Applied Soft Computing 13 (2013) 3106–3117

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A particle swarm optimisation algorithm with interactive swarms for tracking multiple targets

Myo Thida [a,b,*], How-Lung Eng [a], Dorothy N. Monekosso [b,c], Paolo Remagnino [b]

[a] Institute for Infocomm Research, Singapore
[b] Kingston University, United Kingdom
[c] University of Ulster, Jordanstown, United Kingdom

ARTICLE INFO

Article history:
Received 21 December 2011
Received in revised form 1 April 2012
Accepted 16 May 2012
Available online 12 June 2012

Keywords:
Interactive swarms
Particle swarm optimisation
Multi-target tracking
Social behaviour
Crowded scenes

ABSTRACT

We propose a novel particle swarm optimisation algorithm that uses a set of interactive swarms to track multiple pedestrians in a crowd. The proposed method improves the standard particle swarm optimisation algorithm with a dynamic social interaction model that enhances the interaction among swarms. In addition, we integrate constraints provided by temporal continuity and strength of person detections in the framework. This allows particle swarm optimisation to be able to track multiple moving targets in a complex scene. Experimental results demonstrate that the proposed method robustly tracks multiple targets despite the complex interactions among targets that lead to several occlusions.

1. Introduction

Multi-target tracking in crowded scenes still remains a challenging problem due to several aspects. The complex interactions and inter-occlusion between targets are major challenges encountered for the problem of tracking in a crowd. To address this problem, several methods [1–5] have proposed to integrate the social interactions among targets in the tracking algorithms. This direction has shown promising performance to track multiple targets in crowded scenes. An early example which models the social interaction of targets is Markov chain Monte Carlo (MCMC) based particle filter [2]. Their method models social interactions of targets using Markov random field and adds motion prior in the sampling process of a joint particle filter.

In this paper, we address the problem of multi-target tracking based on the particle swarm optimisation framework. Recently, particle swarm optimisation (PSO) [6] has gained attentions of many researchers because of its nature of interacting particles has proved to be effective in finding the optimum in a search space. In contrast to the particle filter [7] where particles move independently, PSO allows particles to interact; each particle, which is a candidate solution, searches the optimum using both social interaction and cognitive knowledge [8,9]. This idea of PSO is inspired by behaviour models of bird flocking where each bird finds its target (food) in the search space by sharing information with other birds in the swarm. This underlying phenomenon resembles the social interaction of pedestrians in a crowd and motivates us to employ the particle swarm optimisation framework for tracking multiple targets in a crowded scene.

However, the standard PSO is generally used to find a single optimum in a static search space. In contrast, the nature of tracking is dynamic where optima change over time. Thus, the standard PSO cannot be directly used to address the problem of tracking multiple targets. The idea of multiple independent swarms within a PSO, as in our earlier work [10], is not a viable option to track multiple interacting targets. In this paper, we formulate the problem of multi-target tracking as an optimisation problem of finding dynamic optima (pedestrians) where these optima interact frequently. We incorporate motion prediction and social interaction in the PSO framework such that each swarm finds the best local optimum based on its best knowledge and exchanges information with others.

The main contributions of our method can be summarised as follows: (1) introducing an idea of multiple interactive swarms to the standard PSO to track moving targets in a crowd; (2) incorporating higher level information such as social behaviour (motion information among pedestrians) in the process of finding optima in a high dimensional space; (3) integrating constraints provided

* Corresponding author.
E-mail address: mthida@i2r.a-star.edu.sg (M. Thida).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.05.019

by temporal continuity of target tracks and the strength of person detections; (4) initialising a separate swarm for each new person entering the scene.

The rest of the paper is organised as follows. Section 2 describes related work on multi-target tracking using particle swarm optimisation framework. In Section 3, the standard particle swarm optimisation algorithm is briefly explained. Section 4 explains details of our proposed method. Experimental results are presented in Section 5 and followed by conclusions in Section 6.

2. Related work

Related work to our method can be divided into two main strands of research.

The first strand is tracking algorithms which incorporate social interactions of targets in the tracking process. This idea of integrating social interactions of targets in tracking algorithms is motivated by the behaviour of targets in a crowd. In crowded scenarios, the behaviour of each individual target is influenced by the proximity and behaviour of other targets in the crowd. Several methods [3–5,11] have proposed to integrate the social interactions among targets in the tracking algorithms. This direction has shown promising performance to track multiple targets in crowded scenes. An early example which models the social interaction of targets is Markov chain Monte Carlo (MCMC) based particle filter [11]. Their method models social interactions of targets using Markov random field and adds motion prior in a joint particle filter. The traditional importance sampling step in the particle filter is replaced by a MCMC sampling step. French et al. [3] extended the method in [11] by adding social information to compute the velocity of particles. In [5], the authors formulated the tracking problem as a problem of minimising an energy function. The energy function is defined based on the both social information and physical constraint in the environment. Their preliminary results indicate that social information provides an important cue for tracking multiple targets in a complex scene.

The second strand is tracking algorithms that employ the PSO framework. PSO was first introduced to the problem of target tracking by Kölsch and Turk [12]. Particles were represented by the positions of *KLT* (Kanade, Lucas, and Tomasi) feature points [13]. The movement of particles were spatially confined based on the swarm behaviour using two thresholds: first one to define the maximum distance between feature points and the other to define the minimum distance between a particle and the swarm. A similar approach can be found in [14] where the object of interest is represented by N pixels. A swarm with N particles is initialised to track the target in an image space. The above approaches define a particle as a point and hence, their search space is limited to a 2 dimensional space. A higher dimensional search space is considered in [15] where the target is represented by the centroid, the width and the height of its bounding box. Similarly, Zhang et al. [16] proposed a sequential PSO algorithm where temporal information is incorporated into the standard PSO. In [17], Yang et al. incorporated PSO algorithm into unscented particle filter-based tracking to avoid an impoverishment problem which is a known problem in particle filter-based tracking. They have demonstrated that incorporating PSO improves the performance of the particle filter-based tracking in terms of accuracy and robustness. Recently, other hybrid trackers that incorporate PSO algorithm into particle filter [18], Kalman filter [19] and mean shift [20] have shown that swarm optimisation improves the performance of the tracker. For instance, compared to the standard mean-shift tracker, PSO-based mean shift provides better performance in tracking fast moving targets.

Our earlier work [10] extended the standard PSO algorithm to track multiple targets. Assuming that the number of targets are known a priori, we track multiple targets using a set of swarms where each swarm tracks a target independently. Our preliminary work in [10] has shown that the PSO algorithm is a promising framework for tracking multiple targets. Zhang et al. [21] proposed a species-based PSO where the global swarm is divided into many species to track multiple targets. These species track targets independently and interact only when the overlapping area between targets is greater than a particular threshold. Hence, their method [21] requires occluded targets to be detected explicitly and a selective appearance updating scheme is used to handle occlusion. In addition, the number of targets is assumed to be fixed and known a priori, which is hard to achieve in real applications. This limits its applicability and may fail in crowded situations with heavy interactions and frequent occlusions.

In this paper, we extend our earlier work [10] to track multiple pedestrians in a crowd. In contrast to [10] and [21], our method tracks multiple targets using a set of *interactive* swarms. A new velocity-updating mechanism is proposed which incorporates motion prediction and social interaction among pedestrians and thus, naturally handles the occlusion problem and improves tracking accuracy and precision.

