Human Body Pose Estimation with PSO

Spela Ivekovic and Emanuele Trucco

***Abstract*— In this paper we describe the application of Particle Swarm Optimisation to the problem of human body pose estimation from multiple view video sequences. We use a subdivision body model with an underlying skeleton layer to estimate and illustrate the body pose. The optimisation looks for the best match between the silhouettes extracted from the original video sequence and the silhouettes generated by the projection of the model in a pose suggested by the PSO. The original PSO algorithm is applied hierarchically and combined with the full overall optimisation to decrease the effects of error propagation. Results demonstrate the ability of PSO to reliably recover the correct body pose from 4-viewpoint video sequences.**

I. Introduction

Human body pose estimation remains an active research area with solutions applicable in many domains including surveillance, motion capture, behaviour analysis, medical analysis, human-computer interaction and animation.

We address the problem of upper-body pose estimation for application in immersive videoconferencing [1]. The main concept behind the idea of immersiveness in videoconferencing is the novel view synthesis [2], a method of creating new, previously unseen views of the scene, from the available video data. To achieve the impression of immersiveness, one has to make use of the available views and render the scene from a perspective expected by the conference participant, thereby leading them to believe that they are actually present in a conference room with all other participants seated at the same conference table.

Participant's perspective changes dynamically throughout the videoconferencing session. The change is detected in real time with the use of gaze tracking methods and the scene is then appropriately rendered on the videoconferencing screen providing the impression of 3D motion parallax and, very importantly, eye contact between the individual conference participants engaged in a conversation.

Advanced high-quality view synthesis techniques rely heavily on accurate and dense stereo disparity maps to compute the novel view. Stereo correspondence problem and disparity map computation have been thoroughly researched and a number of solutions has been reported in the literature [3]. The common issue with all disparity estimation algorithms is that they fail to perform well at depth discontinuities. Additionally, it is theoretically impossible to recover disparity information for areas of the scene only visible in one of the multi-view images, the so-called *occluded* areas. In view synthesis for videoconferencing these two problems significantly influence the quality of the final result.

Spela Ivekovic and Emanuele Trucco are with the EECE department, EPS, Heriot-Watt University, Edinburgh EH14 4AS, United Kingdom. (email: si1,e.trucco@hw.ac.uk)

In order to address the problem of disparity map reliant view synthesis quality, we decided to work on improving the accuracy and reliability of the already computed dense stereo disparity maps used as a base for view synthesis. In our approach we took advantage of the *a-priori* knowledge about the scene and used a human body model to represent the pose of the conference participant. In order to assess the accuracy of computed dense disparities and fill in the missing disparity regions with the data coming from the model, the model first had to be deformed to match the pose of the person in the video sequence. We approached the body pose estimation from the multi-view video sequences with an algorithm making use of a subdivision surface layered human body model and particle swarm optimisation.

In the remainder of this paper we first give an overview of the related work in Section II, then describe the PSO algorithm used in our tests in Section III and briefly explain the idea of subdivision in Section IV. Section V elaborates on the algorithm combining the body model and PSO to solve the pose estimation problem and Section VI details the experiments and results obtained using our approach. We conclude with a discussion and future work in Section VII.

II. Related Work

Human body pose estimation has been attempted in various ways including with and without explicit body models, using image data and 3D scanner data, and in the case of image data, using single or multiple viewpoint sequences. A not-so-recent survey can be found in [4] and partly also in [5]. Our work uses an explicit human body model and estimates the pose from multi-view video sequences.

Body pose estimation from images using a human body model has been addressed by various researchers. Plänkers and Fua [6] report using an implicit surface body model (metaballs) which they fit to 3D stereo data constrained by silhouette contours. They use an implementation of Levenberg-Marquardt optimisation method to fit the model to the stereo data obtained from 3 views. Carranza *et al.* [7] use a triangular mesh body model enhanced with a 1D Bezier spline and downhill optimisation constrained with silhouettes to recover the body pose. Poppe *et al.* [8] mention a similar application area to ours, virtual

environments, and work on monocular video sequences using a simple body model composed of cylinders.

