Real-Time Multi-view Human Motion Tracking Using Particle Swarm Optimization with Resampling

Bogdan Kwolek[3,2], Tomasz Krzeszowski[3,2], André Gagalowicz[1], Konrad Wojciechowski[2], and Henryk Josinski[2]

[1] INRIA Paris-Rocquencourt, Rocquencourt 78153, France
[2] Polish-Japanese Institute of Information Technology Koszykowa 86, 02-008 Warszawa, Poland,
[3] Rzeszów University of Technology 35-959 Rzeszów, Poland, bkwolek@prz.edu.pl

Abstract. In this paper we propose a particle swarm optimization with resampling for marker-less body tracking. The resampling is employed to select a record of the best particles according to the weights of particles making up the swarm. The algorithm better copes with noise and reduces the premature stagnation. Experiments on 4-camera datasets show the robustness and accuracy of our method. It was evaluated on nine sequences using ground truth provided by Vicon. The full body motion tracking was conducted in real-time on two PC nodes, each of them with two multi-core CPUs with hyper-threading, connected by 1 GigE.

1 Introduction

In 1995, Dyer et al. [6] in Motion Capture White Paper, which was issued by SGI wrote that "motion capture is one of the hottest topics in computer graphics today". As stated in the mentioned paper, motion capture involves measuring an object's position and orientation in physical space, then recording that information in a computer-usable form. Since then it has been published many survey papers, for example [12][13], and the number of papers grows exponentially.

In the last years many motion capture (also known as MoCap) systems have been developed for gaming and animation [14], sport [1], rehabilitation treatment and medical applications [15][18]. In optical MoCap systems either passive reflective markers or active markers are attached to a performer. The locations of the markers on the suit are designed such that the required body parts (e.g. joints) are covered, and a system of fixed cameras records the position of such markers. In general, existing commercial MoCap systems are only suitable for a well controlled indoor environment. In contrast, marker-less human motion capture consists in capturing human motion without any markers, by operating on image sequences only. The recovery of human body movements from image data without using markers is a very challenging task. The major difficulties are due to large number of degrees of freedom (DOFs) of the human body pose

that needs to be recovered, large variability in human appearance, noisy image observations, self-occlusion, and complex human motions.

Most previous work on human motion tracking has focused on the use of 3D articulated models of various complexity to recover the position, orientation and joint angles. Such models can be perceived as a kinematic chain, where at least eleven elements correspond to the major body parts. Usually, the 3D geometric model is constructed from truncated cones or cylinders and is used to generate contours, which can be compared with edge contours. A lot of hypothetical body poses is generated, which are projected into image plane to find a configuration of the 3D model, whose projection best fits the image observations.

Particle filters [7] are one of the most popular techniques for body tracking. However, given the high-dimensionality of the models to be tracked, the number of required particles to properly populate the space of possible solutions makes the pose tracking computationally very expensive. Deutscher and Reid [5] developed an annealed particle filter (APF), which adopts an annealing scheme together with the stochastic sampling to achieve better concentration of the particle spread close to the extremum. Additionally, a crossover operation is utilized to achieve improved particle's diversity. Recently, particle swarm optimization (PSO) [9] has been successfully applied to body motion tracking [17]. In PSO each particle follows simple position and velocity update equations. Thanks to interaction between particles a collective behavior arises. It leads to the emergence of global and collective search capabilities, which allow the particles to gravitate towards the global extremum. Human motion tracking can be achieved by a sequence of static PSO-based optimizations, followed by re-diversification of the particles to cover the possible poses in the next time step.

In recent work, John et al. propose a PSO-based hierarchical approach for full body pose tracking [8]. However, the discussed algorithm can have difficulties in escaping from local maxima determined in preceding hierarchical levels. In [10] a parallel PSO algorithm for full body tracking in multi-view image sequences has been proposed. However, the tracking has been done using an ordinary PSO, which has been parallelized and then executed on several multi-core CPUs. In more recent work [11], a PSO algorithm with a pool of best particles has been proposed to achieve better tracking. The pool of candidate best solutions has been obtained through smoothing the objective functions in an annealing scheme and then quantizing them. The better performance has been achieved owing to the ability of the algorithm to deal with observation ambiguities and noise.

