Chapter 4
Multi-object Tracking Using Particle Swarm Optimization on Target Interactions

Bogdan Kwolek

Abstract In this work, a particle swarm optimization based algorithm for multi-target tracking is presented. At the beginning of each frame, the objects are tracked individually using highly discriminative appearance models among different targets. The task of object tracking is considered as a numerical optimization problem, where a particle swarm optimization is used to track the local mode of the similarity measure. The objective function is built on the region covariance matrix and multi-patch based object representation. The target locations and velocities that are determined in such a way are further employed in a particle swarm optimization based algorithm, which refines the trajectories extracted in the first phase. Afterwards, a conjugate method is used in the final optimization. Thus, the particle swarm algorithm is utilized to seek good local minima and the conjugate gradient is used to find the local minimum accurately. At this stage, we optimize complex energy functions which represent the presence, movement and interaction of all targets in sequence of recent frames within a temporal window. The algorithm has been evaluated on publicly available datasets. The experimental results show performance improvement over relevant algorithms.

4.1 Introduction

Visual tracking of multiple objects is a challenging problem. The aim is to infer the states of all targets in the scene and to maintain their identity over time. Despite significant progress in this area, reliable tracking of multiple targets is still a great challenge, particularly in crowded scenes. Many different methods [2, 6, 10, 17, 22, 23] have been proposed in the last decade. One solution to multiple object tracking is the use of multiple trackers, where each tracker is responsible for tracking one object. The so-called tracking-by-detection algorithms [8] gained considerable attention in this area of the research. A widely used approach to multi-target tracking consists in exploiting a joint state-space representation, which concatenates all of the targets'

B. Kwolek (✉)
Rzeszow University of Technology, Al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland
e-mail: bkwolek@prz.edu.pl

A. Chatterjee et al. (eds.), *Advances in Heuristic Signal Processing and Applications*,
DOI 10.1007/978-3-642-37880-5_4,

states together [23], or inferring this joint data association problem by estimating all possible associations between the targets and the observations [17, 24]. In contrast to the above mentioned approaches, in order to achieve multi-target tracking, the multiple parallel filters where a single filter per target has its own state space were proposed in [9]. However, when the interactions among the moving targets take place, difficulties in maintaining the correct object identities might arise. Therefore, modeling the interactions among targets and occlusion reasoning play an incredibly important role in multi-target tracking. Khan et al. [17] use a Markov Random Field (MRF) motion prior to modeling the interactions among targets. Andriyenko et al. [2] propose a model for global occlusion reasoning. In an approach that is based on particle swarm optimization [30], the object interactions are modeled as species competition and repulsion. Particle Swarm Optimization (PSO) is a population based stochastic optimization technique [16] which shares many similarities with evolutionary computation techniques. It has been shown to perform well on many nonlinear and multimodal optimization problems.

Visual object tracking is an important ingredient of any multi-object tracking algorithm. Particle filters [13] are one of the most efficient techniques for object tracking. They were successfully applied in many visual tracking applications [28], including multi-object tracking [8, 23]. The task of object tracking can be considered as a numerical optimization problem, where a local optimization is used to track the local mode of the similarity measure in a parameter space of translation, rotation, and scale. In [29], it was shown that, in tasks consisting in tracking a face or a human, a particle swarm optimization-based tracker outperforms a tracker built on a particle filter in terms of accuracy.

Visual object tracking using particle swarm optimization has been an active research area for several years [18, 19]. Recently, particle swarm optimization was proposed to achieve full body motion tracking [14, 20, 31]. The particle swarm optimization, which is a population-based searching technique, has high search efficiency by combining a local search (using self-experience) and a global one (using neighbor experience). In particular, a few simple rules result in high effectiveness of exploration of a high-dimensional search space. In contrast, in a particle filter, the samples do not exchange information and do not communicate with each other, and thus they have reduced capability of exploring huge search spaces.

In this work, we present a PSO based algorithm for multi-target tracking. At the beginning of each frame, the targets are tracked individually using highly discriminative appearance models among different targets. Each of them is tracked on the basis of separate particle swarm optimizations. The target locations and velocities that are determined by independent trackers are further employed in a particle swarm optimization based algorithm which refines the trajectories extracted in the first phase. Afterwards, a conjugate method is used in the final optimization. At this stage, we utilize a complex energy function which represents the presence, movement, and interaction of all targets within a temporal window consisting of the recent frames.

4.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) [16] is a global optimization algorithm to find the minimum of a numerical function. PSO is a derivative-free, stochastic and population-based computational method often used to optimize functions in rather unfriendly non-convex, non-continuous search spaces. It maintains a swarm of particles, where each one represents a candidate solution. Particles are placed in the search space and move through such a space according to rules which take into account each particle's personal knowledge and the global knowledge of the swarm. Every particle moves with its own velocity in the multidimensional search space, determines its own position, and calculates its fitness using an objective function $f(x)$. Each particle follows simple position and velocity update equations; yet, as particles interact, the collective behavior arises, and the interactions between particles lead to the emergence of global and collective search capabilities, which allow the particles to gravitate towards the global extremum.