Our work differs from the mentioned related work in two aspects. Unlike other reported work, we use a subdivision surface body model, the choice of which was motivated by the requirements of our application, and we recover the pose parameters using a global optimisation algorithm, the Particle Swarm Optimisation (PSO).

PSO has been successfully applied to various problems, however, we were not able to find many references to its use for body pose estimation. The closest related work found is by Schutte *et al.* [9] who report a parallel PSO implementation and illustrate its performance on an example of a simple kinematic chain similar to our boundary case example described in section VI.

Human body pose estimation has been tackled with many different optimisation methods. Our analysis of the nature of the evaluation function, detailed in section VI, revealed that a global optimisation approach is required to adequately address the dimensionality and multimodality of the problem while keeping it fully automatic. The recently reported success of the application of PSO to various problems motivated our decision to use it as a global optimiser for our problem. As shown in Section V, formulating the pose estimation problem in the context of PSO is actually fairly straightforward. The simplicity of the PSO algorithm versus its ability to find global minima is in itself a very appealing argument for its use.

The choice of the body model was motivated by the recent popularity of the subdivision modelling in the area of Computer Graphics. Unlike regular mesh models, the subdivision surfaces exhibit no unwanted artefacts, such as cracks and gaps, when deforming the mesh. They are inherently multiresolutional which is a great advantage for our application as we have to keep in mind that the result of model fitting will have to be transmitted across the network to other computers taking part in the videoconference. Last but not least, it has been shown that is possible to adaptively subdivide and deform the coarse base mesh adding an arbitrary level of detail to the subdivision model [10] and in this way allowing it to accurately represent not only the pose but also the shape of the modelled person, which is one of the future goals of this work.

III. Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) is an evolutionary computation technique introduced by Kennedy and Eberhart in 1995 [11]. The idea originated from the simulation of a simplified social model where the agents were thought of as collision-proof birds and the original intent was to graphically simulate the unpredictable choreography of a bird flock.

The original PSO algorithm was later modified by the authors and other researchers to improve its search capabilities and convergence. Several successful applications of PSO were also reported in the literature. For an overview of the relevant research in this area an interested reader will find a good starting point in [12].

One of the important modifications of PSO was introduced in 1998 by Shi and Eberhart [13]. They changed the velocity update equation of the swarm by adding an additional parameter called *inertia weight*, w. The aim of this parameter was to guide the search behaviour of the swarm. The larger the inertia parameter value, the more global the search, and vice versa. Several other modifications were added later on but for the purpose of this paper we focus on the contribution of [13], as it is also the version of PSO which we used in our experiments.

In the following we give a brief overview of the PSO algorithm using inertia weight parameter.

A. PSO Algorithm with Inertia Weight Parameter

Assume an n-dimensional search space $\mathbb{S} \subseteq \mathbb{R}^n$, a swarm consisting of N particles and a fitness function $f : \mathbb{S} \rightarrow \mathbb{R}$ defined on the search space. The i-th particle is represented as an n-dimensional vector $X_i = (x_{i1}, x_{i2}, ..., x_{in})^T \in \mathbb{S}$. The velocity of this particle is also an n-dimensional vector $V_i = (v_{i1}, v_{i2}, ..., v_{in})^T \in \mathbb{S}$. The best position encountered by the i-th particle so far (*personal* best) is denoted as $P_i = (p_{i1}, p_{i2}, ..., p_{in})^T \in \mathbb{S}$ and the value of the fitness function at that position $pbest_i = f(P_i)$. The index of the particle with the overall best position so far (*global* best) is denoted as g and $gbest = f(P_g)$. Let us also denote the optimum of the fitness function f by $sol = f(P_s)$, where the index s denotes the solution position in the search space. The PSO algorithm can then be stated as follows.

