Marker based Human Pose Estimation
Using Annealed Particle Swarm Optimization with Search Space Partitioning

Ashraf Sharifi, Ahad Harati, Abedin Vahedian
Department of Computer Engineering
Ferdowsi University of Mashhad
Mashhad, Iran
sharifi.a7@stu.um.ac.ir, a.harati@um.ac.ir, vahedian@um.ac.ir

Abstract —**In this paper, a marker based human pose estimation from multi-view video sequences is presented. The pose estimation problem is defined as optimization of the 45 parameters which define body pose model and is solved using particle swarm optimization (PSO). The objective of this optimization is to maximize a fitness function which formulates how much body model matches with 2D marker's coordinate in video frames.**

In this algorithm a sampling covariance matrix is used in the first part of the velocity equation of PSO that is annealed with iterations. One of the major problems of this algorithm is the high number of parameters that define the pose of the body model. To tackle this problem, we divide the optimization into six stages that exploit the hierarchical structure of the model. The first stage optimizes the six parameters that define the global orientation and position of the body. Other stages are related to optimization of right and left hand, right and left leg and head orientation, respectively.

Experimental results on PEAR[1] dataset [1] indicate that the proposed algorithm can achieve lower estimation error in tracking human motion compared with Annealed Particle Filter (APF) and Standard Particle Filter (PF).

Keywords- marker based human motion tracking; particle swarm optimization (PSO)

I. INTRODUCTION

Human pose estimation is the process of determining the configuration (orientation and location) of body parts which is an important problem in human–computer interaction, computer graphic and animation, medical gait analysis and so on.

3D human motion tracking can be formulated as a nonlinear optimization problem. Gradient based methods are faster than gradient free methods but one of the major problems in these methods is the local optima convergence that yields poor tracking results [2]. Particle filter (PF) can alleviate this problem, but a major challenge is that for successful tracking, the required particles' count increases exponentially with the dimension of the state space [3].

In 2000, Deutscher [4] introduced a modified particle filter termed Annealed Particle Filter (APF), that required fewer particles and could get better results compared with standard particle filter. To better explore the search space, instead of using single weighting function in PF, APF uses a series of weighting functions (W_0 to W_M) in which each W_m differs slightly from W_{m-1}. The first function (W_m) is designed so smooth, to represent the overall trend of search space while W_0 emphasizes the local features of it, thus the initial searching area is global at first and gradually becomes local with in layers.

In 2005, an improved version of APF was proposed that uses crossover operation and search space partitioning [5], this version of APF was used as the baseline algorithm in HumanEva framework [6].

In [7], GA algorithm was proposed to estimate upper-body pose and Zhao and Liu presented an annealed genetic algorithm (AGA) for 3D human motion analysis [8].

Particle Swarm Optimization (PSO) is an evolutionary optimization algorithm that was used in 2006 to solve upper-body pose estimation [9] and has recently got more attention for full-body pose estimation [2, 10-12,16-17]. The particles in PSO communicate with each other, therefore, searching is more efficient than the crossover operation in GA [5].

In this paper, we use PSO for full body pose estimation. One of the major challenges in this problem is the high number of DOF that has to be recovered. An effective method to reduce search complexity in such problem is the search space decomposition. In this method, one section of search space is optimized independently and its result is used as a constraint to limit the rest of search space [3].

In this work, we propose to split the 45-dimensional search space into six stages so as to overcome the mentioned problem.

To our knowledge, this is the first time that combination of search space decomposition and PSO algorithm is used in marker based human pose estimation.

The performance of the proposed algorithm is assessed by means of objective metrics that are defined in the framework of HumanEva dataset [6] and metrics introduced in [13].

The rest of this paper is organized as follows. The background is presented in Section I. In Section II, we explain proposed tracking algorithm. Experimental results are shown in Section III, and Section IV is devoted to conclusion.

II. BACKGROUND

Articulated human body model that is used in tracking process is first reviewed. Next propagation model is explained followed by presentation of likelihood model.

[1] Pose Estimation and Action Recognition

A. Human Body Model

The kinematic structure of the human body is usually represented as a 3D kinematic tree where each node corresponds to a joint in the human body. Torso joint of the human body is usually assumed as the root node in kinematic tree. Every node has up to three rotational DOF, while the root node also has three translational DOF that determines the global position of the body. The joint angles are variable parameters during the tracking process while length of body limbs is assumed to be constant. In this paper we used 45 parameters, including global position and joint angles, to represent the full body model.

