Particle Swarm Optimization with Soft Search Space Partitioning for Video-Based Markerless Pose Tracking

Patrick Fleischmann[1], Ivar Austvoll[2], and Bogdan Kwolek[3]

[1] Institute for Communication Systems ICOM,
University of Applied Sciences of Eastern Switzerland,
8640 Rapperswil, Switzerland
patrick.fleischmann@hsr.ch
[2] Dept. of Electrical and Computer Engineering,
University of Stavanger, N-4036 Stavanger, Norway
ivar.austvoll@uis.no
[3] Rzeszów University of Technology, 35-959 Rzeszów, Poland
bkwolek@prz.edu.pl

Abstract. This paper proposes a new algorithm called soft partitioning particle swarm optimization (SPPSO), which performs video-based markerless human pose tracking by optimizing a fitness function in a 31-dimensional search space. The fitness function is based on foreground segmentation and edges. SPPSO divides the optimization into two stages that exploit the hierarchical structure of the model. The first stage only optimizes the most important parameters, whereas the second is a global optimization which also refines the estimates from the first stage. Experiments with the publicly available Lee walk dataset showed that SPPSO performs better than the annealed particle filter at a frame rate of 20 fps, and equally well at 60 fps. The better performance at the lower frame rate is attributed to the explicit exploitation of the hierarchical model structure.

Keywords: Video Processing, Particle Swarm Optimization, Motion Capture

1 Introduction

Pose tracking, also known as motion capture, is the process of sequentially estimating the pose and position of a human subject in a sequence of images. The applications of video-based markerless human pose tracking range from computer games to medical gait analysis. The ubiquitous presence of computer vision hardware in the modern world offers many opportunities for video based human motion capture.

Markerless pose tracking is a hard problem due to ambiguities and self-occlusions arising from the mapping of 3D body poses to 2D images. The high dimensionality of the parameter space is another major problem in all approaches

that use articulated body models. The required number of parameters for a full body model is often over 30, even for coarse models. The most successful pose tracking algorithms are interacting simulated annealing (ISA) [5] and the annealed particle filter (APF) [4]. They are both optimization algorithms that aim to find the maximum of the posterior probability for tracking.

Particle Swarm Optimization (PSO) [9] is well suited for parameter optimization problems like pose estimation. It was first applied to markerless pose estimation by Ivekovic and Trucco in 2006 [7]. They only performed static pose estimation of the upper body in this work. In two recent papers, Ivekovic et al. describe a hierarchical approach using PSO for full body pose tracking [7, 8]. They use the articulated human model of the Lee walk dataset [1] and divide the 31-dimensional parameter space into 12 hierarchical subspaces to overcome the problem of high dimensionality. This approach has some shortcomings because the optimization cannot escape from local maxima found in preceding hierarchical levels. Moreover the final solution tends to drift away from the true pose, especially at low frame rates [8].

Krzeszowski et al. propose a global local PSO (GLPSO) [10] where the PSO-based optimization is divided into two stages. The first stage is a global optimization of the pose and the second stage is a local refinement of the limb configuration. This is done for the legs and arms separately. Kwolek et al. [11] combine the global-local approach with a modified PSO named global local annealed PSO (GLAPSO). The most notable property of this variant of PSO is the quantization of the fitness function. Instead of one global best particle, the algorithm maintains a pool of candidates, which improves the algorithm's ability to explore the search space. This modification improves the tracking performance and allows the use of fewer fitness evaluations.

Robertson and Trucco use an approach where the number of optimized parameters is iteratively increased so that a superset of the previously optimized parameters is optimized at every hierarchical stage [13]. This approach, like SPPSO, exploits the hierarchical structure of the body model while avoiding the error accumulation problem of other hierarchical approaches. The main difference to SPPSO is that they use 3D observations in contrast to 2D for SPPSO and they require 6 optimization stages for an upper body model, whereas SPPSO requires only two stages for a full body model.