1) **Initialisation**:
   - Initialise a population of particles $\{X_i\}, i = 1...N$, with random positions and velocities in the search space $\mathbb{S}$. For each particle evaluate the desired fitness function and set $pbest_i = f(X_i)$. Identify the best particle in the swarm and store its index as g and its position as P_g.
2) **Repeat** until $|sol - gbest| < \epsilon$ for some predefined ϵ or the number of iterations reaches a predefined limit:
   - Move the swarm by updating the position of every particle according to the following two equations:

$$\begin{aligned} V_i &= wV_i + \varphi_1(P_i - X_i) + \varphi_2(P_g - X_i) \\ X_i &= X_i + V_i \end{aligned} \quad (1)$$

     where φ_1 and φ_2 are random numbers defined by an upper limit which is a parameter of the system and w is the inertia weight parameter.
   - For $i = 1...N$ update $pbest_i$ and $gbest$.

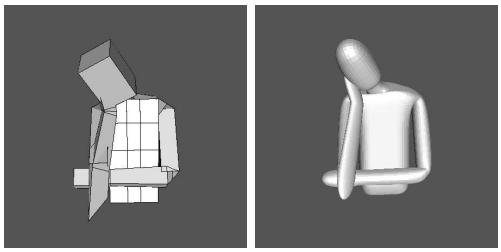

Fig. 1. Subdivision body model base mesh and the corresponding smooth surface after 3 iterations of Catmull-Clark subdivision.

The value of the inertia weight w can remain constant throughout the search or change with time, depending on the nature of optimisation.

IV. Subdivision Surfaces

Subdivision surfaces were introduced in 1978 by Catmull and Clark [14] and Doo and Sabin [15]. The algorithm for modelling smooth 3-D objects starts with a coarse polyhedron p_0 approximating the shape of the desired object. This coarse model can then be refined using subdivision rules to produce increasingly faceted approximations to the associated smooth shape. If these rules are represented by the operator $\mathcal{S}$, this process has the form [16]

$$p^k = \mathcal{S}p^{k-1}. \tag{2}$$

Applying $\mathcal{S}$ to an initial model p^0 yields a sequence of polygonal models $p^1, p^2, \ldots$. The rules comprising $\mathcal{S}$ specify how the polygonal faces of p^{k-1} are split, as well as how the vertices of p^k are positioned in terms of the vertices p^{k-1}. If these rules are chosen carefully, the limit of this process is a smooth surface p^∞ that approximates the coarse model p^0.

Depending on the type of the mesh describing the coarse polyhedron, different rules are applied to produce the final smooth surface. The two basic and most frequently used subdivision methods are *Loop* subdivision, defined on a triangular mesh, and *Catmull-Clark* subdivision, defined on a quadrilateral mesh. A description of both can be found in [16]. These two subdivision schemes are applicable generally as an arbitrary mesh can be reduced to a triangular or quadrilateral mesh after one subdivision pass.

The coarse polyhedron or the *base mesh* of our body model is represented with a quadrilateral mesh as illustrated in Figure 1 and we therefore use *Catmull-Clark* subdivision rules to produce the final smooth body model.

V. Pose Estimation Algorithm

In this section we describe the components of the pose estimation algorithm. We begin with the body model definition, then express the pose estimation as a PSO problem, define the fitness function and finally describe the inertia weight parameter change model.

A. Model

In our target application the person is seated at a conference table with only the upper body visible. We are therefore only interested in modelling the human body from the waist up. The body model is a layered subdivision surface model, illustrated in Figure 1, consisting of two layers, the skeleton and the skin.

The skeleton layer is defined as a set of transformation matrices which encode the information about the position and orientation of every joint with respect to its parent joint in the hierarchy:

$$Skeleton = \{T_1^2, T_2^3, ..., T_{N-1}^N\}. \tag{3}$$

N is the number of joints in the skeleton and T_i^j is a homogeneous transformation matrix encoding the orientation of the coordinate system of joint j with respect to the coordinate system of joint i. The top of the hierarchy is the root joint which branches out into three kinematic sub-chains, one for each arm and one for the neck and head. The clavicle joint is the root of all three sub-branches and is defined to have one location and three different orientations, one for each of the sub-branches. The entire kinematic structure consists of 12 joints.

