Comparing Evolutionary Algorithms and Particle Filters for Markerless Human Motion Capture

Enrique Yeguas-Bolivar*, Rafael Muñoz-Salinas, Rafael Medina-Carnicer, Angel Carmona-Poyato

Department of Computing and Numerical Analysis. Maimonides Institute for Biomedical Research (IMIBIC). University of Córdoba. Córdoba (Spain)

Abstract

Markerless Human Motion Capture is the problem of determining the joints' angles of a three-dimensional articulated body model that best matches current and past observations acquired by video cameras. The problem of Markerless Human Motion Capture is high-dimensional and requires the use of models with a considerable number of degrees of freedom to appropriately adapt to the human anatomy.

Particle filters have become the most popular approach for Markerless Human Motion Capture, despite their difficulty to cope with high-dimensional problems. Although several solutions have been proposed to improve their performance, they still suffer from the *curse of dimensionality*. As a consequence, it is normally required to impose mobility limitations in the body models employed, or to exploit the hierarchical nature of the human skeleton by partitioning the problem into smaller ones.

Evolutionary algorithms, though, are powerful methods for solving continuous optimization problems, specially the high-dimensional ones. Yet, few works have tackled Markerless Human Motion Capture using them. This paper evaluates the performance of three of the most competitive algorithms in continuous optimization – Covariance Matrix Adaptation Evolutionary Strategy, Differential Evolution and Particle Swarm Optimization – with two of the most relevant particle filters proposed in the literature namely, the Annealed Particle Filter, and the Partitioned Sampling Annealed Particle Filter.

The algorithms have been experimentally compared in the public dataset HumanEva-I by employing two body models with different complexities. Our work also analyzes the performance of the algorithms in hierarchical and holistic approaches, i.e., with and without partitioning the search space. Non-parametric tests run on the results have shown that: (i) the evolutionary algorithms employed outperform their particle filter counterparts in all the cases tested; (ii) they can deal with high-dimensional models thus leading to better accuracy; and (iii) that the hierarchical strategy surpasses the holistic one.

Keywords:
Markerless Motion Capture, Tracking, Particle filters, Evolutionary Algorithms

1. Introduction

Markerless Motion Capture (MMOCAP) is an emerging field with potential applications in areas like the animation industry [1], medical rehabilitation [2], and video surveillance [3], amongst others. The goal is to automatically determine the pose of a subject from the images of a video sequence. For that purpose, a human body model is rendered on the images so as to determine the model's configuration that best matches the input images. The sequential nature of the problem allows us to use the solutions obtained in one frame as the starting point for the search in the next one. In order to cope with three-dimensional ambiguities and occlusions, the most effective set up consists of several cameras simultaneously observing the scene from different points of view. Ultimately, the problem is formulated as a continuous optimization problem that seeks the best angles for the joints of the human body model given the information available in the previous and the current images.

MMOCAP is considered in the computer vision community as a difficult problem because of the many challenges that it presents. First, it is a high-dimensional problem. In order to obtain accurate results, the body model employed needs to have a suitable number of joints so as to adapt properly to all human poses. Second, the computing power is a limiting factor. The operations needed to evaluate solutions are computationally very expensive. Therefore, it is of paramount importance to obtain good solutions in as few iterations as possible. Finally, an appropriate balance is required between local and global search. While in most of the cases local search achieves good solutions, occlusion and ambiguities in the camera configuration might require the use of global optimization to discover the correct solution once the conflicting situation has finished.

Most of the solutions emerging from the computer vision community are particle filtering variants [4–7], which try to cope with the high dimensionality of the model. Some solutions partition the search space taking advantage of the hierarchical

*Corresponding author.
Email addresses: `eyeguas@uco.es` (Enrique Yeguas-Bolivar), `rmsalinas@uco.es` (Rafael Muñoz-Salinas), `rmedina@uco.es` (Rafael Medina-Carnicer), `ma1capoa@uco.es` (Angel Carmona-Poyato)

nature of the problem, [5, 6]. Other solutions tackle the problem by dividing the search into multiple stages, [4, 7]. In spite of the advances achieved, most of the proposed solutions suffer from the curse of dimensionality and they need to rely on simple human models (which lead to suboptimal tracking results), or require a high number of evaluations to achieve accurate results.

Over the past few years, the field of global optimization has been very active, producing different kinds of deterministic and stochastic algorithms for optimization in the continuous domain [8]. Among the stochastic approaches, Evolutionary Algorithms [9] offer a number of exclusive advantages that make them an attractive choice, e.g., robust and reliable performance, global and local search capability, little or no information requirement, etc. Despite of their advantages, there are few works in the literature devoted to test the performance of evolutionary algorithms on the MMOCAP problem, [10–12].

This work performs an experimental comparison of three relevant evolutionary algorithms namely Covariance Matrix Adaptation Evolutionary Strategy (CMAES) [13], Differential Evolution (DE) [14], and Particle Swarm Optimization (PSO) [15], with two popular variants of particle filters proposed for the MMOCAP problem namely Annealed Particle Filter (APF) [4] and Partitioned Sampling Annealed Particle Filter (PSAPF) [6]. The selection of these evolutionary algorithms is based on the results reported in the IEEE International Conferences on Evolutionary Computation (CEC) competitions on real parameter optimization [16], for which these three algorithms have continued to secure top ranking. With regards to the particle filters selected, the first one (APF) can be considered the base line algorithm taking into account the high number of cites obtained. The PSAPF has been selected for being considered the hierarchical version of the APF.

This work analyzes the behavior of the evolutionary algorithms using both holistic and hierarchical strategies so as to decide which strategy is preferable. Additionally, two different articulated models have been employed. The first one, comprised of 28 degrees of freedom (DOF), is similar to those used in most of the previous approaches in the literature. The second one, comprised of 39 DOF, aims at evaluating the capability of the tested algorithms to deal with more realistic (and thus complex) body models.

The experiments have been conducted on nine sequences of the HumanEva-I dataset [17], and analyzed by means of a nonparametric statistical analysis [18]. The results obtained show that:(i) the evolutionary algorithms tested provide significantly better results than particle filters; (ii) the 39-DOF body model employed in this work obtains better results than the other one; and (iii) the hierarchical strategy surpasses the holistic one in all the evolutionary algorithms applied. Additionally, our experiments determine the minimum number of evaluations required for an optimal trade-off between precision and speed for the different algorithms in the experiments.

The remainder of this paper is structured as follows. Section 2 revises the related work. Section 3 formulates the problem of pose estimation, describes the body models, cost function and evaluation strategies. Section 4 introduces the ba-

sis of the particle filters and the evolutionary algorithms employed. Finally, Section 5 explains the experimentation carried out, while Section 6 draws some conclusions.

2. Related works

Most of the solutions proposed for the MMOCAP problem fall into two main categories: holistic and hierarchical approaches. The former employs a global optimization approach to fit the model's parameters. As a consequence, the computational requirements are high and the search method may suffer from premature convergence because of the high dimensionality. On the other hand, hierarchical strategies exploit the underlying hierarchical structure of the articulated model assuming that body parts can be localized independently from each other. Then, the problem is divided into smaller problems that can be more easily solved. However, hierarchical approaches have several drawbacks. First, the optimal partitioning may not be obvious (e.g. arms first or legs first?) and it may change over time (e.g. when the legs cross during walking). Second, while partitioning might not make a difference for noise-free simulation data, in practice an incorrectly estimated early partition due to noisy data may irrevocably mislead the outcome, [6].

2.1. Computer Vision approaches

The first solutions emerging from the computer vision community consist in the use of particle filters. In particular, the Condensation algorithm is the most prevalent of such algorithms and has been widely employed for the tracking task, [19]. However, when applied to the MMOCAP problem, it has been repeatedly shown that it suffers from the *curse of dimensionality*. Therefore, Deutscher and Reid proposed the APF [4], which combines the ideas of the Condensation and the Annealed search so as to improve the tracking results. With the aim of reducing the complexity of the search in many dimensions, the particles are evaluated in layers. In the first layers, the objective function are smoothed so as to allow escaping from local minima. In the last layers, the objective function is more peaked thus concentrating on exploiting the best solutions. Corazza *et al.* propose also a custom version of adapted fast simulated annealing [20] for body tracking using as input data a visual hull reconstruction and an a priori model of the subject.

Another popular approach for tracking articulated objects is the use of Partitioned Sampling (PS) [5]. The technique was initially employed for tracking several objects using particle filters, but then successfully applied to hand tracking. Unlike the APF, PS imposes a strong partition of the search space. As previously indicated, the main problem consists in determining the optimal partitioning. Bandouch *et al.* proposed the PSAPF [6] as an attempt to combine the strengths of PS and the Annealed Search. To do so, they incorporate the APF within a PS framework by applying an appropriate weighted resampling in each sub-space. As they report, they are able to cope with high-dimensional models, but at the cost of employing a very high number of evaluations per frame (around 8000). Recently

Gall *et al.* [7] have proposed a multi-layer framework that combines stochastic optimization, filtering, and local optimization search. Their approach runs on several stages. First, a simulated annealing is employed. And then, the solutions obtained are refined by filtering and local optimization. Although they report good tracking results, the main drawback of their approach is the limitation imposed to the body model tested (only 28 DOF).

Chang *et al.* [21] proposed a progressive particle filter. The main idea was to combine a standard particle filter with the mean shift strategy [22] and a hierarchical search. However, their solution was only tested on video sequences with a single camera in a non-public dataset. It is not clear if their approach would work on sequences with multiple views.

In spite of the advances achieved over the last years, mobility limitations often are imposed to the body models employed so as to obtain reasonable performance in manageable computing times. For instance, there are works [4, 7, 10, 21] which employ models with no more than 32 DOF and assume no mobility in dorsal spine, hands and feet. This simplification of the human anatomy allows a tractable computation of the model while achieving acceptable results for some applications. However, some other applications require a more precise modelling of the human body so as to measure biomechanical parameters [23–26].

2.2. Evolutionary Computation approaches

Many real-world problems, such as the MMOCAP problem, may be formulated as optimization problems of parameters with variables in continuous domains. Over the past few years, the field of global optimization has been very active, producing different kinds of evolutionary algorithms for continuous optimization problems. Such algorithms have been widely applied, showing better performance than other approaches, to diverse scientific and engineering problems [8]. Nonetheless, few authors have applied the ideas from Evolutionary Computation to the MMOCAP problem.