Hierarchical optimization, as well as global local PSO, divides the optimization into multiple stages, in which a subset of the parameters is optimized while the rest of the parameters are fixed. This is a hard partitioning of the search space. The term soft partitioning was introduced by Deutscher et al. to describe the way the annealed particle filter automatically adjusts the sampling variance of individual parameters [4]. In contrast to hard partitioning, soft partitioning means that some parameters are allowed more variance than others, but no parameters are completely fixed. The annealed particle filter adjusts the variance fully automatic. It uses no prior information about the hierarchical structure of the body model and is therefore a very general approach. SPPSO on the other hand, explicitly exploits the hierarchical structure.

SPPSO belongs to the class of direct model use (generative) algorithms. That means it incorporates a 3D model of the human body in an analysis-by-synthesis fashion [12]. The kinematic structure is modelled by a kinematic tree with the joint angles as the variable parameters during tracking. A kinematic tree for a full body model requires around 30 parameters. Such a high number of degrees of freedom (DoF) makes pose estimation and tracking a very hard problem. Examples of direct model use algorithms can be found in [2, 6, 14]. When the type of motion (e.g. walking) is known, a strong (action specific) motion model can be used to predict possible poses in the next frame. When the motion is arbitrary, a weak motion model must be used. SPPSO uses a zero motion model, which is suitable for any type of motion as long as it is not too fast. Algorithms that use a weak motion model generally require multiple camera views to alleviate the ambiguities and self-occlusion problems [12]. SPPSO was evaluated using four camera views.

In this work, human pose tracking is achieved by Particle Swarm Optimization. The major contribution is a novel PSO with soft search space partitioning for markerless human pose tracking in multi-view videos. The algorithm has been compared with the state-of-the-art annealed particle filter in qualitative and quantitative evaluations using the Lee walk dataset which includes multi-view video and ground truth poses.

2 SPPSO-based Human Pose Tracking

2.1 Body Model

The body model used by SPPSO is a modified version of the publicly available model used by Balan et al. [1]. It uses a kinematic tree with 31 parameters to describe a human pose. The first six parameters determine the global position and orientation of the model and the remaining parameters are relative joint angles. The outer shape of the body is modelled with ten truncated cones (henceforth called cylinders), which are fixed to the kinematic tree. Figure 1 shows the kinematic tree and the cylinder model. The dimensions of the cylinders were determined by Balan et al. using marker-based motion capture and are kept constant during tracking. The cylinder model is used to project the silhouettes and edges to the four camera views, where they are compared to the silhouettes and edges that were extracted from the four videos using image processing.

2.2 Fitness Function

The body model described above is used to compute the fitness of candidate poses, defined by the parameters $x_1 - x_{31}$. The fitness indicates how well a candidate pose matches the observations, i.e. the images from all four views at the time instant. The fitness $f = f_s + f_e$ is the sum of two terms: the *silhouette fitness* f_s and the *edge fitness* f_e. Both terms are normalized to lie in the range