B. Propagation Model

Pose tracking process is performed by maximizing a fitness function in image sequences. Population in each frame, is generated from the estimated pose in previous frame. To initialize optimization in each frame, a motion model is employed to propagate particles to the new frame.

Tracking algorithms presented in [2, 5, 14, 15] are examples of algorithms that use zero motion with additional Gaussian noise as motion model:

$$x_i^t \leftarrow N(\hat{x}^{t-1}, \Sigma)\ \ ,\Sigma = \begin{pmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{45}^2 \end{pmatrix}, \sigma = \begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_{45} \end{pmatrix} \quad (1)$$

The standard deviation σ_d in the Σ matrix is equal to the maximum absolute inter-frame differences of the joint angles that are almost determined in a training process.

Experiments on the motion model conducted in [6], show that tracking accuracy is largely dependent on the amount of standard deviation values in the Σ matrix. These experiments show that most accurate results were obtained when style-specific motion model was used, For example, to track walking motion, only data related to walking motion were used to learn the sampling covariance matrix (Σ).

In proposed algorithm, we use a generic motion model in which the amount of standard deviation for all joints and for all motions is set to a low value (0.1 radian in this paper). This motion model eliminated the need for training process. As we show in Section III, using this motion model, our algorithm can track different motions with no prior knowledge of motions type.

C. Likelihood Evaluation

In order to evaluate the likelihood between the body pose and observation data, we use the method that is presented in [14] which is briefly explained.

$2D$ coordinate of markers attached to the body of the performer onto N_C camera views (four views in this paper), provide observation data (z_t) in this tracking system.

If $D_n = \{d_1, d_2, \dots, d_{Qn}\}$ is the set of Q_n, $2D$ coordinate of markers in the $n-th$ view and $X = \{x_1, \dots, x_M\} \in R^3$ is the $3D$ position of the body joints and the end of the limbs, using forward kinematics, the computed fitness function should measure how well these $2D$ coordinates fit as projections of $3D$ coordinate of the set X . For every element x_m in the set X, its projection onto every camera view is computed in (2).

$$P_{m,n} = P_n(x_m)\ , \quad 1 \le m \le M\ , 1 \le n \le N_c \quad (2)$$

Where P_n is the perspective projection operator from 3D to $2D$ on the $n-th$ view. Next, the set $T_m = \{t_1, \dots, t_{NC}\}$ containing the closest measurement in every camera view associated to every element x_m is constructed according to (3).

$$t_n = min_{d_q} \|p_{m,n} - d_q\|\ ,\ d_q \in D_n, \forall n \quad (3)$$

Then the extension of the symmetric epipolar distance for $k \ge 2$ points in k different views ($d_{s\varepsilon}(x^1, \dots, x^k)$) is used to compute fitness value. $d_{s\varepsilon}(x^1, \dots, x^k)$ can be computed as (4):

$$d_{s\varepsilon}(x^1, \dots, x^k) = \sqrt{\sum_{i=1}^{k-2} \sum_{j=i+1}^{k-1} d_{s\varepsilon}^2(x^i, x^j)} \quad (4)$$

Where $d_{s\varepsilon}(x^j, x^j)$ is symmetric epipolar distance between two points x^i and x^j in the two views i and j and is defined in (5):

$$d_{s\varepsilon}(p^i, p^j) \triangleq \sqrt{d^2(l(x^i, j), x^j) + d^2(l(x^j, i), x^i)} \quad (5)$$

$l(x^i, j)$ in (5) is the epipolar line generated by the point x in a given view i onto another view j and $d(l(x^i, j), x^j)$ is defined as Euclidean distance between the epipolar line $l(x^i, j)$ and the point x^j.

When the $2D$ points are projected from the same $3D$ point, this distance decreases. With this definition, the score s_m and weighting function is formulated in (6-7).

$$s_m(z_t, x_m) \equiv s_m(z_t, T_m) \ \propto d_{s\varepsilon}(T_m) \quad (6)$$

$$w(z_t, y) = \exp\left(-\frac{1}{M} \sum_{m=1}^{M} s_m(z_t, x_m)\right) \quad (7)$$

III. Optimization Algorithm

PSO is a computational method that is used to solve optimization problems which is based on a population of candidate solutions.

