Multiple Object Tracking Via Species-Based Particle Swarm Optimization

Xiaoqin Zhang, Weiming Hu, *Senior Member, IEEE,* Wei Qu, *Member, IEEE,* and Steve Maybank, *Senior Member, IEEE*

***Abstract*—Multiple object tracking is particularly challenging when many objects with similar appearances occlude one another. Most existing approaches concatenate the states of different objects, view the multi-object tracking as a joint motion estimation problem and search for the best state of the joint motion in a rather high dimensional space. However, this centralized framework suffers from a high computational load. We bring a new view to the tracking problem from a swarm intelligence perspective. In analogy with the foraging behavior of bird flocks, we propose a species-based particle swarm optimization algorithm for multiple object tracking, in which the global swarm is divided into many species according to the number of objects, and each species searches for its object and maintains track of it. The interaction between different objects is modeled as species competition and repulsion, and the occlusion relationship is implicitly deduced from the "power" of each species, which is a function of the image observations. Therefore, our approach decentralizes the joint tracker to a set of individual trackers, each of which tries to maximize its visual evidence. Experimental results demonstrate the efficiency and effectiveness of our method.**

***Index Terms*—Multiple object tracking, particle swarm optimization.**

I. Introduction

MULTIPLE object tracking in videos is one of the most important problems in many emerging applications, such as surveillance, intelligent transportation, human-computer interface, and video analysis. Due to its crucial value in these applications, many efforts have been made to solve this problem in the recent decades [1]–[26]. Most notably, two influential methods were proposed in the early days: the multiple hypothesis tracker [1] and the joint probabilistic data association filter [2]. Recent years have witnessed great advances in multiple object tracking in the context of computer vision [3]–[26]. According to whether the stationary-camera environments are needed, these multiple object tracking algorithms can be roughly divided into two categories: stationary and non-stationary.

A. Multiple Object Tracking with Stationary Cameras

In this category, the tracking task is conducted under stationary cameras. Different background subtraction techniques are usually employed to obtain prior information about the positions of moving objects. BraMBLe [3], a Bayesian multi-blob tracker, combines a multi-blob likelihood function with particle filters. In this paper, the multi-blob likelihood function is learned from both a foreground model and a background model. In [4], Zhao and Nevatia proposed a multiple human tracking algorithm for crowded scenes, where the Markov chain Monte Carlo (MCMC) technique is used to estimate the state and the number of objects. In [5] and [6], enumeration strategies are used to achieve the best data association in the multiple object tracking when different objects merge or split. Bose *et al.* [7] used a graph model to solve the labeling problem in case of merging and splitting. Their algorithm c deal with track creation, confirmation, occlusion, and deletio However, the computational cost grows exponentially with the number of objects. To avoid enumerating all possible solutions to the labeling problem, Yu *et al.* [8] proposed a spatio-temporal MCMC data association algorithm to sample the solution space efficiently. In [9], color, texture, and motion are combined into a unified distance measure, aiming at making the trackers more robust and reducing the errors in assignment during occlusions. Ishiguro *et al.* [10] assumed that the type of target motions can be classified using a few distinct motion models. They use a switching dynamic model in a number of target trackers. Song *et al.* [11] proposed an on-line supervised learning-based algorithm for tracking multiple interactive targets. Moreover, many approaches exploit the information of multiple cameras to overcome identity assignment errors. One strategy is to model positions of objects on a discrete occupancy grid, and use dynamic programming to globally optimize the trajectories of the objects [12]. Alternatively, Khan and Shah [13] proposed a multiple occluding people tracking method by localizing on multiple scene planes. A planar homography occupancy constraint and

Manuscript received February 24, 2010; revised June 7, 2010 and July 28, 2010; accepted August 19, 2010. Date of publication October 14, 2010; date of current version November 5, 2010. This work was supported in part by the National Natural Science Foundation of China, under Grants 60825204 and 60935002, and the National 863 High-Tech Research and Development Program of China, under Grant 2009AA01Z318. This paper was recommended by Associate Editor P. L. Correia.

X. Zhang is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100864, China, and also with the College of Mathematics and Information Science, Wenzhou University, Zhejiang 325035, China (e-mail: xqzhang@wzu.edu.cn).

W. Hu is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: wmhu@nlpr.ia.ac.cn).

W. Qu is with the School of Information Science and Engineering, Chinese Academy of Sciences, Beijing 100190, China (e-mail: quweiusa@gmail.com).

S. Maybank is with the School of Computer Science and Information Systems, Birkbeck College, London WC1E 7HX, U.K. (e-mail: sjmaybank@dcs.bbk.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2010.2087455

the foreground likelihood information extracted from different views are combined to tackle the occlusion problem. However, fusion of information from different cameras would be very time consuming and thus not practical for a real-time system. Although the above algorithms achieve a good performance in the multiple object tracking problem, the requirement of using only stationary cameras greatly limits their applications.

B. Multiple Object Tracking with Non-Stationary Camera

Methods in this category do not need background information, thus they have a wider range of applications. MacCormick and Blake [14] developed a probabilistic exclusion principle to solve the association problem in multiple object tracking, but it can only be applied to pairs of objects. Khan *et al.* [15] proposed an MCMC-based particle filter which uses a Markov random field to model the objects' motion interaction, but their model cannot deal with occlusion. Yu *et al.* [16] proposed a collaborative tracking algorithm for multiple objects which models the objects' joint prior probability distribution by a Markov random network to solve the identity maintenance problem. Qu *et al.* [17] suggested an interactively distributed multi-object tracking algorithm using a magnetic-inertia potential model to solve the multiple object labeling problem in the presence of occlusions. In [18], the spatio-temporal context of each object is used to maintain the correct identification of the object. Nillius *et al.* [19] proposed a method to resolve multiple hypotheses via Bayesian networks and a novel solution is obtained by belief propagation techniques. In [20], a linear programming relaxation algorithm is proposed for multiple object tracking. This paper views the multiple object tracking as a multi-path searching problem by explicitly modeling the track interaction and objects' mutual occlusion. Yang *et al.* [21] proposed a game-theoretic multiple target tracking algorithm, in which the tracking problem is solved by finding the Nash equilibrium of a game. It can decentralize the joint tracker and uses computational resources efficiently. Another solution to overcome the curse of dimensionality in tracking multiple objects jointly is the variational particle filter proposed by Jin *et al.* [22], where the proposal distribution is based on an approximated posterior obtained by variational inference. In recent years, tracking-by-detection-based methods have become popular in multiple object tracking, especially in pedestrian tracking [23]–[26]. These methods first detect the pedestrians using an off-line learned pedestrian detector, and then assign the detection responses to the tracked trajectories using different data association strategies, e.g., cognitive feedback to visual odometry [23], min-cost flow network [24], hypothesis selection [25], and the Hungarian algorithm [26]. These tracking-by-detection-based methods are applicable even for moving cameras, but their performance greatly depends on the accuracy of pedestrian detection.

Despite the increasing amount of work done in multiple object tracking above, they do not investigate the key points which determine the performance of multiple object tracking in a general and theoretical way, and thus these work cannot be extended to the general cases and provide a theoretical guidance for designing more effective multiple object tracking algorithms.

C. Our Work

Recently, particle swarm optimization (PSO) [27], [28], a new population-based stochastic optimization technique, has received much attention because of its considerable success. Unlike the independent particles in the particle filter, the particles in PSO interact locally with one another and with their environment in analogy with the cooperative and social aspects of animal populations, e.g., as found in birds flocking. Starting from a diffuse population (or "swarm"), individuals (or "particles") tend to move in the state space and eventually cluster in regions where optimal states are located. The advantages of this mechanism are, on the one hand, the robustness and sophistication of the obtained group behavior and, on the other hand, the simplicity and low cost of the computation associated with each particle.

Inspired by the foregoing discussions, we propose a species-based PSO algorithm for multiple object tracking, where the global particle swarm is divided into several species according to the number of objects. The main contributions of the proposed tracking algorithm are summarized as follows.

1) We propose an annealed Gaussian-based PSO algorithm. Compared with the conventional PSO, it has two major merits: 1) a big reduction in the number of parameters—only one single annealing parameter needed, and 2) while maintaining a comparable performance, it converges much faster (see Section VI-A).
2) A species concept is introduced into the PSO framework to extend it to multiple object tracking. The particles are divided into species so that each species corresponds to one of the objects. The occlusion between different objects is modeled as species competition, and the occlusion relationship is implicitly deduced from the power of each species. Meanwhile, a repulsion force is employed to prevent the particles in one species from being miss-attracted by other species. As a result, the joint tracker can be decentralized to individual trackers, which try to maximize their visual evidence.
3) In order to investigate the decisive factors of multiple object tracking performance, we first derive the detailed form of the "optimal" importance proposal distribution for the state of an object in case of occlusion from a sequential Monte Carlo sampling view, and then show that our algorithm is an effective approximation of the sampling results from the "optimal" importance proposal distribution. This theoretical analysis provides a guidance for designing a robust multiple object tracking algorithm.
4) The appearances of objects under occlusion are carefully updated according to the reconstruction errors of the subspace-based appearance models. Thus, an object emerging from severe occlusion can be successfully reacquired after occlusion.