3. Standard particle swarm optimisation

In this section, we briefly introduce a standard PSO and its notations. PSO is a population-based optimisation technique in which a set of particles $\{\mathbf{x}_i\}_{i=1}^N$ iteratively find the optimum solution in a search space. Each particle is a candidate solution equivalent to a point in a d-dimensional space, so the ith candidate can be represented as $\mathbf{x}_i = (x_{i1}, x_{i2}, \cdots, x_{id})$. The movement of each particle depends on two important factors: $\mathbf{x}_i^b$ the best position that the ith candidate has found so far and $\mathbf{x}^g$ the global best position found by the whole swarm (all particles). Based on these two factors, each candidate updates its velocity and position in the $(n+1)$th iteration as follows:

$$\mathbf{v}_i^{n+1} = \omega \mathbf{v}_i^n + \varphi_1 r_1 (\mathbf{x}_i^b - \mathbf{x}_i^n) + \varphi_2 r_2 (\mathbf{x}^g - \mathbf{x}_i^n) \quad (1)$$

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \mathbf{v}_i^{n+1} \quad (2)$$

where ω is the inertia weight, the parameters φ_1 and φ_2 are positive constants, which balance the influence of the individual best and the global best position. The parameters, $r_1, r_2 \in (0, 1)$ are uniformly distributed random numbers. Over the last decade, many variants of particle swarm optimisation algorithms have been proposed [22] and some algorithms have addressed dynamic optimisation problems [23,24]. One of the influential approach is the one proposed by Clerc and Kennedy [25] where a factor is introduced to avoid the unlimited growth of the particles' velocity. Eq. (1) is then modified as:

$$\mathbf{v}_i^{n+1} = \chi(\mathbf{v}_i^n + \varphi_1 r_1 (\mathbf{x}_i^b - \mathbf{x}_i^n) + \varphi_2 r_2 (\mathbf{x}^g - \mathbf{x}_i^n)) \quad (3)$$

where $\chi < 1$ is a constraining factor and defined as:

$$\chi = \frac{2}{\|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\|}, \quad \text{where } \varphi = (\varphi_1 + \varphi_2) > 4.0. \quad (4)$$

This method has been used by many researchers subsequently due to its stability and convergence ability in high-dimensional problems [22,26]. From Eq. (3), it can be observed that the movement of each particle depends on three components: inertial velocity, cognitive effect and social effect. The first component maintains the direction of the particle during the optimisation process while the second component allows each particle to move based on its own information, i.e., its best known position $\mathbf{x}_i^b$ in the previous iteration. The third component gives the social effect where the particle moves towards the global best position $\mathbf{x}^g$ defined by all members of the swarm.

Table 1
Notations adopted in our proposed method.

$\mathbf{X}_k^t$	A swarm corresponds to target k at time t
$\mathbf{x}_{i,k}^{b,t}$	Individual best for target k at time t
$\mathbf{x}_k^{g,t}$	Global best of the swarm for target k at time t
$\mathbf{x}_k^d$	State of the target k given by the object detector
K	Number of targets (number of swarms)
N	Number of particles for each swarm
i	Particle index
n	Iteration index
k	Target index
t	Frame (time) index

Here, the individual best, $\mathbf{x}_i^b$ and the global best, $\mathbf{x}^g$ positions are chosen at each iteration by evaluating a fitness function at the position $\mathbf{x}_i$. In general, the higher the fitness value, the closer to the optimum position. Hence, each particle will update its best position only if the fitness value of the current position is greater than the previous best value, otherwise the previous best position will be kept. The global best is the position that has the highest fitness value among all individual best positions. Mathematically, this can be formulated as follows:

$$\mathbf{x}_i^b = \begin{cases} \mathbf{x}_i^n, & \text{if} f(\mathbf{x}_i^n) > f(\mathbf{x}_i^b); \\ \mathbf{x}_i^b, & \text{otherwise.} \end{cases} \tag{5}$$

$$\mathbf{x}^g = \arg\max_{\mathbf{x}_i^b} f(\mathbf{x}_i^b) \tag{6}$$

where $f(\mathbf{x}_i^n)$ is the fitness value at the position $\mathbf{x}_i^n$.

4. A modified PSO with interactive swarms for multi-target tracking

Our method tracks multiple targets using a set of interactive swarms in the PSO framework. In order to handle the dynamic optimisation problem effectively, we introduce three major stages here: (1) a scheme for diversifying particles and swarms to maintain diversity over time, (2) a novel optimisation process that integrates the concepts of multiple swarms where the PSO updating equation is modified to incorporate temporal continuity information and social interaction among targets, and (3) a swarm initialisation and termination strategy to accommodate targets entering and leaving the scene. The detailed information of each major stage is elaborated in the following sections.

Section 4.1 explains the process of particles and swarms diversification while Section 4.2 explains the optimisation process of our proposed multiple swarms PSO in details. The swarm initialisation and termination strategy, which accommodates targets entering and leaving the scene, is described in Section 4.3. The notations adopted in this paper are listed in Table 1.

4.1. Particle and swarm diversification

Particle diversity: In order to allow a swarm to track a dynamic optimum (moving target), it is important to maintain particles diversity within the swarm over time. In our proposed method, a swarm $\mathbf{X}_k = \{\mathbf{x}_{i,k}\}_{i=1}^N$ is initialised for every new target entering the scene. Each swarm has N particles where each member $\mathbf{x}_{i,k} = (x_c, y_c, w, h)$ is a potential best state of the pedestrian represented by its centroid location and the width and height of the bounding box. These particles are sampled from a Gaussian distribution at the beginning of the PSO iteration in every frame as follows:

$$\{\mathbf{x}_{i,k}\}^t \sim N(\mathbf{x}_{k,pred}^t, \Sigma) \tag{7}$$

where Σ is a diagonal matrix and its entries are given by $\mathbf{v}_{k,pred}^t$. The predicted position $\mathbf{x}_{k,pred}^t$ of the target k at time t is given by:

$$\mathbf{x}_{k,pred}^t = \begin{cases} 0 & \text{when the swarm is first created} \\ \mathbf{x}_k^{g,t-1} + \mathbf{v}_{k,pred}^t & \text{otherwise.} \end{cases} \tag{8}$$

where the predicted velocity for target k is estimated as:

$$\mathbf{v}_{k,pred}^t = \mathbf{v}_{k,ind}^t + \mathbf{v}_{soc}^t \tag{9}$$

where $\mathbf{v}_{k,ind}^t$ refers to individual velocity of target k and $\mathbf{v}_{soc}^t$ refers to social velocity of the group. Thus, the motion of a target is predicted based on its personal information $\mathbf{v}_{k,ind}^t$ and the movement of other members of its social group $\mathbf{v}_{soc}^t$. Here, the individual velocity for target k is estimated by:

$$\mathbf{v}_{k,ind}^t = \mathbf{x}_k^{g,t-1} - \mathbf{x}_k^{g,t-2} \tag{10}$$

where $\mathbf{x}_k^{g,t-1}$ and $\mathbf{x}_k^{g,t-2}$ are states of the target k at time $t-1$ and $t-2$ respectively. Then, the social velocity is computed by sharing information among targets which have been moving generally in the same direction as follows:

$$\mathbf{v}_{soc}^t = \frac{1}{K_n} \sum_{j=1}^{K_n} (\mathbf{x}_k^{g,t-1} - \mathbf{x}_j^{g,t-2}) \tag{11}$$

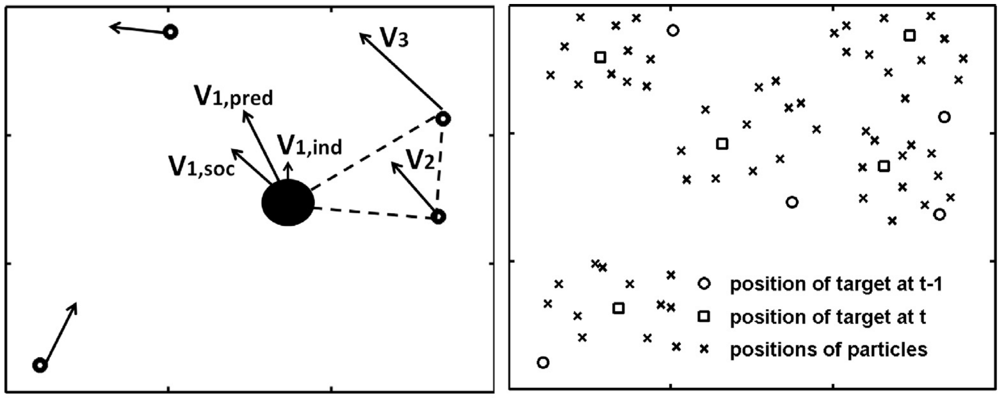
where K_n is the total number of neighbours of the target k. In this paper, two targets are considered as neighbours if they are in close proximity and have similar motion direction and speed for a time-overlap window of δT frames. More precisely, given a pair of trajectories for target k_1 and k_2 with the time-overlap window (in which both targets appear) of δT (in this paper, $\delta T = 10$), we compute the similarity score based on the absolute difference in their position, direction and speed. Two targets are defined as neighbours when their similarity score is larger than a particular threshold.