PSO was initially used for upper-body pose estimation problem in 2006 [9]. One of the most important properties of PSO is that unlike the other particle based methods, such as Particle Filter and its variants, particles share their information with each other and with the best particle in the whole population therefore the searching is more efficient than the crossover operation in GA[5].

PSO is based on a swarm consisting of N particles. Let x^i and v^i be the $i-th$ particle and velocity of it, respectively, and p^i be the best position of $i-th$ particle that encountered so far and p^g be the best known position of the entire swarm.

With these assumptions, The PSO algorithm can be explained as follows:

1. Initialize the population's position and velocity randomly in the search space. Set p^i for each particle and identify best particle in the swarm and set as p^g.
2. Repeat until stopping criterion is satisfied:

– compute velocity of particles according to (8) and update the position of every particle x^i based on (9):

$$v_{t+1}^i = v_t^i + \varphi_1\left(p_t^i - x_t^i\right) + \varphi_2\left(p_t^g - x_t^i\right) \quad (8)$$

$$x_{t+1}^i = x_t^i + v_{t+1}^i \quad (9)$$

Where subscript t denotes the time step (iteration). Parameters φ_1 and φ_2 are random numbers drawn from [0; 1].

– For $\mathrm{i} = 1 \ldots \mathrm{N}$ update p^i , p^g
Criterion for termination is usually the maximum number of iterations.
The first component in (8) is known as "habit" or "history" and makes particles to continue moving in the same direction that they has been traveling so far [18]. As mentioned above, in the classical PSO this component is initialized randomly. The experiments show that this component leads to produce many impossible body poses. One reason could be that because of the complex motion of body parts, history of particles may not be much reliable.
To solve this problem, we use the method presented in [12]. In this method the sampling covariance matrix is used to initialize the first part in (8) that annealed during iterations. Thus the initial searching area is global at first and gradually becomes local during the iterations.
So the velocity equation (8) is updated as (10).

$$v^{i,n+1} = P_n + \varphi_1(p^i - x^{i,n}) + \varphi_2(g - x^{i,n}) \quad (10)$$

Given sampling factor $\alpha_n < 1$, the covariance matrix P_n is evolved as follows:

$$P_n = \alpha_n * P_0 \quad (11)$$

Where P_0 is covariance matrix described in section *1.B*. In proposed algorithm, sampling factor α_n is formulated in (12).

$$a = -0.8 * \left(\frac{n}{M}\right) + 1 \quad (12)$$

In this equation n is the current number of iterations and M is the maximal number of iterations.
As mentioned before one of the major challenges in this problem is the high number of DOF that has to be recovered. An effective method to reduce search complexity in such problem is search space decomposition [3]. So we use optimization of body pose in six stages that each one is optimized using PSO algorithm that is described above. In the first stage, six parameters relative to global position and orientation of torso are optimized. Other five stage are relative to optimization of left and right hand, left and right leg and head orientation, respectively. To avoid error accumulation [2] the standard deviations for first six optimized parameters are reduced to one tenth in these five stages and a trivial search about the result of stage one is performed in the other stages.

IV. Experimental And Results

In order to test the proposed algorithm we used PEAR dataset [1]. This dataset contains synchronized motion capture and 16 visual streams from different views at resolution 704 *576 pixels and frame-rate of 25fps.
PEAR dataset contains 5 subjects performing 6 predefined actions. The predefined actions include walk, jump, skip, wave, stretch, jog and Ground-truth data is available.
To carry out the test of performance the metrics presented in [6] were used which included mean-average Euclidean distance between the estimated landmark and ground truth- and standard deviation (σ) of the estimation error and two metrics (MMTA[2] and MMTP[3]) that are proposed in [13].
Let $\hat{x}$ be the landmark position associated to estimated pose and x be the ground truth position, with this assumption MMTA is percentage of $\hat{x}$ positions that are closer than δ (10cm in this paper) to the ground truth positions:

$$\in = \|x_m - \widehat{x_m}\| < \delta \quad (13)$$

MMTP is the average of distance between $\hat{x}_m$ andx_m for all these pairs. Finally average of this metrics for all frames is computed.
In this section, various experiments are analyzed. First the effect of algorithm parameters such as number of particles, number of iterations and performance of marker detection algorithm are examined, then performance of algorithm in low frame rate and computational time for each stage of optimization is reported finally performance of algorithm is compared with the annealed particle filter and particle filter.

