Markerless Multi-View Articulated Pose Estimation Using Adaptive Hierarchical Particle Swarm Optimisation

Spela Ivekovic, Vijay John and Emanuele Trucco

School of Computing, University of Dundee, Dundee DD1 4HN
{spelaivekovic,vijayjohn,manueltrucco}@computing.dundee.ac.uk

Abstract. In this paper, we present a new adaptive approach to multi-view markerless articulated human body pose estimation from multi-view video sequences, using Particle Swarm Optimisation (PSO). We address the computational complexity of the recently developed hierarchical PSO (HPSO) approach, which successfully estimated a wide range of different motion with a fixed set of parameters, but incurred an unnecessary overhead in computational complexity. Our adaptive approach, called APSO, preserves the black-box property of the HPSO in that it requires no parameter value input from the user. Instead, it adaptively changes the value of the search parameters online, depending on the quality of the pose estimate in the preceding frame of the sequence. We experimentally compare our adaptive approach with HPSO on four different video sequences and show that the computational complexity can be reduced without sacrificing accuracy and without requiring any user input or prior knowledge about the estimated motion type.

1 Introduction

Video-based markerless articulated pose estimation is an important problem in computer vision which has been given much attention recently [1,2]. Established commercial systems for accurate articulated pose estimation, e.g., Vicon [3], require subjects to wear tight Lycra suits and optical or magnetic markers, an expensive, intrusive and time-consuming solution [2]. Video-based markerless motion capture promises an unintrusive, cheaper and less time-consuming alternative for articulated motion capture.

If the markerless pose estimation is to become truly useful in practice, a black-box solution is necessary which won't require the user to have a detailed knowledge of its internal structure and parameter values. From the user's perspective, regardless of the type of articulated motion being estimated, the algorithm should accept a video sequence as the input and produce a high-quality articulated pose estimate as the output.

In this paper, we present such a black-box approach to articulated pose estimation from multi-view video sequences. We use a powerful global optimisation approach, called particle swarm optimisation (PSO), which has been shown to outperform other search methods (e.g., simulated annealing) on large, difficult and non-linear optimisation problems [4]. John *et al.* [5] use the PSO framework to formulate the articulated pose estimation as a hierarchical search in a constrained search space. Their approach, called HPSO, works as a black box in the sense that it generalises well to different types of motion with fixed PSO parameter settings. However, this ability comes at the price of unnecessarily large computational complexity (although still smaller than the competing techniques). In this paper, we present an adaptive extension of HPSO, called APSO, designed to reduce the search complexity of the HPSO approach without affecting its black-box nature.

This paper is organised as follows. We begin with a discussion of recent related work in Section 2. We describe the PSO algorithm in Section 3, followed by a description of the body model, PSO parametrisation and fitness function in Section 4. The HPSO approach is described in Section 5, and the adaptive extension, APSO, in Section 6. We quantitatively compare our approach with HPSO on several multi-view sequences in Section 7 and conclude in Section 8.

2 Related work

The literature on markerless human body motion capture is large; recent surveys are [1,2]. Here, we discuss the recent work with respect to the use of motion models and search algorithms.

Motion models. The key motivation behind using motion models is to reduce the dimensionality of an otherwise very expensive or unfeasible search problem. We can regard motion models for human motion tracking as instantaneous or global. *Instantaneous models* predict pose on a frame-by-frame basis; examples are Kalman filtering approaches [6] and particle filtering and variations [7,8,9]. *Global models* seek to describe whole actions (e.g., walking, sitting down) [10,11,12] to provide a context strongly constraining the next pose. The price is reduced generality, as tracking becomes specific to a dictionary of pre-determined actions. Recently, solutions have been proposed which make use of global optimisation to remove the dependency on the pre-trained motion models [5,13], which is also the research direction this paper pursues.

