Real-Time Tracking of Full-body Motion using Parallel Particle Swarm Optimization with a Pool of Best Particles

Tomasz Krzeszowski[2,1], Bogdan Kwolek[2,1], Boguslaw Rymut[2,1], Konrad Wojciechowski[1], and Henryk Josinski[1]

[1] Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warszawa, Poland
bytom@pjwstk.edu.pl
[2] Rzeszów University of Technology
W. Pola 2, 35-959 Rzeszów, Poland
bkwolek@prz.edu.pl

Abstract. In this paper we present a particle swarm optimization (PSO) based approach for marker-less full body motion tracking. The objective function is smoothed in an annealing scheme and then quantized. This allows us to extract a pool of candidate best particles. The algorithm selects a global best from such a pool to force the PSO jump out of stagnation. Experiments on 4-camera datasets demonstrate the robustness and accuracy of our method. The tracking is conducted on 2 PC nodes with multi-core CPUs, connected by 1 GigE. This makes our system capable of accurately recovering full body movements with 14 fps.

1 Introduction

Tracking of 3D articulated body motion in image sequences plays an important role due to wide variety of potential applications. The aim of articulated body tracking is to estimate the joint angles of the human body at any time. The recovery of human body movements from image sequences is a very challenging problem. The difficulties arise mainly due to the high dimensionality and non-linearity of the search space, large variability in human appearance, noisy image observations, self-occlusion, and complex human motions. To cope with such difficulties, much previous work has focused on the use of 3D human body models of various complexity to recover the position, orientation and joint angles from 2D image sequences [3][4][9][10]. An articulated human body can be perceived as a kinematic chain consisting of at least eleven parts, corresponding to the main body parts. Usually such a 3D human model is built on very simple geometric primitives like truncated cones or cylinders. Given the 3D model, a lot of hypothetical body poses are generated and then projected into the image plane in order to find the configuration of the 3D model, whose projection matches best the current image observations. Multiple cameras and simplified backgrounds are commonly used to ameliorate some of practical difficulties arising due to occlusions and depth ambiguities [9][10].

2 Searching Schemes for Human Motion Tracking

In tracking of the articulated human motion the particle filtering is utilized in the majority of the trackers. Particle filters [5] are recursive Bayesian filters that are based on Monte Carlo simulations. They approximate a posterior distribution for the human pose on the basis of a series of observations. The high dimensionality of the search space entails vast number of particles to approximate well the posterior probability of the states. Moreover, sample impoverishment may prevent particle filters from maintaining multimodal distribution for longer periods of time. Therefore, many efforts have been devoted to confining the search space to promising regions that contain the true body pose. Deutscher and Reid [3] developed an annealed particle filter, which adopts an annealing scheme together with the stochastic sampling to achieve better concentration of the particle spread close to modes of the probability distribution. To achieve this the fitness function is smoothed using annealing factors $0 = \alpha_1 < \alpha_2, \ldots, < \alpha_n = 1$, and in consequence the particles migrate towards the extremum without getting stuck in local minima. In addition, a crossover operation is employed in order to obtain an improved particle's diversity.

The annealed particle filter greatly improves the tracking performance in comparison to the ordinary particle filtering. However, a considerable number of particles it still required. In contrast, the particle swarm optimization (PSO) [7], which is population-based searching technique, has higher searching capabilities owning to combining the local search and global one. A basic variant of the PSO algorithm is built on particles representing candidate solutions. These particles are moved around in the search-space according to a few very simple rules. 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. Particles move through the solution space, and undergo evaluation according to some fitness function $f()$. While the swarm as a whole gravitates towards the global extremum, the individual particles are capable of ignoring many local optima. In the dynamic optimization the aim is not only to seek the extrema, but also to follow their progression through the space as closely as possible. Since the object tracking is a kind of dynamic optimization, the tracking can be attained through incorporating the temporal continuity information into the ordinary PSO. In consequence, the 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 $\hat{x}_{t-1}$, $x_t^{(i)} \leftarrow \mathcal{N}(\hat{x}_{t-1}, \Sigma)$.