John *et al* [10] apply PSO to the MMOCAP problem using a 31-DOF articulated model with great success. As they report, their approach outperforms the results of APF and PSAPF. The main advantages of the PSO algorithm become particularly evident when tracking fast movements, since it has demonstrated a good performance without requiring any motion prior. Shen *et al.* [11] propose their own evolutionary algorithm and test it with a 33-DOF model. The experiments performed show that their approach compares favourly to the APF algorithm. One of the weaknesses of their work is that their method is not evaluated in a public dataset. Zhao and Liu [12] propose a Hierarchical Annealed Genetic Algorithm to infer the three-dimensional pose from a single monocular camera. The main novelty of their approach is the use of Principal Component Analysis (PCA) to reduce the dimensionality. To that end, the algorithm is trained to be employed in specific motions that are learnt in advance. In this work, however, we are interested in evolutionary algorithms that can be employed without prior knowledge of the motions to be tracked.
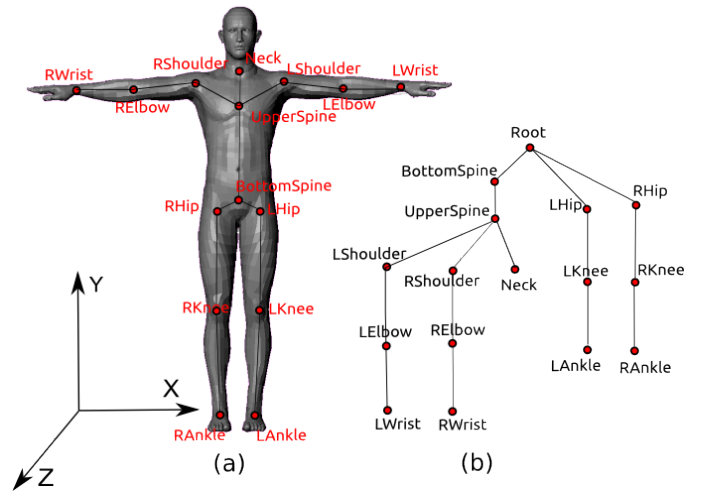

Figure 1: **Body model employed: (a) Skin and Skeleton Models.** The skin is a 3D model representing the surface of the body while the skeleton represents the internal structure of the articulations (joints and bones). **(b) Hierarchical Structure of the Skeleton Model.** Each node represents a joint which is subject to rotations in the three axes. The transformation of a node affects all its children.

The goal of this work is to perform a comparative study of the performance of the evolutionary algorithms and particle filters paradigms when applied to the MMOCAP problem. For that purpose, we have selected three of the most relevant evolutionary approaches (PSO, CMAES and DE), and the two most widely employed particle filters (APF and PSAPF). As we show in the experimental section, the evolutionary approaches tested are generic optimization algorithms that provide significantly better results than their particle filter counterparts.

3. Problem formulation

Markerless Motion Capture (MMOCAP) consists in determining the pose of a subject from a video sequence. For that purpose, either a specific or a generic articulated body model of the subject is matched against the images. In particular, we are concerned with the problem of estimating the three-dimensional pose of the subject. Therefore, the subject is recorded using a set of synchronized video cameras whose relative position to a common reference system is obtained by calibration. As a consequence, it is possible to determine the three-dimensional projection of the model in the images.

3.1. Body models

The body model employed is generally comprised of an outer model or *skin* and an inner model or *skeleton*. In this work, the skin consists of a set of meshes representing the surface of the body while the skeleton represents the internal structure of the articulations. The skeleton is modelled as a hierarchical structure where each node represents a joint which is subject to rotations in the three axes (R_x, R_y and R_z). It is employed to apply the body movements to the skin model in such a way that the transformation of a node affects all its children.

Fig. 1(a) shows the skin and skeleton models employed in our work, whereas Fig. 1(b) depicts its hierarchical structure. As can be seen, the hierarchical model has a root node (root joint) from which the rest depends. It is employed to define the global rotations and translations of the model. In total, our model is comprised of 16 joints, so that a complete transformation of such model is defined by 3 translation components (T_x, T_y and T_z) plus 16×3 rotations (R_x^j, R_y^j, R_z^j), $j = 1, \ldots, 16$, i.e., a total of 51 parameters (DOF) constituting the dimension D of the problem.

$$x_{51DOF} = \{T_x, T_y, T_z, R_x^1, R_y^1, R_z^1, \ldots, R_x^{16}, R_y^{16}, R_z^{16}\} \in \mathbb{R}^{D=51}.$$

However, because of the biomechanical structure of the human body, not all these rotations are valid and the model can be simplified. Inspired in the work by John *et al* [10], Table 1 defines the valid axes of rotation for each joint of our model. So, the valid transformation of the model is then defined by 39 DOF which may be represented by the vector:

$$x_{39DOF} = \{T_x, T_y, T_z, R_x^1, R_y^1, R_z^1 \ldots, R_x^{16}, R_y^{16}\} \in \mathbb{R}^{D=39}.$$

The higher the number of DOF of the proposed model the better the adaptation to the human anatomy. The 39-DOF model employed here has higher dimensionality than those in previously cited works. For instance, Deutscher and Reid [4] or John *et al* [10] employed models, with 31 and 29 DOF respectively, which impose mobility limitations in parts such as dorsal spine, hands and feet. To assess the importance of using body models that adapt well to the human anatomy, we compare the tracking results of our 39-DOF model with a simplified version that resembles those presented in previous literature [4, 7, 10, 21]. This simplified version is created by removing the four joints of wrists and ankles, plus the one labelled *UpperSpine* (see the joints marked with an asterisk in Table 1). As a consequence, the simplified model has a total of 28 DOF.

3.2. Model evaluation

The goal of MMOCAP is to estimate the most likely model configuration x^t at each time step t, given the set of images captured. For that purpose, it is assumed that there has been defined a fitness function $f(x)$ which indicates the likelihood of a model configuration to be correct. For each new frame, the minimization procedure relies on the results obtained in the previous one so as to improve the results. For the first frame, an initial body configuration x^0 is provided.

Many fitness functions have been proposed in the literature, but the ones based on silhouettes have shown to provide an appropriate trade-off between robustness and speed. The main idea is to match the projection of a model configuration x (silhouettes) to the foreground information obtained by background subtraction.

Next, we will define the silhouettes-based fitness function used in our research. For that purpose, let us denote by

$$\mathcal{I}^t = \{\mathcal{I}_v^t \mid v = 1, \ldots, V\}, \tag{1}$$

Joint	DOF	
Root	$T_x, T_y, T_z, R_x, R_y, R_z$	6
BottomSpine	R_x, R_y, R_z	3
UpperSpine *	R_x, R_y, R_z	3
LShoulder	R_x, R_y, R_z	3
LElbow	R_y	1
LWrist *	R_y, R_z	2
RShoulder	R_x, R_y, R_z	3
RElbow	R_y	1
RWrist *	R_y, R_z	2
Neck	R_x, R_y, R_z	3
LHip	R_x, R_y, R_z	3
LKnee	R_x	1
LAnkle *	R_x, R_y	2
RHip	R_x, R_y, R_z	3
RKnee	R_x	1
RAnkle *	R_x, R_y	2
	Total	39

Table 1: Biomechanically valid DOF (rotation axes) for the human model's joints. The reduced model is created removing rotation axes in the joints marked with an asterisk.

bib:hansen06 the set of images captured at time instant t with the V cameras available.

In an initial phase, a background model which captures the color statistics of each pixel is created. Then, using background subtraction techniques, the foreground images are obtained indicating which pixels belong to the moving objects in the scene. Let us denote by

$$\mathcal{F}^t = \{\mathcal{F}_v^t \mid v = 1, \ldots, V\}, \tag{2}$$

to the set of foreground images, so that a pixel $\mathcal{F}_v^t(p)$ is 1 if it belongs to the foreground and 0 otherwise.

Finally, we shall denote by

$$\mathcal{M}(x) = \{\mathcal{M}_v(x) \mid v = 1, \ldots, V\}, \tag{3}$$

to the projections of the body model configuration x in the camera images. Likewise, a pixel $\mathcal{M}_v(x, p)$ is 1 if it belongs to the model's silhouette and 0 otherwise.

Fig. 2 helps to clarify the above explanation. While Fig. 2(a) shows an image captured by a camera ($\mathcal{I}_v^t$), Fig 2(b) shows its corresponding foreground image ($\mathcal{F}_v^t$). Finally, Fig. 2(c) represents the projection of a model configuration in the image $\mathcal{M}_v(x)$.

Using the above defined concepts, a model can be evaluated by matching the degree of overlap of its projected silhouette and the foreground image. Thus, let us define the evaluation function as:

$$f_v^t(x) = \frac{1}{2|\mathcal{D}(\mathcal{M}_v(x))|} \sum_{p \in \mathcal{D}(\mathcal{M}_v(x))} \mathcal{M}_v(x, p) - \mathcal{F}_v^t(p) +$$

$$\frac{1}{2|\mathcal{D}(\mathcal{F}_v^t)|} \sum_{p \in \mathcal{D}(\mathcal{F}_v^t)} \mathcal{F}_v^t(p) - \mathcal{M}_v(x, p) \tag{4}$$
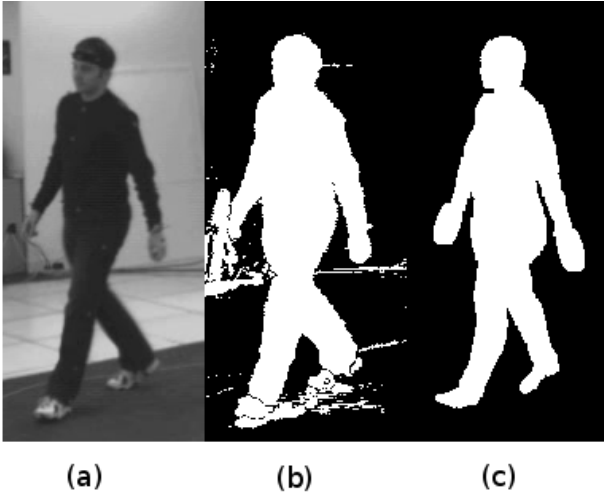
Figure 2: **Images taking part in the model evaluation: (a) Camera image** $\mathcal{I}_v^t$, captured at time instant t with camera v. **(b) Foreground image** $\mathcal{F}_v^t$, obtained from $\mathcal{I}_v^t$ by background subtraction techniques and showing in white pixels belonging to the moving objects in the scene. **(c) Model silhouette** $\mathcal{M}_v(x)$, resulting of the projection of the body model (skin in Fig. 1(a)) in $\mathcal{I}_v^t$. White pixels belong to the model's silhouette.

where the function $\mathcal{D}(\cdot)$ indicates only these pixels with value 1. Thus, the first term of Eq. 4 accounts for these model points that project on foreground points, i.e., it decreases as the degree of overlap between the model and the mask increases. On the other hand, the second term of the equation accounts for the pixels of the foreground image that are not covered by the model's projection. Consequently, the function $f_v^t(x) = 0$ when the model projection fits exactly the foreground mask. On the contrary, $f_v^t(x)$ tends to 1 as the degree of overlap decreases.