Search. Nearly invariably, pose estimation or tracking is cast as a search in a high-dimensional parameter space, so that an efficient optimiser is of paramount importance. In addition to statistical estimation of dynamic systems (e.g., Kalman and particle filtering), search algorithms reported include iterative closest point (ICP) and variations [14], constrained non-rigid factorization [15], Markov models [12] and gradient boosting [16]. In terms of evolutionary approaches, articulated pose estimation from video sequences has been reported with genetic algorithms [17,18], particle swarm optimisation [4,5,19], and indirectly from voxel data using evolutionary algorithms [20].

In this paper, we extend the pose estimation work recently reported by John *et al.* [5], primarily addressing the computational complexity of the black-box hierarchical PSO search proposed in [5].

3 Particle Swarm Optimisation

PSO is a swarm intelligence technique introduced by Kennedy and Eberhart [21]. The original PSO algorithm has since been modified by several researchers to improve its search capabilities and convergence properties. In this paper, we use the PSO algorithm with an inertia weight parameter, introduced by Shi and Eberhart [22].

3.1 PSO Algorithm with Inertia Weight Parameter

Assume an n-dimensional search space $\mathbb{S} \subseteq \mathbb{R}^n$, a swarm consisting of N particles, each particle representing a candidate solution to the search problem, and a fitness function $f : \mathbb{S} \rightarrow \mathbb{R}$ defined on the search space. The i-th particle is represented as an n-dimensional vector $\mathbf{x}^i = (x_1, x_2, ..., x_n)^T \in \mathbb{S}$. The velocity of this particle is also an n-dimensional vector $\mathbf{v}^i = (v_1, v_2, ..., v_n)^T \in \mathbb{S}$. The best position encountered by the i-th particle so far (*personal* best) is denoted by $\mathbf{p}^i = (p_1, p_2, ..., p_n)^T \in \mathbb{S}$ and the value of the fitness function at that position $pbest^i = f(\mathbf{p}^i)$. The index of the particle with the overall best position so far (*global* best) is denoted by g and $gbest = f(\mathbf{p}^g)$. The PSO algorithm with inertia weight can then be stated as follows:

1. **Initialisation**:
   - Initialise a population of particles $\{\mathbf{x}^i\}, i = 1 \ldots N$, with random positions and velocities in the search space $\mathbb{S}$. For each particle evaluate the desired fitness function f and set $pbest^i = f(\mathbf{x}^i)$. Identify the best particle in the swarm and store its index as g and its position as $\mathbf{p}^g$.
2. **Repeat** until stopping criterion (see below) is satisfied:
   - Move the swarm by updating the position of every particle $\mathbf{x}^i$ according to

$$\begin{aligned} \mathbf{v}^i_{t+1} &= w\mathbf{v}^i_t + \varphi_1(\mathbf{p}^i_t - \mathbf{x}^i_t) + \varphi_2(\mathbf{p}^g_t - \mathbf{x}^i_t) \\ \mathbf{x}^i_{t+1} &= \mathbf{x}^i_t + \mathbf{v}^i_{t+1} \end{aligned} \tag{1}$$

     where subscript t denotes the time step (iteration) and φ_1, φ_2 are defined below.
   - For $i = 1 \ldots N$ update $\mathbf{p}^i$, $pbest^i$, $\mathbf{p}^g$ and $gbest$.

The usual stopping criterion is either that the maximum number of iterations is reached or that the $gbest$ improvement in subsequent iterations becomes small enough. The parameter w is the *inertia weight*. The parameters $\varphi_1 = c_1 rand_1()$ and $\varphi_2 = c_2 rand_2()$, where c_1, c_2 are constant and $rand()$ is a random number drawn from $[0, 1]$, influence the *social* and *cognition* components of the swarm behaviour, respectively. In line with [21], we set $c_1 = c_2 = 2$, which gives the stochastic factor a mean of 1.0 and causes the particles to "overfly" the target about half of the time, while also giving equal importance to both social and cognition components.

The inertia weight We model the inertia change over time with an exponential function which allows us to use a constant sampling step α to gradually guide the swarm from a global to more local exploration:

$$w(\alpha) = \frac{A}{e^{\alpha}}, \quad \alpha \in [0, \ln(10A)], \tag{2}$$

where A denotes the starting value of w, when the sampling variable is $\alpha = 0$. The step α is incremented by a constant $\Delta\alpha = ln(10A)/C$, where C is the chosen number of inertia weight changes per search. The optimisation ends when $w(\alpha)$ falls below 0.1.