A. Number of Particles and Iterations

The base configuration (number of particles and iteration) of proposed algorithm is shown in Table I. As mentioned, stages 1 to 6 are related to Torso position and orientation, left and right hand, left and right leg and head orientation, respectively.
Table II and Fig. 1, shows result of algorithm with base configuration. In order to test repetitively of algorithm, each experiment was carried out several times (5 runs in this paper) and mean of error is reported. The result of this table indicate that in %97 of evaluated frames, difference between ground truth and estimated pose is below 10cm and mean of error is 9mm. As can be seen in this table the algorithm is less efficient for more complex motions (jump and skip) compared to motions like wave and walk.
In order to test the number of particles effect on the performance of pose tracking algorithm, we ran the proposed algorithm with different particle size while

[2] Multiple Marker Tracking Accuracy

[3] Multiple Marker Tracking Precision

keeping total number of fitness evaluation equal to the base configuration. Table III and Table IV shows different configuration for this test and mean result for 5 runs, respectively. As indicated in Table IV, the performance of tracking with low particle size (config1) or low iteration number (config4) is not good enough, while the algorithm performs well in medium size of particles and iteration (base configuration and configuration 2).

The "NAN" value for MMTP metric in this table means that in one or more frames, none of the estimated landmark positions are closer than 10cm to ground truth positions.

B. Marker Detection Algorithm

In this section we analyze the effect of marker detection accuracy on the performance of proposed pose estimation algorithm. For this purpose, performance of marker detection method is assessed by three factors: detection rate ($\overline{DR}$, miss detection), False positive rate ($\overline{FP}$, false measurement) and the variance of estimation error (σ^2 adding a Gaussian noise to all 2D coordinates of markers).

In the first experiment, it is assumed that $\overline{FP}$=0 and $\sigma^2 = 0$ and effect of DR factor on algorithm's performance is analyzed. Fig.3 shows result of this experiment.

In the second experiment, it is assumed that DR=0.9 and result for the case $FP = \{0\ ,20{,}40\}$ and $\sigma = \{0{,}20{,}40\}$ is shown in Fig.4 and Fig.5.

As it can be seen in these figures, the mean tracking error for most of the motions and frames, is almost close together. So we can say that our tracking algorithm is fairly robust to changes in marker detection accuracy.

C. Performance In Low Frame Rate

In this section, the performance of proposed algorithm is evaluated in low frame rate. To do this test, the value of matrix P_0 in (11) used for 25fps are doubled and are used as initial covariance matrix in (10).

Table V, shows result of proposed algorithm with base configuration and in 12fps frame rate. To test the repetitively of the algorithm, each experiment was carried out for 3 runs and mean of error is reported. As it can be seen in this table, this algorithm achieves good performance in stretch, wave and walk motions and acceptable performance in more complex motions like jump and skip motions.

D. Computational Time

The experiments have been tested on a 2.67 GHz Intel core2 CPU with 8 GB RAM, using MATLAB 2013. Computational time for each stage is represented in Table VI.

E. Comparison with PA and APF

In this section we compare the proposed algorithm with APF and Parametric Annealing (PA) algorithm which is based on APF and presented in [19]. Experiments of APF were performed with 700 particles and 5 annealing layers for APF algorithm and "vh" setup in [19] for PA algorithm.

The result of this experiment is reported in Table VII. As it can be seen in this table, performance of the proposed tracking method is significantly better compared to APF and PA algorithms.

TABLE I. BASE CONFIGURATION

STAGE	*PARTICLES*	*ITERATION*
1-5	80	10
6	30	10

.

TABLE II. RESULT OF PROPOSED TRACKING ALGORITHM WITH BASE CONFIGURATION IN 25 FPS

MOTION	MMTA	MMTP(MM)	μ(MM)	σ (MM)
STRETCH	**96.78**	**7.61**	**12.69**	**1.00**
WAVE	**99.69**	**6.14**	**6.51**	**4.52**
WALK	**99.22**	**5.22**	**3.38**	**4.65**
JUMP	**94.85**	**13.53**	**19.90**	**16.14**
SKIP	**96.43**	**16.02**	**21.40**	**14.72**
MEAN	**97.39**	**9.70**	**12.77**	**10.21**

TABLE III. DIFFERENT CONFIGURATION WITH DIFFERENT PARTICLES SIZE

	Config1		Config2		Config3	
Stages	Particles	Iteration	Particles	Iteration	Particles	Iteration
1-5	20	40	40	20	400	2
6	30	10	30	10	30	10