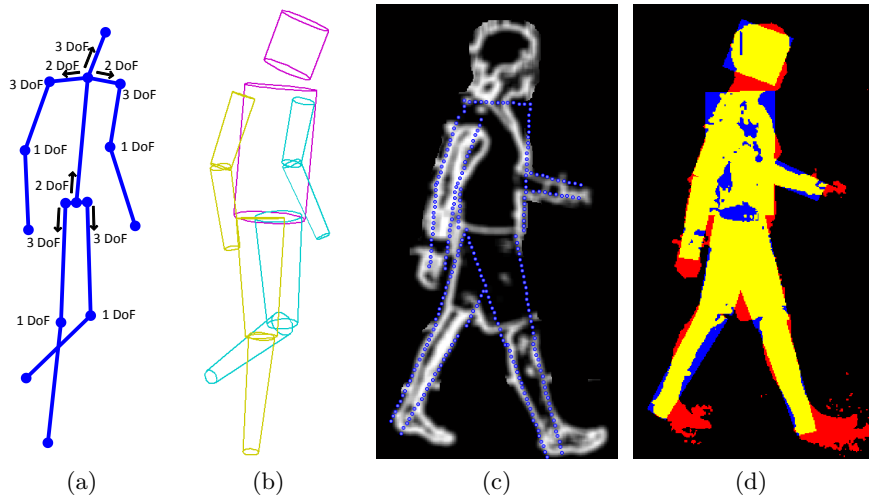


Fig. 1. (a) Kinematic tree of the body model with 31 degrees of freedom. (b) Cylinder model with ten cylinders (right limbs in yellow, left limbs in cyan). (c) Sampling points for the edge fitness function, overlaid on the edge map. (d) Image segmentation for the silhouette fitness. Red: in observed silhouette but not in projected, blue: in projected but not in observed, yellow: overlap of both silhouettes

between 0 and 1, where 0 means no match and 1 means complete match. Both partial fitness terms are defined on a single observed image. The total fitness is computed by first averaging the partial fitness values of all four views and then summing the two averaged partial fitness values.

Edges are a robust image feature and human subjects usually produce strong edges along the outline of the body and individual limbs. Edges are therefore a valuable feature for pose tracking [4]. The edge map is obtained by edge detection, blurring, and rescaling to the range between 0 and 1. The edge fitness f_e is then computed by sampling the edge map at discrete points along the visible edges of the candidate pose, similar to the *edge likelihood* used by Balan et al. [1]. The edge fitness can be computed for all, or for individual cylinders. At the first stage of SPPSO, only the torso cylinder is considered for the edge fitness. Except for the torso cylinder, only the edges parallel to the cylinder axes of the model are considered. The upper edge of the torso is also sampled because it provides a valuable hint for the z-location of the model. The head cylinder is never used for the edge fitness because the cylindrical shape is only a very crude approximation of the head's shape. Figure 1c depicts the sampling points for the edge fitness overlaid on the edge map.

The silhouette fitness f_s measures the overlap of the observed silhouette and the projected silhouette. The *observed silhouette* is a binary image, obtained by foreground-background segmentation of the observed image. The *projected*

silhouette is obtained by projecting the cylinders of the candidate pose into the respective view.

A good silhouette fitness must be bidirectional [14]. This means that it must measure how much of the projected silhouette falls into the observed, as well as how much of the observed silhouette falls into the projected. This is necessary to prevent unreasonably high fitness values for poses that have overlapping limbs. SPPSO uses a silhouette fitness based on the bidirectional silhouette log-likelihood used by Sigal et al. [14]. It is computed as follows: Let R be the area that lies in the observed, but not in the projected silhouette, B the area that lies in the projected, but not in the observed silhouette, and Y the overlap area of both silhouettes (See Fig. 1d for an illustration of this segmentation.). The silhouette fitness is then computed as $f_s = \frac{1}{2}\frac{Y}{B+Y} + \frac{1}{2}\frac{Y}{R+Y}$. Hence, f_s is 1 when the two silhouettes are identical, and 0 when there is no overlap.

2.3 Optimization

SPPSO maximizes the fitness for every new frame with a particle swarm optimization. The estimated pose from the previous frame is used to initialise the optimization. Hence, the tracking process is a series of static optimizations. These optimizations are divided into two hierarchical stages and both stages use a constricted PSO, as introduced by Clerc and Kennedy [3].