At the beginning of the optimization, each individual is initialized with a random position and velocity. While seeking for the best fitness, every individual is attracted towards a position which is affected by the best position p_i found so far by itself and the global best position g found by the whole swarm. In every iteration k, each particle's velocity is first updated based on the particle's current velocity, the particle's local information, and global swarm information. Then, each particle's position is updated using this velocity. The position and velocity of particle i are calculated as follows:

$$v^{(i,k+1)} = \omega v^{(i,k)} + c_1 r_1 \left(p^{(i)} - x^{(i,k)}\right) + c_2 r_2 \left(g - x^{(i,k)}\right), \tag{4.1}$$

$$x^{(i,k+1)} = x^{(i,k)} + v^{(i,k+1)}, \tag{4.2}$$

where the constants c_1 and c_2 are used to balance the influence of the individual's knowledge and that of the group, respectively, r_1 and r_2 are uniformly distributed random numbers, $x^{(i)}$ is position of the ith particle, $p^{(i)}$ is the local best position of particle i, whereas g stands for the global best position, and ω is an inertia constant. The swarm stops the search when a termination criterion is met.

Particles can be attached to each other by any kind of neighborhood topology represented by a graph. In the fully connected neighborhood topology, which is represented by a fully connected graph, all particles in the swarm are connected to one another. Each particle in a swarm represents a candidate solution of the problem. With respect to a fitness function, the best location that has been visited thus far by a particle is stored in the particle's memory. The fitness values corresponding to such best positions are also stored. Additionally, the particles have access to the best location of the whole swarm, i.e., a position that yielded the highest fitness value. A particle therefore employs the best position encountered by itself and the best position of the swarm to move itself toward the optimal value of the objective function.

4.3 PSO-Based Object Tracking

The visual object tracking can be perceived as a dynamic optimization problem. In the PSO-based tracking, in each frame, the object's state is determined using a fitness function expressing the object's appearance. In order to cover possible state changes between consecutive images, the particles are propagated according to a weak transition model. In this section, we show how single object tracking can be accomplished by PSO. We present the fitness function as well as the re-diversification of the swarm to cover the object state changes between the consecutive images.

4.3.1 Multi-patch Based Object Tracking Using Region Covariance

The fitness function is based on the region covariance matrix (RC). The object is represented by an image template consisting in several non-overlapping image patches. For every pixel i in such a patch of size $M \times N$, we calculate a feature vector b_i

$$b_i = (x \quad y \quad R \quad G \quad B \quad I_x \quad I_y)^T \tag{4.3}$$

where x, y represent the Cartesian coordinates of pixel i, whereas R, G, B stand for color components, and I_x, I_y are image derivatives. The RC descriptor is given by:

$$C = \frac{1}{MN-1} \sum_{i=1}^{MN} (b_i - \overline{b})(b_i - \overline{b})^T \tag{4.4}$$

where $\overline{b}$ denotes the vector of means of the corresponding features for the pixels in the template. The region covariance descriptor has many advantages. In particular, RC indicates both spatial and statistical properties of the objects, it allows combining multiple modalities and features, and last but not least, it is capable of relating regions of different sizes. This descriptor is also robust to the variations in illumination conditions, pose, and view. Although the covariance matrices are positive semi-definite in general, in practice they should be regularized by adding a small constant multiple of the identity matrix, making them strictly positive.

In [5], a Log-Euclidean Riemannian metric has been introduced to obtain statistics on symmetric positive definite matrices. The Singular Value Decomposition (SVD) of a symmetric matrix A of size $n \times n$ is $U \Sigma U^T$, where U is an orthonormal matrix, and $\Sigma = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ is diagonal matrix with nonnegative eigenvalues. The matrix exponential $\exp(A)$ of a symmetric matrix is given by: $\exp(A) = U \cdot \mathrm{diag}(\exp(\lambda_1), \ldots, \exp(\lambda_n)) \cdot U^T$; conversely, the matrix logarithm of a symmetric positive definite matrix is calculated according to: $\log(A) = U \cdot \mathrm{diag}(\log(\lambda_1), \ldots, \log(\lambda_n)) \cdot U^T$. Each symmetric matrix is associated to a tensor by the exponential, conversely, a tensor has a unique symmetric matrix logarithm.

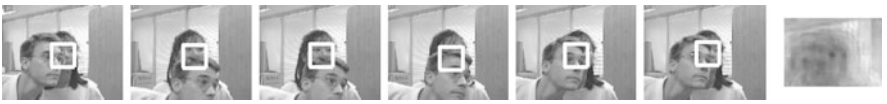

Fig. 4.1 PSO based tracking using multi-patch object representation. Frames #431, 441, 453, 455, 460, 461, and the probability image of the target in frame #431

The distance between two symmetric positive definite matrices X and Y under the Log-Euclidean Riemannian metric can be expressed as follows:

$$\text{dist}(X, Y) = \left\| \log(X) - \log(Y) \right\|_2. \tag{4.5}$$

The Riemannian mean of several elements is an arithmetic mean of matrix elements. Using the Log-Euclidean metric, the algorithm [25] for the incremental subspace update can be employed directly.