TABLE IV. TRACKING RESULT FOR DIFFERENT PARTICLES SIZE

	Base configuration				Config1			
	MMTA	MMTP	μ	σ	MMTA	MMTP	μ	σ
Mean	97.39	9.70	12.77	10.21	68.90	NaN	145.51	121.09
	Config2				Config3			
	MMTA	MMTP	μ	σ	MMTA	MMTP	μ	σ
Mean	95.78	10.34	16.49	13.68	80.19	NaN	100.16	98.31

TABLE V. RESULT OF PROPOSED TRACKING ALGORITHM WITH BASE CONFIGURATION IN FRAME RATE 12 FPS

MOTION	**MMTA**	**MMTP(MM)**	**μ(MM)**	**σ (MM)**
STRETCH	**93.57**	**8.89**	**21.68**	**22.46**
WAVE	**100**	**6.75**	**6.75**	**1.74**
WALK	**90.29**	**7.13**	**25.49**	**22.75**
JUMP	**72.15**	**20.85**	**73.27**	**58.22**
SKIP	**88.24**	**17.95**	**41.42**	**52.25**
MEAN	**88.85**	**12.32**	**33.72**	**31.48**

TABLE VI. TIME CONSUMPTION FOR EACH STAGES

stage	Time for each frame(second)
Torso position and orientation	0.4
Left and right hand	0.7
Left and right leg	0.55
head	0.32

TABLE VII. COMPARISON OF MEAN TRACKING ERROR(MM) OF PROPOSED TRACKING ALGORITHM TO APF AND PF

Motion	Proposed Algorithm	PA	APF
Stretch	10.66	56.63	69.08
Wave	6.77	75.85	81.94
Walk	5.65	42.68	79.67
Jump	19.87	138.45	311.05
Skip	15.48	69.11	118.01
Mean	11.69	76.55	131.95

V. Conclusions

In this paper a combination of annealed PSO algorithm and partition sampling is presented for marker based pose estimation. The proposed algorithm used a generic motion model that eliminated the need for training process. According to experimental results it is able to track different motion with no prior knowledge about motion type.

Utilizing a gradient based method as a local refinement stage after optimization with the proposed algorithm, can be one area of future work, which is expected to improve the efficiency of pose estimation algorithm.

References

[1] X. Zhao and Y. Liu, "PEAR: Synchronized 16-Views Visual and 3D Human Pose Dataset for Pose Estimation and Action Recognition," Shanghai Jiao Tong University Technical Report2012.

[2] P. Fleischmann, I. Austvoll, and B. Kwolek, "Particle swarm optimization with soft search space partitioning for video-based markerless pose tracking," presented at the Proceedings of the 14th international conference on Advanced Concepts for Intelligent Vision Systems, Brno, Czech Republic, 2012..

[3] J. MacCormick and M. Isard, "Partitioned Sampling, Articulated Objects, and Interface-Quality Hand Tracking," presented at the Proceedings of the 6th European Conference on Computer Vision-Part II, 2000.

[4] J. Deutscher, A. Blake, and I. Reid, "Articulated Body Motion Capture by Annealed Particle Filtering," in Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, Hilton Head Island, SC 2000, pp. 126 - 133.

[5] J. Deutscher and I. Reid, "Articulated body motion capture by stochastic search," International Journal of Computer Vision, pp. 185-205, 2005.

[6] L. Sigal, A. O. Balan, and M. J. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," International Journal of Computer Vision, pp. 4-27, 2010..

[7] J. Ohya and F. Kishino, "Human posture estimation from multiple images using genetic algorithm.," in Conference A: Computer Vision & Image Processing, Proceedings of the 12th IAPR International Conference on, 1994, pp. 750-753.

[8] X. Zhao and Y. Liu, "Generative tracking of 3D human motion by hierarchical annealed genetic algorithm," Pattern Recognition, vol. 41, pp. 2470-2483, 2008.

[9] S. Ivekovic and E. Trucco, "Human body pose estimation with pso," in Evolutionary Computation, 2006. CEC 2006. IEEE Congress on Vancouver, BC 2006, pp. 1256 - 1263

[10] S. Ivekovic, V. John, and E. Trucco, "Markerless Multi-View Articulated Pose Estimation Using Adaptive Hierarchical Particle Swarm Optimisation," presented at the Proceedings of the 2010 international conference on Applications of Evolutionary Computation, Istanbul, Turkey, 2010.