Resampling is perhaps the most obvious and simple approach to deal with noise. It is one of the techniques to improve the performance of evolutionary algorithms (EAs) in noisy environment [4]. Motivated by the work mentioned above we elaborated a particle swarm optimization algorithm with resampling to achieve full body tracking in real-time. In contrast to [4], which handles multi-objective problems through mutation operator whose range of action varies over time, our algorithm relies on resampling. During tracking a repository of best particles is selected according to importance weights. One of the contributions of this paper is a parallel particle swarm algorithm with resampling, which allows us to track the full body in real-time using multi-view images.

2 Annealed PSO with Resampling

The ordinary PSO algorithm consists of particles representing candidate solutions. Particles move through the solution space, and undergo evaluation according to some fitness function $f()$. The movements of the particles are guided by their own finest known locations in the search-space as well as the entire swarm's best location. While the swarm as a whole gravitates towards the global extremum, the individual particles are capable of ignoring many local optima. The object tracking can be realized by a sequence of static PSO-based optimizations, followed by re-diversification of the particles to cover the possible poses in the next time step. The re-diversification of the particle i can be realized on the basis of normal distribution concentrated around the state estimate determined in the previous frame.

In the ordinary PSO, the convergence of particles towards its attractors is not guaranteed. In order to ensure convergence and to fine-tune the search, Clerc and Kennedy [3] employed a constriction factor ω in the following form of the formula expressing the i-th particle's velocity:

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

where constants c_1 and c_2 are responsible for balancing the influence of the individual's knowledge and that of the group, respectively, r_1 and r_2 stand for uniformly distributed random numbers, x^i denotes position of the i-th particle, p^i is the local best position of particle, whereas g is the global best position.

In our approach the value of ω depends on annealing factor α as follows:

$$\omega = -0.8\alpha + 1.4 \tag{2}$$

where $\alpha = 0.1 + \frac{k}{K+1}$, $k = 0, 1, \ldots, K$, and K is the number of iterations. The annealing factor is also used to smooth the objective function. The larger the iteration number is, the smaller is the smoothing. In consequence, in the last iteration the algorithm utilizes the non-smoothed function. The algorithm termed as annealed PSO with resampling (RAPSO) can be expressed as follows:

1. For each particle i
2. initialize $v_t^{i,0}$
3. $x_t^{i,0} \sim \mathcal{N}(g_{t-1}, \Sigma_0)$
4. $p_t^i = x_t^{i,0}$, $f_t^i = f(x_t^{i,0})$
5. $u_t^i = f_t^i$, $\tilde{u}_t^i = (u_t^i)^{\alpha_0}$
6. $i^* = \arg\min_i \tilde{u}_t^i$, $g_t = p_t^{i^*}$, $y_t = u_t^{i^*}$
7. For $k = 0, 1, \ldots, K$
8. update ω_α on the basis of (2)
9. For all particles, linearly scale $\tilde{u}_t^i$
10. For each particle, on the basis of $\tilde{u}_t^i$ compute weight w_t^i
11. Normalize weights w_t^i
12. Draw particles according to importance weights w_t^i and insert them to G
13. For each particle i

14. Select a particle from $\{G \cup g_t\}$ and assign it to g_t^i
15. $v_t^{i,k+1} = \omega_\alpha[v_t^{i,k} + c_1 r_1(p_t^i - x_t^{i,k}) + c_2 r_2(g_t^i - x_t^{i,k})]$
16. $x_t^{i,k+1} = x_t^{i,k} + v_t^{i,k+1}$
17. $f_t^i = f(x_t^{i,k+1})$
18. if $f_t^i < u_t^i$ then $p_t^i = x_t^{i,k+1}$, $u_t^i = f_t^i$, $\tilde{u}_t^i = (u_t^i)^{\alpha_k}$
19. if $f_t^i < y_t$ then $g_t = x_t^{i,k+1}$, $y_t = f_t^i$

The initialization of the algorithm takes place at the beginning of each frame, see lines 1-6 of the pseudo-code. Given the location g_{t-1} of the best particle in time $t-1$, for each particle i the algorithm determines initial location x_t^i and initial velocity v_t^i. Afterwards, best particle locations p_t^i and corresponding fitness values u_t^i are determined. For such fitness values the smoothed values $\tilde{u}_t^i$, the best location g_t and the corresponding best fitness value y_t are calculated. At the beginning of each iteration k, the algorithm updates ω_α and linearly scales $\tilde{u}_t^i$ to range $[0, 1]$. It then calculates the normalized weights w_t^i. Finally, a record of best particles G is selected according to importance weights w_t^i, see 12th line in the pseudo-code. Each particle selects best particle form such a record, determines its own velocity and location, see lines 14-16 in the pseudo-code. Finally, best locations with the corresponding fitness values are updated.