In object tracking, we should seek in each frame a location for which the covariance matrix within the object template is most similar to the covariance matrix of the model template. Hence, we should find an object location x^* for which the distance $\text{dist}(\cdot, \cdot)$ between the corresponding covariance matrix X and model covariance matrix $\overline{X}$ assumes the minimal value, i.e., we have to minimize

$$x^* = \arg\min_x \text{dist}(X_x, \overline{X}). \tag{4.6}$$

This is a nonlinear optimization problem that is solved using the PSO algorithm, which in each frame seeks for the best match.

Figure 4.1 depicts some tracking results that were obtained using the multi-patch object representation and a PSO consisting of 10 particles and executing 10 iterations. The tracking of a woman's face was done on color images of size 128×96.[1] We employed both horizontal and vertical patches. The horizontal patches were constructed through dividing vertically the object template into two adjoining patches. Then such patches were divided into 10 horizontally oriented patches, in fives in each of the two vertically oriented patches. The vertical patches were created analogously. The right most image depicts the probability image of the target in frame #431. The detection of outliers is achieved through sorting the scores of the patches and then omitting the poorest ones. The fitness function $f_g(x)$ is the average of K such best matches between the patches of the template at the location x^* and the corresponding patches of the model template.

A tracking algorithm built on the covariance score and with multi-patch object representation can recover after substantial temporal occlusions or large movements. Figure 4.2 illustrates some tracking results that were obtained on the image sequence 'S2L1_View_1' from PETS 2009 database [12], see also Fig. 4.3. As we can observe, the walking woman is successfully tracked despite considerable and multiple temporal occlusions with the static road sign and the pedestrians.

[1] Sequence obtained from http://robotics.stanford.edu/birch/headtracker.

Fig. 4.2 Sub-images with object undergoing tracking in frames #129, 130, 131, 149, 150, 151, 152, 153, 154, 155

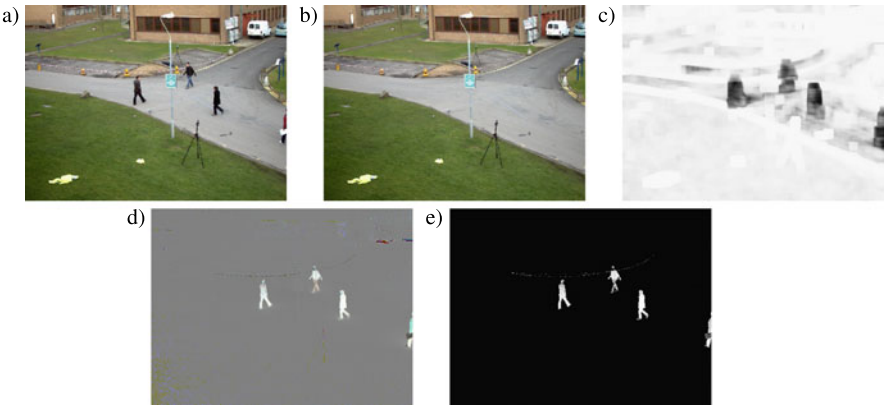

Fig. 4.3 Input image (**a**), reference image (**b**), NCC-based probability image between the reference image and the input image (**c**), color ratios between reference and current image (**d**), and image foreground (**e**)

4.3.2 Foreground Prior

In multiple object tracking, the targets usually become completely or partially occluded. This results in the lack of evidence consisting in non-observability of an occluded target in the image data. In PETS 2009 datasets, some occlusions by the road sign (see images in Fig. 4.2) are relatively long-lasting. As a consequence, the above presented tracker was unable to successfully track some targets in the whole time span, i.e., from entering the scene until exiting the tracking area. Moreover, in a few cases, after loosing the target, the tracker concentrated by mistake on some background areas. In order to cope with such undesirable effects and to decrease the probability of concentrating of the tracker on some non-target areas, we extended the feature vector b_i by a term expressing the object prior. The seventh element of the extended feature vector expresses the object probability which is determined by a foreground segmentation algorithm.

4.3.3 Foreground Segmentation

Our foreground segmentation algorithm is based on a color reference image, which is foreground-free and is extracted automatically in advance, given a sequence of

images with moving targets. Afterwards we employ both region and pixel cues which handle the illumination variations. In addition, we accommodate online the reference image against the illumination and scene changes. The reference image is extracted on the basis of the median of pixel values in some temporal widow. For the 'S2L1_View_1' sequence, the number of images that were needed to extract the foreground free images was equal to 40. Figure 4.3(b) depicts the reference image which was extracted using pixel intensities and the above mentioned number of images.

The normalized cross-correlation *NCC* was used to extract brightness and contrast invariant similarity between the reference image and the current image. It was computed very efficiently using integral images. The *NCC* was used to generate the probability images between the reference images and the current image, see Fig. 4.3(c).