Each particle in the PSO constitutes a candidate pose. Its position vector consists of the variable parameters of the body model (i.e. the position and angles of the kinematic tree). The initial particle positions x_i^t are sampled from a multivariate normal distribution, centred around the estimated pose from last frame $\widehat{x}^{t-1}$.

$$x_i^t \leftarrow \mathcal{N}(\widehat{x}^{t-1}, \Sigma), \quad \Sigma = \begin{pmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_{31}^2 \end{pmatrix}, \quad \sigma = \begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_{31} \end{pmatrix} . \tag{1}$$

Σ is the same diagonal covariance matrix as used for the first annealing layer in the annealed particle filter of Balan et al. [1]. The standard deviations σ_d in Σ, where d stands for dimension, are equal to the maximum absolute inter-frame differences of the parameters in a training set of motion capture data at 60 fps. For example: σ_4 (x-translation) is 13.7 mm and σ_{10} (left knee angle) is 0.093 rad. Therefore, the distribution $\mathcal{N}(\widehat{x}^{t-1}, \Sigma)$ can be interpreted as a prior probability for the parameters at time t and it is reasonable to sample the initial particle set from this distribution.

The used training set focuses primarily on walking motions. Therefore, this covariance matrix can be regarded as a weak model for walking motions. This bias towards walking motions could be removed by using a training set with more diverse motions. But this would enlarge the search space and therefore make the tracking of a walking subject more difficult. For experiments at slower frame rates than 60 fps, σ is always upscaled accordingly. For example, σ is multiplied by three when tracking at a frame rate of 20 fps.

The particle velocity is limited to two times the standard deviation in every dimension to prevent unreasonable poses. The initial particle velocities are sampled from a uniform distribution in the interval $[-\sigma, +\sigma]$.

The optimization is subject to two constraints: (i) The angles must remain inside anatomical joint limits and (ii) the limbs may not inter-penetrate. These constraints are equal to the *hard priors* of Balan et al. [1]. They were found to improve the tracking performance significantly by Balan et al. because they reduce the search space. The constraints are enforced by resampling the particle velocity until either the constraints are met or the maximum number of 10 attempts is exceeded.

Algorithm 1 shows the PSO that is used at the two stages of SPPSO. The coefficients $c_1(k)$ and $c_2(k)$ are linearly increased from 2.05 to 2.15 during the optimization to gradually increase the algorithm's tendency to converge. Consequently, the constriction factor $\chi(k)$ is adapted according to (3) for every iteration to ensure convergence [3]. This can be seen as an annealing scheme which was introduced to enforce swarm convergence even with a limited number of iterations N.

$$c_1(k) = c_2(k) = \frac{0.1}{N-2}(k-2) + 2.05 \ . \tag{2}$$

$$\chi(k) = \frac{2}{\left|2 - \varphi(k) - \sqrt{\varphi(k)^2 - 4\varphi(k)}\right|}, \ \varphi(k) = c_1(k) + c_2(k) \ . \tag{3}$$

Algorithm 1 Constricted PSO with enforced constraints for one stage of SPPSO.

sample particle positions $x_i \leftarrow \mathcal{N}(\widehat{x}^{t-1}, \Sigma)$
sample particle velocities $v_i \leftarrow \mathcal{U}(-\sigma, \sigma)$
calculate particle fitness: $f(x_i) = f_s(x_i) + f_e(x_i)$
update particle best p_i and global best p_g
for each iteration $k = 2$ **to** N **do**
  for each particle i **in the swarm do**
    repeat
      for each dimension d **do**
        $v_{id} = \chi(k)(v_{id} + c_1(k)\epsilon_1(p_{id} - x_{id}) + c_2(k)\epsilon_2(p_{gd} - x_{id}))$
      end for
      limit abs(v_i) **to** 2σ
      $x_i = x_i + v_i$
    until x_i **meets constraints**
    calculate particle fitness: $f(x_i) = f_s(x_i) + f_e(x_i)$
    update particle best p_i **and global best** p_g
  end for
end for

2.4 Soft Partitioning Stages

The optimization of the pose is divided into two hierarchical stages. Both stages are complete optimizations with the above described PSO and the estimated pose from the first stage is used as the initialisation for the second stage.