The fitness score is calculated on the basis of following expression: $f(x) = 1 - (f_1(x)^{w_1} \cdot f_2(x)^{w_2})$, where w denotes weighting coefficients that were determined experimentally. The function $f_1(x)$ reflects the degree of overlap between the extracted body and the projected model's into 2D image. The function $f_2(x)$ reflects the edge distance-based fitness.

The calculation of the objective function is the most consuming operation. Moreover, in multi-view tracking the 3D model is projected and then rendered in each camera's view. Therefore, in our approach the objective function is calculated by OpenMP threads [2], which communicate via the shared memory. The PSO thread has access to the shared memory with the objective function values, which were determined by the local threads as well as the values of the objective functions that were calculated by the cooperating swarm on another cores or computational nodes. We employ asynchronous exchange of the best particle location and its fitness score. In particular, if a sub-swarm, which as the first one finished object tracking in a given frame, it carries out the re-diversification of the particles using its current global best particle, without waiting for the global best optimum determined by the participating sub-swarms. It is worth mentioning that in such circumstances the estimate of the object state is determined using the available global best locations of cooperating sub-swarms.

3 Experimental Results

The proposed algorithm was evaluated on several image sequences, which were acquired by four synchronized and calibrated cameras. Each pair of the cameras is approximately perpendicular to the other camera pair. The cameras acquire color images of size 1920×1080 at 25 fps. In experiments we employed images

sub-sampled by 2 in both the x and y directions. A commercial motion capture (MoCap) system from Vicon Nexus was employed to provide the ground truth data. The system uses reflective markers and sixteen cameras to recover the 3D location of such markers. The system is capable of differentiating overlapping markers from each camera's view. The data are delivered with rate of 100 Hz and the synchronization between the MoCap and multi-camera system is achieved using hardware from Vicon Giganet Lab. The location of the cameras and layout of laboratory is depicted in Fig. 1.

Fig. 1: Layout of the laboratory with four cameras. The images illustrate the initial model configuration, overlaid on the image in first frame and seen in view 1, 2, 3 and 4.

Our algorithm was tested on a variety of sequences with walking humans, observed from different viewpoints. To provide quantitative evaluation, the pose of walking subject was estimated by our algorithm. On the basis of the pose estimates, the configuration of the 3D model was determined. The model was then overlaid on the images. In each image sequence the same actor performed two walks, consisting in following a virtual line joining two opposite cameras and following a virtual line joining two nonconsecutive laboratory corners. The first subsequence is referred as 'straight', whereas the second one is called 'diagonal'. Figure 2 depicts some results that were obtained in a sequence #1 with a person

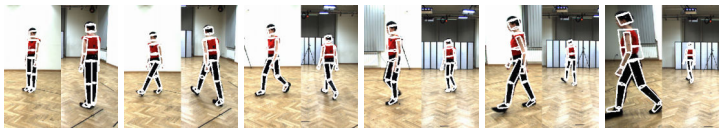

Fig. 2: Articulated 3D human body tracking in four camera setup. Shown are results in frames #0, 20, 40, 60, 80, 100. The left sub-images are seen from view 1, whereas the right ones are seen from view 2.

following a line joining two nonconsecutive laboratory corners. The degree of overlap of the projected 3D model on the performer illustrates the accuracy of the tracking. Focusing on tracking of the torso and the legs, we estimated also the head's pose as well as the pose of both arms. The configuration of the body is parameterized by the position and the orientation of the pelvis in the global coordinate system and the angles between the connected limbs. The human body model consists of eleven rigid body parts, each of which is represented by a

truncated cone. The model represents pelvis, upper torso, two upper arms, two lower arms, two thighs, two lower legs and the head, see also Fig. 2.

In Fig. 3 are depicted some results that were obtained in the same image sequence, but with the performer following a virtual line between two opposite cameras. The figure depicts model overlaid on the image from right profile view and the frontal view. The discussed results were obtained by RAPSO algorithm in 20 iterations per frame, consisting of 300 particles.

Fig. 3: Articulated 3D human body tracking in four camera setup using frontal and side views. Shown are results in frames #0, 20, 40, 60, 80, 100, 120. Left sub-images are seen from view 1, whereas the right ones are seen from view 2.