This paper is organized as follows. The motivation of our approach is given in Section II. A brief introduction of the traditional PSO algorithm is presented in Section III. In Section IV, we show the details of our proposed tracking approach. The theoretical analysis of our algorithm is shown

in Section V. Experimental results are shown in Section VI, and Section VII is devoted to conclusion.

II. Motivation

A. Single Object Tracking from the Biological Swarm Intelligence Viewpoint

First, we define the following analogies: 1) the groundtruth state of an object and its support region are viewed as ecological resources (e.g., food); 2) the particles in state space correspond to a certain animal (e.g., bird); and 3) the observation likelihood associated with each particle is analogous to the fitness ability of an individual animal to detect the resource. Then the single object tracking problem can be viewed in the following way: suppose that a group of particles (birds) are randomly generated in the image (state space), and none of the particles (birds) knows where the object (food) is. But each particle (bird) knows how far it is from the object (food) by evaluating the observation (fitness ability) in each iteration. What is the best strategy to find the object (food), and how can the information obtained by each particle (bird) be used efficiently? The PSO framework [27], [28], inspired by the swarm intelligence of birds flocking, provides an effective way to answer these questions.

B. Extended to Multiple Object Tracking

When the multiple objects are separated, the mechanism in Section II-A for single object tracking can be easily extended to multiple object tracking by creating a tracker for each object, and conducting these trackers independently. If the objects move close together and even occlude each other, these independent single object trackers may fail. As mentioned in Section II-A, the support regions of objects are analogous to ecological resources, e.g., food. If occlusion happens between two objects, their support regions overlap, which means that the overlap part is the resource needed by both species. Consequently, the competition and repulsion between these two species arise as they compete for this part of the resource, and the stronger species may have a higher probability of winning the competition. From the discussions of the relationship between multiple object tracking and biological swarming, we find that our assumptions and analogies are reasonable and tractable.

In the following two sections, we first briefly review the traditional PSO algorithm, and then give a detailed description of the multiple species-based PSO tracking algorithm.

III. Particle Swarm Optimization

PSO [27] is a population-based stochastic optimization technique, which is inspired by the social behavior of bird flocking. In detail, a PSO algorithm is initialized with a group of random particles $\{x^{i,0}\}_{i=1}^{N}$ (N is the number of particles). Each particle $x^{i,0}$ has a corresponding fitness value $f(x^{i,0})$ and a relevant velocity $v^{i,0}$, which is a function of the best state found by that particle (p^i, for individual best), and of the best state found so far among all particles (g, for global best).

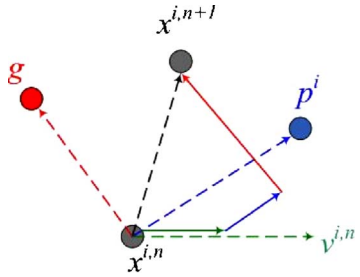


Fig. 1. Vector space schematic diagram for the nth iteration of particle i. The dashed lines represent the vector directions of the three terms on the right-hand side of (1); the solid lines represent the scales of the three terms on the right-hand side of (1).

Given these two best values, the particle updates its velocity and state in the nth iteration as follows:

$$v^{i,n+1} = w^n v^{i,n} + \varphi_1 u_1 (p^i - x^{i,n}) + \varphi_2 u_2 (g - x^{i,n}) \quad (1)$$

$$x^{i,n+1} = x^{i,n} + v^{i,n+1} \quad (2)$$

where w^n is the inertial weight, the φ_1, φ_2 are acceleration constants, and $u_1, u_2 \in (0, 1)$ are uniformly distributed random numbers. The inertial weight w^n is usually a monotonically decreasing function of the iteration n. For example, given a user-specified maximum weight w_{max}, a minimum weight w_{min} and the initialization of $w^0 = w_{max}$, one way to update w^n is as follows:

$$w^{n+1} = w^n - dw, \quad dw = (w_{max} - w_{min})/T \quad (3)$$

where T is the maximum iteration number.

As shown in Fig. 1, the three terms on the right-hand side of (1) represent inertial velocity, cognitive effect, and social effect, respectively, where cognitive effect refers to the evolution of the particle according to its own observations, and social effect refers to the evolution of the particle according to the cooperation between all particles.

After the fitness value of each particle $f(x^{i,n})$ is evaluated, the individual best and the global best of particles are updated as follows:

$$p^i = \begin{cases} x^{i,n}, & \text{if } f(x^{i,n}) > f(p^i) \\ p^i, & \text{else} \end{cases} \quad (4)$$

$$g = \arg\max_{p^i} f(p^i). \quad (5)$$

In analogy with the foraging behavior of the bird flocks, here the optimal state of $f(\cdot)$ corresponds to food, and the particles in state space correspond to birds.

As a result, the particles interact locally with one another and with their environment in analogy with the "cognitive" and "social" aspects of animal populations, and eventually cluster in the regions where the local optima of $f(\cdot)$ are located.

IV. Proposed Tracking Algorithm

In our tracking algorithm, the motion of a tracked object between two consecutive frames is approximated by a set of affine parameters $x = (t_x, t_y, \theta, s, \alpha, \beta)$, where $\{t_x, t_y\}$ denote the 2-D translation parameters and $\{\theta, s, \alpha, \beta\}$ are deformation

parameters. A particle is a sample from the affine parameter space and its fitness value is evaluated by a subspace-based appearance model [30]. In the following parts, we first introduce the incremental subspace learning-based appearance model, and then give a detailed description of the proposed multiple object tracking algorithm.

A. Incremental Subspace Learning-Based Appearance Model

In this section, we introduce a subspace-based appearance model [30] for observation evaluation, which models the appearance of an object by incrementally learning a low-order eigenspace representation.

1) *Incremental Subspace Learning of Object Appearance:* Let matrix $A = \{o_1, \ldots, o_t\}$ be the image observations of an object appearance up to time t, where each column o_i is the image observation of the object in the ith frame. Let $A = U\Sigma V^T$ be the singular value decomposition (SVD) of matrix A, and U is the subspace of the object appearance up to time t. Let $E = \{o_{t+1}, \ldots, o_{t+m}\}$ be the m subsequent image observations of the object after tracking m frames. Now the problem of incremental subspace learning of the object appearance is defined as follows: given only U and E, how can we incrementally learn the subspace of the object appearance at time $t+m$?

The R-SVD algorithm [31] is an effective tool to compute the SVD of the matrix $A' = (A|E) = U'\Sigma'V'^T$ based on the SVD of A. The details are as follows.

1) Perform an orthonormalization process on the matrix $(U|E)$, yielding the columns orthonormal matrix $\tilde{U} = (U|\tilde{E})$.
2) Let $\tilde{V} = \begin{pmatrix} V & 0 \\ 0 & I_m \end{pmatrix}$ be a $(t+m)\times(t+m)$ matrix, where I_m is an $m \times m$ identity matrix.
3) Let $\tilde{\Sigma} = \tilde{U}^T A' \tilde{V} = \begin{pmatrix} U^T \\ \tilde{E} \end{pmatrix} (A|E) \begin{pmatrix} V & 0 \\ 0 & I_m \end{pmatrix} = \begin{pmatrix} U^T AV & U^T E \\ \tilde{E}^T AV & \tilde{E}^T E \end{pmatrix}$. Note that $U^T AV = \Sigma$ and $\tilde{E}^T AV = 0$. Then $\tilde{\Sigma}$ becomes $\begin{pmatrix} \Sigma & U^T E \\ 0 & \tilde{E}^T E \end{pmatrix}$.
4) Compute the SVD of $\tilde{\Sigma}$: $\tilde{\Sigma} = \hat{U}\hat{\Sigma}\hat{V}^T$ and the SVD of A' is

$$A' = \tilde{U}(\hat{U}\hat{\Sigma}\hat{V}^T)\tilde{V}^T = (\tilde{U}\hat{U})\hat{\Sigma}(\hat{V}^T\tilde{V}^T) = U'\Sigma'V'^T$$

where $U' = \tilde{U}\hat{U}$, $\Sigma' = \hat{\Sigma}$, $V'^T = \hat{V}^T\tilde{V}^T$.

In this way, the R-SVD algorithm computes the new eigenbasis efficiently. Therefore, the subspace of an object appearance U can be incrementally updated during the tracking process.