Pose estimation which is divided into hierarchical stages with hard partitions suffers from error accumulation. This happens because the fitness function for one stage cannot be evaluated completely independently from subsequent stages. SPPSO uses a soft partitioning scheme to avoid error accumulation. Figure 2 illustrates the principle of soft partitioning compared to hard (hierarchical) partitioning. As in hierarchical schemes, the search space is partitioned according to the model hierarchy. The most important parameters are optimized first, while the less important are kept constant. The crucial difference to hard partitioning is that the previously optimized parameters are allowed some variation in the following stage. Soft partitioning reduces the search space not as much as hard partitioning but the search space is much smaller than in a global optimization. This allows a much more efficient search for the optimal pose.

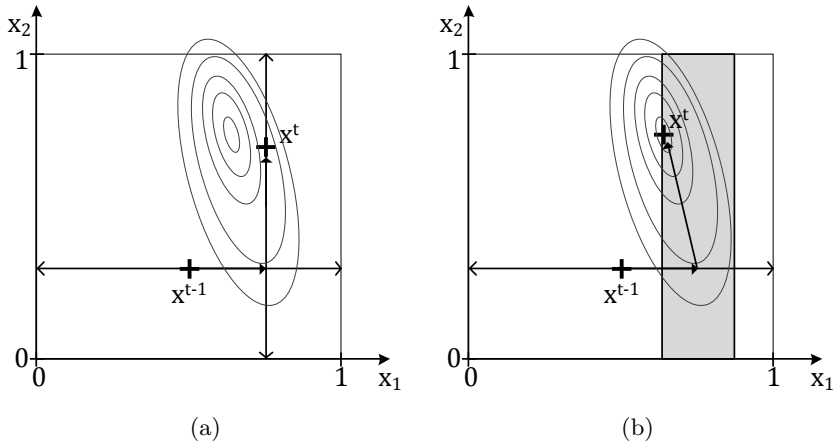


Fig. 2. (a) Optimization with hard partitioning. At the first stage, x_1 is optimized while x_2 is kept constant. At the second stage, x_1 is kept constant while x_2 is optimized. Consequently, the optimizer cannot correct the suboptimal estimate of x_1 from the first stage. (b) Soft partitioning. The first stage is identical to the hierarchical scheme, but x_1 is allowed some variation at the second stage. Therefore, the optimizer finds a better estimate

SPPSO has two hierarchical stages, where only the first six parameters $x_1 - x_6$ (global orientation and position) are optimized in the first stage. The second stage is a global optimization of all parameters where the standard deviations for $x_4 - x_6$ (global position) are divided by an empirically determined factor

of ten. Experiments showed that the tracking performance is not significantly increased when the optimization is further divided into three stages. However, the soft partitioning scheme performs much better than global optimization or hard partitioning.

3 Experimental Results

This section presents the experimental evaluation of SPPSO. After showing general results and establishing the maximum obtainable tracking accuracy with the used body model, SPPSO is compared to the annealed particle filter (APF), which is the benchmark algorithm of the HumanEva framework [1, 14]. Finally, the computation time for different parts of the SPPSO algorithm is analysed.

Table 1 shows the number of particles and iterations per optimization stage as well as the used fitness functions for the experiments with 1000 fitness evaluations. Keeping the number of evaluations fixed allows a fair comparison to other algorithms because fitness evaluations (including rendering) dominate the total processing time. 1000 evaluations per frame is the standard number of evaluations in the HumanEva framework [14].