We construct an image of color ratios between the reference image and the current image, where the value of each pixel at location x_1 is given by [4]:

$$\left[\arctan\left(\frac{R^c_{x1}}{R^r_{x1}}\right) \quad \arctan\left(\frac{G^c_{x1}}{G^r_{x1}}\right) \quad \arctan\left(\frac{B^c_{x1}}{B^r_{x1}}\right)\right]^T \tag{4.7}$$

where c and r denote the current and reference image, respectively, whereas R, G, B stand for color components of the *RGB* color space. Such color ratios are independent of the illumination, change in viewpoint, and object geometry. Figure 4.3(d) depicts an example image of color ratios. We can observe that for the pixels belonging to the background the color assumes gray values. This happens because the color channels in the *RGB* color space are highly correlated. Moreover, the color ratios are far smaller in comparison to the ratios between foreground and background. However, as we might observe in the color ratio image there are noisy pixels. The majority of such noisy pixels can be excluded from the image using the probability images, extracted by the normalized cross-correlation.

In our algorithm, we compute online the reference image using the running median. Afterwards, given such an image, we compute the difference image. The difference image is then employed in a simple rule-based classifier, which extracts the foreground objects and shadowed areas. In the classifier, we utilize also the probability image extracted via normalized cross-correlation, as well as the color ratios. The classifier makes decision if pixel is from the background, shadow, or foreground. For shadowed pixels the normalized cross-correlation assumes values near to one. The output of the classifier is the enhanced object probability image. Optionally, in the final stage, we employ the graph-cut optimization algorithm [7] in order to fill small holes in the foreground objects.

4.3.4 Re-diversification of the Swarm

At the beginning of each frame, in some surrounding of the swarm's best location g_t the algorithm selects possible object candidates. Such object candidates are delineated using the foreground blobs. A simple heuristics, which is based on blob

Fig. 4.4 Sub-images with object being tracked in frames #106, 107, 109, 112, 130, 140, and the binary sub-image in frame #112

areas and height-to-width ratios in connection to location of the object at the ground plane, is carried out to select the object candidates. For the videos that were recorded using the calibrated cameras, we project the person locations on the ground onto 3D world coordinates. Such 3D person's location is calculated on the basis of the center of the bottom edge belonging to the bounding box of the blob. Then we employed such information, together with the projected blob sizes, to enhance the delineation of the target candidates as well as to determine the occlusions and splits of the blobs representing the pedestrians into multiple blobs. Afterwards, the particles are initially placed in the gravity centers of the object candidates selected in such a way. The positions of the remaining particles of the swarm are initialized on the basis of the normal distribution which is concentrated around the state estimate at time $t-1$:

$$x_t^{(i)} \leftarrow \mathcal{N}(g_{t-1}, \Sigma) \tag{4.8}$$

where g_{t-1} denotes the location of the best particle that was determined in the previous frame at time $t-1$ and Σ denotes the covariance matrix of the Gaussian distribution whose diagonal elements are proportional to the predicted velocity $v_t = g_{t-1} - g_{t-2}$.

In Fig. 4.4, we can observe the behavior of the tracker with such a swarm re-diversification. As one can notice, the tracking temporally failed in frame #109. Thanks to placing the particles at both candidate objects (see the rightmost image on Fig. 4.4), the tracker correctly recovered the identity of the person in frame #112. It is worth noting that, owing to the object prior in the covariance matrix, the bounding box was placed on the person undergoing tracking and not on the background areas, see frame #109. In order to enhance the object candidate selection, we employed also a person detector [11]. Overall, the person detector found 4550 objects in the 'S2L1_ View_1' dataset. To further enhance the re-diversification of the swarm, the particles were initially placed on the locations determined by the person detector.

4.4 Multiple Object Tracking

The ordinary PSO is not well suited to achieve multiple object tracking. One possible approach to tackle such a problem might be to utilize a PSO that is built on highly discriminative appearance models among different targets, for instance, like those in [10], together with an association framework to achieve better maintaining the identities over time. However, in practice, complex interactions between targets

often lead to difficulties in resolving ambiguities between them. In general, it is relatively easy to track the distinctive objects, but it is much more difficult to achieve reliable tracking when occlusion happens, particularly when the targets have similar appearances. Another approach to this problem might be to represent the positions of feature points by individual particles and to track them using spatial constraints like the maximum distance between feature points together with the maximum distance to the best particle, as it was done in the seminal work [19] (that introduced the PSO for object tracking), and then to select the reliable trajectories on the basis of forward–backward errors [15]. Taking into account the high effectiveness of the PSO when seeking in high-dimensional spaces the problem of multi-object tracking might be formulated as optimization of an energy function, for instance, like those in [3], and estimating the joint state. Recently, the power of the PSO has been fully exploited in multi-object tracking [30], where species-based trackers are employed and each of them tracks one object. In the approach mentioned above, the object interactions are modeled as species competition and repulsion. The occlusion is implicitly inferred using the power of each species and the image observations. Our approach to multiple object tracking is also based on multiple particle swarms. Each object is tracked by a separate swarm. Given the initial tracklets that were determined by the swarms, the refinement of the object's trajectories is done by a PSO-based optimization algorithm. In contrast to [2], which starts an optimization of the energy function from relatively good initial object trajectories and then maintains the identities through the global optimization, in our approach a local optimization takes place in a moving time window. The initial tracklets, which are determined by the swarms, are further distilled in the PSO-based optimization stage that in turn resolves between-object interactions. In the energy function, all target locations belonging to the current time window are considered.