It should be highlighted that the motion of a target is predicted based on its personal information as well as the social knowledge. In this way, the position and motion of an occluded target can be estimated from its social group and particles are distributed at the likely position in the next frame. As a result, the proposed method can recover the target which is being occluded for a period of time. This situation is illustrated in Fig. 1.

Swarm diversity: When multiple targets are being tracked, there is a high probability that two targets occlude one another, especially in a crowded scene. This results in two different swarms competing for the same target or to cluster at the same location. To prevent this, we introduce the idea of *swarm diversity*: diversity among swarms that are close to each other. We first compute the distance between the global best states found by two different swarms to decide whether they are competing for the same target or cluster at the same location. When two swarms compete for the same target, we gradually expands the search space of the swarm with the lower fitness value. As a result, the target can be recovered even after the target is being occluded for a period of time.

4.2. Swarm optimisation

In the standard PSO, each particle is a candidate solution and finds the optimum by updating its position based on three components: inertial velocity, cognitive effect and social effect. In this paper, we propose a novel PSO updating rule where each particle adjusts its speed and position in the search space based on its personal knowledge, a shared information among its own swarm members, and the social activity among swarms. In addition, we incorporate detection responses $\mathbf{x}_k^d$ in the PSO framework to drive particles to find the new state in the direction given by the pedestrian detector and hence boost the convergence rate.



(a) $t = t_0$ (b) $t = t_0 + 1, n = 0$

Fig. 1. Effects of different components of the predicted velocity on initialising particles. (a) Shows how the motion of a target is predicted. Dotted lines indicate group membership. Target 1 is being occluded and its motion is estimated from other members of its social group. (b) Shows the distribution of particles based on its predicted position and motion. The current states of targets are shown in black circles while the previous states of targets are shown in black squares. Particles are marked by black cross symbols.

Our proposed velocity and position updating equations for a particle at time t are defined as follows:

$$\mathbf{v}_{i,k}^{n+1} = \chi[\mathbf{v}_{i,k}^{n} + c_1 r_1(\mathbf{x}_{i,k}^{b} - \mathbf{x}_{i,k}^{n}) + c_2 r_2(\mathbf{x}_{k}^{g} - \mathbf{x}_{i}^{n}) + c_3 r_3(\mathbf{x}_{k}^{d} - \mathbf{x}_{i}^{n})] \tag{12}$$

$$\mathbf{x}_{i,k}^{n+1} = \mathbf{x}_{i,k}^{n} + \mathbf{v}_{i,k}^{n+1} \tag{13}$$

where $\mathbf{v}_{i,k}^{n}$ and $\mathbf{x}_{i,k}^{n}$ are the velocity and the state of particle i of swarm (target) k at iteration n at time t. Here, we omit the subscript t for simplification. The first component $\mathbf{v}_{i,k}$ is the motion prior-based inertial velocity that integrates both individual and social velocity among targets. In contrast to the traditional PSO [6] where the inertial velocity is initialised to zero in the first iteration $n = 0$, our method incorporates the motion prediction based on the individual and the social behaviour of targets as follows:

$$\mathbf{v}_{i,k}^{0,t} = \begin{cases} 0 & \text{when the swarm is first created}\,(n=0) \\ \mathbf{v}_{k,pred} & \text{otherwise.} \end{cases} \tag{14}$$

where the predicted velocity for the target k is given by Eq. (9). The second component $(\mathbf{x}_{i,k}^{b} - \mathbf{x}_{i,k}^{n})$ corresponds to the cognitive effect where each particle moves to its best known position $\mathbf{x}_{i,k}^{b}$. The third component $(\mathbf{x}_{k}^{g} - \mathbf{x}_{i}^{n})$ then gives the social effect where the particle moves towards the global best position $\mathbf{x}_{k}^{g}$ defined by its own swarm.

Compared to the standard PSO [6], we introduce a new component based on the detection response $\mathbf{x}_{k}^{d}$. This component constrains particles to find the new state in the direction given by a state-of-the-art detector. The details procedures of selecting the detection response will be explained in Section 4.2.2. The parameters r_1, r_2 and r_3 are random numbers uniformly distributed in (0, 1), generated at every iteration. The parameter $\chi < 1$ confines the velocity of particles within a reasonable range and is defined as: $\chi = 2/\|2 - c - \sqrt{c^2 - 4c}\|$ where $c = c_1 + (c_2 + c_3)$. The parameters c_1, c_2 and c_3 are positive constants and balance the influence of cognitive, social and detection information respectively. In this paper, we set $c_1 = (c_2 + c_3) = 2.05$ [22]. This allows each particle of the swarm to use the social knowledge and the detection information collectively but still retain its personal knowledge as independent knowledge. In addition, we set the parameter $c_3 \in (0, 2.05)$ using the normalised matching score between the detection x_k^d and the best state of target k at previous frame $t - 1$ such that the influence of the detection information is high only when the selected detection is a good match to the target k.

In the following we describe how the individual best state $(\mathbf{x}_{i,k}^{b})$, the global best state $(\mathbf{x}_{k}^{g})$ and the detection response $(\mathbf{x}_{k}^{d})$ are selected for the target k.

4.2.1. Identifying individual and global best

The individual best $(\mathbf{x}_{i,k}^{b})$ and the global best $(\mathbf{x}_{k}^{g})$ states of particles are updated every iteration during the optimisation process by evaluating a fitness (cost) function. In our method, we define a fitness function based on a localised *HSV* (hue, saturation and value) colour histogram. Given the state of a particle i for target k at time t, we model the appearance of target k defined by the bounding rectangle $(x_c - (w/2), y_c - (h/2), x_c + (w/2), x_c + (h/2))$ as follow: (1) we first divide the target region into M equal parts (in this paper, $M = 9$); (2) each part is then represented by a $8 \times 8 \times 4$ histogram in *HSV* colour space. Mathematically, the target model for a given state $\mathbf{x}_i$ is given as $\mathbf{h}(\mathbf{x}_i) = \{\mathbf{h}_m\}_{m=1}^{M}$ where $\mathbf{h}$ is $8 \times 8 \times 4$ histogram. Then, the fitness function is defined by:

$$f(\mathbf{x}_i) = \frac{1}{M}\sum_{m=1}^{M} d(\mathbf{h}_m(\mathbf{x}_i), \mathbf{h}_m(\mathbf{x}_k)) \tag{15}$$

where $\mathbf{h}_m(\mathbf{x}_i)$ and $\mathbf{h}_m(\mathbf{x}_k)$ are the model and candidate histograms computed at the local part m, M is total number of parts and d is the Bhattacharyya distance between two histograms. Hence, $f(\mathbf{x}_i)$ will be low if two histograms are similar. The next step is to find the individual best and global best state by evaluating the fitness function at different states. A particle updates its current state as the best position (individual best) if the histogram at current state is more similar to the model (i.e., the fitness value at the current state $\mathbf{x}_i^n$ is lower than the value evaluated at the previous state $\mathbf{x}_i^{n-1}$). Otherwise, the previous best state will be kept. Mathematically,

$$\mathbf{x}_i^b = \begin{cases} \mathbf{x}_i^n, & \text{if} f(\mathbf{x}_i^n) < f(\mathbf{x}_i^b); \\ \mathbf{x}_i^b, & \text{otherwise.} \end{cases} \tag{16}$$

Once all particles update their best individual states, the global best among swarm members is identified as:

$$\mathbf{x}_k^g = \arg\min_{\mathbf{x}_{i,k}^b} f(\mathbf{x}_{i,k}^b) \tag{17}$$

where $i = (1, 2, \cdots, N)$ is a member of the swarm for target k. Fig. 2 illustrates a simulation of how to identify individual and global best states in a 2D search space. At first iteration $n = 0$ (Fig. 2(a)),