Table 1. Configuration of SPPSO for experiments with 1000 fitness evaluations

Stage	Particles	Iterations	Edge fitness	Silhouette fitness
1	10	20	only torso	full body
2	20	40	full body	full body

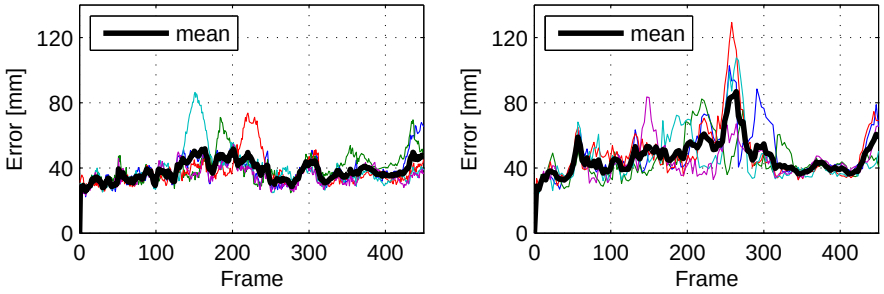
The experiments were performed on the Lee walk dataset which contains video at 60 fps from four views, showing a human subject that walks in a circle. The ground truth poses contained in the dataset were obtained using a marker-based motion capture system. All reported tracking errors are computed using the standard error measure of Balan et al. [1], which is the average 3D-distance of 15 marker joints to the ground truth positions. Figure 4 shows that the tracking has some errors at 20 fps, but it can generally follow the body configuration. However, at 60 fps the tracking is accurate enough.

Generally, the tracking gets more accurate with higher frame rates and more evaluations per frame. This is also illustrated by Table 2, which shows the mean and maximum tracking error of several tracking runs that were performed to establish the minimal obtainable tracking error. The minimal mean error is reached at about 35 mm and is limited by two factors: First, the ground truth poses are not perfectly accurate. Second, the body model is very coarse.

Figure 3 shows error plots produced by SPPSO with 1000 evaluations per frame at 60 fps and 20 fps. The mean error is a little smaller at 60 fps and the tracking is generally more stable. The outliers in the graph at 60 fps come from a temporarily lost arm. At 20 fps, the maximum error is much higher

Table 2. Tracking error of SPPSO at 60 fps with different evaluation rates. The table shows mean and maximum 3D error on the first 450 frames of the Lee walk sequence

evaluations/frame	1000	2000	4000
runs	5	3	1
mean error [mm]	38.0	37.3	35.7
max error [mm]	86.4	82.3	56.7



(a) 60 fps, $D_{\text{mean}} = 38$ mm (b) 20 fps, $D_{\text{mean}} = 46$ mm

Fig. 3. 3D tracking error of SPPSO with 1000 evaluations per frame for the Lee walk sequence. The graphs show five individual runs and the mean error

because SPPSO often loses track of the legs around frame 250 and also loses arms frequently. Figure 5 illustrates how SPPSO loses multiple limbs but reacquires them after some frames at 20 fps. The ability to recover from tracking failures is an important feature for pose tracking algorithms. As expected, the tracking is always very good during the period when the subject stands still (frame 330 to 430).

3.1 Comparison to the Annealed Particle Filter

SPPSO was compared to the annealed particle filter, which currently is the state-of-the-art algorithm for human pose tracking. It is also the benchmark algorithm of the HumanEva framework [14]. Figure 6 shows the results of an experiment with 1000 evaluations per frame at 60 fps and 20 fps, with the same body model and fitness function for both algorithms. Furthermore, both algorithms used the same parameter covariance Σ. SPPSO performs significantly better at 20 fps and equally well at a frame rate of 60 fps. The better performance of SPPSO at the lower frame rate is attributed to the direct exploitation of the hierarchical model. The APF on the other hand, relies on an automatic soft partitioning [4].