Table 1. Body model joints and their corresponding DOF. There are 31 DOF in total.

JOINT (index)	#	DOF	JOINT (index)	#	DOF
Global body position (1)	3	r_x, r_y, r_z	Right shoulder orientation (7)	3	$\alpha_x^7, \beta_y^7, \gamma_z^7$
Global body orientation (1)	3	$\alpha_x^1, \beta_y^1, \gamma_z^1$	Right elbow orientation (8)	1	β_y^8
Torso orientation (2)	2	β_y^2, γ_z^2	Head orientation (9)	3	$\alpha_x^9, \beta_y^9, \gamma_z^9$
Left clavicle orientation (3)	2	α_x^3, β_y^3	Left hip orientation (10)	3	$\alpha_x^{10}, \beta_y^{10}, \gamma_z^{10}$
Left shoulder orientation (4)	3	$\alpha_x^4, \beta_y^4, \gamma_z^4$	Left knee orientation (11)	1	β_y^{11}
Left elbow orientation (5)	1	β_y^5	Right hip orientation (12)	3	$\alpha_x^{12}, \beta_y^{12}, \gamma_z^{12}$
Right clavicle orientation (6)	2	α_x^6, β_y^6	Right knee orientation (13)	1	β_y^{13}

4 Body Model, PSO Parametrisation and Fitness Function

To enable performance comparison, we use the same body model and fitness function as the HPSO technique and originally proposed by Balan et al. [23]. In this section, we provide a short summary for completeness and refer the reader to [5,23] for details.

4.1 Body model

The human body shape is modelled as a collection of truncated cones (Figure 1(a)). The underlying articulated motion is modelled with a kinematic tree containing 13 nodes, each node corresponding to a specific body joint. For illustration, the indexed joints are shown overlaid on the test subject in Figure 1(b). Every node can have up to 3 rotational degrees of freedom (DOF), while the root node also has 3 translational DOF. In our model, we use a total of 31 DOF, detailed in Table 1.

4.2 PSO Parametrisation

The PSO particle position vector represents an articulated body pose and hence consists of 31 parameters corresponding to the 31 DOF in Table 1:

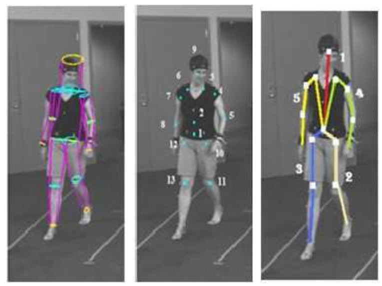
$$\mathbf{x}^i = (r_x, r_y, r_z, \alpha_x^1, \beta_y^1, \gamma_z^1, \ldots, \beta_y^{13}). \tag{3}$$

4.3 Fitness function

The fitness function $f(\mathbf{x}^i)$ measures how well a candidate pose $\mathbf{x}^i$ matches the pose of the person in the sequence. It consists of two parts, an edge-based part and a silhouette-based part:

$$f(\mathbf{x}^i) = MSE_{edge}(\mathbf{x}^i) + MSE_{silhouette}(\mathbf{x}^i), \tag{4}$$

where MSE denotes the mean-square error. The edge-based part penalises the distance between the projections of truncated cone edges and the edges in the edge maps obtained from the input images. In the silhouette-based part, a predefined number of points on the surface of the articulated model is projected into the silhouette images and the overlap estimated.



(a) (b) (c)

Fig. 1. (a) The truncated-cone body model. (b) Joint positions. (c) Kinematic tree.

5 The Hierarchical PSO

The HPSO algorithm by John *et al.* [5] splits the 31-dimensional search space into 12 disjoint subspaces which are searched in a pre-defined hierarchical sequence dictated by the hierarchical structure of the kinematic tree representing the human body motion. The subspaces are defined in such a way that only one limb segment at a time is being optimised (see Table 2). Formulating the search in this way significantly reduces its complexity.