4.4.1 Multiple Object Tracking by Multiple Particle Swarms

In the first phase, the targets are tracked individually. The between-object interactions are initially determined on the basis of our foreground extraction algorithm and a blob analysis. Given the location of a blob in the image as well as the size of its bounding box in relation to the area of the connected component, we decide if a blob represents a single target. In general, a single blob may include multiple objects, while one object may split into multiple blobs. In case of occlusions, two or more swarms responsible for tracking different objects compete for the same target or cluster at the same location. After the end of the occlusion, the swarms should recognize the object identities and continue tracking the objects.

Assuming that in the considered test sequences the people walk on a known ground plane, the location of a candidate target on the ground plane is utilized in evaluation of the expected object area as well as its height and width. This information helps us to decide if the considered target is occluded or if eventually the considered blob is fragmented into several blobs. During the decision making process,

we examine also the distance between the edges of the corresponding rectangles that model the locations and sizes of the objects. Two or more objects are considered as possibly occluded if the distance between the closest edges of the boxes is below a threshold, which in turn depends on the location of the objects on the ground plane. The larger the distance of the object from the camera, the smaller the threshold. At this stage we take into account the distance between the locations of the global best particles in the previous frame, too. The information about the matching of individual patches composing the object templates with the reference templates is considered in the decision process mentioned above and helps us to decide which object or objects are occluded and which are occluding. The search space of the particle swarm with the smaller fitness value is gradually expanded to allow the recovery of the target after occlusion. In scenes with a layout like a corridor with a long vertical passage, with many pairs of pedestrians, etc., where a probability of long term occlusion and the lack of evidence in a longer period of time is considerable, we extract the targets that are close to each other and have similar motion directions. In case of such long term occlusions, we estimate the location (motion) of the occluded object on the basis of the location of the occluder.

As we already mentioned, at this stage the targets are tracked individually. A swarm responsible for tracking a single person is created at the moment of entering the tracked area. The swarm finishes the tracking if the person leaves the tracking scene. Such a scenario greatly simplifies the resolving of interactions, as in each time instant we known the number of the targets. In the presented approach, the position of the target is always defined.

The object tracking is done using the algorithm discussed in Sect. 4.3. In contrast to a typical approach for object tracking, where a model of the object appearance is accommodated over time, in our approach we maintain a pool of models expressing the object appearance at various poses or in different camera views. The object location is determined on the basis of the most similar object model from such a pool of the object models. Each target maintains a constant number of the models in the pool. If the target is not occluded, i.e., the area of the blob as well as the size of the surrounding blob is consistent with the location of the target on the ground-plane, the person detector successfully sought a person in the proximity of the considered person location, the value of the objective function is above an assumed threshold, we replace the pre-selected in advance model by a model determined at the best object location. At the end of the occlusion, or optionally when a target leaves the pre-specified area surrounding the road sign in the 'S2L1_View_1' sequence, we perform the object back-tracking using the above-mentioned pool of the object models. If the back-tracker arrives to a different object, on the basis of the pool of the object model we calculate the sum of the fitness values on both trajectories and choose the trajectory with better fitness. The size of the template modeling the object is determined with regard to its location on the ground-plane.

4.4.2 Refinement of the Tracklets by Particle Swarm Optimization

Particle swarm optimization demonstrated to be an efficient global search method for nonlinear complex systems without a priori knowledge about the system structure. Here, we employ its potential in the optimization of the complex energy function which represents the presence, movement, and interaction of all targets in a sequence of last frames within a temporal window. If the calibration data are available, the tracking is done in the world coordinates. This means that object locations at the ground-plane that were determined by individual trackers are projected to 3D.

Our energy function consists of three terms expressing the pedestrian presence, priors for the pedestrian motion, and mutual exclusion:

$$E(X) = \alpha E_l + \beta E_v + \gamma E_c. \tag{4.9}$$

The vector X consists of the ground-plane coordinates of all targets being in the scene from current time t to time $t - T$. This means that the energy is minimized in a temporal window comprising the last T frames.

The energy should be smaller for the trajectories going around regions of high pedestrian likelihood. Thus, the term expressing the pedestrian presence is given by:

$$E_l(X) = -\sum_{\tau=t}^{t-T}\sum_{id=1}^{P}\exp\left(-\sigma_l^2\sum_{h=1}^{H(t)}\left\|x_\tau^{(id)} - d_\tau^{(h)}\right\|^2\right) \tag{4.10}$$

where t stands for the current time, P is the number of the targets, whereas $H(t)$ denotes the number of the detections in frame τ, and the $d_\tau^{(h)}$ is the location of the detection h in frame τ. The term expressing the motion of the target favors movement with a constant velocity:

$$E_v(X) = \sum_{\tau=t}^{t-T}\sum_{id=1}^{P}\left\|v_\tau^{(id)} - v_{\tau-1}^{(id)}\right\|^2. \tag{4.11}$$

The term expressing the mutual exclusion should penalize the trajectory configurations if two targets approach each other. It assumes the following form:

$$E_c(X) = \sum_{\tau=t}^{t-T}\sum_{id_i\neq id_j}\frac{s_c}{\|x_\tau^{(id_i)} - x_\tau^{(id_j)}\|^2} \tag{4.12}$$

where s_c is a scale factor.