The skin layer represents the second layer in the model and is connected to the skeleton through the joints' local coordinate systems. Each of the joints controls a certain area of the skin. Whenever a joint or limb moves, the corresponding part of the skin moves and deforms with it. The skin can therefore be described as a set of transformation matrices from the skeleton layer plus the sets of points influenced by each of the transformations:

$$Skin = \{\{T_1^2, P_{T_1^2}\}, \{T_2^3, P_{T_2^3}\}, ..., \{T_{N-1}^N, P_{T_{N-1}^N}\}\} \tag{4}$$

In order to generate a smooth skin surface of the model, all the skin points $P_{T_i^j}$ have to be transformed into a common coordinate system such as the world coordinate system:

$$P_w = T_w^1 * T_1^2 * \cdots * T_i^j * P_{T_i^j}, \quad \forall\ i,\ j \in [1, \ldots, N] \tag{5}$$

The points (vertices) P_w^i are connected with edges into faces F to form a base mesh:

$$M_0 = \{V, F\}, \text{where } V = \{P_w^i\}, F = \{P_w^{i_1}, P_w^{i_2}, P_w^{i_3}, P_w^{i_4}\} \tag{6}$$

which is then subdivided to obtain the smooth limit surface, i.e., the skin:

$$M_\infty = \mathcal{S}_\infty \ldots \mathcal{S}_1 \mathcal{S}_0 M_0 \tag{7}$$

B. PSO parametrisation of the problem

In PSO, each particle represents a potential solution in the search space. Our search space is the space of all plausible skeleton configurations. We represent the body pose with 20 degrees of freedom, 3 translations and 17 rotations, as detailed in Table I. Limb lengths are fixed. The individual

TABLE I

SKELETON DEGREES OF FREEDOM

JOINT (index)	DOF
Root location (root)	3
Root orientation (0)	3
Clavicle-neck orientation (1)	2
Clavicle-left orientation (2)	2
Left Shoulder orientation (3)	3
Left Elbow orientation (4)	1
Clavicle-right orientation (5)	2
Right Shoulder orientation (6)	3
Right Elbow orientation (7)	1
TOTAL	20

particle's position vector in the search space is specified as follows:

$$X_i = (root_x, root_y, root_z, \alpha_x^0, \beta_y^0, \gamma_z^0, \alpha_x^1, \gamma_z^1, ..., \alpha_x^7), \quad (8)$$

where $root_x, root_y, root_z$ denote the position of the root joint with respect to the world coordinate system, α_x^0 denotes a rotation around x-axis of the root joint coordinate system for angle α, γ_z^1 denotes a rotation around z-axis of the clavicle-neck joint for angle γ, etc.

The position and velocity update equations of the PSO algorithm described in Section III-A contain three parameters, w, φ_1, φ_2, which, alongside the evaluation function, define the behaviour of the swarm. The inertia weight parameter w influences the exploratory behaviour of the particles. It can remain constant throughout the optimisation or change with time. In our experiments a changing inertia weight parameter was used and its change modelled with an exponential function described in more detail in Section V-D.

The parameters φ_1 and φ_2 influence the *social* and *cognition* components of the swarm behaviour [13]. They are composed of a random number and a constant and can be written as $\varphi_1 = c_1 rand_1()$ and $\varphi_2 = c_2 rand_2()$, where c_1 and c_2 are two constants and $rand_1()$ and $rand_2()$ two random numbers in the interval $[0, 1]$. In our experiments the values of the constants c_1 and c_2 were both set to integer 2, as recommended by [13], [11], which on average made the weights for social and cognition components of the swarm equal to 1. Throughout the experiments with pose estimation we concentrated on the influence of the inertia parameter on the swarm behaviour and didn't experiment with the social or cognition bias that could be introduced through manipulating the values of φ_1 and φ_2.