Due to illumination changes and color similarities between the subject and the background, it is unlikely to achieve a perfect match between the model and the foreground images (see Fig 2(b)). The use of multiple cameras helps not only to alleviate these problems but also to infer the three-dimensional configuration of the subject. The evaluations of the model in all the cameras are fused as:

$$f^t(x) = \frac{1}{V} \sum_v f_v^t(x). \qquad (5)$$

Therefore, values of Eq. 5 near to 0 indicate that x is a good solution and values near 1 indicate that x is a poor solution.

3.3. Evaluation strategies

As previously indicated, the representation of the problem has a hierarchical nature (see Fig. 1(b)). It means that the modification of a node affects its children but not its parents. As a consequence, there are two strategies that can be applied to solve the problem: holistic and hierarchical.

In the holistic approach, all variables are optimized jointly. This approach is, in theory, the most appropriated one since it makes no assumptions about the independence of the body parts

and it is more robust to errors. In practice, however, the high dimensionality of the problem makes the holistic approach inappropriate in situations having real-time requirements. It must be considered that the elevated computing time required for evaluating a solution imposes strong restrictions on the maximum number of evaluations for some applications. The main drawback of the holistic approach, then, is the exponential growth of the search space with the number of variables. In addition, since changes in the upper nodes of the structure affects to their children, good solutions obtained for the lower nodes in early stages of the search might be discarded later by changes in one of their parents.

On the other hand, the hierarchical search takes advantage of that situation by considering the existence of independent branches in the human body, namely the limbs and head. Therefore, it is possible to apply a hierarchical search in which there are independent search processes working on smaller spaces. However, hierarchical approaches have several drawbacks too. First, the optimal partitioning may not be obvious and it may change over time. Second, incorrect estimations in a node (due to noise or occlusion) affect to all its children, i.e. an error in the estimation of the first node compromises the rest of the nodes irrevocably.

This paper aims at evaluating the performance of the algorithms proposed using the two strategies. In particular, we will test the hierarchical strategy using a six-step approach which we consider that provides an appropriate balance between global and local search. The six steps are: the *Root* (translation and rotation) first, then torso and head, and finally, the branches corresponding to the limbs, which are optimized independently.

4. Algorithms Evaluated

This section provides an overview of the algorithms employed: particle filters and evolutionary algorithms.

4.1. Particle filters

Particle filtering techniques are widely employed for estimating the state of a dynamic system x^t from sequential observations of it [19, 27–29]. For that purpose, they use a weighted set of particles $\{(x_i^t, \pi_i^t)\}$, where x_i^t represents a possible state of the system, and π_i^t is a non negative numerical factor called *importance weight* which represents an estimation of the observation density, i.e., $\pi_i^t \propto 1 - f(x_i^t)$. Particle filters are especially interesting because they can deal naturally with systems where both the posterior density and the observation density are non-Gaussian.

Particle filters have been proposed for articulated body tracking, including the APF which is a holistic approach, [4], and PSAPF constituting a hierarchical approach, [6]. A brief introduction to these two techniques, is given below.

4.1.1. Annealed Particle Filter

The most prevalent algorithm in the Computer Vision field is the Condensation algorithm [19]. However, the algorithm is known to have difficulty with high-dimensional state spaces and

unknown motion models. For that reason, Deutscher and Reid proposed the APF [4], as an extension to the original algorithm aimed to deal with problems of high dimensionality and specifically with the MMOCAP problem. Since then, it has become one of the most relevant algorithms in its field [3].

The outline of the APF is presented in Fig. 3. At the beginning, the algorithm is provided by an initial sample set $X(0)$ of N particles, where all of them are equally weighted. At each iteration, the algorithm uses the sample set $X(t-1)$ to create a new one $X(t)$. For that purpose, each iteration is divided in a total of L steps, or *layers*. In each layer, the algorithm selects N particles with replacement. The idea is that good particles are selected with a higher probability. Then, each particle is evaluated using the fitness function modified by an exponent β_l that increases at each layer, i.e., $\beta_1 > \beta_2 > \beta_L$. The exponent plays the role of the annealing procedure. In the first layers, an small exponent creates a soft search surface causing a high exploration. Then, the exponent increases so that the search space becomes more peaked thus forcing exploitation.

Perturbation of particles is made using the noise matrix $\mathbf{B}_t$. As suggested by the authors, a good choice for $\mathbf{B}_t$ is a normal distribution with zero mean and covariance:

$$\mathbf{B}_t \propto \frac{1}{N} \sum_i (x_i^t - \hat{x}_i^t) \cdot (x_i^t - \hat{x}_i^t)^T, \qquad (6)$$

proportional to the population's covariance. In Eq. 6, the parameter $\hat{x}_i^t$ represents the mean of all particles. The idea behind the equation is as follows. If a variable exhibits a low variance among the population, then, a good solution has already been obtained for it. As a consequence, it is preferable not to alter the good values already achieved. However, variables with high variance correspond to parts of the space for which a good solution is not yet obtained. Then, a higher perturbation is employed so as to seek better solutions.

The algorithm also provides a method to automatically obtain the values of β_l from the particles evaluated. For that purpose, it is employed the concept of *particle survival rate*

$$\delta = \frac{1}{N \sum_{i=1}^{N} \pi_i^2} \qquad (7)$$

which is the proportion of particles that pass to the next layer. For instance, $\delta = \frac{1}{N}$ if the same particle is always chosen in the selection process; and $\delta = 0.5$ if half of the particles in one layer are selected to the next one. The idea is then to set a desired δ and choose β_l for the next layer as:

$$\arg \min_{\beta_l} \left| \frac{1}{N \sum_{i=1}^{N} \pi_i^2} - \delta \right|. \qquad (8)$$

In our experiments, it is always used $\delta = 0.5$ as stated in the original paper. For further information, the interested reader is referred to the work of Deutscher and Reid [4].

Considering the pseudocode in Fig. 3, the run-time complexity for the worst-case scenario of the APF algorithm is linear with respect to the dimensionality of the problem.

4.1.2. Partitioned Sampling Annealed Particle Filter

Partitioned Sampling (PS) is a technique proposed to overcome the poor performance of particle filters in high-dimensional problems [5]. Initially, it was proposed for hand tracking, but latter applied to MMOCAP. PS is a generic term for a strategy which consists of dividing the state space into two or more partitions, and sequentially applying the dynamics for each partition followed by an appropriate weighted resampling operation. The prerequisites for using PS in a problem are: the configuration space can be partitioned as a cartesian product, the dynamics of one partition do not influence the dynamics of the hierarchically preceding partition, and the weighting function can be evaluated locally for each partition. All these conditions are met in the MMOCAP problem.

The work of Bandouch *et al.* [6] consists in combining the APF and PS algorithms. To that end, the authors partition the search space, and then employ an APF in each partition, instead of using a standard particle filter. The dimensional complexity of the algorithm is also linear. As a consequence, the PSAPF can be seen as the hierarchical version of the APF algorithm. In this work, we are employing the 6-steps hierarchy explained in Section 3.3.

4.2. Evolutionary algorithms

Over the past few years, an increasing interest has arisen in solving continuous optimization problems using evolutionary algorithms, [30]. This family includes real-coded genetic algorithms, evolution strategies, evolutionary programming, memetic algorithms, swarm intelligence, and differential evolution. These algorithms are able to adapt to the dimensionality of the search space and have the ability of being scalable for high-dimensional problems which are essential requirements for the MMOCAP problem.

$X(0) = \{(x_i^0, \pi_i^0)| i = 1, \ldots, N\}|\ \pi_i^0 = \frac{1}{N}$
Create a new sample set $X(t)$ from $X(t-1)$ as
for $l = 1$ to L **do**
   Select N samples from the set $X(t, l-1)$ with replacement so as to obtain $X(t, l)$
   Propagate each selected sample:

$$x_i^t = x_i^t + \mathbf{B}^t$$

   Evaluate the importance of each sample

$$\pi_i^t = (1 - f(x_i^t))^{\beta_l}.$$

   Normalise so that $\sum_{i=1}^{N} \pi_i^t = 1$.
   Recalculate β_l so that $\delta = 0.5$.
end for
Estimate the mean of the state $X(t)$ as:

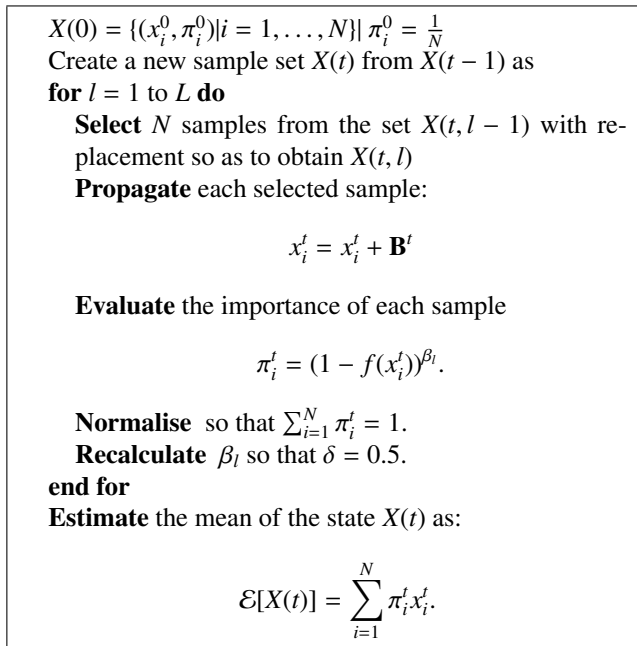$$\mathcal{E}[X(t)] = \sum_{i=1}^{N} \pi_i^t x_i^t.$$

Figure 3: Annealed Particle Filter.

Some of the algorithms which have emerged as very competitive forms of evolutionary computing are CMAES representing evolution strategies, [13], DE which is itself an evolutionary paradigm, [14], and PSO representing swarm intelligence, [15]. We can also observe that no other single search paradigms were able to secure competitive rankings in all CEC competitions, [16].

In this section, we give a brief description of each of these three evolutionary algorithms considered in this work for being applied to the MMOCAP.

4.2.1. Covariance Matrix Adaptation Evolutionary Strategy

The Covariance Matrix Adaptation Evolutionary Strategy (CMAES) is an evolutionary algorithm (belonging to the family of evolution strategies) for difficult non-linear non-convex optimization problems in continuous domains. Its main applications are oriented to unconstrained or bounded constraint optimization problems, and search space dimensions between three and a hundred. Its main objective is to adapt the full covariance matrix of a normal search distribution. The method is feasible on non-separable and/or badly conditioned problems and on non-smooth and even non-continuous problems, as well as on multimodal and/or noisy problems. It has been demonstrated to be reliable and highly competitive in local, [31], and global optimization, [32].