Table 2. 12 hierarchical steps of HPSO (*Cf.* Table 1.)

(Step 1)	Body position 3 DOF: r_x, r_y, r_z	**(Step 5)**	Left lower arm 2 DOF: γ_z^4, β_y^5	**(Step 9)**	Left upper leg 2 DOF: $\alpha_x^{10}, \beta_y^{10}$
(Step 2)	Body orientation 3 DOF: $\alpha_x^1, \beta_y^1, \gamma_z^1$	**(Step 6)**	Right upper arm 4 DOF: $\alpha_x^6, \beta_y^6, \alpha_x^7, \beta_y^7$	**(Step 10)**	Left lower leg 2 DOF: $\gamma_z^{10}, \beta_y^{11}$
(Step 3)	Torso 2 DOF: β_y^2, γ_z^2	**(Step 7)**	Right lower arm 2 DOF: γ_z^7, β_y^8	**(Step 11)**	Right upper leg 2 DOF: $\alpha_x^{12}, \beta_y^{12}$
(Step 4)	Left upper arm 4 DOF: $\alpha_x^3, \beta_y^3, \alpha_x^4, \beta_y^4$	**(Step 8)**	Head 3 DOF: $\alpha_x^9, \beta_y^9, \gamma_z^9$	**(Step 12)**	Right lower leg 2 DOF: $\gamma_z^{12}, \beta_y^{13}$

The HPSO algorithm is designed to be used as a black box, that is, the user is not required to tweak the search parameters in order to customise the search for a particular type of motion. Instead, the parameter values are set in a way that guarantees that a very wide range of motion, for example, walk, jog, kick, jump, etc. can be estimated without requiring any adjustments. The range of motion that can be successfully estimated depends on the value of the inertia parameter - the higher the starting inertia value (A in Equation (2)), the more agile the motion can be.

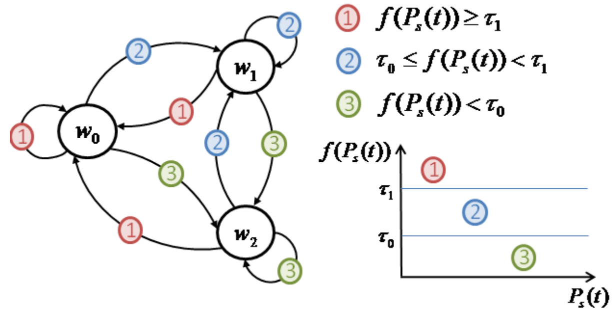


Fig. 2. Adaptive inertia state transition diagram for step s in the hierarchy. At the end of the search, the best pose estimate $P_s(t)$ is evaluated against two fitness function thresholds, τ_0 and τ_1. The higher the $f(P_s(t))$, the better the pose estimate and the smaller the starting inertia for this hierarchical step in the next frame.

6 The Adaptive PSO

When the range of motion we want to estimate with the same parameter settings is very wide, for example, from a simple slow walk to a fast karate kick, the easy solution is to set the starting inertia value A high enough to guarantee that the exploration (rather than exploitation) is given sufficient priority and therefore the fastest motion will be estimated reliably. While the high inertia value is indeed necessary for sequences with fast and sudden motion, it is excessive in sequences where the subject is only walking. In such slow sequences, the high starting inertia value introduces an unnecessary computational overhead. To address this inconsistency, we formulate an adaptive extension of the HPSO approach, the APSO, where the starting inertia value A is adjusted on a frame-by-frame basis.

6.1 APSO Algorithm

In order to adjust the value of A automatically, online and without user interference, the adjustment process must exploit the information about the search performance in the preceding frame. The APSO approach therefore adaptively changes the next-frame *starting* inertia value for *every hierarchical step* in Table 2 by making use of two quality thresholds, τ_0 and τ_1: when the pose estimate $P_s(t)$ for a hierarchical step s in the current frame is evaluated as good, $f(P_s(t)) \geq \tau_1$, where f is the fitness function, the search region in the next frame is kept small ($A_s^{t+1} = w_0$) as the search is thought to be on target; when the pose estimate is very bad, $f(P_s(t)) < \tau_0$, the search is losing the target and hence the search region in the next frame is expanded significantly ($A_s^{t+1} = w_2$). When the estimate is average, $\tau_0 \leq f(P_s(t)) < \tau_1$, the search region is expanded moderately ($A_s^{t+1} = w_1$), where $w_0 < w_1 < w_2$. The process of adaptively changing the inertia value is illustrated with a state transition diagram in Figure 2.