2) *Observation Likelihood:* As shown in Section IV-A1, the subspace of the object U is learned from the observations of previous tracking results. The observation likelihood is defined based on the reconstruction error of the observation candidates with respect to U as follows: in the current image frame t, an observation candidate o_t is generated by warping the image according to its corresponding particle state x_t, and then the reconstruction error of the observation o_t with respect to the object subspace U is calculated as follows:

$$RE = ||o_t - UU^T o_t||^2. \tag{6}$$

As a result, the observation likelihood is naturally formed as follows:

$$p(o_t|x_t) = \exp(-RE). \tag{7}$$

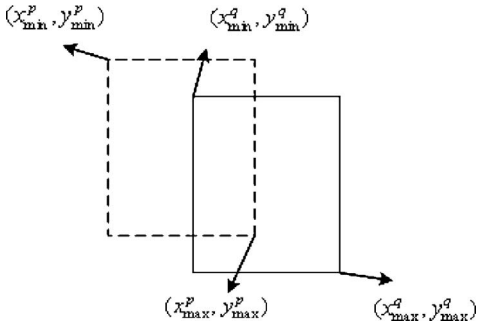


Fig. 2. Schematic diagram of occlusion detection; the dashed box is the candidate region p, the solid box is the candidate region q.

B. Multiple Object Tracking Algorithm

As stated in Section II-B, in the multiple object tracking case, the observations of different objects may overlap during occlusion, and the correspondences between objects and their features become ambiguous. To overcome this difficulty, we propose a multiple species-based PSO algorithm. The fundamental idea of the proposed algorithm is to divide the particles into several species according to the number of objects, and effectively model the interactions and the occlusions between different species.

Below, we give a detailed description of our algorithm which contains the following parts: 1) problem formulation; 2) competition and repulsion model; 3) annealed Gaussian-based PSO; and 4) selective updating for the appearance model.

1) *Problem Formulation:* Let us recall the symbols for the states and observations $\mathcal{X} = \{x^{i,n}_{t,k}, i = 1, \ldots, N, k = 1, \ldots, M\}$, $\mathcal{O} = \{o^{i,n}_{t,k}, i = 1, \ldots, N, k = 1, \ldots, M\}$, where the symbols i, n, t, and k represent the ith particle, the nth iteration, the tth image frame, and the kth object, respectively. Correspondingly, N is the number of particles and M is the number of objects.

Before introducing the problem formulation of multiple object tracking, the implementation of occlusion detection is first illustrated as follows: given a candidate state $x^{i,n}_{t,k_1}$ of object k_1, there is a left-top point (x_{min}, y_{min}) and a right-bottom point (x_{max}, y_{max}) in the candidate observation region $o^{i,n}_{t,k_1}$ corresponding to $x^{i,n}_{t,k_1}$. For the candidate p and candidate q, as shown in Fig. 2, we have two sets of points as follows: (x^p_{min}, y^p_{min}), (x^p_{max}, y^p_{max}) and (x^q_{min}, y^q_{min}), (x^q_{max}, y^q_{max}), and the occlusion relationship between the candidate p and the candidate q is formulated as

$$flag = max(x^p_{min}, x^q_{min}) < min(x^p_{max}, x^q_{max})$$

$$\&\&max(y^p_{min}, y^q_{min}) < min(y^p_{max}, y^q_{max})$$

where $flag = True$ means that occlusion happens, while $flag = False$ means that no occlusion happens.

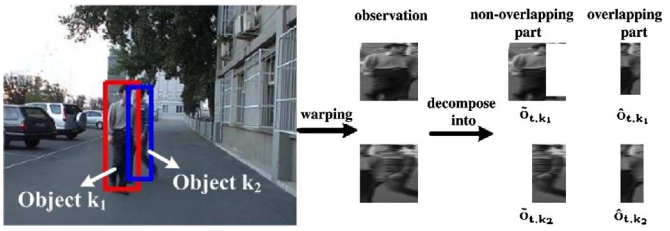


Fig. 3. Observation decomposition of two objects under occlusion (left: original image, middle: the warped observation, right: the observation decomposition).

In our paper, the multiple object tracking problem can be formulated as follows:

$$\mathcal{X}^* = \arg\max_{\mathcal{X}} p(\mathcal{O}|\mathcal{X}). \tag{8}$$

If no occlusion happens, the above optimization problem can be simplified by maximizing the individual observation likelihood independently (here, we drop the superscript i, n for simplicity) as follows:

$$x^*_{t,k} = \arg\max_{x_{t,k}} p(o_{t,k}|x_{t,k}), k = 1, \ldots, M. \tag{9}$$

If occlusion happens between objects k_1 and k_2, as shown in Fig. 3, we divide the observations of object under occlusion into two parts: 1) non-overlapping part $\tilde{o}_{t,k}$, and 2) overlapping part $\hat{o}_{t,k}$. Then the tracking problem of these two objects can be formulated as follows:

$$x^*_{t,k_1} = \arg\max_{x_{t,k_1}} p(\tilde{o}_{t,k_1}|x_{t,k_1}) p(\hat{o}_{t,k_1}|x_{t,k_1}, x_{t,k_2}) \tag{10}$$

$$x^*_{t,k_2} = \arg\max_{x_{t,k_2}} p(\tilde{o}_{t,k_2}|x_{t,k_2}) p(\hat{o}_{t,k_2}|x_{t,k_2}, x_{t,k_1}) \tag{11}$$

where $p(\hat{o}_{t,k_1}|x_{t,k_1}, x_{t,k_2})$ and $p(\hat{o}_{t,k_2}|x_{t,k_2}, x_{t,k_1})$ are the interactive likelihood of the corresponding object on the overlapping part, respectively. The (10) and (11) are iteratively computed until convergence. The occlusion between three or more objects can be formulated similarly.

2) *Competition and Repulsion Model:*

Competition Model: When occlusion between different objects happens, the corresponding support regions may overlap (see Fig. 3). In this case, the two species compete for the overlapping part. The question is how to effectively model the competition phenomenon when occlusion happens.

In order to answer the above question, we first need to tie the visual problem to this phenomenon, and model the detail of the competition process. Before introducing our model, we first discuss the two related works, [17] and [21]. Although they both model the interactions among the objects through the observations, the detailed models are very different from ours. In [17], the interaction is modeled using the whole support of the observation region, not just the overlapping part. This is a little unreasonable, since the competition only happens in the overlapping region, and the effectiveness of the model may be diluted by the non-overlapping regions. In contrast, [21] models the interference only using the overlapping part. However, the model focuses on the pixel level. It may not be very robust when the interacting objects have a similar color or are under severe occlusion. In this paper, we view the overlapping part as a whole and project it onto the corresponding part of the learned subspace of each object (see Fig. 4). Meanwhile, the fitness value on the overlapping part is evaluated as the competition ability. As a result, the power of each species is defined as follows:

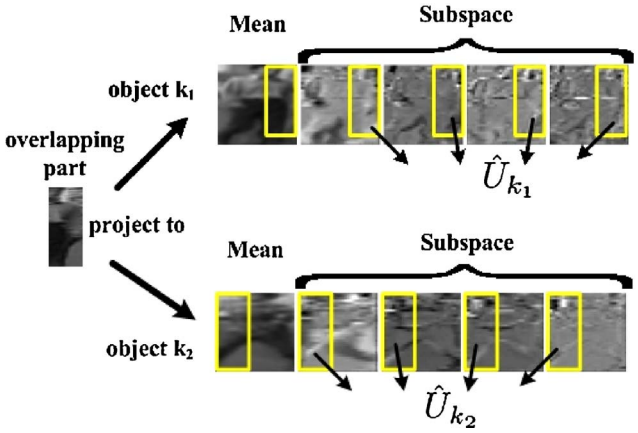


Fig. 4. Project the overlapping part to the corresponding part of the learned subspaces; $\hat{U}_{k_i}$ in the region with solid lines is the corresponding subspace of the overlapping part of the object k_i, $i = 1, 2$.

$$power^{k_i} = p(\hat{o}_{t,k_i}|x_{t,k_i}) = exp(-||\hat{o}_{t,k_i} - \hat{U}_{k_i}\hat{U}^T_{k_i}\hat{o}_{t,k_i}||^2) \tag{12}$$

where $\hat{U}_{k_i}$ is the the corresponding subspace of the overlapping part of the object $k_i, i = 1, 2$. Consequently, the interactive likelihood $p(\hat{o}_{t,k_1}|x_{t,k_1}, x_{t,k_2})$ of object k_1 on the overlapping parts can be obtained[1] as follows:

$$p(\hat{o}_{t,k_1}|x_{t,k_1}, x_{t,k_2}) = \frac{power^{k_1}}{\sum_{i=1,2} \cdot power^{k_i}}. \tag{13}$$

The competition ability can be described by the interactive likelihood for each species. A species with higher competition ability is more likely to win the competition, which means that the object corresponding to this species is more likely to be the one occluding other objects. We will validate this conclusion through the experiments (see Section VI-B2).