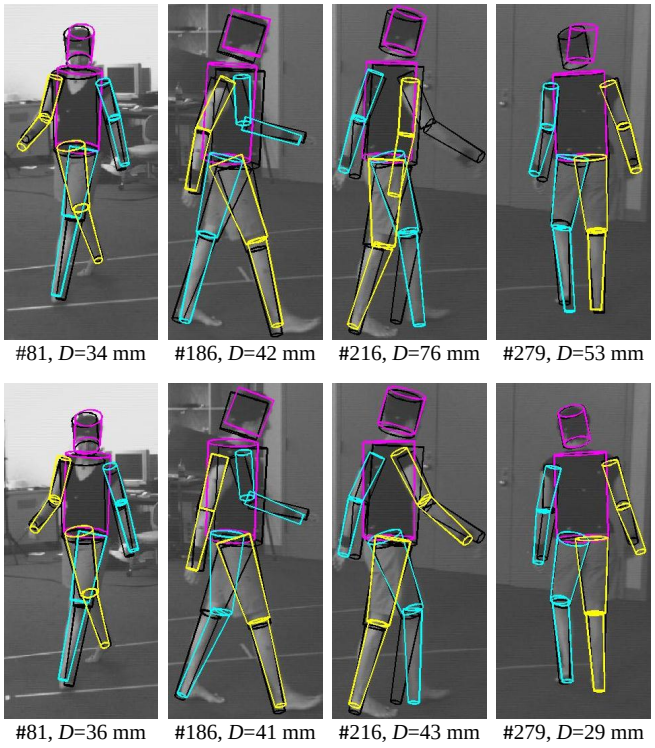


Fig. 4. (top) SPPSO tracking results with 1000 evaluations per frame at 20 fps. (bottom) Tracking with 1000 evaluations per frame at 60 fps. Ground truth cylinders are shown in black, estimated cylinders are coloured to distinguish left and right limbs. Results are shown at frames 81, 186, 216, and 279. D denotes the tracking error at the depicted frame

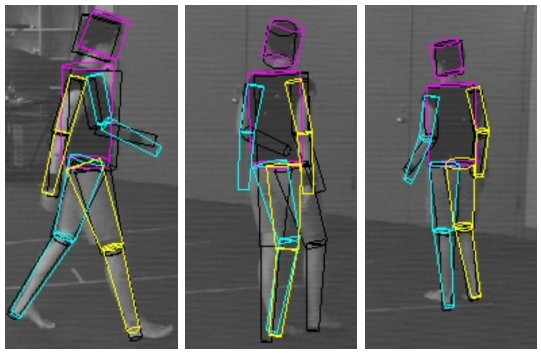
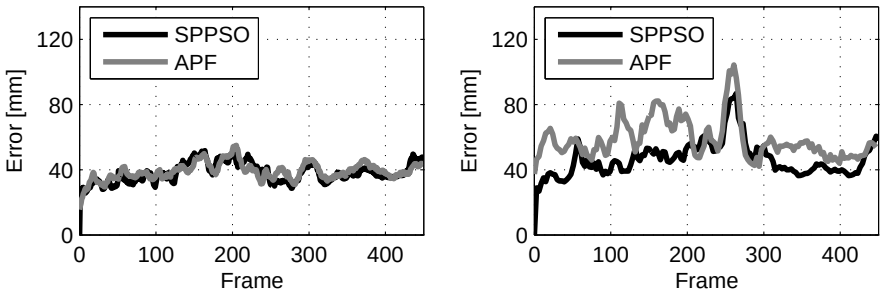


Fig. 5. SPPSO tracking results with 1000 evaluations per frame at 20 fps. The tracker temporarily loses the legs and one arm but can recover in later frames



(a) 60 fps (b) 20 fps

Fig. 6. Tracking error of SPPSO and APF with 1000 evaluatins per frame for the Lee walk sequence. The graphs show mean errors from five individual runs

3.2 Computation Time

SPPSO is based on the HumanEva framework [1, 14] and therefore almost completely implemented in Matlab, including the rendering of the 3D model. Consequently, the algorithm needs 20 seconds to process one frame (1000 evaluations) and is therefore far from being real-time, which would be necessary for many applications. The most processor-intensive tasks in SPPSO are model rendering and fitness evaluation, which account for 50% and 39% of the processing time. These tasks could be performed very fast by graphics processing hardware.