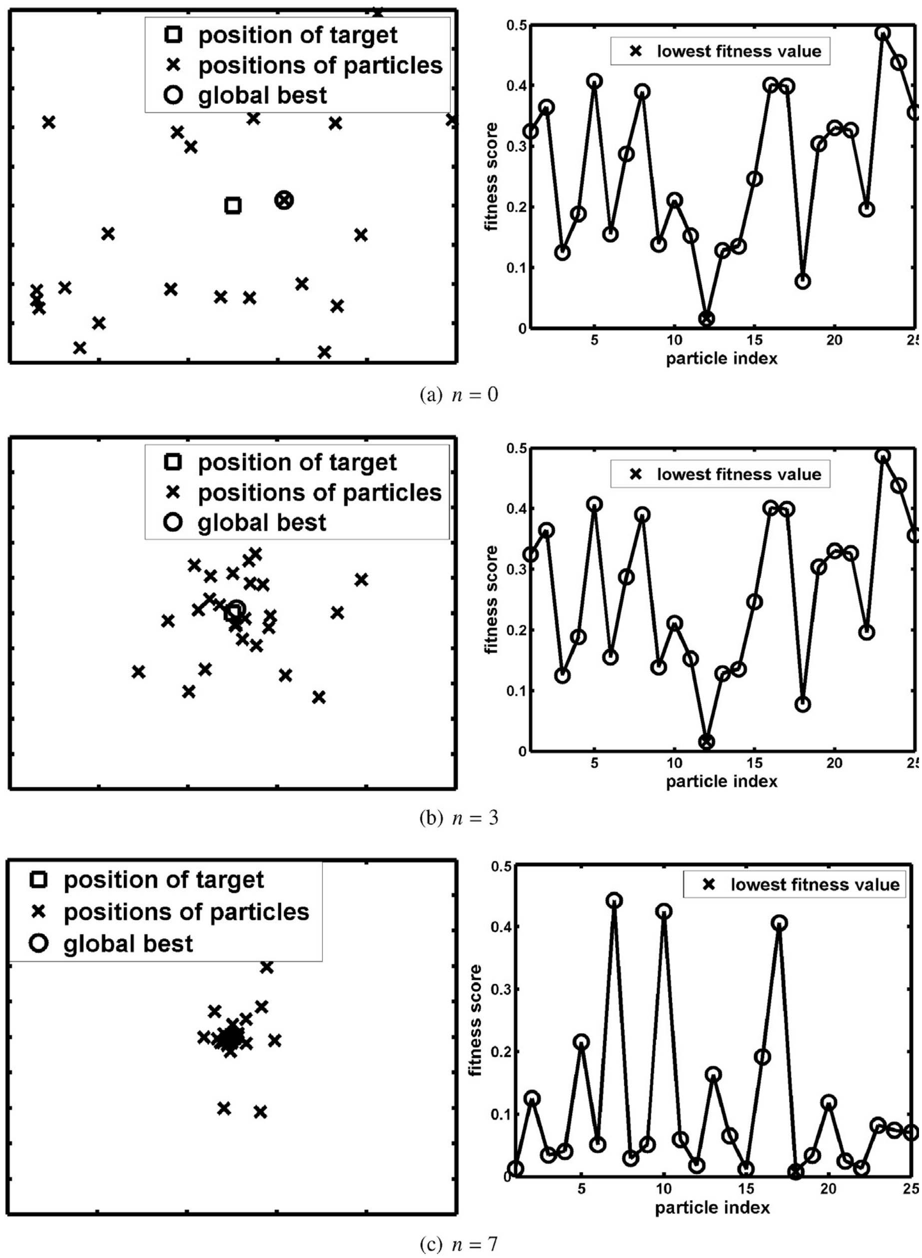


Fig. 2. Simulation of identifying individual and global best states in 2*D* search space. ((a) – left) Shows the distribution of particles at the first iteration $n = 0$. The fitness values against the states of particles is plotted (right image) while the global best is marked with a small black '×'. (b) and (c) Show the iteration process of particles in the subsequent iterations.

each particle takes its current state as the individual best state and the state which is the most similar to the target (nearest to the target in this simulation) is selected as the global best. In subsequent iterations, particles move to new positions based on the individual best and the global best of the swarm. Then, positions of the global best and individual best states are updated by evaluating the fitness function. The movements of particles in subsequent iterations are shown in Fig. 2(b) and (c) respectively.

4.2.2. Identifying detection response

As explained above, we introduce a new component "$c_3(\mathbf{x}_k^d - \mathbf{x}_i^n)$" in Eq. (12) to incorporate a detection response ($\mathbf{x}_k^d$) in the swarm optimisation process. This term computes the distance between the particle $\mathbf{x}_i^n$ and the associated detection $\mathbf{x}_k^d$ and guides the particles to search the optimum in the region given by an object detector. Unlike the individual best ($\mathbf{x}_{i,k}^b$) and the global best ($\mathbf{x}_k^g$) which are updated at every iteration, the state of the detection response ($\mathbf{x}_k^d$) is fixed during the iteration process. The parameter c_3 tunes the influence of the detection response on the movement of particles. In our method, we first obtain the detections in each frame using the histograms of oriented gradients (HOG) based human detector [27]. Fig. 3 shows the results of HOG detector in a scene. Please note that all detection results are not reliable; yielding false positives and missed detections.

Given $\{\mathbf{x}_m\}_{m=1}^{K_d}$ detection results at time t by the HOG detector, the next step is to identify a detection response to guide the tracker to a particular target. In order to decide which detection should guide the current tracker, we compute the matching score between detections and the current state of the tracked target k based on the spatial proximity, size and the appearance similarity as follows:

$$A(\mathbf{x}_k, \mathbf{x}_m) = A_s(\mathbf{x}_k, \mathbf{x}_m) \times A_f(\mathbf{x}_k, \mathbf{x}_m) \times A_d(\mathbf{x}_k, \mathbf{x}_m) \tag{18}$$

Fig. 3. Sample detection results given by the HOG detector [28] (please note that all detection results are not reliable; yielding false positives and missed detections).

where $x_m \in \{\mathbf{x}_m\}_{m=1}^{K_d}$ is a detection result given by the HOG-based detector [27]. Here, the respective matching score A_s is given by the overlapping area between targets k and the detection m while the score $A_f = \exp\{-d(\mathbf{h}(\mathbf{x}_k), \mathbf{h}(\mathbf{x}_m))\}$ is computed based on the Bhattacharyya distance between two histograms. Finally, the score A_d is computed using the Euclidean distance between centroid locations of the tracked target k and the detection response m. Next, the detection response, which is the best match to the current state of the tracked target, is identified by finding the maximum matching score:

$$\mathbf{x}_k^{d,t} = \arg\max_{\mathbf{x}_m} A(\mathbf{x}_k, \mathbf{x}_m), \quad m = \{1, 2, \cdots, K_d\} \tag{19}$$

where K_d is the number of detections given at time t. The matching score $A(\mathbf{x}_k, \mathbf{x}_m)$ between the selected detection and the tracked target k is normalised and used as c_3, a weighting parameter of detection component in Eq. (12). It can be seen that the matching score or the parameter c_3 will be large only if the selected detection and the current state of the tracked target have high correlations in their position, feature and size. In this way, we ensure that the output from a detector is integrated in the swarm optimisation only if the selected detection is a good match to the tracked target.

4.2.3. Convergence criteria

In general, it is assumed that the convergence of a PSO algorithm is achieved when all particles ultimately stop at the global best position. This ensures that the optimisation process grantees the best accuracy at a high computational cost. However, for tracking applications, it is essential to compromise between the tracking accuracy and the computational cost as the ultimate goal of a tracking algorithm is to achieve a reasonable tracking accuracy in real time. By considering both the tracking accuracy and the computation time, the proposed algorithm considers that the optimisation reaches a convergent state when one of the following criteria is achieved:

1. $\mathbf{x}_i^{b,n} \approx \mathbf{x}_i^{b,n+1}$ and $\mathbf{x}_i^b \to \mathbf{x}^g$ for all particles $i \in (1:N)$
2. $\mathbf{x}^{g,n} \approx \mathbf{x}^{g,n+1}$ and $f(\mathbf{x}^g) < TH$
3. the pre-defined maximum number of iterations is reached.

The parameter TH can be defined and updated online by studying the trend of the feature discrepancy of the target k over time. In most experiments presented in this chapter, a good tracking result can be achieved within 10 iterations.

4.3. Swarm initialisation and termination

Our method automatically initialises a new swarm for each person subsequently detected for T frames. In order to reduce false positive detections, we compute a matching score for each detected target over T frames (18). Then, only the associated detections with a matching score higher than the threshold are used to initialise a new swarm. The length of the observation window T can be determined based on the frame rate of the video and the prior knowledge of the monitoring scene, for instance, T should be set to a low value (1,10) for a crowded scene where targets enter and leave the scene frequently. Here, we set $T = 5$ for video sequences tested in this paper.