C. Evaluation function

The evaluation function compares the silhouettes of the original images acquired by the cameras and the silhouettes generated by the model in its current pose. We acquire the original images from four different viewpoints, left, centre, right, and top. Each of the original images is foreground-background segmented and binarised to obtain a silhouette. Let the images containing the original silhouettes be denoted as I_l^o, I_c^o, I_r^o, and I_t^o for left, centre, right, and top *original* silhouette, respectively. Similarly, let I_l^m, I_c^m, I_r^m, and I_t^m denote images of the *model* silhouettes from the left, centre, right, and top view, respectively. The evaluation function can then be written as follows:

$$E = \alpha \sum_{1}^{row} \sum_{1}^{col} (I_l^o \ \& \ I_l^m) \quad + \quad \beta \sum_{1}^{row} \sum_{1}^{col} (I_c^o \ \& \ I_c^m) + \gamma \sum_{1}^{row} \sum_{1}^{col} (I_r^o \ \& \ I_r^m) \quad + \quad \delta \sum_{1}^{row} \sum_{1}^{col} (I_t^o \ \& \ I_t^m) \quad (9)$$

where row and col denote the image dimensions, i.e., number of rows and columns, respectively, and $\&$ denotes the logical *AND* operation. Coefficients $\alpha, \beta, \gamma, \delta$ are used to normalise the contribution of every camera to the total error count. Let T_l, T_c, T_r, T_t denote the total number of pixels for the original silhouettes of the left, centre, right and top view, respectively. The values of the coefficients are then $\alpha = 1/T_l$, $\beta = 1/T_c$, $\gamma = 1/T_r$ and $\delta = 1/T_t$. We use the same evaluation function in all experiments.

D. Inertia weight parameter

As already mentioned in Section III, inertia weight plays an important role in directing the exploratory behaviour of the particles. Higher inertia values push the particles to explore more of the search space and emphasise their individual velocity. This kind of behaviour is useful when trying to coarsely explore the entire search space to find a good starting point for a multimodal optimisation. On the other hand, lower inertia values force particles to focus on a smaller search area and move towards the best solution found so far. This approach makes sense when the global optimum region has been successfully identified and all that remains is finding the exact optimum location in the search space.

In Section VI we describe the application of PSO to a boundary case of parameter estimation for a 2 DOF kinematic chain. When estimating the parameters, we make an extensive use of the inertia parameter to guide the particle behaviour at different stages of the optimisation. Shi and Eberhart discussed the influence of different inertia values on the exploratory abilities of the swarm in [13]. They used a constant inertia change function and an inertia change function which decreased linearly with time. They tested inertia values in the interval $[0, 1.4]$ and found that, when using a constant inertia value, a medium value of w, i.e., $0.8 < w < 1.2$, had the best chance to find the global optimum while also taking a moderate number of iterations. Large values of w, i.e., $w > 1.2$ made PSO behave more like a global search method always trying to exploit new search areas. In order to decide which inertia value best worked for our problem, we first ran the optimisation by

linearly decreasing the inertia with every iteration. Results showed that only a small number of large inertia values were necessary for the swarm to locate the section of the fitness function containing the global minimum. The optimisation rapidly converged towards the minimum once the inertia value reached 0.5 and lower values.

We decided to model the inertia change with an exponential function. By doing so, we achieved that, using a constant sampling step, at the beginning of the optimisation the inertia values were large, allowing the swarm to explore, but not for too long, as the value was set to rapidly decrease as soon as the swarm failed to find a better optimum with the current inertia value. Towards the end of the optimisation the inertia values were small and decreasing slowly, thereby allowing the optimisation enough time to converge and find an accurate optimum location.

The following simple exponential function was used to model the inertia change:

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

where A denotes the starting value of w when $x = 0$. The optimisation terminated when $w(x)$ fell below 0.1. The findings in [13], the shape of $w(x)$ and the dimensionality of the search space influenced the decision to set $A = 2.0$ in order to force the swarm to explore possible search areas of the multi-dimensional search space before focusing on the best optimum discovered and finding its precise location. The sampling variable x was incremented by $\Delta x = ln(10A)/N$, where N is the desired number of inertia weight changes.