The adaptive inertia scheme is also used to correct for the effects of an occasional bad starting inertia proposal. For example, in a case where the search is on target in frame t, $f(P_s(t)) \geq \tau_1$, and hence w_0 is proposed for the search in frame $t+1$, but from frame t to frame $t+1$ a sudden, large motion occurs, for which w_0 is not sufficient. We deal with this case as follows. After the search for a particular hierarchical step has been completed, we check the quality of the final pose estimate. If the estimate is bad or average, $f(P_s(t)) < \tau_1$ and the proposed starting inertia value that was used was not the highest inertia value available, $A_s^t = w_i, i < 2$, then the starting inertia value is increased to the next higher value, $A_s^t = w_{i+1}$ and the search for this hierarchical step is repeated. The process repeats until *either* the highest inertia value has been reached, $A_s^t = w_2$, *or* the pose estimate is sufficiently good, $f(P_s(t)) \geq \tau_1$. The value A_s^{t+1} for the next frame is then determined the same way as described in the previous paragraph and illustrated in Figure 2.

The rationale behind the use of this adaptive scheme is in the observation that even fast and sudden actions like, for example, karate kick, consist of segments with slow, medium and fast motion, and therefore searching with the highest inertia value in every frame would be excessive. The adaptive scheme favours a smaller inertia weight and as the experimental results in Section 7 demonstrate, this is not a bad assumption; the search time indeed decreases in comparison with HPSO without sacrificing the accuracy of the estimates. In fact, given the stochastic nature of the PSO, in our limited experimental evaluation the accuracy actually slightly increases owing to the search repeat strategy which corrects for bad starting values. Making the starting inertia value dependent on the quality of the pose estimate very effectively prevents the search from losing the target and ensures that even very erratic and sudden motion can be followed without diverging.

6.2 Setting τ_0 and τ_1

As a first attempt, we determined the values for τ_0 and τ_1 from a video sequence accompanied with ground truth optical motion capture data. The ground truth poses were used to evaluate the fitness function over a 200-frame sequence and the highest and lowest value of the fitness function were recorded. The interval between the highest and lowest value was then split into three identical bands and the boundaries of the middle band were used as τ_0 and τ_1. As we show with the experimental results, specifying τ_0 and τ_1 in this way does improve the efficiency of the pose estimation, however, we must stress that this is by no means the final solution. Further research is necessary to find a principled way of setting these thresholds which will allow an optimal choice of the search region for every frame of the sequence.

7 Experiments

In this section we compare the performance of the proposed APSO algorithm with that of HPSO.

Datasets. In our experiments, we used 4 datasets: the *Lee walk* sequence included in the Brown University evaluation software [23] and 3 datasets courtesy of the University of Surrey: *Jon walk*, *Tony kick* and *Tony punch*. The *Lee walk* dataset was captured with 4 synchronised grayscale cameras with resolution 640×480 at 60 fps and came with the

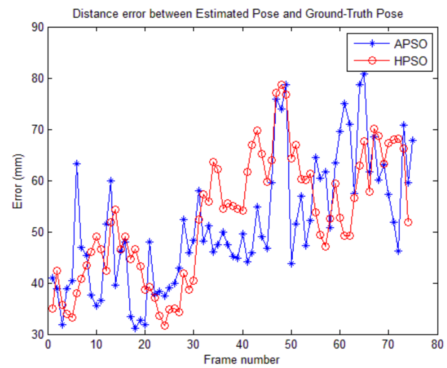


Fig. 3. The error graph on the Lee Walk 30fps sequence.