It is important to update the target model $\mathbf{h}(\mathbf{x}_k)$ as the appearance of the target model is expected to have slight variations over time. In this method, the online updating model is proposed to address the appearance variations of targets. Given the previous T appearances (histograms) of the target k, $\{\mathbf{h}(\mathbf{x}_k)^{t_1}, \mathbf{h}(\mathbf{x}_k)^{t_2}, \cdots, \mathbf{h}(\mathbf{x}_k)^T\}$, the minimum appearance change of the target k at time t can be computed as follows:

$$\delta(\mathbf{x}_k^t) = \min_{t'} d(\mathbf{h}(\mathbf{x}_k^t), \mathbf{h}(\mathbf{x}_k^{t'})) \tag{20}$$

where $t' = \{t_1, t_2, \cdots, T\}$ is a time index for the previous frames. This value gives the smallest appearance change of target k at time t from the previous observations. Then, the next step is to check if the appearance change of the target $\delta(\mathbf{x}_k^t)$ is significantly different from the previous appearance changes $\delta(\mathbf{x}_k^{t'})$. The target model should not be updated if there is a significant appearance change in the current frame as this can indicate that the tracker is stuck at the wrong person or the scene condition has changed suddenly. However, if the change is small, the target model must be updated to accommodate the slight appearance changes.

In this method, the probability density function *PDF* of the appearance changes of the target over T previous frames is estimated using the kernel-based density function. Then, the probability score for the appearance change of the target k at time t is computed as:

$$p(\delta(\mathbf{x}_k^t)) = \frac{1}{k}\sum_{t'=t_1}^{T} \exp\left[-\frac{(\delta(\mathbf{x}_k^t) - \delta(\mathbf{x}_k^{t'}))^2}{2\sigma^2}\right] \tag{21}$$

where σ is the standard variation of the of appearance changes in the pervious frames. The high probability score indicates the slight changes in the appearance target model and hence, the target model $\mathbf{h}(\mathbf{x}_k)$ is updated using an adaptive filter as:

$$\mathbf{h}(\mathbf{x}_k) = \alpha\mathbf{h}(\mathbf{x}_k) + (1-\alpha)\mathbf{h}(\mathbf{x}_k^t) \tag{22}$$

where $\alpha \in (0, 1)$ is the learning rate.

In a scenario where multiple targets are entering and leaving a scene, it is important to terminate the tracking process when the swarm (tracker) loses its target for a number of subsequent frames. The probability score computed in Eq. (21) indicates the degree of the appearance change of the target at time t from the previous frames. The small probability score states that the target has a significant changes from the previous frames or the tracker has lost its target. Based on this observation, a swarm is terminated when the probability score is lower than a particular threshold value for T subsequent frames.

4.4. Algorithm summary

The proposed method can be summarised as follows:

Algorithm 1. Pseudo-code of the proposed algorithm

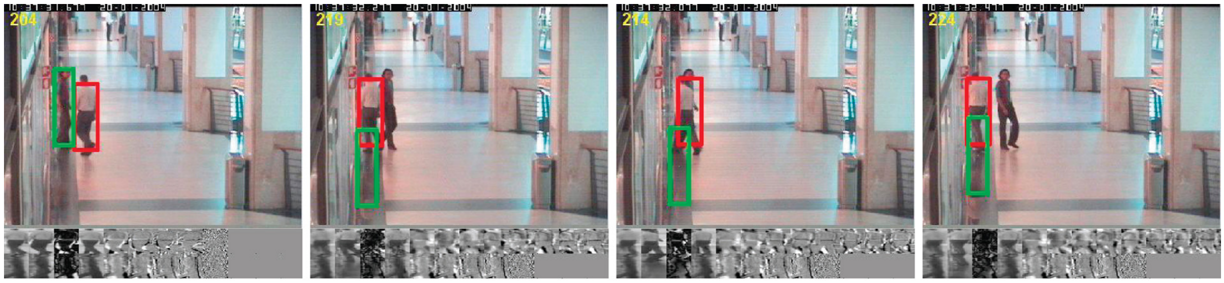
Input: New Image Frame at time t and Existing trackers: $\{\mathbf{x}_k\}_{k=1}^{K}$
Perform HOG detection: $\{\mathbf{x}_m\}_{m=1}^{K_d}$
Compute similarity scores between HOG result and states of existing trackers using equation (18).
for *all targets* **do**
  if *new target* **then**
    //Initialize a new tracker
    Randomly generate a new swarm: $:\{\mathbf{x}_i\}_{i=1}^{N}$
    Increase total number of trackers: K= K + 1;
  end
  else
    Find associated detection result: $\mathbf{x}_k^{d,t}$ using equation (19).
    Predict target velocity at time t using equation (9).
    Initialise a new swarm: $\{\mathbf{x}_{i,k}\}^t \sim N(\mathbf{x}_{k,pred}^t, \Sigma)$ using equation (7).
  end
  //Initialisation process
  foreach *Particle* $i \in 1 \to N$ **do**
    $\mathbf{x}_{i,k}^b = \mathbf{x}_{i,k}^n$
    compute the fitness value $f(\mathbf{x}_{i,k}^b)$
  end
  $\mathbf{x}^g = \operatorname{argmax}_{\mathbf{x}_{i,k}^b} f(\mathbf{x}_{i,k}^b)$
  //Iteration process
  for $n = 1$ *to maximum number of iterations* **do**
    foreach *Particlei* $\in 1 \to N$ **do**
      update velocity using equation (12)
      update position using equation (**??**)
      if $f(\mathbf{x}_{i,k}^n) < f(\mathbf{x}_{i,k}^b)$ **then**
        $\mathbf{x}_{i,k}^b = \mathbf{x}_{i,k}^n$
      end
      compute the fitness value $f(\mathbf{x}_{i,k}^b)$
    end
    $\mathbf{x}^g = \operatorname{argmax}_{\mathbf{x}_{i,k}^b} f(\mathbf{x}_{i,k}^b)$
    if *convergencecriteria are met* **then**
      Exit from Iteration process;
    end
  end
  Output: the global best position: $\mathbf{x}_k^t = \mathbf{x}_k^g$
  if $p(\delta(\mathbf{x}_k^t)) > th_1$ **then**
    update target model using equation (22)
    else if $(p(\delta(\mathbf{x}_k^{t'})) < th_2, t' = \{t-1, t-2, \cdots, t-T\})$ **then**
      terminate the tracker k;
    end
  end
end
Output: target states at time t: $\{\mathbf{x}_k^t\}_{k=1}^{K}$

5. Experiments

In this section, we present experimental results in two different contexts. First, we assess the performance of our proposed PSO against the other state-of-the-art PSO-based algorithms. Next, we evaluate our method in multi-target tracking context. We compare the performance of our method with other state-of-the-art methods in tracking domain. Here, we evaluate our method on public data-sets of crowded scenes with different crowd densities. All our experiments are carried out using visual C++on a platform with a dual-Core 3 GHz processor and 4 GB RAM.

5.1. Proposed method vs. species-based PSO

The first experiment compares the performance of our proposed multi-swarm PSO algorithm with the species-based PSO algorithm [21]. To the best of our knowledge, [21] is the only PSO-based method which tested on tracking multiple targets. To make a fair comparison, we employ the same video sequence (EnterExitCrossingPaths2cor.mpg from CAVIAR [29]) used in [21] and assume that there is no detection output for the proposed method. Assuming



(a) Species-based PSO [21]. Reported results are extracted from their paper.

(b) Results achieved by the Proposed Method

Fig. 4. Qualitative comparisons of the proposed method with species-based PSO [21].

that the number of targets are fixed and known a prior, the targets are initialised manually in the first frame. Fig. 4 shows the qualitative results achieved by species-based PSO and the proposed method. Please note that there is no quantitative results reported in [21] and we have limited information to re-implement the species-based PSO. As a result, we focus our comparison only on qualitative results for this particular experiment. We can observe that species-based PSO failed to recover the targets after occlusions (see Fig. 4(a)). The authors tackled the problem by incorporating the selective part-based appearance updating model which updates only the non-occluded parts of the targets. On the other hand, our proposed method, by accounting for social interaction among pedestrians, tracks and recovers targets after occlusion without a need to detect occluded regions (see Fig. 4(b)).