Species Repulsion: Generally, multiple object tracking algorithms suffer from the the well-known coalescence problem during occlusions, where a tracker loses its associated object and mistakenly tracks other objects. While in the real world, the stronger species repels other species and tries to take up all the resources. In order to tackle the coalescence problem, we need to define a repulsion model for the objects under occlusion. When occlusion happens between objects k_1 and k_2, the repulsion force from object k_2 to object k_1 is defined as follows:

$$F_{\overrightarrow{k_2, k_1}} = p(\hat{o}_{t,k_2}|x_{t,k_1}, x_{t,k_2}) V_{k_1} \tag{14}$$

where V_{k_1} is the velocity vector of object k_1. The scale parameter $p(\hat{o}_{t,k_2}|x_{t,k_1}, x_{t,k_2})$ is determined by the competition ability of object k_2, representing the intensity of the repulsion force. The species repulsion model means that the species with a larger competition ability repels other species nearby with a larger force.

[1]Here, we also assume that the occlusion happens between object k_1 and object k_2.

This repulsion force is added to the particle evolution process (see Section IV-B3) to prevent the particle from being miss-attracted by other species, and thus to maintain the diversity among the species. In this way, the competition model is incorporated into the particle evolution process, thus alleviating the coalescence problem.

3) *Annealed Gaussian-Based PSO:* In the traditional PSO algorithm, there are several parameters to be tuned: inertial weights w^n, acceleration constants φ_1, φ_2, constriction factor $\mathcal{R}$, and maximum velocity v^{max}. It lacks a mechanism to control these parameters, which fosters the danger of swarm explosion and divergence especially in high dimensions. Therefore, we propose an annealed Gaussian-based particle swarm optimization (AGPSO) algorithm, in which the particles and their velocities are updated as follows:

$$v^{i,n+1} = |r_1|(p^i - x^{i,n}) + |r_2|(g - x^{i,n}) + \epsilon \tag{15}$$

$$x^{i,n+1} = x^{i,n} + v^{i,n+1} \tag{16}$$

where $|r_1|$ and $|r_2|$ are the absolute values of the independent samples from the Gaussian probability distribution $\mathcal{N}(0, 1)$, and ϵ is the zero-mean Gaussian perturbation noise which prevents the particles from becoming trapped in local optima. The covariance matrix of ϵ is changed in an adaptive simulated annealing way [32] as follows:

$$\Sigma_\epsilon = \Sigma e^{-cn} \tag{17}$$

where Σ is the covariance matrix of the predefined transition distribution, c is an annealing constant, and n is the iteration number. The elements in Σ_ϵ decrease rapidly as the iteration number n increases which enables a fast convergence rate.

If occlusion happens between object k_1 and k_2 at time t, we add a repulsion force to the particle evolution process, and then the iteration form for object k_1 becomes as follows:

$$v_{t,k_1}^{i,n+1} = |r_1|(p_{t,k_1}^i - x_{t,k_1}^{i,n}) + |r_2|(g_{t,k_1} - x_{t,k_1}^{i,n}) + |r_3| F_{\overrightarrow{k_2, k_1}} + \epsilon \tag{18}$$

$$x_{t,k_1}^{i,n+1} = x_{t,k_1}^{i,n} + v_{t,k_1}^{i,n+1} \tag{19}$$

where r_3 is a Gaussian random number sampled from $\mathcal{N}(0, 1)$. The third term on the right-hand side of the above equation represents the repulsion effect of object k_2 on object k_1.

In summary, our approach models the competition phenomenon on the observation level and models competition effect on the state space to guide the evolution process of object state. Experimental results show that our model is reasonable.

4) *Selective Updating for Appearance Model:* In most multiple tracking algorithms, updating of the appearance model is stopped during occlusions. However, if an object appearance changes during occlusions, the tracker may fail to reacquire this object after the occlusion. In this paper, we design a selective updating scheme to accommodate the appearance changes during occlusion: 1) as shown in the Fig. 3, pixels in the non-overlapping part of objects are incrementally updated in the normal way, and 2) pixels in the overlapping part are projected back to the corresponding subspace of each object (see Fig. 4) and the reconstruction errors are calculated as follows:

$$R = ||\hat{o}_{t,k} - \hat{U}_k \hat{U}_k^T \hat{o}_{t,k}||^2. \tag{20}$$

If the reconstruction error of a pixel inside the overlapping part is smaller than a predefined threshold, then it is updated to the corresponding subspace.

C. Algorithm Summary

Our proposed tracking algorithm is summarized as follows.[2]

1: Initialization: $t = 0$, the states of the multiple objects are manually initialized as the global best for species $\{g_{t,k}\}_{k=1}^M$. The individual best $\{p_{t,k}^i\}_{i=1}^N$ are set equal to $g_{t,k}$.
2: **while** $t = 1, 2, \ldots$ **do**
3: Check occlusions among $\{g_{t-1,k}\}_{k=1}^M$, e.g. occlusion between g_{t-1,k_1} and g_{t-1,k_2} is detected.
4: Randomly propagate the particles to enhance their diversities within the species according to the following transition model

$$x_{t,k}^{i,0} \sim \mathcal{N}(p_{t-1,k}^i, \Sigma_k)$$

where Σ_k is the covariance matrix of the Gaussian-transition distribution for the kth object.
5: **for** $n = 1, 2, \ldots, T$ **do**
6: Carry out the PSO evolution for object k_1

$$v_{t,k_1}^{i,n+1} = |r_1|(p_{t,k_1}^i - x_{t,k_1}^{i,n}) + |r_2|(g_{t,k_1} - x_{t,k_1}^{i,n}) + |r_3| F_{\overrightarrow{k_2, k_1}} + \epsilon$$

$$x_{t,k_1}^{i,n+1} = x_{t,k_1}^{i,n} + v_{t,k_1}^{i,n+1}.$$

7: Evaluate the fitness values by the observation model and the interactive model

$$f(x_{t,k_1}^{i,n+1}) = p(\tilde{o}_{t,k_1}|x_{t,k_1}^{i,n+1}) p(\hat{o}_{t,k_1}|x_{t,k_1}^{i,n+1}, g_{t,k_2}).$$

8: Update the individual best of each particle and global best of all particles and the annealing parameter.
9: Carry out the similar procedure for object k_2 (other trackers are independently carried out without interactive part).
10: Check the convergence criteria: $f(g_{t,k_i}) > Th$ and $f(g_{t,k_i})$, $i = 1, 2$ changes little from previous iteration.
11: If the convergence criteria for the object k_i is satisfied, stop its iteration;
12: **end for**
13: Update the appearance model-based the visible parts and the corresponding reconstruction error.
14: Output the object states at time t: $\{g_{t,k}\}_{k=1}^M$
15: **end while**

[2]Here, we take pairwise occlusion as an example. The occlusion between three or more objects can be formulated similarly. We only show how the tracking process is conducted on objects under occlusion. The trackers for unoccluded objects are conducted independently without the interaction part.

V. Algorithm Analysis

In this section, we first derive the "optimal" importance proposal distribution for the state of object k_1 in case of occlusion from a sequential Monte Carlo sampling point of view, and show that our algorithm in Section IV-C is an effective approximation of the sampling results from the "optimal" importance proposal distribution.

A. Optimal Importance Proposal Distribution

Doucet *et al.* [35] have proved that the "optimal" importance proposal distribution for particle filter is $p(x_t|x^i_{t-1}, o_t)$ in the sense of minimizing the variance of the importance weights. However, in practice, it is impossible to use $p(x_t|x^i_{t-1}, o_t)$ as the proposal distribution in the nonlinear and non-Gaussian cases, since it is difficult to sample from $p(x_t|x^i_{t-1}, o_t)$ and to evaluate $p(o_t|x^i_{t-1}) = \int p(o_t|x_t)p(x_t|x^i_{t-1})dx_t$.

In tracking applications, if no occlusion happens between object k_1 and other objects, the "optimal" importance proposal distribution for object k_1 is $p(x_{t,k_1}|x^i_{t-1,k_1}, o_t)$ obviously. But if occlusion happens between object k_1 and other objects, e.g., occlusion happens between object k_1 and object k_2, the "optimal" importance proposal distribution for object k_1 is $p(x_{t,k_1}|x^i_{t-1,k_1}, o_t, g_{t,k_2})$.

Proposition 1*:* If occlusion happens between object k_1 and object k_2, $q(\cdot) = p(x_{t,k_1}|x^i_{t-1,k_1}, o_t, g_{t,k_2})$ is the "optimal" importance proposal distribution for the state of object k_1 in the sense of minimizing the variance of the importance weights.

