (2007). Ensemble tracking. IEEE transactions on pattern analysis and machine intelligence, 29(2), 261-271.

[5] Grabner, H., Bischof, H. (2006, June). On-line boosting and vision. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) (Vol. 1, pp. 260-267). IEEE.

[6] Grabner, H., Leistner, C., Bischof, H. (2008, October). Semi-supervised on-line boosting for robust tracking. In European conference on computer vision (pp. 234-247). Springer Berlin Heidelberg.

[7] Kalal, Z., Matas, J., Mikolajczyk, K. (2010, June). Pn learning: Bootstrapping binary classifiers by structural constraints. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (pp. 49-56). IEEE.

[8] Ross, D. A., Lim, J., Lin, R. S., Yang, M. H. (2008). Incremental learning for robust visual tracking. International Journal of Computer Vision, 77(1-3), 125-141.

[9] Wang, S., Lu, H., Yang, F., Yang, M. H. (2011, November). Superpixel tracking. In 2011 International Conference on Computer Vision (pp. 1323-1330). IEEE.

[10] Yang, F., Lu, H., Chen, Y. W. (2010, August). Bag of features tracking. In Pattern Recognition (ICPR), 2010 20th International Conference on (pp. 153-156). IEEE.

[11] Xia, G., Ludwig, S. A. (2016). Object-tracking based on particle filter using particle swarm optimization with density estimation. In Evolutionary Computation (CEC), 2016 IEEE Congress on (pp. 4151-4158). IEEE.

[12] Isard, M., Blake, A. (1998). Condensationconditional density propagation for visual tracking. International journal of computer vision, 29(1), 5-28.

[13] Wu, Y., Huang, T. S. (2004). Robust visual tracking by integrating multiple cues based on co-inference learning. International Journal of Computer Vision, 58(1), 55-71.

[14] Zhang, T., Ghanem, B., Liu, S., Ahuja, N. (2012, June). Robust visual tracking via multi-task sparse learning. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on (pp. 2042-2049). IEEE.

[15] Tong, G., Fang, Z., Xu, X. (2006, July). A particle swarm optimized particle filter for nonlinear system state estimation. In 2006 IEEE International Conference on Evolutionary Computation (pp. 438-442). IEEE.

[16] Klamargias, A. D., Parsopoulos, K. E., Vrahatis, M. N. (2008, July). Particle filtering with particle swarm optimization in systems with multiplicative noise. In Proceedings of the 10th annual conference on Genetic and evolutionary computation (pp. 57-62). ACM.

[17] Zhang, X., Hu, W., Maybank, S., Li, X., Zhu, M. (2008, June). Sequential particle swarm optimization for visual tracking. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on (pp. 1-8). IEEE.

[18] Zhao, J., Li, Z. (2010). Particle filter based on Particle Swarm Optimization resampling for vision tracking. Expert Systems with Applications, 37(12), 8910-8914.

[19] Oron, S., Bar-Hillel, A., Levi, D., Avidan, S. (2015). Locally orderless tracking. International Journal of Computer Vision, 111(2), 213-228.

[20] Karavasilis, V., Nikou, C., Likas, A. (2011). Visual tracking using the Earth Mover's Distance between Gaussian mixtures and Kalman filtering. Image and Vision Computing, 29(5), 295-305.

[21] Boltz, S., Nielsen, F., Soatto, S. (2010, September). Earth mover distance on superpixels. In 2010 IEEE International Conference on Image Processing (pp. 4597-4600). IEEE.

[22] Adam, A., Rivlin, E., Shimshoni, I. (2006, June). Robust fragments-based tracking using the integral histogram. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) (Vol. 1, pp. 798-805). IEEE.

[23] Zhao, Q., Yang, Z., Tao, H. (2010). Differential earth mover's distance with its applications to visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(2), 274-287.

[24] Ling, H., Okada, K. (2006, May). EMD-L 1: an efficient and robust algorithm for comparing histogram-based descriptors. In European Conference on Computer Vision (pp. 330-343). Springer Berlin Heidelberg.

[25] Karel Zuiderveld. 1994. Contrast limited adaptive histogram equalization. In Graphics gems IV. Academic Press Professional, Inc., San Diego, CA, USA 474-485.

[26] Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika, 82(4), 711-732.

[27] Liu, J. S., Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. Journal of the American statistical association, 93(443), 1032-1044.

[28] Musso, C., Oudjane, N., Le Gland, F. (2001). Improving regularised particle filters. In Sequential Monte Carlo methods in practice (pp. 247-271). Springer New York.

[29] Pitt, M. K., Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. Journal of the American statistical association, 94(446), 590-599.

[30] Chang, C., Ansari, R. (2005). Kernel particle filter for visual tracking. IEEE Signal Processing Letters, 12(3), 242-245.

[31] Tong, G., Fang, Z., Xu, X. (2006, July). A particle swarm optimized particle filter for nonlinear system state estimation. In 2006 IEEE International Conference on Evolutionary Computation (pp. 438-442). IEEE.

[32] Klamargias, A. D., Parsopoulos, K. E., Vrahatis, M. N. (2008, July). Particle filtering with particle swarm optimization in systems with multiplicative noise. In Proceedings of the 10th annual conference on Genetic and evolutionary computation (pp. 57-62). ACM.

[33] Zhao, J., Li, Z. (2010). Particle filter based on Particle Swarm Optimization resampling for vision tracking. Expert Systems with Applications, 37(12), 8910-8914.

[34] Doucet, A., Godsill, S., Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and computing, 10(3), 197-208.

[35] Wu, Y., Lim, J., Yang, M. H. (2013). Online object tracking: A benchmark. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2411-2418).

[36] Gordon, N., Salmond, D., Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEE Proceedings F Radar and Signal Processing, vol. 140, no. 2, p. 107.