The deterministic optimization algorithms like gradient descent converge rapidly, but may get stuck in local minima of multimodal functions. In the vicinity of a local optimum, the deterministic algorithms converge faster than stochastic search algorithms because stochastic search algorithms waste computational time doing a random search. On the other hand, the PSO may avoid becoming trapped in local optima and find the global optimum. Therefore, in our algorithm the energy function

is first optimized by a PSO and then by a conjugate gradient algorithm [26]. The search area of the PSO is sufficiently large to cover promising configurations. In the PSO, we employ 40 particles, and the maximum number of the iterations is set to 300. The locations determined by the individual person trackers are employed to initialize the PSO, whereas the output of the PSO is used as starting trajectory of the conjugate gradient optimization algorithm which is responsible for the final refinement of the trajectories. Thus, the particle swarm algorithm is utilized to seek good local minima and the conjugate gradient is used to find the local minimum accurately. The optimization is done using person coordinates and velocities from a sequence of the last frames. Thus, the state vector X consists of the person locations determined in the current frame by individual trackers and the refined locations of all persons in a sequence of the last frames.

We achieved considerable improvement of the results by running the optimization on only last 20 frames. For each person entering the tracking area, the optimization starts in the seventh frame. In the eight frame, the optimization algorithm runs on the current locations determined by individual trackers and the refined locations from frames #2–7, etc. Substantial improvement of the tracking accuracy was observed in scenarios with considerable temporal occlusions. In such scenarios, the blobs representing the pedestrians are frequently fragmented, the trackers temporally loose the tracks, making uncoordinated jumps from one object to another. Owing to the energy optimization which considers the interactions of all targets in a sequence of the last frames, the trajectories are far smoother, and most importantly, they pass through regions of high pedestrian likelihood.

4.5 Experiments

The algorithm was evaluated on two publicly available video sequences. The performance of our PSO-based algorithm for multi-object tracking was compared with the performance of the available PSO-based algorithm [30] for tracking multiple objects. In this recently proposed algorithm, species-based trackers are employed and each tracking one object. The object interactions are modeled as species competition and repulsion, whereas the occlusion is implicitly inferred using the power of each species and the image observations. The discussed method has been evaluated on 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 tracking performance of our algorithm was compared with the performance of the algorithm mentioned above on an image sequence that is called 'ThreePastShop2cor', which consists of color RGB images of size 384×288, recorded with 25 frames per second. Figure 4.5 depicts some key frames, where three pedestrians are tracked through occlusion. All three persons were correctly tracked in 108 frames. Thanks to patch-based representation of the object template, the algorithm is able to select the occluding object.

The algorithm was compared with state-of-the-art algorithms for multi-object tracking by analyses carried out both through qualitative visual evaluations as well

Fig. 4.5 Tracking three persons undergoing occlusions. Frames #422, 455, 465, 480, 488, 518

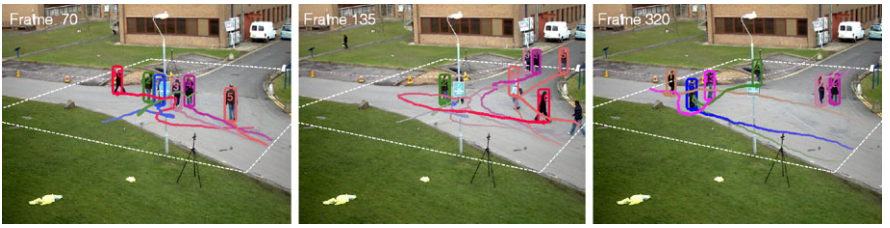

Fig. 4.6 Tracking results on the PETS 2009 *S2L*1_*View*_1 dataset with trajectory refinement using PSO. Frames #70, 130, 320

Table 4.1 Quantitative comparison of our method with state-of-the-art methods on the *S2L*1_*View*_1 sequence from PETS 2009 data set

Metric	[6]	[3]	[2]	Current work
MOTA	79.0 %	81.4 %	88.3 %	90.4 %
MOTP	59.0 %	76.1 %	75.7 %	85.2 %
MT	–	82.6 %	87.0 %	91.3 %
ML	–	0.0 %	4.4 %	4.4 %

as quantitatively using the latest VS-PETS benchmark from 2009 [12]. The experiments were carried out on the sequence 'S2L1_View_ 1', which was recorded at 7 frames per second and contains 795 color images of size 768×576.

The algorithm was evaluated using *CLEAR* metrics [27]. The Multi-Object Tracking Accuracy (*MOTA*) counts all missed targets, false positives, and identity mismatches. It is normalized to tracking all targets such that 100 % means no errors. The Multi-Object Tracking Precision (*MOTP*) expresses the normalized distance between the ground truth location and the estimated location. Mostly Tracked (*MT*) accounts for the percentage of ground-truth trajectories that are covered by the tracker for more than 80 % in length, whereas Mostly Lost (*ML*) is the percentage of the ground-truth trajectories that are covered by the tracker for less than 20 % in length [21]. Table 4.1 illustrates the accuracy and precision, as well as the number of mostly tracked and mostly lost trajectories. The accuracy is a bit higher than 90 %. When no optimization was used, the accuracy was somewhat below 75 %. The percentage of mostly tracked trajectories is nearly 4.5 % higher in comparison to the best reported results.