Proof: When occlusion happens between object k_1 and object k_2, the observation o_t of the object k_1 is not independent to the state of object k_2 given the the state of object k_1. Denote the state of object k_2 as g_{t,k_2}, then the observation model for the object k_1 can be formulated as $p(o_t|x_{t,k_1}, g_{t,k_2})$. For the ith particle x^i_{t,k_1} of the object k_1, its importance weight w^i_t is calculated as follows:

$$w^i_t = w^i_{t-1}\frac{p(o_t|x^i_{t,k_1}, g_{t,k_2})p(x^i_{t,k_1}|x^i_{t-1,k_1})}{q(\cdot)}. \tag{21}$$

Thus, the variance of w^i_t is calculated as follows:

$$var_{q(\cdot)}(w^i_t) = E((w^i_t)^2) - E^2(w^i_t)$$

where $E(\cdot)$ is the expectation operator, and straightforward calculation yields

$$E((w^i_t)^2) = (w^i_{t-1})^2\int \frac{(p(o_t|x^i_{t,k_1}, g_{t,k_2})p(x^i_{t,k_1}|x^i_{t-1,k_1}))^2}{q(\cdot)}dx_{t,k_1}$$

$$E(w^i_t) = w^i_{t-1}\int p(o_t|x^i_{t,k_1}, g_{t,k_2})p(x^i_{t,k_1}|x^i_{t-1,k_1})dx_{t,k_1}$$

$$= w^i_{t-1}\int p(o_t|x^i_{t,k_1}, g_{t,k_2})p(x^i_{t,k_1}|x^i_{t-1,k_1}, g_{t,k_2})dx_{t,k_1}$$

$$= w^i_{t-1}p(o_t|x^i_{t-1,k_1}, g_{t,k_2}).$$

Thus

$$var_{q(\cdot)}(w^i_t) = (w^i_{t-1})^2\,[P - Q]$$

where

$$P = \int \frac{(p(o_t|x^i_{t,k_1}, g_{t,k_2})p(x^i_{t,k_1}|x^i_{t-1,k_1}))^2}{q(\cdot)}dx_{t,k_1}$$

$$Q = p^2(o_t|x^i_{t-1,k_1}, g_{t,k_2}).$$

This variance is zero for $q(\cdot) = p(x_{t,k_1}|x^i_{t-1,k_1}, o_t, g_{t,k_2})$.

Here, we find that the key points which determine the performance of multiple object tracking are twofold: 1) $p(o_t|x_{t,k_1}, g_{t,k_2})$, which needs to be robust to ambiguous observations, and 2) $p(x_{t,k_1}|x^i_{t-1,k_1}, o_t, g_{t,k_2})$, the optimal importance distribution. However, it is impossible to take $p(x_{t,k_1}|x^i_{t-1,k_1}, o_t, g_{t,k_2})$ as the proposal distribution for the similar reason of $p(x_t|x^i_{t-1}, o_t)$. So the question is, how to incorporate the current observation o_t and the state of object k_2 into the transition distribution $p(x_{t,k_1}|x_{t-1,k_1})$ of object k_1 to form an effective proposal distribution at a reasonable computational cost.

B. Hierarchical Sampling Strategy

Here, we investigate the particle evolution process of our algorithm in Section IV-C, and show that it is a two-stage sampling strategy to generate samples that approximate to the sampling results from the "optimal" proposal distribution $p(x_{t,k_1}|x^i_{t-1,k_1}, o_t, g_{t,k_2})$: first, the particles are sampled from the state transition distribution $p(x_t|x_{t-1})$, and second, the sampled particles evolve through the PSO iterations to obtain the final sampling results. In the particle filtering view, we can see that our strategy is essentially a hierarchical importance sampling. In the coarse sampling stage, the particles are first sampled from the state transition distribution as in conventional particle filters to enhance their diversity

$$x^{i,0}_{t,k} \sim \mathcal{N}(p^i_{t-1,k}, \Sigma_k). \tag{22}$$

In the fine sampling stage, the particles evolve through PSO iterations, and are updated according to the current observations. In fact, this is essentially a latent multilayer importance sampling process with an implicit proposal distribution. Let us focus on one PSO iteration as follows:

$$v^{i,n+1}_{t,k_1} = |r_1|(p^i_{t,k_1} - x^{i,n}_{t,k_1}) + |r_2|(g_{t,k_1} - x^{i,n}_{t,k_1}) + |r_3|F_{\overrightarrow{k_2,k_1}} + \epsilon \tag{23}$$

$$x^{i,n+1}_{t,k_1} = x^{i,n}_{t,k_1} + v^{i,n+1}_{t,k_1} \tag{24}$$

where r_1, r_2, and r_3 are random numbers sampled independently from the Gaussian probability distribution $\mathcal{N}(0, 1)$, and ϵ is a zero-mean Gaussian perturbation noise vector with covariance matrix Σ_ϵ. Suppose that $x_t \in \mathbb{R}^d$ is a d-dimensional state, the distribution of the lth element in the vector $|r_1|(p^i_{t,k_1} - x^{i,n}_{t,k_1})$ is as follows:

$$|r_1|(p^i_{t,k_1} - x^{i,n}_{t,k_1})_l$$

$$\sim \begin{cases} 2\mathcal{N}(0, (p^i_{t,k_1} - x^{i,n}_{t,k_1})^2_l)\ [0, +\infty), & \text{if } (p^i_{t,k_1} - x^{i,n}_{t,k_1})_1 \geq 0 \\ 2\mathcal{N}(0, (p^i_{t,k_1} - x^{i,n}_{t,k_1})^2_1)\ (-\infty, 0), & \text{else} \end{cases}$$

where $l = 1, \cdots, d$. So the distribution of $|r_1|(p^i_{t,k_1} - x^{i,n}_{t,k_1})$ is as follows:

$$|r_1|(p^i_{t,k_1} - x^{i,n}_{t,k_1}) \sim R_1 = 2\mathcal{N}(0, \Sigma_1)$$

$$\Sigma_1 = \begin{pmatrix} (p^i_{t,k_1} - x^{i,n}_{t,k_1})^2_1 & \mathbf{0} \\ \ddots & \\ \mathbf{0} & (p^i_{t,k_1} - x^{i,n}_{t,k_1})^2_d. \end{pmatrix}.$$

The field of definition is either on $[0, +\infty)$ or $(-\infty, 0)$, depending on the sign of the elements in $(p^i_{t,k_1} - x^{i,n}_{t,k_1})$. Similarly available

$$|r_2|(g_{t,k_1} - x^{i,n}_{t,k_1}) \sim R_2 = 2\mathcal{N}(0, \Sigma_2)$$

$$\Sigma_2 = \begin{pmatrix} (g_{t,k_1} - x^{i,n}_{t,k_1})^2_1 & \mathbf{0} \\ \ddots & \\ \mathbf{0} & (g_{t,k_1} - x^{i,n}_{t,k_1})^2_d \end{pmatrix}$$

$$|r_3|F_{\overrightarrow{k_2, k_1}} \sim R_3 = 2\mathcal{N}(0, \Sigma_3)$$

$$\Sigma_3 = \begin{pmatrix} (F_{\overrightarrow{k_2, k_1}})^2_1 & \mathbf{0} \\ \ddots & \\ \mathbf{0} & (F_{\overrightarrow{k_2, k_1}})^2_d \end{pmatrix}.$$

Together with $\epsilon \sim R_4 = \mathcal{N}(0, \Sigma_\epsilon)$, the implicit proposal distribution behind the PSO iteration in (23) and (24) is $R = R_1 * R_2 * R_3 * R_4$[3] with a translation by $x^{i,n}_{t,k_1}$, as shown in (24). Here, $*$ stands for convolution operator. Although we cannot obtain an explicit form of the proposal distribution, all the four parts in (23) are fused into the proposal distribution through convolution.

As shown in steps 7 and 8 in Section IV-C, the fitness value of each particle is evaluated by $p(\tilde{o}_{t,k_1}|x^{i,n+1}_{t,k_1})p(\hat{o}_{t,k_1}|x^{i,n+1}_{t,k_1}, g_{t,k_2})$, and used to update the individual best and the global best. In this way, the PSO iterations can naturally take the current observation o_t and the state g_{t,k_2} of the interactive object into consideration. Therefore, beginning with a coarse importance sampling stage from the state transition distribution $\mathcal{N}(p^i_{t-1,k}, \Sigma_k)$, the hierarchical sampling process can approximate to the optimal sampling from $p(x_{t,k_1}|x^i_{t-1,k_1}, o_t, g_{t,k_2})$.

VI. Experimental Results

In our implementation, each candidate image corresponding to a particle is warped to a 20×20 patch, and the feature is a 400-dimension vector of gray level values subjected to zero-mean-unit-variance normalization. The above algorithm is implemented using MATLAB on a P4-3.2G computer with 512M random access memory.

[3]Since the analytical form of R is not available, we called it a latent sampling process.