Figure 4 outlines the complete algorithm whose complexity is of the order $O(D)$, [31, 33]. For more details, the interested reader is referred to the work by Hansen [13]. Default strategy parameters values are given in Table 2.

The general principle of the algorithm is to obtain, for generation $g + 1$, λ offspring sampled independently according to a multi-variate normal distribution

$$x_k^{(g+1)} \sim \mathcal{N}\left(m^{(g)}, \left(\sigma^{(g)}\right)^2 C^{(g)}\right) \text{ for } k=1, \ldots, \lambda, \quad (9)$$

where $\mathcal{N}(m, C)$ denotes a normally distributed random vector with mean m and covariance matrix C.

The definition of the complete iteration step supposes to calculate $m^{(g+1)}$, $\sigma^{(g+1)}$ and $C^{(g+1)}$ for the next generation $g+1$. The correspondent stages are respectively selection and recombination for the mean, step size control for the standard deviation and adaptation for the covariance matrix. The two last stages are also called adaptation of the mutation parameters.

In the first stage, the new mean $m^{(g+1)}$ of the search distribution is a weighted average of the μ best offspring

$$m^{(g+1)} = \sum_{i=1}^{\mu} w_i x_{i:\lambda}^{(g+1)}, \quad (10)$$

where the positive weights $w_i \in \mathbb{R}$ sum to one. Recombination is done by taking a weighted sum of μ individuals and selection by choosing $\mu < \lambda$ and/or assigning different weights.

To control the step size, $\sigma^{(g+1)}$, an evolution path $p_c^{(g+1)}$ (a whole path taken by the population over a number of generations, [31]) is utilized. In general, the expected length of that evolution path depends of its direction and a conjugate evolution path $p_\sigma^{(g+1)}$ is constructed

$$p_\sigma^{(g+1)} = (1 - c_\sigma)p_\sigma^{(g)} +$$
$$\sqrt{c_\sigma(2 - c_\sigma)\mu_{eff}} C^{(g)^{-\frac{1}{2}}} \frac{m^{(g+1)}-m^{(g)}}{\sigma^{(g)}} \quad (11)$$

where $c_\sigma < 1$ is the learning rate for the cumulation (summation of successively selected mutation steps) of the step size control and μ_{eff} the variance effective selection mass (defined by the weights w_i).

The update rule for $\sigma^{(g)}$ obeys to a comparison between $\|p_\sigma^{(g+1)}\|$ with its expected length $E\|\mathcal{N}(0, I)\|$

$$\sigma^{(g+1)} = \sigma^{(g)} exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|p_\sigma^{(g+1)}\|}{E\|\mathcal{N}(0, I)\|} - 1\right)\right) \quad (12)$$

where $c_\sigma < 1$ is the learning rate for the cumulation, $d_\sigma \approx 1$ is called the damping parameter and scales the change magnitude of $\sigma^{(g)}$, and the factor c_σ/d_σ is based on in-depth investigations of the algorithm, [13].

Finally, in the third stage, the covariance matrix $C^{(g)}$ is adapted by the μ weighted difference vectors between the recent parents and $m^{(g)}$ and by the evolution path $p_c^{(g+1)}$.

$$p_c^{(g+1)} = (1 - c_c)p_c^{(g)} +$$
$$H_\sigma^{(g+1)} \sqrt{c_c(2 - c_c)} \frac{\sqrt{\mu_{eff}}}{\sigma^{(g)}} \left(m^{(g+1)} - m^{(g)}\right), \quad (13)$$

where c_c is the learning rate for cumulation (in the same sense as in the step size control stage) of the covariance matrix and $H_\sigma^{(g+1)}$ is the Heaviside function, which stalls the update of p_c if $\|p_\sigma\|$ is large.

The adaptation rule is defined as follows

$$C^{(g+1)} = (1 - c_{cov})C^{(g)} + c_{cov}\frac{1}{\mu_{cov}}p_c^{(g+1)}\left(p_c^{(g+1)}\right)^T +$$
$$\left(c_{cov} - \frac{c_{cov}}{\mu_{cov}}\right)\sum_{i=1}^{\mu}\frac{w_i}{\sigma^{(g)^2}}\left(x_{i:\lambda}^{(g)} - m^{(g)}\right)\left(x_{i:\lambda}^{(g)} - m^{(g)}\right)^T \quad (14)$$

where c_{cov} is the learning rate for the covariance matrix update and μ_{cov} is the parameter for weighting between the two last terms in the sum (called, respectively, rank-one and rank-μ update).

4.2.2. Differential Evolution

Differential Evolution (DE), [14], is a powerful population-based stochastic function optimizer. The algorithm maintains a population of N vectors representing candidate solutions for the problem at every generation g: $\{x_{i=1...N}^{(g)}\}$. The initial population comprises random vectors within the search space. The main loop of the algorithm consists of three steps: mutation, crossover and selection.

First, the mutation operation creates the donor vector $V_i^{(g)}$ as a linear combination of some members of the population,

$$V_i^{(g)} = x_{r_1} + F \cdot (x_{r_2} - x_{r_3}) \quad (15)$$

where $r_1, r_2, r_3 \in [1, N]$ are mutually different integers, not considering i, and F is a positive control parameter for scaling the difference vector.

7

Figure 4: Covariance Matrix Adaptation Evolutionary Strategy.

Table 2: Default parameter settings for CMAES.

Parameter	Value
λ	$\lfloor 4 + 3 \ln D \rfloor$
μ	$\lfloor \lambda/2 \rfloor$
w_i	$\dfrac{\ln(\mu + 1) - \ln i}{\sum_{j=1}^\mu (\ln(\mu + 1) - \ln j)}$ for $i = 1, \ldots, \mu$
μ_{eff}	$\dfrac{1}{\sum_{i=1}^\mu w_i^2}$
c_σ	$\dfrac{\mu_{eff} + 2}{D + \mu_{eff} + 3}$
d_σ	$1 + 2\max\left(0, \sqrt{\dfrac{\mu_{eff} - 1}{D + 1}} - 1\right) + c_\sigma$
c_c	$\dfrac{4}{D + 4}$
μ_{cov}	μ_{eff}
c_{cov}	$\dfrac{1}{\mu_{cov}} \dfrac{2}{(D + \sqrt{2})^2} +$ $\left(1 - \dfrac{1}{\mu_{cov}}\right) \min\left(1, \dfrac{2\mu_{eff} - 1}{(D + 2)^2 + \mu_{eff}}\right)$

After mutation, a trial vector $U_i^{(g)}$ is generated from $x_i^{(g)}$ and $V_i^{(g)}$ in the crossover, which is defined as follows

$$u_i[j] = \begin{cases} v_i[j] & \text{if } (rand[0, 1) \leq CR) \text{ or } (j = unif[1, D]); \\ x_i[j] & \text{otherwise.} \end{cases} \tag{16}$$

where $u_i[j], v_i[j], x_i[j]$ are, respectively, the j-th components of the vectors $U_i^{(g)}, V_i^{(g)}$ and $x_i^{(g)}$, $CR \in [0, 1)$ is the crossover rate controlling the fraction of parameter values chosen from the donor vector, and $unif[1, D]$ is an uniform-randomly sampled index of a vector component in the set $\{1, \ldots, D\}$.

Finally and after evaluating trial vectors, the selection operation occurs. For each member in the population, the best vector between $U_i^{(g)}$ and $x_i^{(g)}$ is chosen to become the new member of the population.

The DE strategy applied in this work, following the work by García-Martínez *et al* [34], proposes two different mechanisms to emphasize the proper selection of vectors for each role in crossover and mutation operations: role differentiation and malleable mating. Four different roles are clearly identified: placing (P), leading (L), correcting (C) and receiving (R) vectors. A placing vector $x_i^{P,(g)}$ points to the search space region where the donor vector is sampled. Leading and correcting vectors, $x_i^{L,(g)}$ and $x_i^{C,(g)}$ respectively, define the search direction used to sample the donor vector within the local neighborhood of the placing vector. The receiving vector $x_i^{R,(g)}$ is the target vector that is crossed over with the donor to generate a new trial solution.

The role differentiation mechanism defines the attributes for which vectors are selected for each role. Malleable mating allows placing vectors to adapt their mating trends to ensure some similarity relations with the leading and correcting vectors. Assortative mating is the natural occurrence of mating between individuals of similar phenotype more or less often than expected by chance. Mating between individuals with similar phenotype more often is called positive assortative mating and less often is called negative assortative mating, [35]. This way, each solution incorporates two new parameters α^L and α^C that control the applied mating mechanism for selecting, respectively, leading and correcting vectors. The aim is to yield better results in high dimensionality problems.

Fig. 5 outlines the DE pseudocode incorporating role differentiation and malleable mating. The complexity of the proposal is linear with regards to the dimension of the problem, i.e. $O(D)$, due to recombination, solution copying and distance calculations.

4.2.3. Particle Swarm Optimization

Particle Swarm Optimization (PSO), [15], is a population-based stochastic optimization technique originally inspired in the way crowds of individuals (bird flocking or fish schooling) move towards predefined objectives. Individuals in the population try to move towards the fittest position known to them and to their informants. The objective is to maximize or minimize a fitness function f.

In the algorithm, each potential solution to the problem is given by a particle position and the population of N particles is called swarm. Each particle position x_i is updated each generation g by means of the equation,

Figure 5: Differential Evolution with role differentiation and malleable mating.

Figure 6: Particle Swarm Optimization.

pling step while gradually guiding the swarm from a global to a more local search, [10],

$$\omega(c) = \frac{\omega_{max}}{e^c}, \Delta c = \ln(10\omega_{max})/g_{max} \qquad (19)$$

where $c \in [0, \ln(10\omega_{max})]$ and g_{max} is the desired number of inertia weight changes.

Fig. 6 presents the algorithm in pseudocode. The dimensional complexity of the algorithm is linear, $O(D)$. This is due to the velocity and position update rules and the evaluation of particles.

5. Experimental Results

This section explains the experiments carried out to compare the algorithms selected, using both holistic and hierarchical strategies. As instances of particle filters, the APF and the PSAPF have been chosen for the holistic and hierarchical approaches, respectively. With respect to the evolutionary algorithms, their hierarchical approaches will be denoted as H-CMAES, H-DE and H-PSO in the remainder of the paper. These hierarchical variants of the evolutionary algorithms consist in running the corresponding algorithms sequentially in each part of the 6-step hierarchy (see Section 3.3).