Table 3. Lee Walk sequence: the mean and standard deviation of the distance from the ground truth

Sequence	HPSO ($\mu \pm \sigma$)	Avg time (5 trials)	APSO ($\mu \pm \sigma$)	Avg time (5 trials)
Lee Walk 30Hz	52.5±11.7mm	1 hr,35min	50.8±10.4mm	1 hr, 5min

ground truth articulated motion data acquired by a Vicon system. The Surrey sequences were acquired by 10 synchronised colour cameras with resolution 720×576 at 25 fps. The test sequences were chosen to cover a range of different body motions.

HPSO and APSO setup. In [5] HPSO was run with only 10 particles; the starting inertia weight was set to $A = 2$, guaranteeing that the particles visit the entire search space, and the stopping inertia was fixed at $w = 0.1$ for all sequences. This amounted to 60 PSO iterations per hierarchical step in HPSO or 7200 fitness function evaluations per frame. In order to enable a meaningful comparison, APSO was also run with only 10 particles, the starting inertia values were set to $w_0 = 0.5$, $w_1 = 1.2$ and $w_2 = 2.0$, the stopping inertia was fixed at 0.1 and pose estimate accuracy thresholds τ_0, τ_1 were derived from the ground-truth pose estimates of the *Lee walk* sequence for every hierarchical step.

Table 4. The silhouette/edge overlap measure for the Surrey sequence. Bigger number means better performance.

Sequence	HPSO Mean ± Std.dev	Avg time (5 trials)	APSO Mean ± Std.dev	Avg time (5 trials)
Jon Walk	1.38±0.01	2hr,30min	1.39±0.01	2hr,15min
Tony Kick	1.29±0.02	1hr,30min	1.31±0.01	1hr,15min
Tony Punch	1.32±0.01	1hr,30min	1.34±0.01	1hr,15min

7.1 Comparison of APSO vs HPSO

Lee Walk Results. We tested on a downsampled frame rate of 30 fps instead of the original 60 fps. The results are shown in Table 3 and Figure 3 and indicate that APSO uses less time while also maintaining the average accuracy of the estimation. Table 3 shows the error calculated as the distance between the ground-truth joint values and the values from the pose estimated in each frame. As the algorithm is stochastic in nature, the results shown are averaged over 5 trials. A larger number of trials would provide a better insight, however due to the computational complexity of the algorithm, running a significantly larger number of trials was not practical as HPSO took 70 sec per frame, while APSO varied between 40 sec and 100 sec per frame.
Surrey Results. The Surrey test sequences contain faster motion than the Lee walk sequence. Again, our results for all tested sequences show that APSO reduces the tracking time. The average overlap and standard deviation for the Surrey sequence over 5 trials are shown in Table 4.
Recovery. John *et al.* [5] remark that HPSO demonstrates the ability to recover from wrong estimates due to error propagation in the hierarchical search within a few frames. APSO, with its search-restart strategy, in fact encounters the error propagation problem a lot less frequently, as any potentially stray estimates are immediately corrected. That explains the slight improvement in APSO estimation accuracy when compared to HPSO.

8 Discussion

Markerless pose estimation from multi-view video sequences is an important problem in computer vision. For any solution to become a useful motion capture tool in practical applications, algorithms are required which can take the burden of parameter tuning and deep algorithmic knowledge away from the intended end-user and work well on a variety of input sequences. A step in the direction of developing such a black-box pose estimation approach was made recently by John *et al.* [5], however, in guaranteeing the black-box property, the HPSO algorithm incurred an unnecessary overhead in computational complexity. In this paper, we presented an adaptive extension of the HPSO approach which reduces the computational complexity of the search. We experimentally demonstrated that the proposed algorithm is computationally more efficient and can still be used on a range of different motions without requiring any parameter tuning by the user.

The adaptive scheme presented in this paper relies only on the information about the quality of the pose estimate in the preceding frame. It does not take into account the information about the speed of the estimated motion which can be extracted from the preceding estimates online, during search. A k-th order autoregressive model trained online to predict the required inertia value based on the estimates in the previous k frames would further improve the efficiency of the search as it would reduce the number of search-restart events which happen due to bad predictions. The choice of inertia values w_0, w_1, w_2 can also be made more principled on the basis of recent work by Poli [24]. We will investigate these improvements in future work.