In the original PSO, convergence of particles towards its attractors is not guaranteed. Clerc and Kennedy [2] studied the mechanisms to improve the convergence speed and proposed constriction methodologies to ensure convergence and to fine-tune the search. They proposed to utilize 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 (APSO) [8] 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. $i^* = \arg\min_i \tilde{u}_t^i$, $g_t = p_t^{i^*}$, $w_t = u_t^{i^*}$
7. For $k = 0, 1, \ldots, K$
8. update ω_α on the basis of (2)
9. $G = \arg\min_i round(num\_bins \cdot \tilde{u}_t^i)$
10. For each particle i
11. Select a particle from $\{G \cup g_t\}$ and assign it to g_t^i
12. $v_t^{i,k+1} = \omega_\alpha[v_t^{i,k} + c_1 r_1(p_t^i - x_t^{i,k}) + c_2 r_2(g_t^i - x_t^{i,k})]$
13. $x_t^{i,k+1} = x_t^{i,k} + v_t^{i,k+1}$
14. $f_t^i = f(x_t^{i,k+1})$
15. if $f_t^i < u_t^i$ then $p_t^i = x_t^{i,k+1}$, $u_t^i = f_t^i$, $\tilde{u}_t^i = (u_t^i)^{\alpha_k}$
16. if $f_t^i < w_t$ then $g_t = x_t^{i,k+1}$, $w_t = f_t^i$

The smoothed objective functions are quantized, see 9th line in the pseudo-code. Owing to this the similar function values are clustered into the same segment of values. In each iteration the algorithm determines the set G of the particles, which after the quantization of the smoothed fitness function from the previous iteration, assumed the smallest values (the best fitness scores), see 9th line in the pseudo-code. For each particle i the algorithm selects the global best particle g_t^i from $\{G \cup g_t\}$, where g_t determines the current global best particle of the swarm. By means of this operation the swarm selects the global best location from a pool of candidate best locations to force the PSO jump out of stagnation. We found that this operation contributes considerably toward better tracking, particularly in case of noisy observations. It is worth noting that in the literature devoted to dynamic optimization the problem of optimization of noisy objective functions is considered very rarely.

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 segmented body parts and the projected model's parts into 2D image. The function $f_2(x)$ reflects the edge distance-based fitness [8].

3 Parallel APSO for Real-time Motion Tracking

PSO is parallel in nature. To shorten the optimization time several studies on parallelizing the algorithm were done so far. However, the majority of the algorithms are for the static optimization. In object tracking 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 [1], which communicate via the shared memory, see Fig. 1. 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.

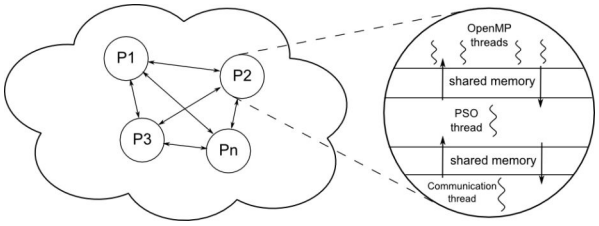

Fig. 1: The communication in parallel PSO for real-time object tracking.

4 Experimental results

The algorithm was evaluated on two image sequences acquired by four synchronized and calibrated cameras. The color images of size 1920×1080 were acquired with rate 25 fps and then subsampled at a factor of 4 both horizontally and vertically. Each pair of the cameras is approximately perpendicular to the other two. A commercial motion capture (moCap) system from Vicon Nexus provides ground truth data at rate of 100 Hz. The system employs reflective markers and sixteen cameras to recover the 3D location of such markers. The synchronization between the moCap and multi-camera system is based on hardware from Vicon.

All computations were conducted on a computer cluster that was composed of 2 identical machines connected with a TCP/IP 1 GigE (Gigabit Ethernet) local area network. Each PC node is equipped with two six-core Intel Xeon 5690

3.46 GHz CPUs. They support Hyper-Threading technology, which enables a single core to act like multiple cores. In this way, a core with Hyper-Threading appears to be more than one core. For example, if the CPU is a dual core processor with Hyper-Threading, the operating system will process as many as four threads through it simultaneously.