Figure 4.6 depicts some tracking results. It also shows ground-plane trajectories. As we can observe, the trajectories are far longer in comparison to trajectories that

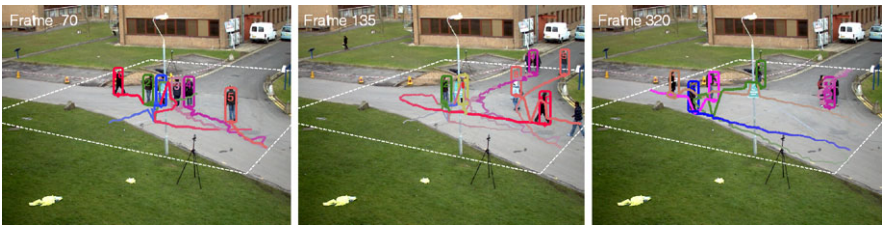

Fig. 4.7 Tracking results on the PETS 2009 *S2L1_View_1* dataset. Frames #70, 130, 330

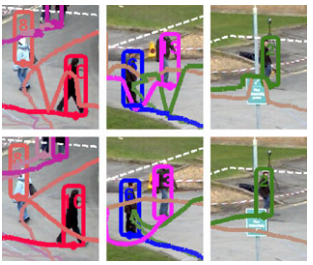

Fig. 4.8 The trajectories without optimization (*top row*) and with the optimization (*bottom row*). Sub-images from frames #135, 320, and 320

are depicted on relevant images in [2]. In almost 40 occlusions like those in frames 129–131 of Fig. 4.2, where the targets undergo temporal occlusion and then split into separate blobs, or the target is occluded by the road sign like in frames 106–112 of Fig. 4.4, the algorithm properly recognized the identities of the targets, avoided clustering on a single target, despite some temporal errors in location or identity estimation.

Figure 4.7 illustrates some tracking results that were obtained using only individual tracking. As we can observe, the trajectories are not so smooth in comparison to the trajectories obtained through the optimization. In particular, one can observe considerable jitters in the trajectories as a result of temporal switches of the identities, see, for instance, a jump close to the road sign in frame #70 of Fig. 4.7.

Our results demonstrate that in multi-object tracking, considerable improvement of the tracking accuracy can be obtained through the use of an optimization algorithm for the refinement of the results obtained by individual trackers, even if they are built on highly discriminative appearance models among different targets. Through formulating an energy function that operates on all targets that are present in a sequence of last frames within a temporal window, and thus takes into account all interactions between them, it is possible to considerably refine the trajectories obtained by individual trackers, see Fig. 4.8.

In our algorithm, in contrast to [3], the joint state is optimized only in some moving temporal widow, which moves forward as the time elapses. The state vector consists of the states determined by the individual trackers in the current frame and the states that were progressively refined in previous frames. In contrast to the algorithm mentioned above, no sophisticated initialization of the optimization algorithm in the form of the pre-calculated trajectories by an Extended Kalman Filter (EKF) or globally optimal discrete tracker based on linear programming [1] is needed. We

also demonstrated that the PSO algorithm is an effective tool for solving such nonlinear and nonconvex energy functions. Since the PSO does not rely on any gradient information, smoothness, or continuity properties, it is possible to employ in the objective functions the terms that employ information, for instance, about the nearest neighbors, identity switches, etc. The PSO-algorithm has also demonstrated great usefulness in single object tracking where swarms consisting of 20 particles and in 10 iterations are able to follow objects, even in case of considerable temporal occlusions. The discussed algorithms were implemented in MATLAB/C.

4.6 Conclusions

We demonstrated that in multi-object tracking, considerable improvement of the tracking accuracy can be obtained through the use of an optimization algorithm for the refinement of the results obtained by individual trackers, even if they are built on highly discriminative appearance models. In the presented algorithm, the joint state is optimized in some moving temporal widow. The state vector consists of the states determined by the individual trackers in the current frame and the states that were progressively refined in the previous frames. We demonstrated that the particle swarm optimization is an effective tool for solving such nonlinear and nonconvex energy functions. Individual object tracking was considered as a numerical optimization problem, where a particle swarm optimization was utilized in searching for the best local mode of the similarity measure.