5.2. Multi-target tracking

In this set of experiments, we evaluate the proposed method in the context of tracking multiple targets. The first experiment tracks fixed number of targets assuming that we have priori knowledge about number of targets to be tracked. In this experiment, we assume that there is no detections results available. The second experiment evaluates our proposed in more challenging situations, several video sequences with different crowd densities are used to evaluate the robustness of the proposed method against the density of the crowd. Before we present experimental results, we first introduce the evaluation matrices adopted in this paper.

5.2.1. Tracking evaluation

In this paper, we use the evaluation metrics from CLEAR [30] and [31]. The adopted metrics are:

1. MOTA: multiple object tracking accuracy [0,1] is calculated using number of false positives, missed targets and identity switches.
2. MOTP: multiple object tracking precision [0,1] is calculated based on average distance between the centroid positions of tracked targets and the ground truth.
3. MT: Mostly tracked: percentage of ground-truth trajectories which are covered by tracker output for more than 80% in length.
4. ML: Mostly lost: percentage of ground-truth trajectories which are covered by tracker output for less than 20% in length. The smaller the better.
5. PT: Partially tracked: 1.0-(MT+ML). The smaller the better.

5.2.2. Tracking fixed and known number of targets

In this section, we evaluate the performance of our method on tracking multiple targets where the number of targets are assumed to be fixed and known a priori. We first tested our proposed method on a single-target tracking using a helicopter video sequence captured by a mobile camera [32]. This video sequence contains 780 images with a size of 240 × 320. The remotely controlled toy helicopter is partially occluded for few frames by the person who is controlling the helicopters. Fig. 5 shows qualitative results of our proposed method on tracking a single target. As can be seen in Fig. 5, our proposed method tracks the helicopter against the illumination changes and occlusion. The target is successfully tracked regardless of heavy occlusions for few frames.

Next, we track three walking persons with strong interactions on a video sequence from CAVIAR data-set [29] where each video frame is a size of 384 × 288. Fig. 6 shows some qualitative examples of our tracked results. Three persons are successfully tracked even though inter-occlusion happens frequently. We can see that our proposed method, using the proposed swarm diversity scheme and social interaction-based velocity, handles inter-occlusion very well.

5.2.3. Tracking unknown and varying number of targets

In this experiment, we evaluate our method for tracking multiple number of targets in a crowd on two video sequences with different levels of crowd density (*S2L1*-view1) from PETS 2009 data-set [34] and Oxford data-set [28]. The PETS video sequence is recorded from an elevated viewpoint at 7 frame per second and contain 795 frames with an image size of 768 × 576 pixels. The ground truth for the sequence is manually annotated by [4]. The second video sequence contains 7500 images with an image size of 1920 × 1080 and the ground truth for the first 4501 image frames are manually annotated by [28]. The quantitative results for each data-set are given in Table 2. We use the detection outputs given

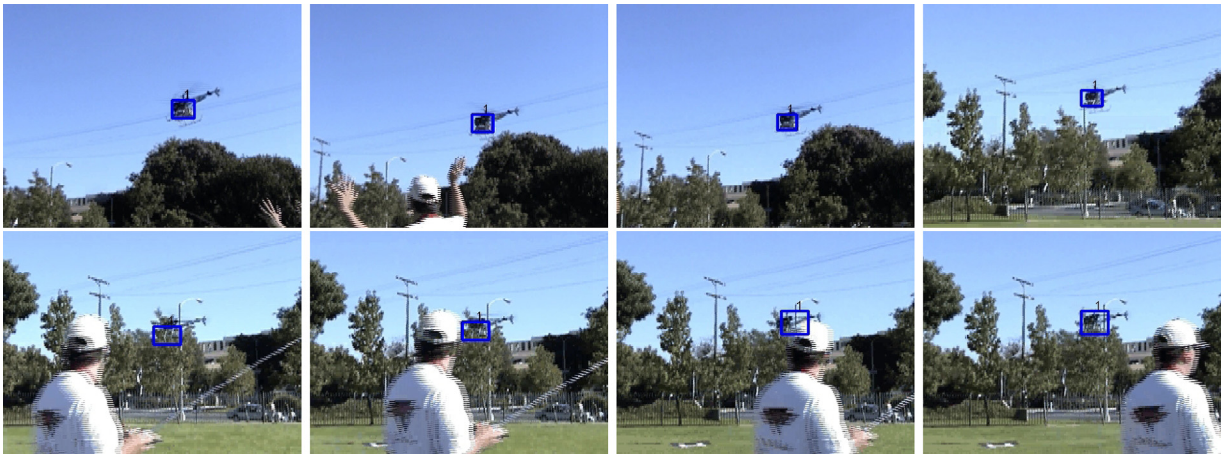


Fig. 5. Qualitative results of our proposed method on the helicopter sequence [32]. More results can be found at http://www.youtube.com/watch?v=-VUNw8CXqIg.

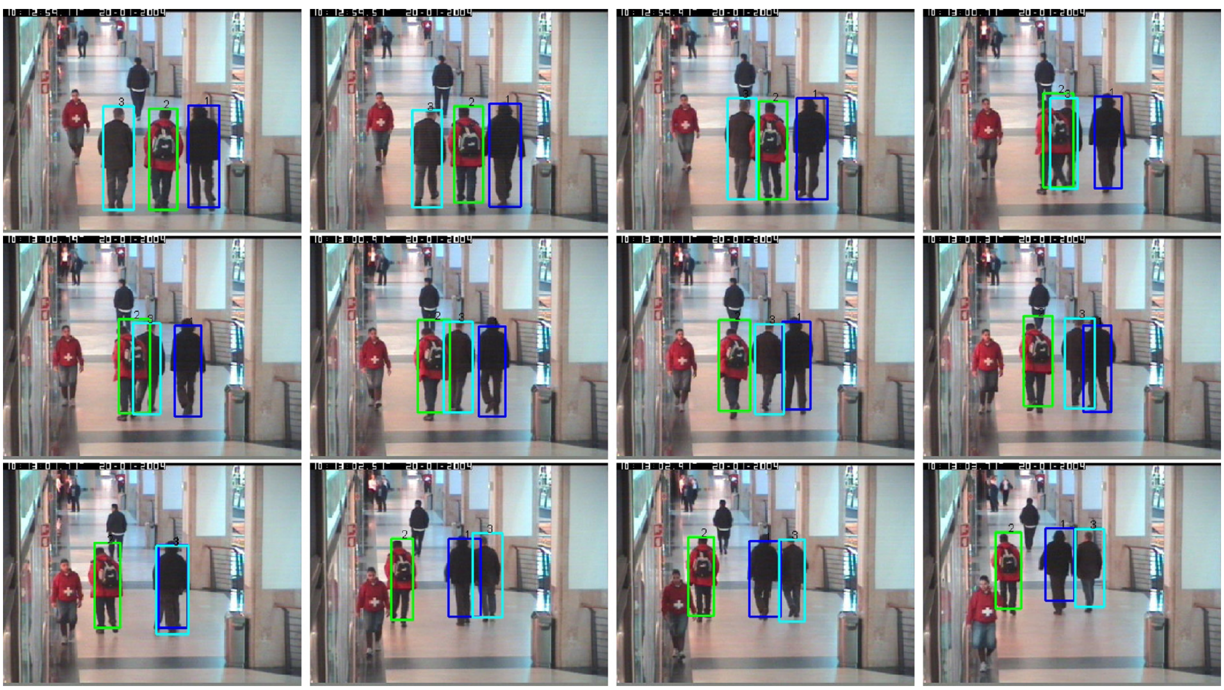


Fig. 6. Qualitative results of our proposed method on CAVIAR data-set [29]. More results can be found at http://www.youtube.com/watch?v=9HLow_Mz9Rg.

Table 2
Quantitative results of our method.

	Oxford data-set	PETS 2009 S2L1
MOTA	84.8%	83.0%
MOTP	80.5%	86.4%
MT	79.6%	94.7%
ML	18.7%	5.3%
PT	1.7%	0.0%

by a state-of-the-art detector[1] for initialisation of new targets. The experimental results indicate that our proposed method tracks the targets with high precision (above 80%) for both sequences. Some qualitative results are given in Fig. 7. We can observe that targets are successfully tracked over time and new targets are initialised automatically.