4 Summary and Conclusions

This paper proposes the SPPSO algorithm for human pose tracking, which has been evaluated on the publicly available Lee walk dataset. These experiments showed that SPPSO performs better than APF at a frame rate of 20 fps with the same number of fitness evaluations.

PSO is a relatively new optimization method for pose tracking (The first source known to the authors is [7]). And there exist only few, more or less successful, attempts to video-based *full body* tracking [8, 10]. It seems that the methods which use a hard partitioning require more fitness evaluations to minimize the problem of error accumulation.

All of the algorithms discussed in the introduction use different hierarchical approaches to overcome the problem of high dimensionality and none of them seems to be clearly superior. With the soft partitioning scheme, SPPSO proposes a novel approach for full body tracking and it has been shown to perform well. In Contrast to the similar approach by Robertson and Trucco [13], SPPSO uses fewer optimization stages for a body model with more parameters.

A further development of SPPSO could add a local refinement stage to the algorithm to achieve a better accuracy with less computational power. This approach has been proven successful for pose tracking by recent research [15, 6].

References

1. Balan, A., Sigal, L., Black, M.: A quantitative evaluation of video-based 3d person tracking. In: Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Jt. IEEE Int. Workshop. pp. 349–356 (2005)
2. Ballan, L., Cortelazzo, G.M.: Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In: 3DPVT (2008)
3. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. 6(1), 58–73 (2002)
4. Deutscher, J., Reid, I.: Articulated body motion capture by stochastic search. Int. J. Comput. Vision 61, 185–205 (2005)
5. Gall, J., Potthoff, J., Schnrr, C., Rosenhahn, B., Seidel, H.P.: Interacting and annealing particle filters: Mathematics and a recipe for applications. J. Math. Imaging Vision 28, 1–18 (2007)
6. Gall, J., Rosenhahn, B., Brox, T., Seidel, H.P.: Optimization and filtering for human motion capture. Int. J. Comput. Vision 87, 75–92 (2010)
7. Ivekovic, S., Trucco, E.: Human body pose estimation with pso. In: Evolutionary Computation, 2006. CEC 2006. IEEE Congr. pp. 1256–1263 (2006)
8. John, V., Trucco, E., Ivekovic, S.: Markerless human articulated tracking using hierarchical particle swarm optimisation. Image Vision Comput. 28(11), 1530–1547 (2010)
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Neural Networks, 1995. Proceedings., IEEE Int. Conf. vol. 4, pp. 1942–1948 (1995)
10. Krzeszowski, T., Kwolek, B., Wojciechowski, K.: Model-based 3d human motion capture using global-local particle swarm optimizations. In: Burduk, R., Kurzynski, M., Wozniak, M., Zolnierek, A. (eds.) Computer Recognition Systems 4, Advances in Intelligent and Soft Computing, vol. 95, pp. 297–306. Springer (2011)
11. Kwolek, B., Krzeszowski, T., Wojciechowski, K.: Swarm intelligence based searching schemes for articulated 3d body motion tracking. In: Blanc-Talon, J., Kleihorst, R., Philips, W., Popescu, D., Scheunders, P. (eds.) Advances Concepts for Intelligent Vision Systems, Lect. Notes Comput. Sci., vol. 6915, pp. 115–126. Springer (2011)
12. Moeslund, T., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Comput. Vision Image Understanding 104(2-3), 90–126 (2006)
13. Robertson, C., Trucco, E.: Human body posture via hierarchical evolutionary optimization. BMVC06 3, 999 (2006)
14. Sigal, L., Balan, A., Black, M.: Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. Int. J. Comput. Vision 87, 4–27 (2010)
15. Sminchisescu, C., Triggs, B.: Covariance scaled sampling for monocular 3d body tracking. In: Computer Vision and Pattern Recognition (CVPR), IEEE Computer Soc. Conf. vol. 1, pp. I–447 – I–454. IEEE (2001)