References

1. Andriyenko, A., Schindler, K.: Globally optimal multi-target tracking on a hexagonal lattice. In: Proc. of the 11th European Conf. on Computer Vision: Part I, pp. 466–479 (2010)
2. Andriyenko, A., Schindler, K.: An analytical formulation of global occlusion reasoning for multi-target tracking. In: IEEE Int. Workshop on Visual Surveillance, pp. 1839–1846 (2011)
3. Andriyenko, A., Schindler, K.: Multi-target tracking by continuous energy minimization. In: IEEE Int. Conf. on CVPR, pp. 1265–1272 (2011)
4. Arsic, D., Lyutskanov, A., Rigoll, G., Kwolek, B.: Multi-camera person tracking applying a graph-cuts based foreground segmentation in a homography framework. In: IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance, pp. 30–37 (2009)
5. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Log-Euclidean metrics for fast and simple calculus on diffusion tensors. Magn. Reson. Med. **56**, 411–421 (2006)
6. Berclaz, E.T.J., Fleuret, F., Fua, P.: Multiple object tracking using k-shortest paths optimization. IEEE Trans. Pattern Anal. Mach. Intell. **33**(9), 1806–1819 (2011)
7. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23**(11), 1222–1239 (2001)
8. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L.J.: Robust tracking-by-detection using a detector confidence particle filter. In: ICCV'09, pp. 1515–1522 (2009)
9. Cai, Y., de Freitas, N., Little, J.J.: Robust visual tracking for multiple targets. In: ECCV, vol. IV, pp. 107–118 (2006)
10. Cheng-Hao, K., Huang, C., Nevatia, R.: Multi-target tracking by on-line learned discriminative appearance models. In: IEEE Int. Conf. on CVPR, pp. 685–692 (2010)

11. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Int. Conf. on CVPR, vol. 1, pp. 886–893 (2005)
12. Ferryman, J., Shahrokni, A.: PETS2009: dataset and challenge. In: IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance, pp. 1–6 (2009)
13. Isard, M., Blake, A.: Condensation—conditional density propagation for visual tracking. Int. J. Comput. Vis. **29**, 5–28 (2006)
14. John, V., Trucco, E., Ivekovic, S.: Markerless human articulated tracking using hierarchical particle swarm optimisation. Image Vis. Comput. **28**(11), 1530–1547 (2010)
15. Kalal, Z., Mikolajczyk, K., Matas, J.: Forward–backward error: automatic detection of tracking failures. In: Int. Conf. on Pattern Rec., pp. 2756–2759 (2010)
16. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. of IEEE Int. Conf. on Neural Networks, pp. 1942–1948 (1995)
17. Khan, Z., Balch, T., Dellaert, F.: MCMC-based particle filtering for tracking a variable number of interacting targets. IEEE Trans. Pattern Anal. Mach. Intell. **27**, 1805–1918 (2005)
18. Koelsch, M., Turk, M.: Flocks of features for tracking articulated objects. In: Kisacanin, T.H.B., Pavlovic, V. (eds.) Real-Time Vision for Human–Computer Interaction. Springer, Berlin (2005). Chap. 9
19. Koelsch, M., Turk, M.: Hand tracking with flocks of features. In: IEEE Int. Conf. on CVPR, vol. 2, p. 1187 (2005)
20. Kwolek, B., Krzeszowski, T., Wojciechowski, K.: Real-time multi-view human motion tracking using 3D model and latency tolerant parallel particle swarm optimization. In: 5th Int. Conf. MIRAGE, pp. 169–180. Springer, Berlin (2011)
21. Li, Y., Huang, C., Nevatia, R.: Learning to associate: hybridboosted multi-target tracker for crowded scene. In: IEEE Int. Conf. on CVPR, pp. 2953–2960 (2009)
22. Li, Y., Huang, C., Nevatia, R.: Stable multi-target tracking in real-time surveillance video. In: CVPR, pp. 2953–2960 (2009)
23. Okuma, K., Taleghani, A., De Freitas, N., Little, J.J., Lowe, D.G.: A boosted particle filter: multi-target detection and tracking. In: ECCV, pp. 28–39 (2004)
24. Rasmussen, C., Hager, G.D.: Probabilistic data association methods for tracking complex visual objects. IEEE Trans. Pattern Anal. Mach. Intell. **23**, 560–576 (2001)
25. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. Int. J. Comput. Vis. **77**(1–3), 125–141 (2008)
26. Steihaug, T.: The conjugate gradient method and trust regions in large-scale optimization. SIAM J. Numer. Anal. **20**, 626–637 (1983)
27. Stiefelhagen, R., Bernardin, K., Bowers, R., Garofolo, J.S., Mostefa, D., Soundararajan, P.: The CLEAR 2006 evaluation. In: CLEAR. LNCS, vol. 4122, pp. 1–44. Springer, Berlin (2006)
28. Yang, H., Shao, L., Zheng, F., Wang, L., Song, Z.: Recent advances and trends in visual tracking: a review. Neurocomputing **74**(18), 3823–3831 (2011)
29. Zhang, X., Hu, W., Maybank, S., Li, X., Zhu, M.: Sequential particle swarm optimization for visual tracking. In: IEEE Int. Conf. on CVPR, pp. 1–8 (2008)
30. Zhang, X., Hu, W., Qu, W., Maybank, S.: Multiple object tracking via species-based particle swarm optimization. IEEE Trans. Circuits Syst. Video Technol. **20**(11), 1590–1602 (2010)
31. Zhang, X., Hu, W., Wang, X., Kong, Y., Xie, N., Wang, H., Ling, H., Maybank, S.: A swarm intelligence based searching strategy for articulated 3D human body tracking. In: IEEE Workshop on 3D Information Extraction for Video Analysis and Mining, pp. 45–50. IEEE, New York (2010)