5.2.4. Comparisons with state of the art methods

In this section, we conduct a comparative study on the performance of our method with the state-of-the-art methods on PETS-S2L1, one of the most widely used sequences from PETS 2009 data-set. As can be seen in Table 3, our method outperforms the state-of-the-art methods for most of the measurements. We achieve the tracking precision of 86% which is nearly 10% higher than the best reported results. The mostly tracked trajectories are increased by about 8% and the partially tracked trajectories are reduced to zero percent. The slightly lower MOTA value can be explained by the frequent interactions of targets and the reliability of the resulting detections; i.e., some targets are detected due to persistent false positives occurred by background structures such as signboards and public phone boxes.

[1] We would like to thank D. Simonnet from Kingston University and Benfold et al. [28] for providing detection outputs for PETS 2009 sequence and Oxford data-set respectively.

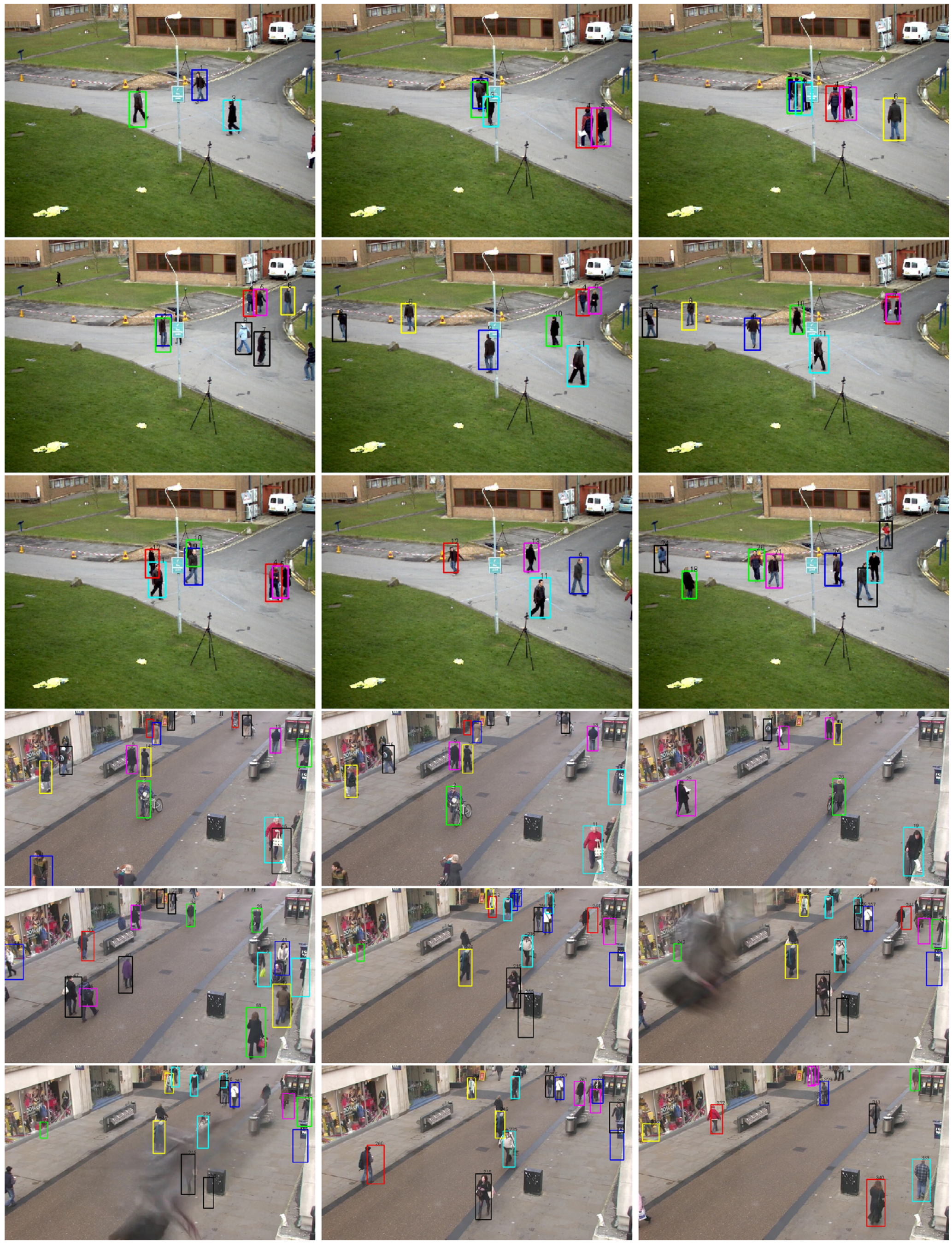


Fig. 7. Qualitative results of our proposed method on PETS data-set (S2L1) and Oxford data-set. More results can be found at http://www.youtube.com/watch?v=wCgtMeKT50U.

Table 3
Quantitative comparisons of our method with state-of-the-art methods on PETS 2009 data-set: S2L1 View 1. OM refers to the occlusion modelling scheme. Reported results for other methods are extracted from their papers. Please note that MT, ML and PT were not reported in [34] and in the last 2 rows, the smaller number indicates better performance.

	[34]	[4]	[35] With OM	Our method
MOTA	79.0%	81.4%	88.3%	83.0%
MOTP	59.0%	76.1%	75.7%	86.4%
MT	–	82.6%	87.0%	94.7%
ML	–	0.0%	4.4%	5.3%
PT	–	17.4%	8.7%	0.0%

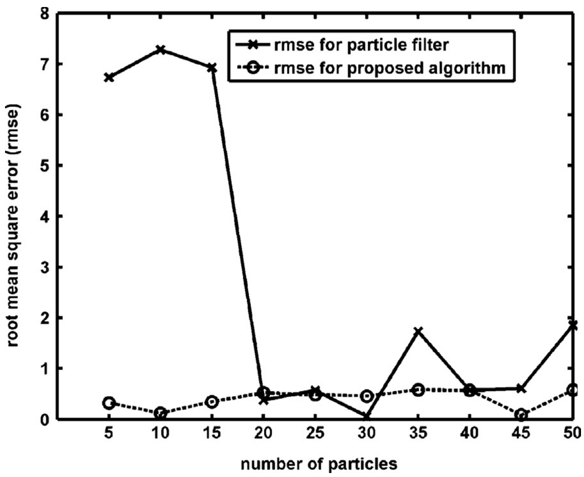


Fig. 8. Root mean square error against the number of particles for the proposed algorithm and PF-based colour tracker [32].

5.3. Robustness test

In this experiment, we evaluate the performance of the proposed method with respect to the number of particles (swarm size *N*). To compare the dependency on number of particles by the proposed algorithm and the particle filter-based colour tracker, we run both algorithms for 10 times with different number of particles. Fig. 8 shows the root mean square error for both algorithms against the number of particles. In contrast to the particle filter where the tracking accuracy is highly depends on the number of particles, the swarm size (number of particles) does not have a significant impact on our method. The proposed method can achieve the comparable tracking accuracy with much lower number of particles.

5.4. Computation cost

This experiment evaluates the computational cost of the proposed method where the number of particles for each swarm is fixed at 15. Fig. 9(a) shows the computation time required for tracking results of helicopter sequence shown in Fig. 5. It can be observed that our method takes about 0.0301 s for each frame while achieving a high tracking accuracy with root mean square error of 0.3470. The number of iterations required for each frame is also shown in Fig. 9(b). On average, the proposed method achieves a good tracking accuracy within 5 iterations.

6. Conclusion and future work

In this paper, we have presented a novel particle swarm optimisation algorithm to track pedestrians in a crowded scene. Through particles and swarms diversification, motion prediction is introduced into multi-swarm PSO, constraining swarm members to the most likely region in the search space. The social interaction among swarm and the output from pedestrians detector are also incorporated into the velocity-updating equation. This allows our proposed method to track multiple targets in a crowded scene with severe occlusion and heavy interactions among targets. Experimental results demonstrate that our proposed method outperforms the state-of-the-art methods to track multiple targets in a crowded scene with high precisions. In future work, we plan to incorporate a more sophisticated model for grouping targets in a crowded scene.