The goal of our experimentation is four-fold. First, we are demonstrating that evolutionary algorithms achieve better results than particle filters for the MMOCAP problem. Second, we are showing that the 39-DOF model proposed can be effectively managed by the evolutionary algorithms obtaining better accuracy than its reduced version of 28 DOF. Third, we are determining the effect of each strategy, holistic and hierarchical, having the same parameter settings for the algorithms. Finally, we are obtaining the number of evaluations for the evolutionary algorithms which gives a suitable trade-off between precision and speed.

Aiming at providing a clear explanation of the experimentation carried out and the results obtained, this section is structured as follows. To begin with, Section 5.1 explains the details

where the factor $v_i^{(g+1)}$ is the velocity of the particle. Velocity constraint is one of the main mechanisms used for controlling the movement of particles through the search space and for balancing the trade-off between exploitation and exploration in the algorithm.

In our case, the inertia weight ω has been used as velocity constraint mechanism, [36]. This parameter controls the trade-off between global and local search. Then, a low inertia value provides the algorithm with exploitation capability and a high inertia promotes the exploration. This way, at the beginning of the search a high inertia ($\omega = \omega_{max}$) is introduced which decreases until reaching the lowest value ($\omega = \omega_{min}$). The velocity update rule stands as follows

$$v_i^{(g+1)} = \omega \cdot v_i^{(g)} + \varphi_1 \cdot r_1 \cdot (p_i^{(g)} - x_i^{(g)})$$
$$+ \varphi_2 \cdot r_2 \cdot (b_i^{(g)} - x_i^{(g)}) \qquad (18)$$

where $p_i^{(g)}$ is the i-particle best solution found so far, $b_i^{(g)}$ is the global best particle or leader that the entire swarm has ever generated, φ_1, φ_2 are specific parameters which control the relative effect of the individual and global best solutions for particles and r_1, r_2 are random values in [0,1]. The two last terms in the sum influence, respectively, the social and cognition components of the swarm behavior, [15].

The change over time for the inertia weight is modelled with an exponential function which allows us to use a constant sam-

$$x_i^{(g+1)} = x_i^{(g)} + v_i^{(g+1)} \qquad (17)$$

of the HumanEva-I dataset employed [17]. Then, Section 5.2 provides an overview of the experimental set-up and Section 5.3 shows the quantitative and some visual results. Finally, Section 5.4 performs a comparative study of the algorithms by employing a series of non-parametric statistical tests.

5.1. HumanEva Dataset

Our experimentation has been carried out using the HumanEva-I dataset, [17], which has been actively used in the community in the last years, [37–39]. It contains 7 calibrated video streams (4 grayscale and 3 color) with 4 subjects performing 6 common actions (e.g. walking, jogging, gesturing, etc.). During the recordings, the subjects worn reflective markers that were captured by a motion capture system. The markers were placed at key positions of the anatomy (such as the head, shoulders, knees, etc.) so as to be able to obtain the three-dimensional pose of the subjects in each frame.

Since the database contains a large number of video sequences, our evaluation has been performed only in a subset of it. In particular, the sequences walking, gesturing and boxing of subjects $S1$, $S2$ and $S3$ have been employed as benchmark. In total, we have worked on more than 8300 video frames (with 7 images each) in 9 different sequences. Since for some of the frames no ground truth data is available, evaluation is performed on a total of 6504 different frames.

A three-dimensional model of each person has been created using the *makehuman* software [40]. For each video sequence, the model has been manually initialized to fit the subject in the first frame. Given the model in its initial position, we added points to the skin model corresponding to the locations of the reflective markers. Therefore, the error in subsequent frames can be obtained as the distance from these points to their ground-truth positions. Ground truth motion of the body in HumanEva-I dataset was captured, as commented previously, using a commercial motion capture system, [17], and it is used as the optimum solution in each frame of the sequences tested. Therefore, we can measure the error of the different algorithms in the MMOCAP process over each frame of the sequence. As proposed by the creators of HumanEva, the error metric employed is the averaged absolute distance between the real positions of the n markers being tracked, X, and their estimated positions $\hat{X}$:

$$d(X, \hat{X}) = \frac{1}{n} \sum_{i=1}^{n} \|x_i - \hat{x}_i\|. \tag{20}$$

Eq. 20 provides an error measure in a single frame of the sequence. It is employed to calculate the tracking error of a complete sequence as the average of all its frames.

5.2. Experimental setup

In order to analyze the algorithms' performance in relation to the number of evaluations and dimensionality (DOF) of the models, each pair model-algorithm has been tested using {500, 1500, 3000} evaluations of the fitness function, presented in Eq. 5, per frame. The dimensional complexity of the different algorithms is linear (see Sections 4.1 and 4.2). The computing

Table 3: Parameter settings for the algorithms tested.

Algorithm	Parameters					
CMAES &	$\sigma^{(0)}$	$\lambda, \mu, w_{i...\mu}, c_\sigma, d_\sigma, c_c, \mu_{cov}, c_{cov}$				
H-CMAES	0.5	Generated from the dimension $D = DOF$. See Table 2.				
PSO &	N	φ_1	φ_2	ω_{max}	ω_{min}	
H-PSO	10	2.0	2.0	2.0	0.1	
DE &	N	N_P	N_L	N_C	CR	F
H-DE	10	5	5	5	0.9	0.5
APF	L					
	10					
PSAPF	L					
	10					

time of the algorithms employed is essentially due to the calculus of the fitness function. For instance, 500 evaluations require 1.18 secs of a single CPU. In fact, image loading, background subtraction and fitness evaluation represents more than 95% of the total computing time. Thus, we can consider the computing time employed by the different algorithms negligible for this problem. It is worth noticing, though, that the computational cost of the fitness function is independent of the DOF of the body model employed. Therefore we choose the number of function fitness evaluations as the representative operation count, [41].

For each test, the algorithms have been run 30 times using different seeds for the random number generator. The search space is delimited by the search radius of the joints, which is set to 0.5 *rad* for all the algorithms.

The parameter settings for the different algorithms are presented in Table 3. The values have been chosen without previous parameter tuning and having as reference the experimentation carried out in previous works:

- For PSO and H-PSO, the parameter setting corresponds to that established by John *et al* [10].

- For DE and H-DE the parameters have been selected according to the work by García-Martínez *et al* [34]. In order to set the variants of DE at the same level as those of PSO, we have chosen the same value for N (hence, the values of the parameters N_P, N_L, N_C have been reduced).

- For CMAES and H-CMAES, the (external) strategy parameters values have been generated from the dimension of the problem according to the expressions given in Table 2. The step size or standard deviation (search radius) has been the only parameter value provided.

- Finally, for the APF and PSAPF, a total of 10 layers were employed as in the work by Deutscher and Reid [4].

The same parameter setting is employed for the holistic and hierarchical approaches. Also, the parameter setting of the algorithms is maintained across all the sequences tested, i.e., no specific tuning has been made for each particular movement or subject.

The whole experimentation was carried out using a total of 40 CPUs running Linux and it took 12 weeks of continuous computation. In particular, we employed 10 CPUs equipped with *Core 2 Quad Q8400* running at 2.66 GHz.
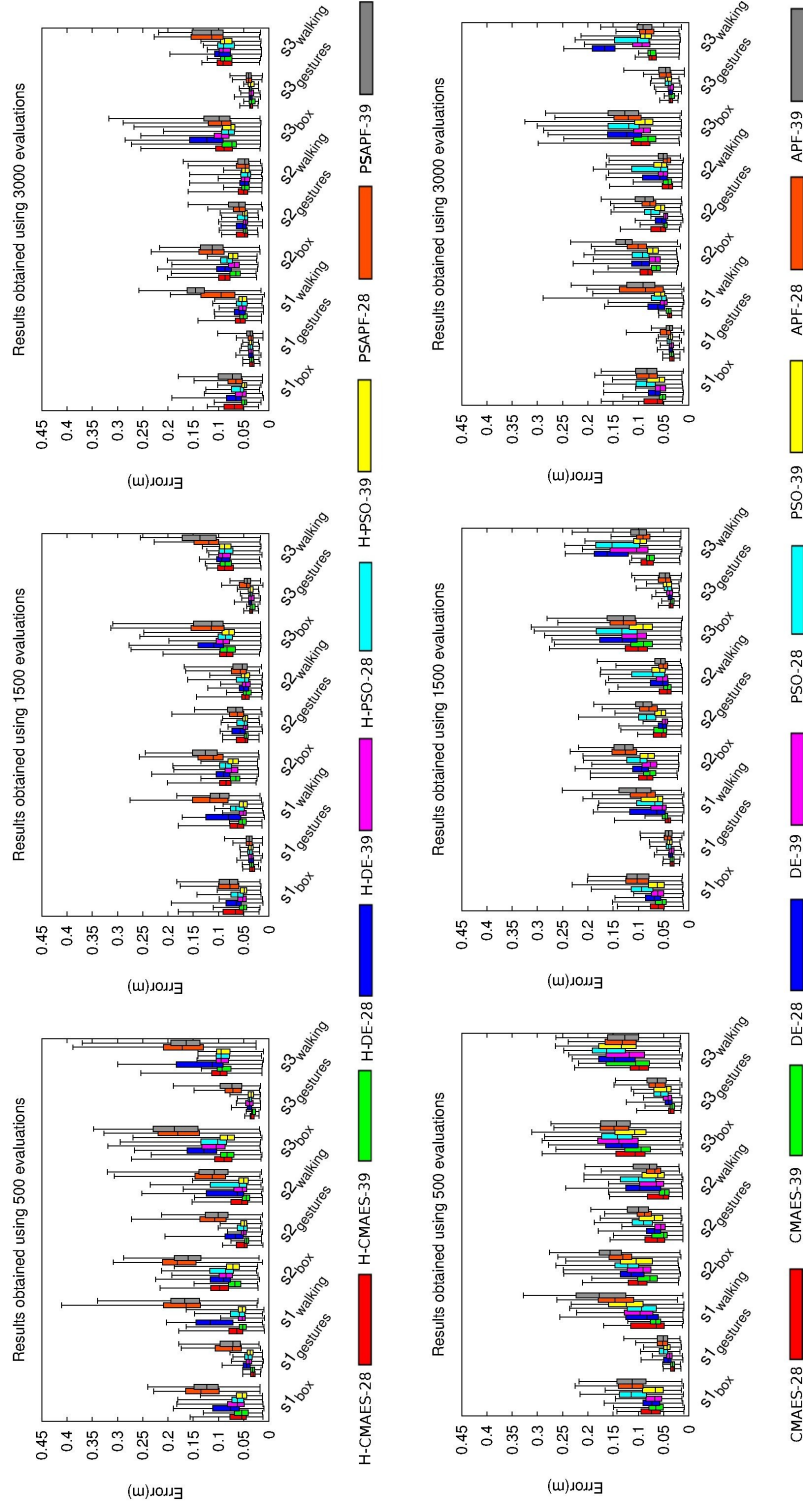
Figure 7: Boxplot showing the tracking error statistics obtained by the hierarchical and holistic algorithms on the 30 runs that each sequence has been tested. Holistic and hierarchical algorithms are presented separatedly and grouped by sequence. The sequences are in the X axis and the tracking error is represented on the Y axis. Each colored box, corresponding to a pair algorithm-model, represents the likely range of variation in the tracking error (interquartile range) and is divided by a line denoting the median value.