[11] V. John, E. Trucco, and S. Ivekovic, "Markerless human articulated tracking using hierarchical particle swarm optimisation," Image and Vision Computing, pp. 1530–1547, 2010.

[12] X. Wang, X. Zou, W. Wan, and X. Yu, "Articulated 3D human pose estimation with Particle Filter based Particle Swarm Optimization " in Audio Language and Image Processing (ICALIP), 2010 International Conference on, Shanghai, 2010.

[13] C. Canton-Ferrer, J. Casas, E. Monte, and M. Pardas, "Towards a Fair Evaluation of 3D Human Pose Estimation Algorithms," Technical University of Catalonia, Barcelona, Spain2009.

[14] C. Canton-Ferrer, J. R. Casas, and M. Pard, "Marker-based human motion capture in multiview sequences," EURASIP Journal on Advances in Signal Processing, vol. 2010, pp. 1-11, 2010.

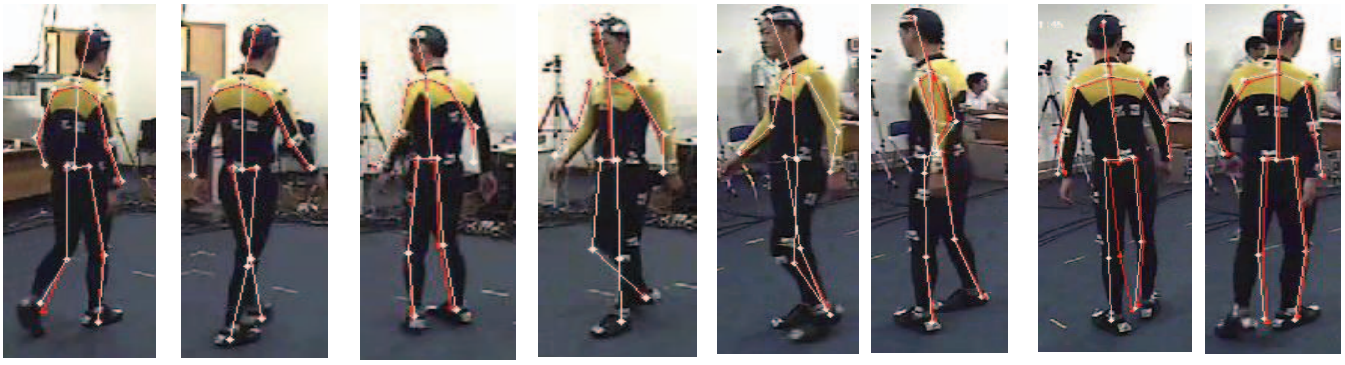
[15] N. H. Lehment, D. Arsic, M. Kaiser, and G. Rigoll, "Automated Pose Estimation in 3D Point Clouds Applying Annealing Particle Filters and Inverse Kinematics on a GPU," in Image Processing (ICIP),17th IEEE International Conference on, Hong Kong, 2010.

[16] X. Zhang, W. Hu, X. Wang, Y. Kong, N. Xie, H. Wang, et al., "A Swarm Intelligence Based Searching Strategy for Articulated 3D Human Body Tracking," in Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, San Francisco, CA, 2010.

[17] Z. Zhang, H. S. Seah, C. K. Quah, and J. Sun, "GPU-Accelerated Real-Time Tracking of Full-Body Motion With Multi-Layer Search," IEEE Transactions on Multimedia, vol. 15, pp. 106-119 2013.

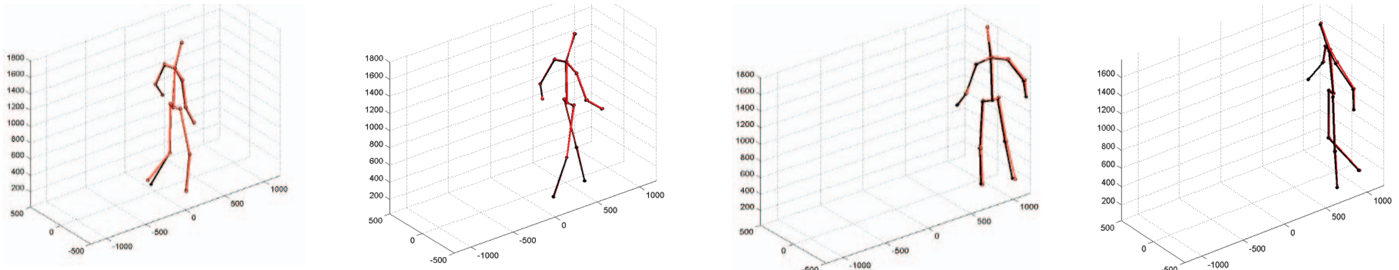
[18] Y. d. Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems," Evolutionary Computation, IEEE Transactions on, vol. 12, pp. 171-195, 2008.