The plots in Fig. 4 illustrate the accuracy of motion estimation for some joints. As we can observe the tracking error of both forearms is something smaller in comparison to the remaining limbs. The average error of both knees is about 50 mm, whereas the maximal errors do not exceed 100 mm for the left knee and slightly exceeds 100 mm for the right one. The discussed results were obtained by RAPSO algorithm in 20 iterations and using 300 particles.

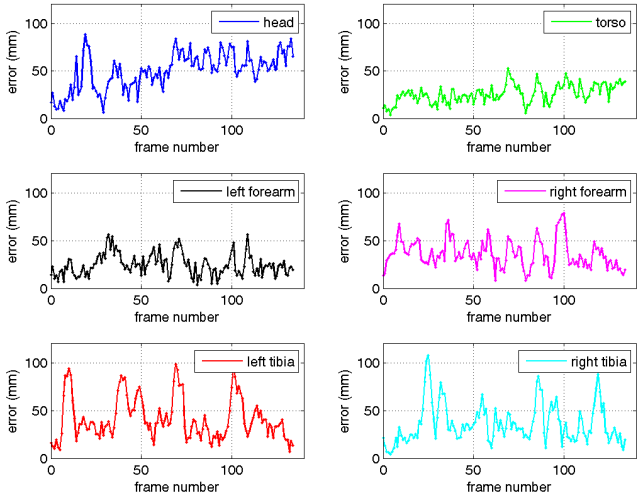

Fig. 4: Tracking errors [mm] versus frame number.

Figure 5 depicts the distance between ankles, which was registered in sequences P2, P3, and P4. In the sequence P4 the performer walked in the direction to the camera, whereas in sequence P2 and P3 the person moved diagonally. The sequences P2 and P4 depict the best results of the diagonal and the straight walks, respectively, whereas the sequence P3 depicts the poorest result of the diagonal walks. On the basis of the motion estimates the gait cycle is can be detected with good precision.

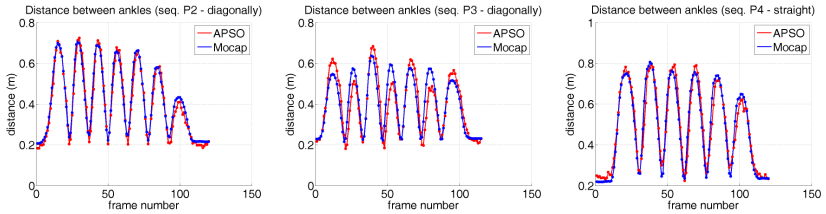

Fig. 5: Distance between ankles during walking in sequences P2 (diagonal), P3 (diagonal) and P4 (straight).

In Tab. 1 are presented some quantitative results that were obtained on four image sequences. The errors were calculated using 39 markers. For each frame they were computed as average Euclidean distance between individual markers and the recovered 3D joint locations [16]. For each sequence they were then averaged over ten runs with unlike initializations.

Table 1: Average errors for $M = 39$ markers in four image sequences.

			Seq. 1 (P1)	Seq. 2 (P1)	Seq. 3 (P6)	Seq. 4 (P6)
	#particles	it.	error [mm]	error [mm]	error [mm]	error [mm]
PSO	100	10	50.5 ± 28.7	56.7 ± 33.0	56.6 ± 37.1	54.4 ± 28.5
	100	20	45.0 ± 23.6	49.6 ± 27.2	46.2 ± 25.9	47.3 ± 23.8
	300	10	45.2 ± 25.5	48.1 ± 25.0	47.6 ± 26.4	48.7 ± 24.4
	300	20	41.9 ± 21.2	45.4 ± 24.4	41.2 ± 22.3	43.3 ± 21.8
APSO	100	10	$\mathbf{44.7} \pm 23.8$	51.1 ± 27.6	$\mathbf{46.7} \pm 26.6$	$\mathbf{45.7} \pm 23.8$
	100	20	$\mathbf{39.5} \pm 20.2$	46.0 ± 26.3	$\mathbf{40.3} \pm 22.8$	40.8 ± 20.1
	300	10	$\mathbf{39.6} \pm 19.7$	$\mathbf{44.7} \pm 23.8$	$\mathbf{40.6} \pm 22.9$	$\mathbf{39.6} \pm 19.3$
	300	20	37.0 ± 18.6	$\mathbf{40.4} \pm 19.9$	$\mathbf{35.8} \pm 19.6$	36.3 ± 16.6
RAPSO	100	10	46.9 ± 27.9	$\mathbf{50.2} \pm 27.2$	48.8 ± 28.0	50.9 ± 26.7
	100	20	40.0 ± 20.7	$\mathbf{45.6} \pm 23.1$	41.3 ± 23.7	$\mathbf{40.1} \pm 19.6$
	300	10	40.6 ± 20.5	45.0 ± 25.2	43.0 ± 25.0	40.5 ± 19.7
	300	20	$\mathbf{36.4} \pm 17.4$	40.4 ± 20.5	36.2 ± 20.1	$\mathbf{35.3} \pm 15.7$