5.3. Quantitative and visual results

The average tracking errors obtained by the different algorithms and models in the experimentation for each sequence, see Eq. 20, are shown in the boxplots in Fig. 7. Holistic and hierarchical algorithms are presented separatedly and grouped by sequence. Each boxplot corresponds to a concrete number of evaluations, showing statistics about the error distribution on all the frames evaluated in the 30 runs. The sequences are in

11

the X axis and the tracking error is represented on the Y axis. Each colored box corresponds to a pair algorithm-model, where the latter is indicated by the DOF. The thin vertical line shows the error range for each pair algorithm-model bounded by the minimum and the maximum values and the horizontal line inside the box indicates the median value. The box itself indicates where most of the cases lie (interquartile range or likely range of variation). The lower side is the first quartile while the upper side is the third quartile.

Results in Fig. 7 show that the performance of the pairs algorithm-model differs from one sequence to another. The best performance and less spread of the error is provided in sequences that affect less parts of the body (gestures). The highest fluctuations correspond to fast limb movements (e.g. fast arm movement in box sequences, see Fig. 8). Moreover, the tracking error is different from one subject to another (the worst results in average are presented by subject S3), although the performance of the algorithms follows a similar pattern for the same type of action. As the number of evaluations increases, the likely range of variation in the error becomes narrower and most of the pairs algorithm-model increase its performance. This is consistent with the fact that the algorithms are able to diversify and explore more promising regions in the solution space. Regarding to the model, the quantitative results presented in the boxplots show how the 39-DOF model yields better results than the 28-DOF model for the same algorithm with the exception of particle filters. A high dimensional model allows a better adaptation to the human anatomy and improves the tracking results.

Evolutionary algorithms surpass particle filters in most of the cases. Particle filters present higher range of variation in the error and worst performance because of the absence of effective mechanisms for intensification (as the number of evaluations increases the convergence is slowly increased) and diversification (the increase in the complexity of the search space, model with more DOF, promotes the stagnation of particle filters). Fig. 9 shows the tracking results by superimposing the body model in some frames of a sequence. When analyzing the particle filter, we can observe that they are unable of successfully tracking the limbs with a short number of evaluations. However, the evolutionary algorithms reveal an important improvement of the tracking performance. The highest fluctuations in the tracking error of the algorithms in each frame of that sequence correspond to particle filters (APF), see Fig. 10.

Evolutionary algorithms are dominated respectively by the holistic and hierarchical version of CMAES. The invariance properties of the algorithm allow a better and faster adaptation of the model to the silhouettes and the prevention of the premature convergence by the step-size control guarantees the diversification in the search, [13]. DE and PSO require a more tedious parameter tuning and need more evaluations to converge. DE works better on large-scale optimization (holistic version), [34], and PSO increases its performance in the hierarchical version due to the effective communication between particles in the swarm search, [10]. A visual example of the results is presented in some demo videos available at Youtube (see `http://www.youtube.com/playlist?list=PL36A29215E7DBACA1`). CMAES presents an uniform level of

error over the entire sequences. The demo videos show more vibrations in the adjusted model for the holistic and hierarchical versions of DE and PSO.

Due to the large amount of tests performed, it is very difficult to obtain meaningful conclusions by analyzing the plots in Fig. 7. Therefore, we have conducted a set of non-parametric tests that are explained in the following section.

5.4. Statistical analysis of the results

In order to obtain a formal comparison of the algorithms, non-parametric tests [18] have been used. First, we have applied an omnibus statistical test to figure out whether the results of the proposed algorithms are statistically different or not. In the case of a statistical significant difference (the null hypothesis of equality of means is rejected) a post-hoc test have been used to compare the best ranked algorithm (control algorithm) against the remaining ones in order to find the concrete pairwise comparisons which produce differences. We have considered as omnibus test the Friedman test, [42], and as post-hoc procedure the Holm test, [43].

First, let us perform a comparative analysis of the algorithms for the two different body models as a function of the number of evaluations. For that purpose, we have run six different statistical tests. Each test compares the results of the eight algorithms using the same body model and the same number of evaluations. The summary of the statistical tests are shown in Tables 4 and 5 for the 39-DOF and 28-DOF models respectively. The labelling convention $ALG(DOF, EVAL)$ has been employed to identify the algorithms, where ALG stands for the algorithm acronym, DOF refers to the dimensionality of the model employed, and $EVAL$ is the number of evaluations.

For the sake of clarity, let us focus on the results of the tests for the eight algorithms using the 39-DOF model and 500 evaluations, which are shown in the first three columns of Table 4. The ranking value of an algorithm indicates the averaged position of its best solutions' fitness when it is compared to the best solutions' fitness of the other algorithms. A value of 1 indicates that the algorithm dominates in all the experiments.

As can be seen, the algorithms are presented in the Table in ascending order of ranking. We can observe that the H-CMAES algorithm is quantitatively better than the others. Nevertheless, obtaining the best ranking does not imply that the algorithm's performance is significantly better than the performance of the others. To assess that, we have calculated the Friedman statistic, [42], that has the value 57.925 as shown in the last row of the Table. This statistic is distributed according to a Chi-square distribution with 7 degrees of freedom (number of algorithms minus one) and if it is greater than the Critical Value of the distribution (for a given α confidence value), it means that there are at least two algorithms with significant differences in their performance. The critical value of the Chi-square distribution with 7 degrees of freedom for $\alpha = 0.05$ is 14.067. So, the Friedman test indicates that there is at least one algorithm (with a 95% of confidence) that shows significant differences in the results with respect to the best algorithm (also referred as the control algorithm from now on). To find out which of the algo-
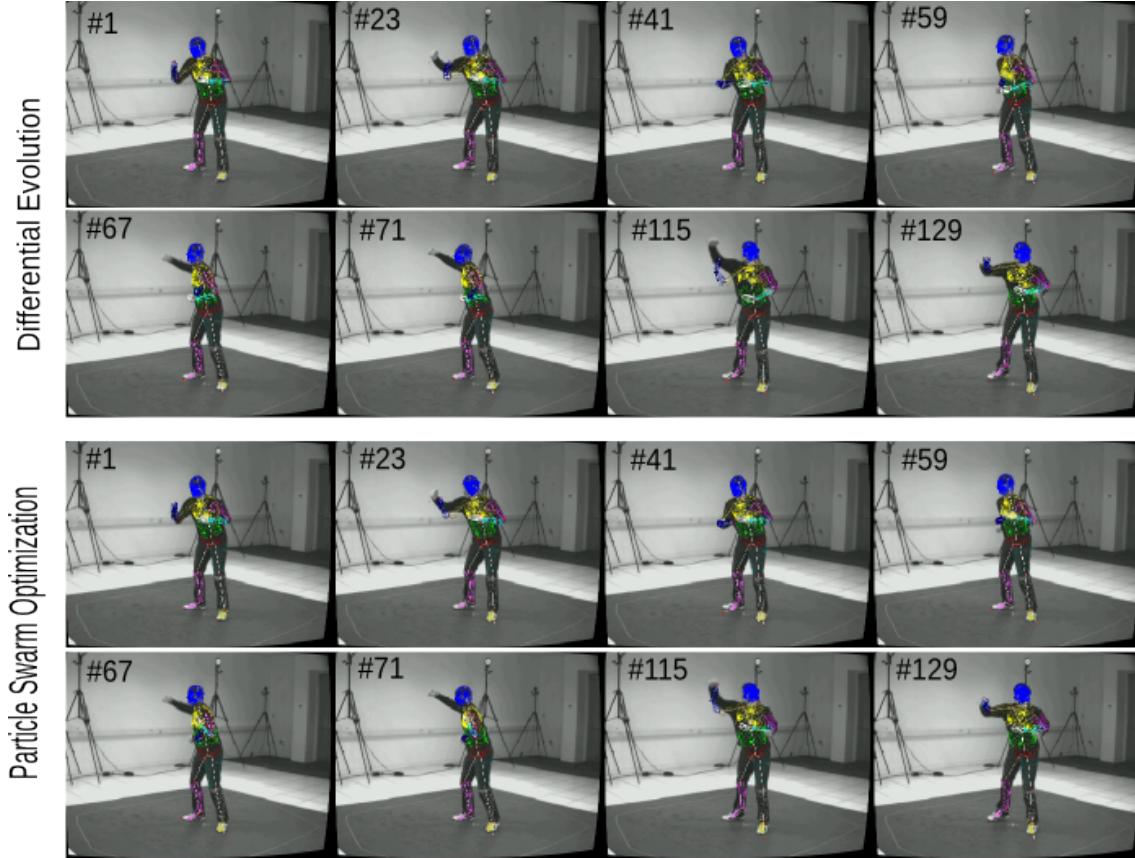
Figure 8: Tracking results obtained in the *box* sequence of subject *S*3. Frames of a particular execution with 3000 evaluations using the 39-DOF model with DE and PSO algorithms. The results are shown by superimposing the body model (skin and skeleton) in some frames of the sequence. Fast arm movements cause tracking failures. Both algorithms require some additional frames to re-track the arm appropriately.

rithms present significant differences, the post-hoc test of Holm has been employed.

The post-hoc test results are presented in third column under the label *p*-value. The analysis of the *p*-values proceeds as follows: if the *p*-value is lower than $\alpha/(r(alg) - 1)$ then the null hypothesis of equality of means for the control algorithm and that corresponding to the row is rejected. The parameter $r(alg)$ represents the rank position of the algorithm, which in this case are $r(H - DE) = 2$, $r(H - PSO) = 3$ and so on. Therefore, using $\alpha = 0.05$, the null hypothesis is rejected in four cases: DE, PSO, APF and PSAPF. The last column of each Table with the label "Holm" shows the threshold values ($\alpha/(r(alg) - 1)$) of the Holm test to consider that there are significant differences among the algorithms.

The same procedure is applied to the five other tests performed shown in Tables 4 and 5. For the sake of simplicity, we have set out in bold font the *p*-values for which the tests indicate that there are significant differences compared to the control algorithm (H-CMAES).