References

1. Moeslund, T.B., Hilton, A., Krueger, V.: A survey of advances in vision-based human motion capture and analysis. CVIU **104**(2-3) (2006) 90–126 1, 2
2. Poppe, R.: Vision-based human motion analysis. CVIU **108**(1-2) (2007) 4–18 1, 2
3. Vicon: Motion capture systems. www.vicon.com (November 2008) 1
4. Ivekovic, S., Trucco, E., Petillot, Y.: Human body pose estimation with particle swarm optimisation. Evolutionary Computation **16**(4) (2008) 1, 2
5. John, V., Ivekovic, S., Trucco, E.: Articulated human tracking using HPSO. In: Proceedings of International Conference on Computer Vision Theory and Applications (VISAPP). Volume 1. (2009) 531–538 1, 2, 4, 5, 8, 9
6. Mikic, I., Trivedi, M., Hunter, E., Cosman, P.: Human body model acquisition and tracking using voxel data. IJCV **53**(3) (2003) 199–223 2
7. Balan, A.O., Sigal, L., Black, M.J., Davis, J.E., Haussecker, H.W.: Detailed human shape and pose from images. In: CVPR'07. (2007) 2
8. Deutscher, J., Reid, I.: Articulated body motion capture by stochastic search. IJCV **61**(2) (2005) 2
9. Wang, P., Rehg, J.M.: A modular approach to the analysis and evaluation of particle filters for figure tracking. In: CVPR'06. Volume 1. (2006) 790–797 2
10. Caillette, F., Galata, A., Howard, T.: Real-time 3-d human body tracking using learnt models of behaviour. CVIU **109**(2) (2008) 112–125 2
11. Rosenhahn, B., Brox, T., Seidel, H.P.: Scaled motion dynamics for markerless motion capture. In: CVPR. (2007) 2
12. Peursum, P., Venkatesh, S., West, G.: Tracking-as-recognition for articulated full-body human motion analysis. In: CVPR'07. (2007) 2
13. Gall, J., Rosenhahn, B., Brox, T., Seidel, H.P.: Optimization and filtering for human motion capture - a multi-layer framework. IJCV (online first) (2008) 2
14. Muendermann, L., Corazza, S., Andriacchi, T.P.: Accurately measuring human movement using articulated icp with soft-joint constraints and a repository of articulated models. In: CVPR'07. (2007) 2
15. Shaji, A., Siddiquie, B., Chandran, S., Suter, D.: Human pose extraction from monocular videos using constrained non-rigid factorization. In: BMVC. (2008) 2
16. Bissacco, A., Yang, M.H., Soatto, S.: Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In: CVPR'07. (2007) 2
17. Ohya, J., Kishino, F.: Human posture estimation from multiple images using genetic algorithm. In: ICPR. Volume 1. (1994) 750–753 2
18. Zhao, X., Liu, Y.: Generative tracking of 3d human motion by hierarchical annealed genetic algorithm. Pattern Recognition **41**(8) (2008) 2470–2483 2
19. Schutte, J.F., Reinbolt, J.A., Fregly, B.J., Haftka, R.T., George, A.D.: Parallel global optimization with the particle swarm algorithm. International Journal for Numerical Methods in Engineering **61(13)** (2004) 2
20. Shen, S., Deng, H., Liu, Y.: Probability evolutionary algorithm based human motion tracking using voxel data. In: Proceedings of IEEE CEC 2008. (2008) 44–49 2
21. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE ICNN. Volume 4. (1995) 1942–1948 2, 3
22. Shi, Y.H., Eberhart, R.C.: A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation. (1998) 69 – 73 3
23. Balan, A.O., Sigal, L., Black, M.J.: A quantitative evaluation of video-based 3d person tracking. In: ICCCN '05. (2005) 349–356 4, 7
24. Poli, R.: Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. IEEE Transactions on Evolutionary Computation **13**(3) (2009) 1–10 9