TABLE I
Experimental Results of State Estimation

Algorithm	MSE Mean	MSE Var	Time (s)
PSO	0.13019	0.044086	10.2087
AGPSO	0.060502	0.06852	6.8005

A. PSO Versus AGPSO

1) *State Estimation:* Our algorithm is first tested on a nonlinear state estimation problem, which is described as benchmark in many papers [36]. Consider the following nonlinear state transition model given as follows:

$$x_t = 1 + \sin(w\pi(t-1)) + \phi_1 x_{t-1} + v_{t-1}, \quad x_t \in \mathbb{R} \tag{25}$$

where v_{t-1} is a Gamma $\mathcal{G}a(3, 2)$ random variable modeling the process noise, and $w = 4e - 2$ and $\phi_1 = 0.5$ are scalar parameters. A non-stationary observation model is as follows:

$$y_t = \begin{cases} \phi_2 x_t^2 + n_t, & t \le 30 \\ \phi_3 x_t - 2 + n_t, & t > 30 \end{cases} \tag{26}$$

where $\phi_2 = 0.2, \phi_3 = 0.5$, and the observation noise n_t is sampled from a Gaussian distribution $\mathcal{N}(0, 0.00001)$. Given only the noisy observation y_t, the goal is to estimate the underlying state sequence x_t for $t = 1 \cdots 60$. Here, we compare AGPSO with traditional PSO [27]. The parameters in APSO and PSO are set as follows: $\Sigma = 0.8, c = 2$, $\varphi_1 = \varphi_2 = 1, w_{max} = 0.8, w_{min} = 0.1, T = 20$. Fig. 5 gives an illustration of the estimates generated from a single run of the two algorithms. As shown in Fig. 5(c), PSO fails at time steps 24 and 25, where the observation is severely contaminated by the noise. While AGPSO introduces an annealing-based random part, thus avoiding being trapped in local optimal. Meanwhile, the numbers of iterations required by two filters are shown in Fig. 5(b), from which we can see that AGPSO only needs 2.4327 iterations to converge on average, while the traditional PSO needs 3.6271 iterations. Since the result of a single run is a random variable, the experiment is repeated 100 times with re-initialization to generate statistical averages. Table I summarizes the performance of the two filters using the following statistics: the means, variances of the mean square error (MSE) of the state estimates and the average execute time for one run. It is obvious that the average accuracy of our algorithm is better than the traditional PSO. In addition, we can see that AGPSO achieves a much faster convergence rate than PSO.

2) *Single Object Tracking:* In this part, we conduct a comparison experiment between the traditional PSO [29] and the proposed AGPSO on single object tracking. Here, the particle number and the covariance matrix of the transition distribution are set to $\{N = 200, \Sigma = diag(8^2, 8^2, 0.02^2, 0.02^2, 0.002^2, 0.002^2)\}$. The same observation model is used for PSO and AGPSO. Fig. 6 shows the tracking performances of PSO and AGPSO on a fast moving face, together with graphs of the RMSE and convergence time, from which we can see that our proposed AGPSO achieves a better tracking accuracy and a much faster convergence rate than the traditional PSO.

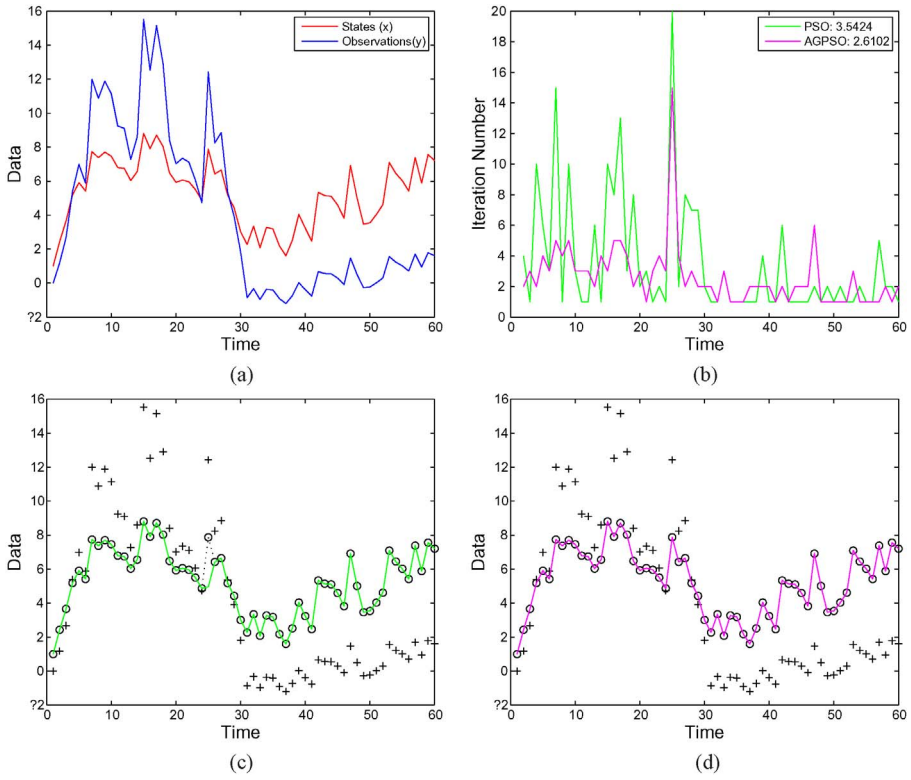


Fig. 5. Illustration of a single run of different filters. (a) True data. (b) Iteration number. (c) PSO. (d) AGPSO.

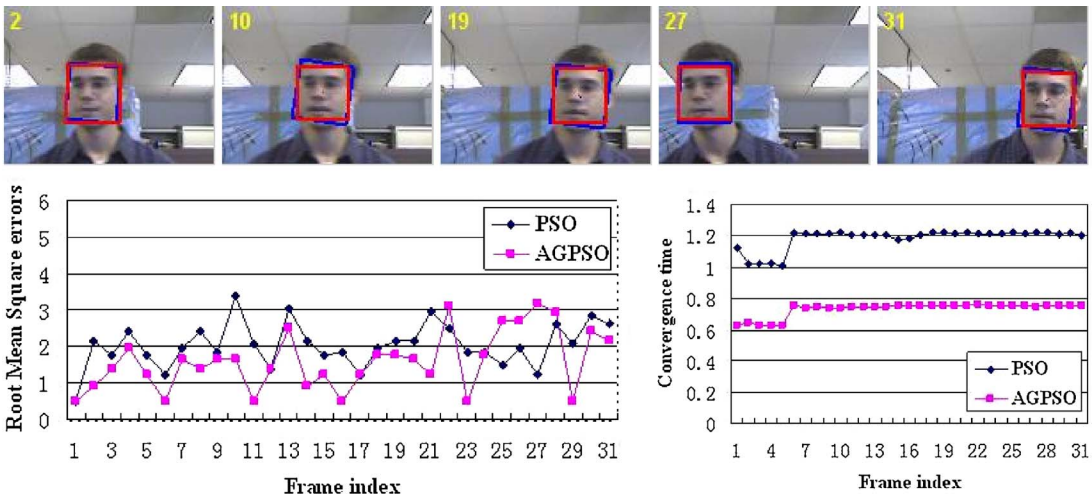


Fig. 6. First row: tracking performances; second row: root mean square error (RMSE) (left), convergence time (right) (blue: PSO, red: AGPSO).

The reason for the above experiments is that the velocity part employed in (1) carries little information, while the annealing part in our PSO iterations enhances the diversity of the particle set and its adaptive effect enables a fast convergence rate.

B. Multiple Object Tracking

In this section, we demonstrate three examples of tracking multiple objects with our method, and then give a summary of the experimental results.