In Tab. 2 are results illustrating the tracking accuracy of APSO and RAPSO algorithms. For each sequence the bold indicates the best results for the diagonal and the straight walks. As we can observe, both algorithms allow full body tracking with similar accuracy.

Table 2: Average errors [mm] for $M = 39$ markers using 300 particles, in 20 iterations.

	APSO		RAPSO	
Person	Straight	Diagonally	Straight	Diagonally
P1	37.0 ± 18.6	$\mathbf{40.4} \pm 19.9$	$\mathbf{36.4} \pm 17.4$	40.4 ± 20.5
P2	45.5 ± 25.8	$\mathbf{56.5} \pm 38.2$	$\mathbf{44.6} \pm 24.7$	56.9 ± 38.9
P3	43.4 ± 23.0	$\mathbf{45.8} \pm 17.7$	$\mathbf{42.7} \pm 20.8$	46.1 ± 19.0
P4	$\mathbf{43.2} \pm 21.3$	43.1 ± 19.3	45.0 ± 23.3	$\mathbf{41.9} \pm 18.3$
P5	$\mathbf{54.1} \pm 19.3$	$\mathbf{52.6} \pm 16.6$	54.6 ± 20.6	53.5 ± 16.7
P6	$\mathbf{35.8} \pm 19.6$	36.3 ± 16.6	36.2 ± 20.1	$\mathbf{35.3} \pm 15.7$
P7	50.0 ± 27.3	48.6 ± 20.5	$\mathbf{49.6} \pm 26.3$	$\mathbf{47.1} \pm 18.8$
P8	$\mathbf{37.8} \pm 25.4$	$\mathbf{38.2} \pm 19.4$	38.2 ± 25.1	38.6 ± 21.2
P9	$\mathbf{45.6} \pm 24.7$	$\mathbf{41.7} \pm 21.2$	45.9 ± 25.3	42.0 ± 20.7

In Tab. 3 are demonstrated the tracking times, which we achieved for various distributions of the sub-swarms/particles into the computational resources. As we can notice, for the same number of the sub-swarms the computation

Table 3: Tracking time [ms] and speed-up for single frame.

		#threads		Seq. 1 (P1)	
#swarms	#particles	PC1	PC2	time [ms]	speed-up
1	300	4	0	326.7	-
2	2×150	8	0	175.3	1.9
2	2×150	4	4	172.5	1.9
3	3×100	12	0	147.7	2.2
3	3×100	8	4	123.1	2.7
4	4×75	16	0	125.6	2.6
4	4×75	8	8	88.4	3.7
6	6×50	12	12	79.2	4.1
8	8×38	16	16	65.9	5.0

time is larger on single computer in comparison to a configuration consisting of two nodes connected by 1 GigE network. This means that the time needed for threads scheduling is larger than the time needed for information exchange in a distributed system. The image processing and analysis takes about 0.2 sec. and it is not included in the times shown in Tab. 3. For the RAPSO algorithm decomposed into two PCs and executing 8 swarms, the motion tracking can be done at about 15 fps. The complete human motion capture system was written in C/C++. The experiments were conducted on two desktop PCs, each equipped with two XEON X5690, 3.46 GHz (6-core) CPUs. The nodes were connected by a TCP/IP 1 GigE (Gigabit Ethernet) local area network. The parallelization of the code was done using OpenMP directives. It is worth noting that on *Lee walk* sequence from Brown University, the processing time of the algorithm proposed in [8], which has been implemented in Matlab, is larger than one hour.