The swarm was allowed to explore the search space with a particular inertia value for as long as every move of the swarm improved the current global optimum estimate. As soon as there was an iteration which failed to improve the estimate, the value of the sampling variable increased and the inertia weight value decreased accordingly. This forced the swarm to identify possible optimum regions at the very beginning, then focus on the best few, and eventually settle down in the most promising region and find the global optimum.

VI. Experiments

A. Setup

In our experiments, we acquired video sequences with a set of four off-the-shelf Unibrain IEEE1394 webcams and off-the-shelf lighting. The setup is shown in Figure 2. Three cameras are positioned in front of the person sitting at the table and one camera above. The system was calibrated with the multi-camera self-calibration method [17] which uses a laser pointer for corresponding points and does not require an explicit calibration object. The back of the cube setup was covered with a blue backdrop to facilitate the foreground-background segmentation and silhouette extraction. We simultaneously acquired 4 views with individual image dimension 640×480. The swarm size was set to 10 particles.

Fig. 2. Cube setup with the off-the-shelf IEEE1394 webcams used in our experiments.

B. Boundary case optimisation

Our skeleton model consists of 3 different types of joints: ball and socket joints, pivot joints and hinge joints. To demonstrate the application of PSO to skeleton pose estimation, we first focus on a boundary case of optimising a kinematic structure with only 2 DOF. We illustrate the approach by estimating the 2 rotation parameters influencing the movement of the lower arm.

In our kinematic skeleton structure the movement of the lower arm is controlled by a combination of two different joint movements - the elbow joint and the shoulder joint. The elbow joint is a hinge joint with 1 DOF, allowing the lower arm to flex towards the upper arm and extend away from it. An accurate description of the lower arm movement requires another degree of freedom which resides in the shoulder joint. The shoulder joint is a ball joint with 3 DOF. Rotation of the shoulder joint around the upper arm axis creates the impression of rotating the elbow joint as during this rotation the shoulder joint and the upper arm appear to be nearly static and the shoulder rotation results in the movement of the lower arm. Figure 4 left illustrates the two rotations.

Figure 3 shows the graph of the evaluation function for the two rotational parameters controlling the lower arm. The evaluation function is multimodal and requires a global optimisation approach to locate the correct minimum.

For this boundary case we performed 100 PSO tests using the exponentially decreasing inertia parameter described in section V-D. We recorded the optimum found in every iteration and performed a statistical analysis of the results. The mean value of the global optimum successfully found in all iterations was at the value -3.3079 which can be seen

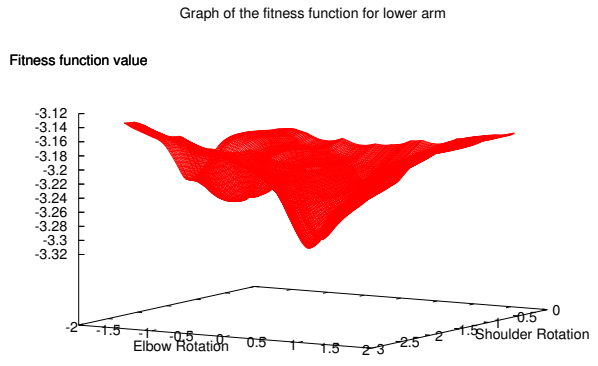

Fig. 3. Graph of the evaluation function for 2DOF

to correspond to the minimum of the evaluation function plotted in Figure 3. The standard deviation of the results was 0.0054, showing that individual optimisation runs quite strongly agreed on the chosen solution. These results very explicitly supported our choice of the inertia value change function and motivated the decision to use the same change function for the full kinematic chain optimisation.