The accuracy of human motion tracking was evaluated experimentally in scenarios with a walking person. Although we focused on tracking of torso and legs, we also estimated the head's pose as well as the pose of both arms. The body pose is described by position and orientation of the pelvis in the global coordinate system as well as relative angles between the connected limbs. The overlap of the projected 3D model on the subject undergoing tracking can be utilized to illustrate the quality of tracking, see Fig. 2, which depicts the frontal

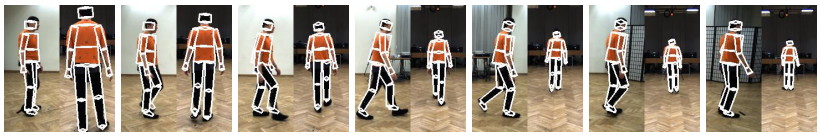

Fig. 2: Articulated 3D human body tracking. Shown are results in frames #20, 40, 60, 80, 100, 120, 140, obtained by APSO. The left sub-images are seen from view 1, whereas the right ones are seen from view 4.

and side views from two nearly perpendicular cameras. As we can see, the overlap of the projected model on both images is quite good. The estimation of the 3D pose was done in 10 iterations using 300 particles. Given the estimated human pose we calculated the location of virtual markers on the model. The location of such markers on the body corresponds to the location of the real markers on the person undergoing tracking. The pose error in each frame was calculated as the average Euclidean distance between corresponding markers. We used 39 markers, where 4 markers were placed on the head, 7 markers on each arm, 12 on the legs, 5 on the torso and 4 markers were attached to the pelvis.

The results obtained on two image sequences were compared by analyses carried out both through qualitative visual evaluations as well as quantitatively by the use of the motion capture data as ground truth. The tracking was done using various number of particle swarms and PC nodes, see Tab. 1. The pool of the particles was distributed evenly among the sub-swarms. The results shown in Tab. 1 demonstrate that the motion tracker based on APSO is better than PSO-based one in terms of the tracking accuracy. As we can observe, the tracking error increases slightly with the number of the swarms. The reason for the poorer accuracy of tracking is that we employ the non-blocking parallel PSO. At two PC nodes the processing time of the blocking parallel PSO is a dozen or so milliseconds larger in comparison to the non-blocking version. The discussed results were obtained in ten runs of the algorithm with unlike initializations.

Table 1: Average errors [mm] for M = 39 markers in two image sequences.

		#threads		Seq. 1		Seq. 2	
#swarms	#particles	PC1	PC2	PSO	APSO	PSO	APSO
1	300	4	0	59.3 ± 33.4	54.9 ± 30.8	57.7 ± 33.6	52.3 ± 27.3
2	2×150	4	4	59.5 ± 33.0	54.2 ± 29.9	61.5 ± 37.0	52.2 ± 28.2
3	3×100	8	4	59.9 ± 35.3	55.3 ± 31.2	62.2 ± 38.6	56.9 ± 37.0
4	4×75	8	8	59.5 ± 34.0	54.8 ± 30.3	60.6 ± 37.3	54.4 ± 33.7
6	6×50	12	12	62.6 ± 36.5	58.2 ± 32.2	62.5 ± 42.1	61.5 ± 43.6
8	8×38	16	16	73.0 ± 46.6	57.7 ± 31.1	69.3 ± 47.9	62.3 ± 43.5

Figure 3 depicts the tracking errors versus frame number that were obtained during motion tracking using APSO and PSO-based motion trackers. The experiments were done on an image sequence acquired by the four camera system. As we can observe in the plots shown at Fig. 3, the tracking accuracy obtained by the APSO-based tracker is much better. In particular, in some frames the accuracy of PSO-based tracker considerably drops. This takes place because the PSO is unable to find the global extremum in a given number of iterations. Similar effect has been observed in many runs of the algorithms with unlike ini-

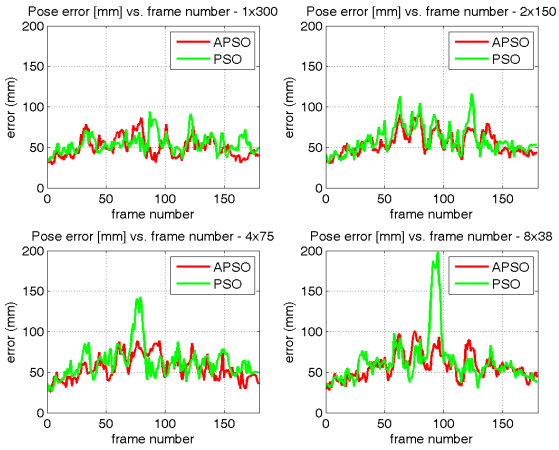

Fig. 3: Tracking errors [mm] versus frame number for PSO and APSO for various number of particles in the sub-swarms.

tializations. In general, APSO performs better than PSO over the whole image sequences, attaining much better accurateness and robustness.