In Tab. 4 are depicted average errors that were obtained on sequence P1 using various number of sub-swarms. As we can observe, the tracking accuracy is far better in comparison to tracking accuracy that was achieved in [10]. The results presented in Tab. 3-4 were achieved on images from sequence P1 sub-sampled by 4 and therefore the tracking accuracy on single computer is worse in comparison to the relevant accuracy shown in Tab. 2.

Table 4: Average errors [mm] obtained with various number of sub-swarms.

#swarms	#particles	error [mm]	std. dev. [mm]
1	300	41.9	23.6
2	2×150	43.0	24.2
3	2×100	44.9	25.6
4	4×75	42.6	22.7
6	6×50	45.4	27.3
8	8×38	50.6	35.8

4 Conclusions

We have proposed a particle swarm optimization with resampling for full body tracking. Owing to resampling the algorithm better copes with noise and reduces premature stagnation. The parallel algorithm, which was executed on two PC nodes with multi-core CPUs allowed us to perform real-time tracking at 15 fps. The accuracy of the algorithm for the real-time full body tracking is better in comparison to recently proposed parallel PSO algorithm. Experimental results on various multi-view sequences of walking subjects demonstrate the effectiveness of the approach.

Acknowledgment. This work has been supported by the National Science Centre (NCN) within the research project N N516 483240 and the National Centre for Research and Development (NCBiR) within the project OR00002111.

References

1. Barris, S., Button, C.: A review of vision-based motion analysis in sport. Sports Medicine 38(12), 1025–1043 (2008)
2. Chapman, B., Jost, G., van der Pas, R., Kuck, D.: Using OpenMP: Portable Shared Memory Parallel Programming. The MIT Press (2007)
3. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Tr. Evolut. Comp. 6(1), 58–73 (2002)
4. Coello, C., Pulido, G., Lechuga, M.: Handling multiple objectives with particle swarm optimization. IEEE Tr. on Evolutionary Computation 8(3), 256–279 (2004)
5. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: IEEE Int. Conf. on Pattern Recognition. pp. 126–133 (2000)
6. Dyer, S., Martin, J., Zulauf, J.: Motion capture white paper, SGI. Available at: `ftp://ftp.sgi.com/sgi/A%7CW/jam/mocap/MoCapWP_v2.0.html`, accessed March 15, 2012
7. Isard, M., Blake, A.: CONDENSATION - conditional density propagation for visual tracking. Int. J. of Computer Vision 29, 5–28 (2006)
8. John, V., Trucco, E., Ivekovic, S.: Markerless human articulated tracking using hierarchical particle swarm optimisation. Image Vis. Comput. 28, 1530–1547 (2010)
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. of IEEE Int. Conf. on Neural Networks. pp. 1942–1948. IEEE Press, Piscataway, NJ (1995)
10. Kwolek, B., Krzeszowski, T., Wojciechowski, K.: Real-time multi-view human motion tracking using 3D model and latency tolerant parallel particle swarm optimization. In: Proc. of the 5th Int. Conf. on Computer Vision/Computer Graphics Collaboration Techniques. pp. 169–180. MIRAGE'11, Springer, vol. 6930 (2011)
11. Kwolek, B., Krzeszowski, T., Wojciechowski, K.: Swarm intelligence based searching schemes for articulated 3D body motion tracking. In: Int. Conf. on Advanced Concepts for Intell. Vision Systems, LNCS. pp. 115–126. Springer, vol. 6915 (2011)
12. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. CVIU 104(2), 90–126 (2006)
13. Poppe, R.: Vision-based human motion analysis: An overview. Comp. Vision and Image Understanding 108(1–2), 4–18 (2007)
14. Quah, C.K., Koh, M., Ong, A., Seah, H.S., Gagalowicz, A.: Video-based motion capture for measuring human movement. In: Digital Sports for Performance Enhancement and Competitive Evolution: Intelligent Gaming Technologies, pp. 209–228. Information Science Publishing (2009)
15. Saboune, J., Charpillet, F.: Markerless human motion capture for gait analysis. Clinical Orthopaedics and Related Research (2005)
16. Sigal, L., Balan, A., Black, M.: HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. Int. Journal of Computer Vision 87, 4–27 (2010)
17. 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 in conjuction with CVPR. pp. 45–50. IEEE (2010)
18. Zhou, H., Hu, H.: Human motion tracking for rehabilitation - a survey. Biomedical Signal Proc. and Control 3, 1–18 (2008)