[19] P. Kaliamoorthi and R. Kakarala, "Parametric annealing: A stochastic search method for human pose tracking," *Pattern Recognition,* vol. 46, pp. 1501–1510, 2012.



Frame: 2 , E= 12.30 Frame: 11 , E= 6.03 Frame:36, E=11.66 Frame:54, E=9.68 Frame:111,E=3.51 Frame:118,E= 11.39 Frame:127,E=43.71 Frame:138, E=25.28

Figure 1. Tracking result for walking motion, projection of ground truth skeleton is showed in white and estimated skeleton is colored red , E denotes mean tracking error(mm) at depicted frame



Frame: 2 , E= 12.30 Frame: 11 , E= 6.03 Frame:36, E=11.66 Frame:54, E=9.68

Figure 2. 3D tracking result for walking motion, ground truth skeleton is showed in black and estimated skeleton is colored red, E denotes mean tracking error(mm) at depicted frame

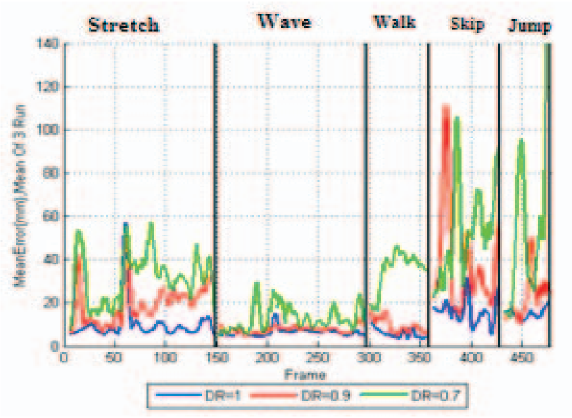


Figure 3. Error graph for different DR value where FP=0 and σ=0

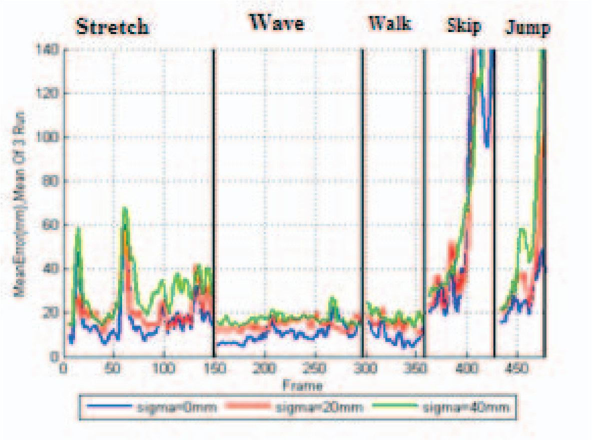


Figure 4. Error graph for different values of σ, where DR=0.9 and FP=20

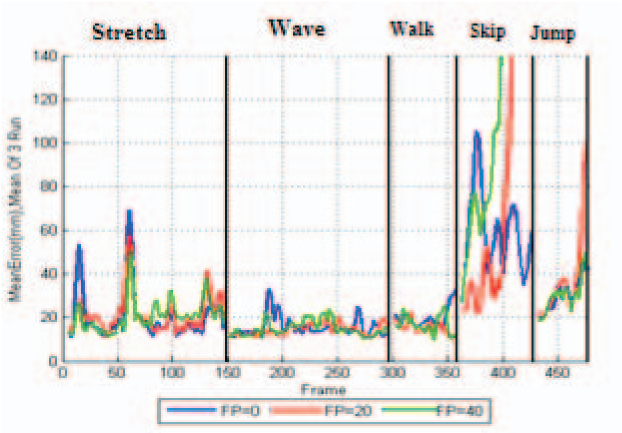


Figure 5. Error graph for different values of FP, where DR=0.9 and σ=20mm