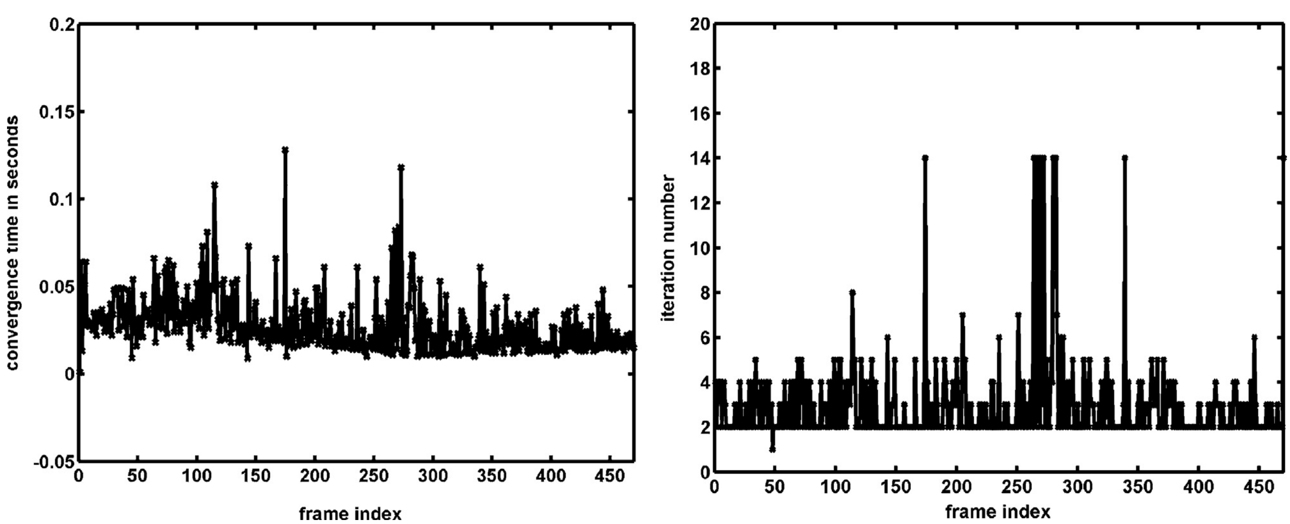


Fig. 9. Computational cost and number of iterations required for tracking results of helicopter sequence.

References

[1] Z. Khan, T. Balch, F. Dellaert, An MCMC-based particle filter for tracking multiple interacting targets, in: Proceedings of European Conference on Computer Vision, 2004, pp. 279–290.

[2] Z. Khan, T. Balch, F. Dellaert, MCMC-based particle filtering for tracking a variable number of interacting targets, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (11) (2005) 1805–1819.

[3] A. French, A. Naeem, I. Dryden, T. Pridmore, Using social effects to guide tracking in complex scenes, in: Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance, 2007, pp. 212–217.

[4] A. Andriyenko, K. Schindler, Multi-target tracking by continuous energy minimization, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, IEEE, Colorado Springs, CO, 2011, pp. 1265–1272.

[5] K. Yamaguchi, A.C. Berg, L.E. Ortiz, T.L. Berg, Who are you with and where are you going? in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, IEEE, Colorado Springs, CO, 2011, pp. 1345–1352.

[6] J. Kennedy, R. Eberhart, Particle swarm optimisation, in: Proceedings of IEEE International Conference on Neural Networks, vol. 4, Perth, WA, Australia, 1995, pp. 1942–1948.

[7] M. Isard, A. Blake, CONDENSATION conditional density propagation for visual tracking International Journal of Computer Vision 29 (1) (1998) 5–28.

[8] K.E. Parsopoulos, M.N. Vrahatis, Recent approaches to global optimisation problems through particle swarm optimisation, Natural Computing 1 (2002) 235–306.

[9] R. Poli, Analysis of the publications on the applications of particle swarm optimisation, Journal of Artificial Evolution and Applications 4 (2008) 1–10.

[10] M. Thida, P. Remagnino, H.-L. Eng, A particle swarm optimisation approach for multi-objects tracking in crowded scene, in: Proceedings of IEEE International Workshop on Visual Surveillance, 2009, pp. 1209–1215.

[11] Z. Khan, T. Balch, F. Dellaert, MCMC-based particle filtering for tracking a variable number of interacting targets, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (11) (2005) 1805–1918.

[12] M. Kölsch, M. Turk, Hand tracking with flocks of features, in: Proceedings of Computer Vision and Pattern Recognition, vol. 2, 2005, p. 1187.

[13] J. Shi, C. Tomasi, Good features to track, in: Proceedings of Computer Vision and Pattern Recognition, vol. 2, 1994, pp. 593–600.

[14] L. Anton-Canalis, M. Hernandez-Tejera, E. Sanchez-Nielsen, Swarmtrack: a particle swarm approach to visual tracking, in: Proceedings of the International Conference on Computer Vision Theory and Applications, vol. 2, Portugal, 2006, pp. 221–228.

[15] Y. Zhang, Y. Meng, Adaptive object tracking using particle swarm optimisation, in: Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2007, pp. 43–48.

[16] X. Zhang, W. Hu, S. Maybank, X. Li, M. Zhu, Sequential particle swarm optimisation for visual tracking, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[17] S. Yang, Q. Ma, W. Huang, Particle swarm optimized unscented particle filter for target tracking, in: Proceedings of International Congress on Image and Signal Processing, 2009, pp. 1–5.

[18] H. Zhou, X. Zhang, H. Li, J. Li, Video object tracking based on swarm optimized particle filter, in: Proceedings of International Conference on Industrial Mechatronics and Automation, vol. 2, 2010, pp. 702–706.

[19] N. Ramakoti, A. Vinay, R.K. Jatoth, Particle swarm optimisation aided Kalman Filter for object tracking, in: Proceedings of International Conference on Advances in Computing, Control, and Telecommunication Technologies, 2009, pp. 531–533.

[20] J. Li, H. Yu, H. Liang, Efficient mean shift tracking via particle swarm optimisation for multi-articulated human body features, in: Proceedings of International Conference on Mechatronics and Automation, 2008, pp. 781–787.

[21] X. Zhang, W. Hu, W. Qu, S. Maybank, Multiple object tracking via species-based particle swarm optimisation, IEEE Transactions on Circuits and Systems for Video Technology 20 (11) (2010) 1590–1602.

[22] M.A. MontesdeOca, T. Stützle, M. Birattari, M. Dorigo, Frankenstein's PSO: a composite particle swarm optimisation algorithm, IEEE Transactions on Evolutionary Computation 13 (5) (2009) 1120–1132.

[23] D. Parrott, X. Li, Locating and tracking multiple dynamic optima by a particle swarm model using speciation, IEEE Transactions on Evolutionary Computation 10 (4) (2006) 440–458.

[24] G.G. Yen, W.F. Leong, Dynamic multiple swarms in multi-objective particle swarm optimisation, IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans 39 (4) (2009) 890–911.

[25] M. Clerc, J. Kennedy, The particle swarm – explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6 (1) (2002) 58–73.

[26] B. Kwolek, Object tracking via multi-region covariance and particle swarm optimisation, in: Proceedings of Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, 2009, pp. 418–423.

[27] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.

[28] B. Benfold, I. Reid, Stable multi-target tracking in real-time surveillance video, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2011, pp. 3457–3464.

[29] CAVIAR data-set, http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/.

[30] Clear evaluation, 2007, http://clear-evaluation.org/.

[31] Y. Li, C. Huang, R. Nevatia, Learning to associate: hybrid boosted multi-target tracker for crowded scene, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 2953–2960.

[32] Particle filter colour tracker, 2007, http://www.mathworks.com/matlabcentral/fileexchange/17960.

[34] PETS, 2009, http://www.cvg.rdg.ac.uk/PETS2009.

[34] E.T. Jérôme Berclaz, François Fleuret, P. Fua, Multiple object tracking using k-shortest paths optimisation, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (9) (2011) 1806–1819.

[35] A. Andriyenko, S. Roth, K. Schindler, An analytical formulation of global occlusion reasoning for multi-target tracking, in: Proceedings of IEEE International Workshop on Visual Surveillance, IEEE, Barcelona, Spain, 2011.