In Fig. 4 are shown the errors that were obtained using single and eight swarms. In a APSO consisting of eight sub-swarms the optimizations were done using 38 particles in each swarm. As we can see, the tracking errors of both legs are something larger in comparison to tracking errors of the torso.

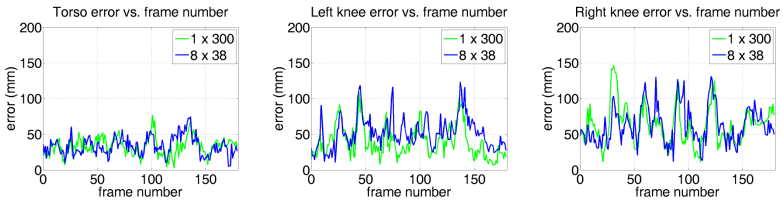

Fig. 4: Tracking errors [mm] versus frame number at 1 and 2 PCs using 1 and 8 particle swarms, respectively.

In Tab. 2 are demonstrated the tracking times that were obtained for various distributions of the sub-swarms into the computational resources. As we can observe, for identical number of the sub-swarms the computation 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 necessary for scheduling the threads is larger in comparison to time needed for information exchange in a

Table 2: Tracking time [ms] and speed-up for a single frame.

		#threads		Seq. 1		Seq. 2	
#swarms	#particles	PC1	PC2	time [ms]	speed-up	time [ms]	speed-up
1	300	4	0	367.0	-	333.2	-
2	2×150	8	0	195.7	1.9	182.5	1.8
2	2×150	4	4	195.9	1.9	183.1	1.8
3	3×100	12	0	163.8	2.2	153.0	2.2
3	3×100	8	4	136.6	2.7	122.4	2.7
4	4×75	16	0	138.9	2.6	125.7	2.7
4	4×75	8	8	126.2	2.9	116.8	2.9
6	6×50	12	12	86.6	4.2	80.5	4.1
8	8×38	16	16	70.9	5.2	67.6	4.9

distributed system. The image processing and analysis takes about 0.2 sec. and it is not included in the times shown in Tab. 2. The complete human motion capture system was written in C/C++ and works in real-time. It is worth noting that in [6], the processing time of *Lee walk* sequence from Brown University is larger than one hour.

5 Conclusions

We presented a marker-less motion capture system for real-time tracking of 3D full body motion. The performance of the proposed algorithms was evaluated on two image sequences captured by 4 cameras. In many quantitative comparisons of APSO and the competing PSO algorithm, APSO expressed better tracking accuracy. APSO shows good global search ability making it well suited for unconstrained motion tracking, where no strong prior or dynamic model is available.

Acknowledgment

This paper has been supported by the research project OR00002111: "Application of video surveillance systems to person and behavior identification and threat detection, using biometrics and inference of 3D human model from video."

References

1. Chapman, B., Jost, G., van der Pas, R., Kuck, D.: Using OpenMP: Portable Shared Memory Parallel Programming. The MIT Press (2007)
2. 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)
3. 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)
4. Deutscher, J., Reid, I.: Articulated body motion capture by stochastic search. Int. J. Comput. Vision 61(2), 185–205 (2005)
5. Doucet, A., Godsill, S., Andrieu, C.: On sequential Monte Carlo sampling methods for bayesian filtering. Statistics and Computing 10(1), 197–208 (2000)
6. John, V., Trucco, E., Ivekovic, S.: Markerless human articulated tracking using hierarchical particle swarm optimisation. Image Vis. Comput. 28, 1530–1547 (2010)
7. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. of IEEE Int. Conf. on Neural Networks. pp. 1942–1948. IEEE Press, Piscataway, NJ (1995)
8. 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)
9. 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)
10. 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)