Based on the experimental results, a set of conclusions can be drawn. First, although the H-CMAES algorithm provides the best quantitative results (best ranked algorithm), we cannot guarantee significant differences with the remaining evolutionary algorithms. Second, significant differences exist between evolutionary algorithms and particle filters for all cases. Third,
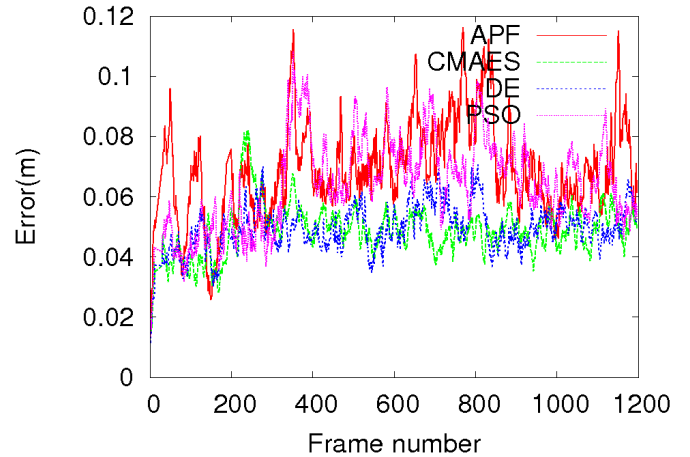

Figure 10: Tracking error of the holistic algorithms in the *walking* sequence of the subject S1 shown in Fig 9. The highest fluctuations correspond to particle filters (APF). The best results are obtained by DE and CMAES. CMAES presents an uniform level of error over the entire sequence.

hierarchical evolutionary approaches yield better quantitative results than their corresponding holistic approaches.

The next point to clarify is whether the results obtained using the 39-DOF model are better than these obtained with the 28-
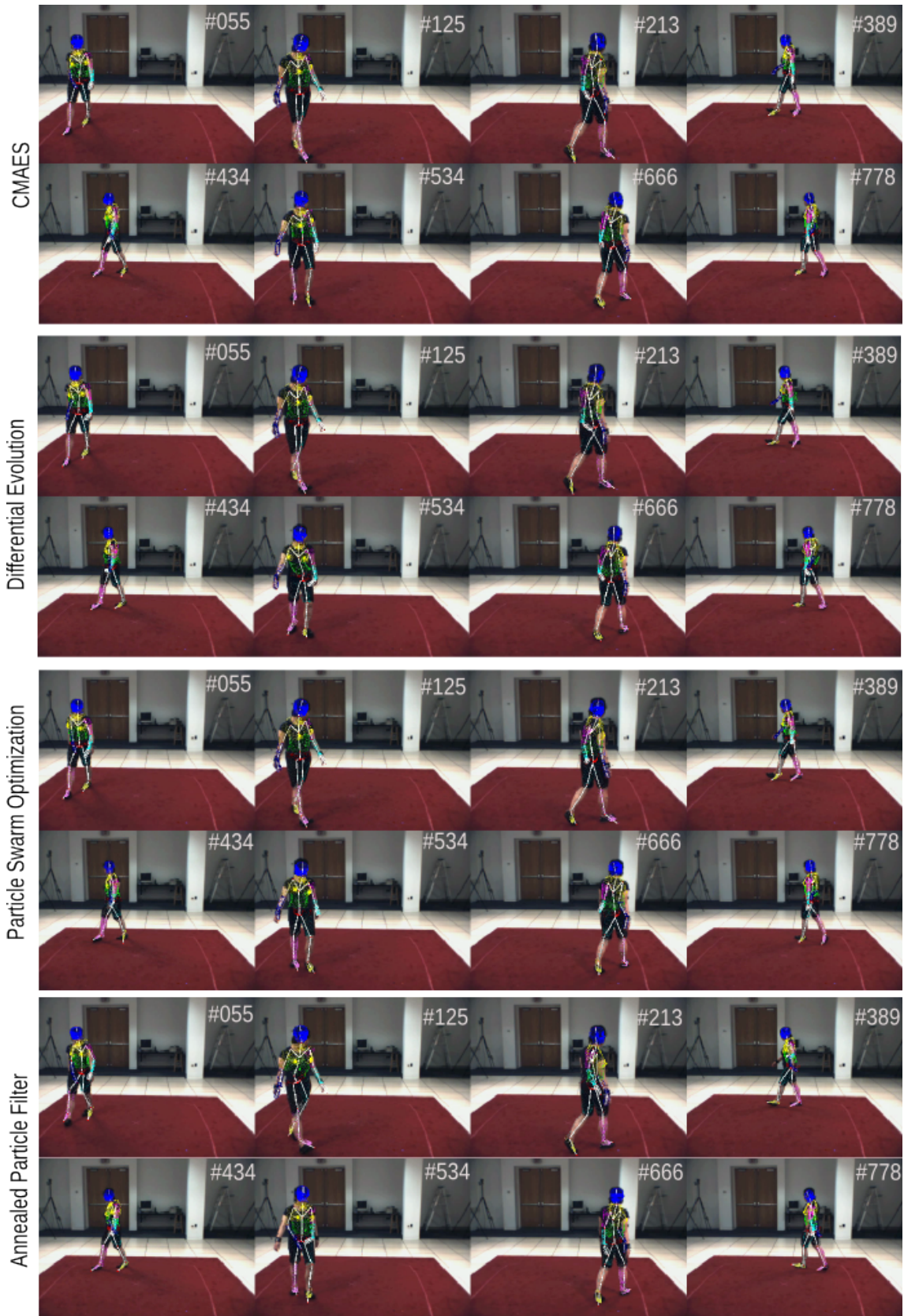
13

Figure 9: Tracking results obtained by the holistic algorithms in the *walking* sequence of the subject S1. Frames of a particular execution with 1500 evaluations using the 39-DOF model. The results are shown by superimposing the body model (skin and skeleton) in some frames of the sequence. The particle filter is unable of successfully tracking the limbs. The evolutionary algorithms reveal an important improvement of the tracking performance and obtain better performance in most of the frames.

Table 4: 39-DOF model. Ranking of the algorithms in the tests performed and results of Friedman and post-hoc Holm tests.

Algorithm	Rank	p-value	Algorithm	Rank	p-value	Algorithm	Rank	p-value	Holm
H-CMAES(39,500)	1.222	-	H-CMAES(39,1500)	2.000	-	H-CMAES(39,3000)	1.777	-	
H-PSO(39,500)	2.333	0.335	CMAES(39,1500)	2.666	0.563	CMAES(39,3000)	2.444	0.563	0.050
CMAES(39,500)	3.000	0.123	H-PSO(39,1500)	2.777	0.500	H-PSO(39,3000)	2.888	0.335	0.025
H-DE(39,500)	3.555	0.043	H-DE(39,1500)	3.111	0.335	H-DE(39,3000)	4.000	0.054	0.016
DE(39,500)	5.111	**7.57E-4**	DE(39,1500)	4.777	0.016	DE(39,3000)	4.222	0.034	0.012
PSO(39,500)	5.888	**5.31E-5**	PSO(39,1500)	6.111	**3.70E-4**	PSO(39,3000)	6.222	**1.18E-4**	0.010
APF(39,500)	7.000	**5.62E-7**	PSAPF(39,1500)	7.000	**1.49E-5**	PSAPF(39,3000)	6.888	**9.58E-6**	0.008
PSAPF(39,500)	7.888	**7.76E-9**	APF(39,1500)	7.555	**1.49E-6**	APF(39,3000)	7.555	**5.62E-7**	0.007
Fried. Stat.	57.925		*Fried. Stat.*	49.148		*Fried. Stat.*	48.851		

Table 5: 28-DOF model. Ranking of the algorithms in the tests performed and results of Friedman and post-hoc Holm tests.

Algorithm	Rank	p-value	Algorithm	Rank	p-value	Algorithm	Rank	p-value	Holm
H-CMAES(28,500)	1.666	-	H-CMAES(28,1500)	2.111	-	H-CMAES(28,3000)	2.444	-	
H-PSO(28,500)	1.888	0.847	CMAES(28,1500)	2.444	0.772	H-PSO(28,3000)	2.444	1.000	0.050
CMAES(28,500)	2.888	0.289	H-PSO(28,1500)	2.666	0.630	CMAES(28,3000)	3.111	0.563	0.025
H-DE(28,500)	4.444	0.016	H-DE(28,1500)	4.111	0.083	H-DE(28,3000)	4.111	0.148	0.016
DE(28,500)	4.666	**0.009**	DE(28,1500)	5.222	**0.007**	DE(28,3000)	5.444	**0.009**	0.012
PSO(28,500)	6.111	**1.18E-4**	PSO(28,1500)	6.333	**2.55E-4**	PSO(28,3000)	5.555	**0.007**	0.010
APF(28,500)	6.222	**7.97E-5**	PSAPF(28,1500)	6.666	**7.97E-5**	PSAPF(28,3000)	5.999	**0.002**	0.008
PSAPF(28,500)	8.000	**4.13E-8**	APF(28,1500)	6.777	**5.31E-5**	APF(28,3000)	6.888	**1.18E-4**	0.007
Fried. Stat.	53.481		*Fried. Stat.*	43.037		*Fried. Stat.*	30.740		

DOF model. To do so, we have evaluated the performance of each algorithm with both models and different number of evaluations. The results of the Friedman and post-hoc tests are shown in Tables 6 and 7. We can notice that significant differences exist between the 39-DOF and 28-DOF models for all evolutionary algorithms (excepting CMAES). In other words, the evolutionary algorithms selected provide better results with a higher dimensionality model. The same cannot be stated for PSAPF and APF because the statistical results show that the 28-DOF model allows to obtain, depending on each case, significantly or quantitatively better results than the 39-DOF model. For the particle filtering algorithms selected, the test reveals that the increment of the DOF actually has a negative impact in the results obtained, i.e., they obtain better solutions with simpler models.

The combinations H–CMAES(39,500), H–DE(39,1500), H–PSO(39,1500), CMAES(39,500), DE(39,1500) and PSO(39,1500) obtain the best trade-off between performance and speed (more DOF and less evaluations without significant difference) for the different evolutionary algorithms.

Finally, we must recall that, as indicated by Bandouch *et al* [6], the PSAPF algorithm improves the APF algorithm, but only as the number of evaluations increases.

6. Conclusions

This paper has compared the performance of particle filters and evolutionary algorithms for the MMOCAP problem. In particular, we have tested the APF particle filter, against the CMAES, DE and PSO evolutionary algorithms. In addition, we have evaluated both the holistic and hierarchical strategies of these algorithms with two body models of increasing complexity (28 DOF and 39 DOF).

Non-parametric statistical tests have been run on the results obtaining the following conclusions. First, the evolutionary algorithms tested, dominated by the hierarchical version of CMAES, perform significantly better than the particle filters employed. Second, the evolutionary algorithms are able to manage the 39-DOF model obtaining better results than when the 28-DOF model is employed. In other words, they can employ a more realistic (and complex) model achieving better results. The same is not true for the particle filters. Finally, the hierarchical strategies rank better than the holistic strategies in all tests.