1) *Example 1:* We test our method with a video sequence from the PETS 2004 database which is an open database for research on visual surveillance, available at http://homepages.inf.ed.ac.uk/rbf/CAVIAR/. The video in this example contains two walking people with severe occlusions. We conduct a comparison experiment between two appearance updating strategies during occlusions: no updating and selective updating. As shown in Fig. 7, we can see that with no updating for the appearance mode, the algorithm fails to track the person being occluded at frame 211 and can not recover the track after occlusions. The reason is that no account is taken of the gradual appearance changes of the man being occluded, and thus the correspondence of pixels between the man and the subspace is not accurate, leading to the tracking failure. In contrast, our selective updating strategy can follow the two people throughout the occlusion and maintain the

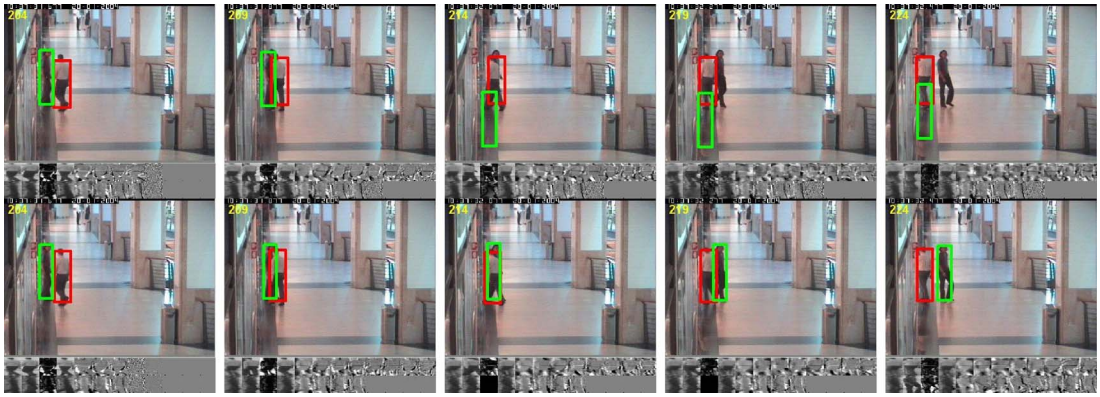


Fig. 7. Tracking two walking men with occlusion (first row: stop updating for the appearance model during occlusion; second row: selective updating for the appearance model during occlusion) for frames 204, 209, 214, 219, and 224.

correct identities. This is because the appearance changes are gradually updated before the object is completely occluded, and all the existing visual evidence is utilized to successfully reacquire the track after the occlusion.

2) *Example 2:* To validate the claimed contributions of our method, we conduct a quantitative evaluation comparison with the two tracking algorithms in [17] and [21], which share some similarities with our work, and furthermore, are respectively conducted in two influential frameworks: particle filter [33] and mean shift [34]. To make a fair comparison, the tracking algorithm in [17] is implemented with the same appearance model and updating scheme as in our work. Fig. 8 shows key frames where three people are tracked through occlusions (person A is tracked with a red window, person B is tracked with a green window, person C is tracked with a blue window), from which we can see that our algorithm handles the interaction and occlusion between different objects very well, while the tracking algorithms in [17] and [21] fail to track person A when he is occluded by person C with a similar appearance, because modeling the species competition on the overlapping part and dealing with it as a whole is more reasonable and robust. Our AGPSO framework is more likely to find the global optima than particle filter and mean shift methods. Fig. 9 shows the recovered occlusion relationships between different persons, where the horizontal axis is the frame index, and the vertical axis is the occlusion relationship. As illustrated in Fig. 9, our method can correctly deduce the occlusion relationship based on the interactive likelihood, and the results support the claim that the object with higher fitness value on the overlapping part is more likely to be the one occluding the other objects.

To further illustrate the advantages of our method, we conduct a quantitative comparison with [17] and [21] in the following aspects: number of frames in which tracking is successful, RMSE between the estimated position and the groundtruth,[4] the average tracking time. Table II shows the quantitative comparison. It is clear that the algorithms in [17] and [21] fail to track person A at frame 501, when he is severely occluded by person C who wears the similar clothes, while our method using the species competition and repulsion model can prevent the coalescence problem and succeeds in tracking throughout the sequence. Additionally, our method achieves a more accurate localization than other the two methods.

3) *Example 3:* This video sequence is also from the PETS 2004 database, and it is more challenging since it contains five walking people with continual occlusions and interactions. Fig. 10 illustrates some key frames where five persons are tracked through the occlusion. The persons are tracked accurately even though the occlusion simultaneously happens among the three persons, as in frames 277–340. We can see that our species competition and repulsion model is also effective in dealing with the occlusion among more than two objects. Besides the species competition and repulsion model, the selective updating of appearances during occlusion also provides an effective contribution to maintain the correct tracking identities in this video sequence.

4) *Example 4:* This part gives an example of tracking failure. As shown in Fig. 11, our algorithm fails to track the man in the green box at the frame 280. The reason for this failure is that the subspace-based appearance model cannot handle sudden changes in appearance, if the variations in the object appearance between consecutive frames are too large, then the algorithm will fail.

C. Summary

The underlying reasons for the above experimental results are discussed in this part. First, the species competition and repulsion force mechanism employed in our method provides a reasonable and effective solution to the interaction and occlusion problems in multiple object tracking. Second, the AGPSO framework is effective at searching for the optima, especially in high dimensions. Third, the carefully designed updating strategy can effectively accommodate the appearance changes while preventing the model from drifting away.

[4]The bounding box in the groundtruth does not fit the object closely, and so contains many background pixels as well as the object. Initialized by such bounding boxes is not suitable for a subspace-based tracking algorithm. Therefore, only the center position of each object in the groundtruth is used for evaluation.

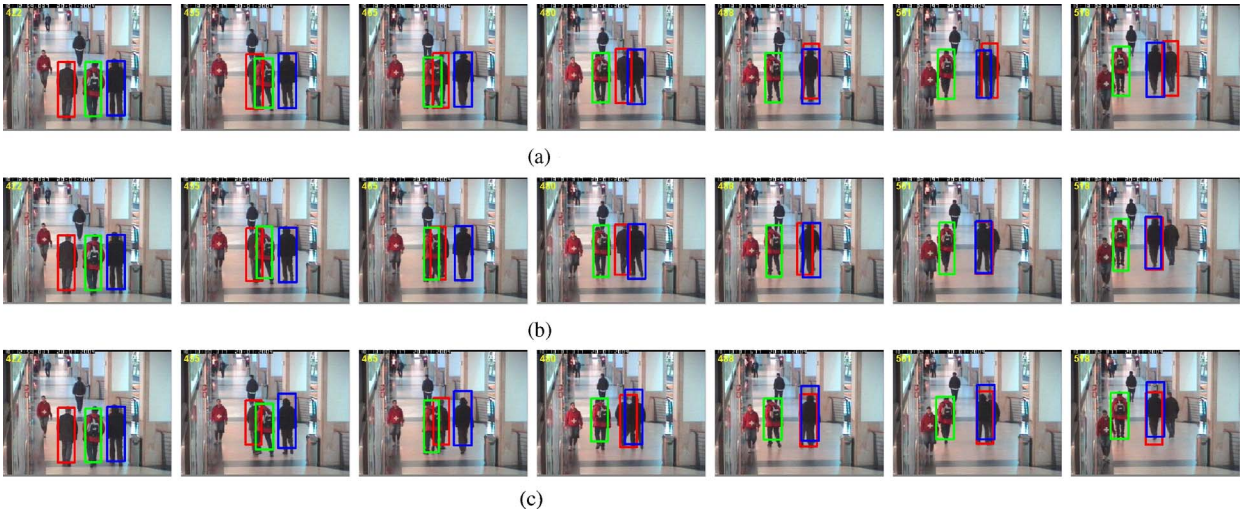


Fig. 8. Tracking performances with occlusion for frames 422, 455, 465, 480, 488, 501, and 518. (a) Our algorithm. (b) Qu's work [17]. (c) Yang's work [21].

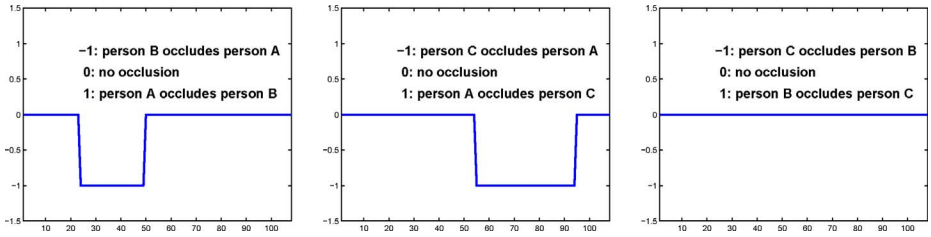


Fig. 9. Recovered occlusion relationship in Example 2.

TABLE II

QUANTITATIVE RESULTS OF OUR APPROACH AND ITS COMPARISON WITH YANG'S AND QU'S WORK

Approaches		Yang's Work	Qu's Work	Our Algorithm
Number of frames in which tracking is successful	Person A (red window)	80/101	80/108	108/108
	Person B (blue window)	108/108	108/108	108/108
	Person C (green window)	108/108	108/108	108/108
RMSE of position (by pixels)	Person A (red window)	12.9768	11.5537	3.6145
	Person B (blue window)	5.4128	4.8482	3.3087
	Person C (green window)	15.2104	2.6483	2.6262

Fig. 10. Tracking people in a shopping center for frames 218, 248, 272, 298, 359, and 406.

Fig. 11. Example of tracking failure (frames 205, 218, 230, 239, 257, and 280).