Acknowledgements

References

[1] F. Multon, R. Kulpa, L. Hoyet, T. Komura, Interactive animation of virtual humans based on motion capture data, Journal of Visualization and Computer Animation 20 (2009) 491–500.

[2] H. Zhou, H. Hu, Human motion tracking for rehabilitation–a survey, Biomedical Signal Processing and Control 3 (2008) 1 – 18.

[3] T. B. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, Comput. Vis. Image Underst. 104 (2006) 90–126.

[4] J. Deutscher, I. Reid, Articulated Body Motion Capture by Stochastic Search, International Journal of Computer Vision 61 (2005) 185–205.

[5] J. MacCormick, M. Isard, Partitioned sampling, articulated objects, and interface-quality hand tracking, in: Proceedings of the 6th European Conference on Computer Vision-Part II, ECCV '00, Springer-Verlag, London, UK, 2000, pp. 3–19.

[6] J. Bandouch, F. Engstler, M. Beetz, Evaluation of hierarchical sampling strategies in 3d human pose estimation, Proceedings of the 19th British Machine Vision Conference, BMVC '08 (2008) 1–10.

[7] J. Gall, B. Rosenhahn, T. Brox, H.-P. Seidel, Optimization and Filtering for Human Motion Capture, International Journal of Computer Vision 87 (2010) 75–92.

[8] M. Lozano, D. Molina, F. Herrera (Eds.), Special issue on scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems, Soft Computing - A Fusion of Foundations, Methodologies and Applications, Springer Berlin / Heidelberg, 2010.

[9] K. A. D. Jong, Evolutionary Computation. A unified approach., MIT Press, 2006.

Table 6: Hierarchical approaches. Friedman and post-hoc Holm tests comparing the performance of the algorithms in relation to the number of evaluations and DOF.

Algorithm	Rank	p-value	Algorithm	Rank	p-value	Algorithm	Rank	p-value	Algorithm	Rank	p-value	Holm
H-CMAES(39,1500)	2.222	-	H-DE(39,1500)	1.666	-	H-PSO(39,3000)	1.555	-	PSAPF(28,3000)	1.333	-	
H-CMAES(39,3000)	2.444	0.801	H-DE(39,3000)	1.777	0.899	H-PSO(39,1500)	1.666	0.899	PSAPF(39,3000)	2.000	0.449	0.050
H-CMAES(39,500)	2.555	0.705	H-DE(28,3000)	3.777	**0.016**	H-PSO(39,500)	3.555	**0.023**	PSAPF(28,1500)	3.222	0.043	0.025
H-CMAES(28,1500)	4.222	**0.013**	H-DE(39,500)	3.888	**0.011**	H-PSO(28,3000)	3.777	**0.011**	PSAPF(39,1500)	3.222	0.043	0.016
H-CMAES(28,3000)	4.666	**0.005**	H-DE(28,1500)	4.000	**0.008**	H-PSO(28,1500)	4.555	**6.69E-4**	PSAPF(39,500)	5.222	**1.03E-5**	0.012
H-CMAES(28,500)	4.888	**0.002**	H-DE(28,500)	5.888	**1.68E-6**	H-PSO(28,500)	5.888	**8.94E-7**	PSAPF(28,500)	5.777	**4.66E-7**	0.010
Fried. Stat.	19.158		*Fried. Stat.*	32.174		*Fried. Stat.*	36.111		*Fried. Stat.*	38.968		

Table 7: Holistic approaches. Friedman and post-hoc Holm tests comparing the performance of the algorithms in relation to the number of evaluations and DOF.

Algorithm	Rank	p-value	Algorithm	Rank	p-value	Algorithm	Rank	p-value	Algorithm	Rank	p-value	Holm
CMAES(39,3000)	1.888	-	DE(39,3000)	1.111	-	PSO(39,3000)	1.222	-	APF(28,3000)	1.444	-	
CMAES(39,1500)	2.666	0.377	DE(39,1500)	2.111	0.256	PSO(39,1500)	2.222	0.256	APF(28,1500)	2.555	0.207	0.050
CMAES(28,3000)	3.555	0.058	DE(28,3000)	3.444	**0.008**	PSO(28,3000)	3.000	0.043	APF(39,3000)	2.555	0.207	0.025
CMAES(39,500)	3.777	0.032	DE(28,1500)	4.000	**0.001**	PSO(39,500)	4.222	**6.69E-4**	APF(39,1500)	3.555	0.016	0.016
CMAES(28,1500)	3.777	0.032	DE(39,500)	4.888	**1.83E-5**	PSO(28,1500)	4.555	**1.57E-4**	APF(28,500)	5.222	**1.83E-5**	0.012
CMAES(28,500)	5.333	**9.39E-5**	DE(28,500)	5.444	**8.94E-7**	PSO(28,500)	5.777	**2.39E-7**	APF(39,500)	5.666	**1.68E-6**	0.010
Fried. Stat.	17.507		*Fried. Stat.*	27.886		*Fried. Stat.*	35.730		*Fried. Stat.*	35.158		

[10] V. John, E. Trucco, S. Ivekovic, Markerless human articulated tracking using hierarchical particle swarm optimisation, Image Vision Comput. 28 (2010) 1530–1547.

[11] S. Shen, M. Tong, H. Deng, Y. Liu, X. Wu, K. Wakabayasi, H. Koike, Model based human motion tracking using probability evolutionary algorithm, Pattern Recognition Letters 29 (2008) 1877–1886.

[12] X. Zhao, Y. Liu, Generative tracking of 3D human motion by hierarchical annealed genetic algorithm, Pattern Recognition 41 (2008) 2470–2483.

[13] N. Hansen, The CMA evolution strategy: a comparing review, in: J. Lozano, P. Larranaga, I. Inza, E. Bengoetxea (Eds.), Towards a new evolutionary computation. Advances on estimation of distribution algorithms, Springer, 2006, pp. 75–102.

[14] K. V. Price, R. M. Storn, J. A. Lampinen, Differential Evolution A Practical Approach to Global Optimization, Natural Computing Series, Springer-Verlag, Berlin, Germany, pp. 37–134. 2005.

[15] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings., IEEE International Conference on Neural Networks. 4 (1995) 1942–1948.

[16] P. N. Suganthan, IEEE Congress on Evolutionary Computation (CEC) Competitions, 2011. http://www3.ntu.edu.sg/home/epnsugan/index_files/cec-benchmarking.htm.

[17] L. Sigal, A. O. Balan, M. J. Black, Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion, Int. J. Comput. Vision 87 (2010) 4–27.

[18] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Information Sciences 180 (2010) 2044–2064.

[19] M. Isard, A. Blake, CONDENSATION – conditional density propagation for visual tracking, Int. J. Computer Vision 29 (1998) 5–28.

[20] S. Corazza, L. Mündermann, A. M. Chaudhari, T. Demattio, C. Cobelli, T. P. Andriacchi, A markerless motion capture system to study musculoskeletal biomechanics: visual hull and simulated annealing approach., Annals of biomedical engineering 34 (2006) 1019–29.

[21] I.-C. Chang, S.-Y. Lin, 3D human motion tracking based on a progressive particle filter, Pattern Recognition 43 (2010) 3621–3635.

[22] D. Comaniciu, P. Meer, Mean shift: A robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 603–619.

[23] A. Cappozzo, U. D. Croce, A. Leardini, L. Chiari, Human movement analysis using stereophotogrammetry: Part 1: theoretical background, Gait and Posture 21 (2005) 186–196.

[24] L. Chiari, U. D. Croce, A. Leardini, A. Cappozzo, Human movement analysis using stereophotogrammetry: Part 2: Instrumental errors, Gait and Posture 21 (2005) 197–211.

[25] A. Leardini, L. Chiari, U. D. Croce, A. Cappozzo, Human movement analysis using stereophotogrammetry: Part 3. soft tissue artifact assessment and compensation, Gait and Posture 21 (2005) 212–225.

[26] U. D. Croce, A. Leardini, L. Chiari, A. Cappozzo, Human movement analysis using stereophotogrammetry: Part 4: assessment of anatomical landmark misplacement and its effects on joint kinematics, Gait and Posture 21 (2005) 226–237.

[27] N. Gordon, D. Salmand, Bayesian State Estimation for Tracking and Guidance Using the Bottstrap Filter, Journal of Guidance, Control and Dynamics 18 (1995) 1434–1443.

[28] M. Isard, A. Blake, Countour Tracking by Stochastic Propagation of Conditional Density, in: European Conference on Computer Vision, 1996, pp. 343–356.

[29] G. Kitagawa, Monte Carlo filter and smoother for non-gaussian nonlinear state space models, Journal of Computational and Graphical Statistics 5 (1996) 1–25.

[30] A. E. Eiben, J. E. Smith, Introduction to Evolutionary Computing, Springer-Verlag, Berlin, Germany, 2003.

[31] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evolutionary Computation 9 (2001) 159–195.

[32] N. Hansen, Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed, in: Workshop Proceedings of the GECCO Genetic and Evolutionary Computation Conference, ACM, 2009, pp. 2389–2395.

[33] N. Hansen, S. Muller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)., Evolutionary Computation 11 (2003) 1–18.

[34] C. García-Martínez, F. Rodríguez, M. Lozano, Role differentiation and malleable mating for differential evolution: an analysis on large-scale optimisation, Soft Computing - A Fusion of Foundations, Methodologies and Applications (2010) 1–18.

[35] C. Fernandes, A. C. Rosa, Self-adjusting the intensity of assortative mating in genetic algorithms, Soft Comput. 12 (2008) 955–979.

[36] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, 1998, pp. 69–73.

[37] G. Rogez, C. Orrite-Uruñuela, J. Martínez-del Rincón, A spatio-temporal 2d-models framework for human pose recovery in monocular sequences, Pattern Recogn. 41 (2008) 2926–2944.

[38] A. Sundaresan, R. Chellappa, Model driven segmentation of articulating humans in laplacian eigenspace, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 1771–1785.

[39] X. Zhao, Y. Liu, Generative tracking of 3d human motion by hierarchical annealed genetic algorithm, Pattern Recogn. 41 (2008) 2470–2483.

[40] M. Bastioni, M. FleRackers, J. Capco, Makehuman, Makehuman Team. http://www.makehuman.org/, 2012.

[41] R. K. Ahuja, J. B. Orlin, Use of representative operation counts in computational testing of algorithms, INFORMS Journal on Computing 8 (1996) 318–330.

[42] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association 32 (1937) 675–701.

[43] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian Journal of Statistics 6 (1979) 65–70.