VII. Conclusion

This paper made an analogy between the tracking problem and the behavior of a flock of birds searching for food, and has proposed a species-based sequential PSO algorithm for multi-object tracking, in which the different species search for their associated objects (food) and track them once found. The occlusion between different objects was modeled as species competition and repulsion. In addition, we have proposed an annealed Gaussian PSO algorithm which is more effective than the traditional PSO algorithm. Unlike the joint tracker, our approach decentralizes the joint tracker, and the individual trackers are conducted for different objects, each of which tries to maximize its visual evidence. Experimental results demonstrate the efficiency and effectiveness of our method.

VIII. Acknowledgment

We would like to thank Dr. M. Yang for the help on implementing the algorithm.

References

[1] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automat. Contr.*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[2] Y. Bar-Shalom, T. Fortmann, and M. Scheffe, "Joint probabilistic data association for multiple targets in clutter," in *Proc. Int. Conf. Inform. Sci. Syst.*, 1980, pp. 404–409.

[3] M. Isard and J. MacCormick, "Bramble: A Bayesian multiple-blob tracker," in *Proc. Int. Conf. Comput. Vision*, 2001, pp. 34–41.

[4] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, vol. 2. Jul. 2004, pp. 406–413.

[5] Y. Huang and I. Essa, "Tracking multiple objects through occlusions," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, vol. 2. Jun. 2005, pp. 1051–1058.

[6] T. Yang, S. Li, Q. Pan, and J. Li, "Real-time multiple objects tracking with occlusion handling in dynamic scenes," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, vol. 1. Jun. 2005, pp. 970–975.

[7] B. Bose, X. Wang, and E. Grimson, "Multi-class object tracking algorithm that handles fragmentation and grouping," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2007, pp. 1–8.

[8] Q. Yu, G. Medioni, and I. Cohen, "Multiple target tracking using spatio-temporal Markov chain Monte Carlo data association," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2007, pp. 1–8.

[9] V. Takala and M. Pietikainen, "Multi-object tracking using color, texture and motion," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2007, pp. 1–7.

[10] K. Ishiguro, T. Yamada, and N. Ueda, "Simultaneous clustering and tracking unknown number of objects," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2008, pp. 1–8.

[11] X. Song, J. Cui, H. Zha, and H. Zhao, "Vision-based multiple interacting targets tracking via on-line supervised learning," in *Proc. Eur. Conf. Comput. Vision*, 2008, pp. 642–655.

[12] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 30, no. 2, pp. 267–282, Feb. 2008.

[13] S. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 31, no. 3, pp. 505–519, Mar. 2009.

[14] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," *Int. J. Comput. Vision*, vol. 39, no. 1, pp. 57–71, 2000.

[15] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-based particle filter for tracking multiple interacting targets," in *Proc. Eur. Conf. Comput. Vision*, 2004, pp. 279–290.

[16] T. Yu and Y. Wu, "Collaborative tracking of multiple targets," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, vol. 1. Jun. 2004, pp. 834–841.

[17] W. Qu, D. Schonfeld, and M. Mohamed, "Real-time distributed multi-object tracking using multiple interactive trackers and a magnetic-inertia potential model," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 511–519, Apr. 2007.

[18] H. Nguyen, Q. Ji, and A. Smeulders, "Spatio-temporal context for robust multitarget tracking," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 29, no. 1, pp. 52–64, Jan. 2007.

[19] P. Nillius, J. Sullivan, and S. Carlsson, "Multi-target tracking-linking identities using Bayesian network inference," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2006, pp. 2187–2194.

[20] H. Jiang, S. Fels, and J. J. Little, "A linear programming approach for multiple object tracking," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Oct. 2007, pp. 1–8.

[21] M. Yang, T. Yu, and Y. Wu, "Game-theoretic multiple target tracking," in *Proc. Int. Conf. Comput. Vision*, 2007, pp. 1–8.

[22] Y. Jin and F. Mokhtarian, "Variational particle filter for multi-object tracking," in *Proc. Int. Conf. Comput. Vision*, 2007, pp. 1–8.

[23] B. Leibe, K. Schindler, and L. V. Gool, "Coupled detection and trajectory estimation for multi-object tracking," in *Proc. Int. Conf. Comput. Vision*, Oct. 2007, pp. 1–8.

[24] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2008, pp. 1–8.

[25] A. Ess, B. Leibe, K. Schindler, and L. V. Gool, "A mobile vision system for robust multi-person tracking," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2008, pp. 1–8.

[26] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in *Proc. Eur. Conf. Comput. Vision*, 2008, pp. 788–801.

[27] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Aug. 1995, pp. 1942–1948.

[28] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.

[29] X. Zhang, W. Hu, S. Maybank, X. Li, and M. Zhu, "Sequential particle swarm optimization for visual tracking," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2008, pp. 1–8.

[30] J. Lim, D. Ross, R. S. Lin, and M. H. Yang, "Incremental learning for visual tracking," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2004, pp. 793–800.

[31] G. H. Golub and C. F. V. Loan, "Matrix computations," in *Johns Hopkins Studies in the Mathematical Sciences*. Baltimore, MD: Johns Hopkins Univ. Press, 1996.

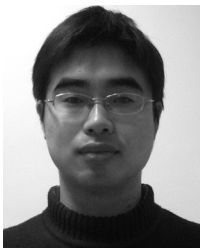
[32] L. Ingber, "Simulated annealing: Practice versus theory," *J. Math. Comput. Modeling*, vol. 18, no. 11, pp. 29–57, 1993.

[33] M. Isard and A. Blake, "Condensation: Conditional density propagation for visual tracking," *Int. J. Comput. Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[34] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 25, no. 5, pp. 234–240, May 2003.

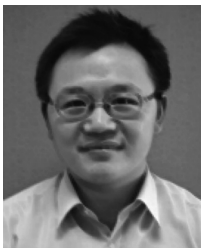
[35] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.

[36] R. van der Merwe, A. Doucet, N. Freitas, and E. Wan, "The unscented particle filter," Dept. Eng., Cambridge Univ., MA, Tech. Rep. CUED/F-INFENG/TR380, Aug. 2000.



Xiaoqin Zhang received the B.S. degree in electronic information science and technology from Central South University, Changsha, China, in 2005, and the Ph.D. degree in pattern recognition and intelligent system from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2010.

He is currently a Lecturer with the College of Mathematics and Information Science, Wenzhou University, Zhejiang, China. He has published more than 30 papers in international and national journals, and international conferences. His current research interests include visual tracking, motion analysis, and action recognition.



Weiming Hu (SM'07) received the Ph.D. degree from the Department of Computer Science and Engineering, Zhejiang University, Zhejiang, China.

From 1998 to 2000, he was a Post-Doctoral Research Fellow with the Institute of Computer Science and Technology, and founder of the Research and Design Center, Peking University, Beijing, China. Since 1998, he has been with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, where he is currently a Professor and a Ph.D. Student Supervisor. He has published more than 100 papers on national and international journals, and international conferences. His current research interests include visual motion analysis and recognition of harmful Internet multimedia.

Wei Qu (S'04–M'06) received the B.S. degree in electrical and computer engineering from the Beijing Institute of Technology, Beijing, China, in 2000, and the M.S. and Ph.D. degrees from the University of Illinois at Chicago, Chicago, IL, in 2005 and 2006, respectively, all in electrical and computer engineering.

He was a Senior Research Scientist with Motorola Laboratories, Schaumburg, IL, and then a Senior Researcher with Siemens Medical Solutions USA, Inc., Malvern, PA, from 2006 to 2009. He joined the Graduate University, Chinese Academy of Sciences, Beijing, in 2009, where he is currently an Associate Professor with the School of Information Science and Engineering, leading a video analysis group. He has authored over 30 technical papers in various journals and conferences, and holds ten U.S. patents pending.

Dr. Qu received the Best Student Paper Award in the IEEE International Conference on Image Processing in 2006. He has also served regularly as a reviewer for different journals and conferences, such as IEEE TRANSACTIONS ON CIRCUIT SYSTEM AND VIDEO TECHNOLOGY, and *Image Processing*.

Steve Maybank (SM'06) received the B.A. degree in mathematics from King's College, Cambridge, MA, in 1976, and the Ph.D. degree in computer science from Birkbeck College, University of London, London, U.K., in 1988.

In 1980, he was with the Pattern Recognition Group, Marconi Command and Control Systems, Frimley, London, U.K. In 1989, he was with the GEC Hirst Research Centre, Wembley, London. From 1993 to 1995, he was a Royal Society/Engineering and Physical Sciences Research Council Industrial Fellow with the Department of Engineering Science, University of Oxford, Oxford, U.K. In 1995, he was a Lecturer with the Department of Computer Science, University of Reading, Reading, U.K. In 2004, he joined as a Professor with the School of Computer Science and Information Systems, Birkbeck College, London. His current research interests include the geometry of multiple images, camera calibration, visual surveillance, information geometry, and the applications of statistics to computer vision.