C. Full optimisation

Although PSO was designed to tackle multidimensional and nonlinear optimisation problems and has been shown to perform well for many demanding optimisation problems, it failed to fulfill the expectations when applied unmodified to our full kinematic chain parameter estimation problem. The search space for the full skeleton structure expanded from the boundary 2 to 20 dimensions. The video sequence was not acquired using any optical markers that could potentially be used as a ground truth. The optimisation results thus had to be evaluated by observation, i.e., by comparing how well the silhouette generated by the optimisation result agreed with the silhouette derived from the real sequence. The results obtained indicated that the evaluation function was extremely multimodal as the optimisation regularly returned local minima, i.e., the estimated pose offered a plausible interpretation of that of the person in the sequence, however, the silhouette overlap of the estimated pose and the real pose was not the best possible.

It is possible that a different inertia parameter change function, larger number of iterations and a larger population would allow the optimisation to successfully locate the global optimum, however, as the fitness function evaluation described in Section V-C is very expensive already and our target application is time-constrained, we decided to follow a different route as described in the next subsection.

D. Hierarchical optimisation

Our optimisation problem posesses an inherent hierarchy. The hierarchy of the upper body's kinematic structure begins at the root joint, the movement of which influences the movement of all other joints. Similarly, the movement of the clavicle joint influences the movement of the shoulder and the rest of the arm. Taking advantage of this hierarchical structure, we can formulate our problem as a sequence of optimisation problems resembling our boundary optimisation case explained in Section VI-B, for which PSO is able to find the correct solution, as demonstrated.

First, the correct hierarchical structure must be identified. The hierarchical structure of the upper body skeleton begins with the root joint and then branches out into three independent sub-chains, one for each of the arms and one for the neck and head. Figure 4 right schematically illustrates the three sub-chains. The clavicle joint is the immediate parent joint of all three sub-chains. It is defined to have one location and three different orientations, each influencing one of the sub-chains. Depending on the subchain that is influenced, the labels for the three different orientations are clavicle-neck (neck and head sub-chain), clavicle-left (left arm) and clavicle-right (right arm).

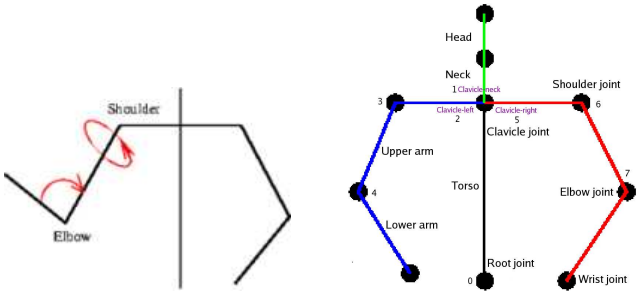

Fig. 4. Left: illustration of the elbow joint parametrisation. Right: hierarchical structure of the skeleton kinematic chain.

Once we identified the hierarchy, we must now decide how many joints we wish to optimise in one stage of the hierarchical approach. Having failed to optimise the whole kinematic structure at once and having successfully shown that the multimodal boundary case optimisation is doable, we opted for optimising each individual joint separately, wherever possible. The rationale behind this decision lies primarily in the fact that 2 DOF which we optimised in the boundary example, is actually the lowest possible dimensionality of any joint present in our hierarchy, as detailed in Table I. As described in Section VI-B, the elbow joint naturally posesses only 1 DOF, however, in order to accurately model its movement, this DOF has to be combined with a DOF coming from the shoulder joint. For the purpose of our analysis we therefore treat the elbow joint as a 2 DOF joint. If we combine several joints and optimise them all at once, the dimensionality of the problem immediately rises up to 4 DOF or more, and based on our findings in the boundary case, we expect the modality of the fitness function for 4 DOF to be very complicated and possibly require a different approach, more similar to the

one mentioned in Section VI-C. However, as shown next, we were not always able to avoid such a high dimensionality.

We perform the hierarchical optimisation in 7 steps (see Table II). First, we optimise the location of the skeleton in space, i.e., the location of the root joint, followed by the root joint orientation. These are both 3 DOF optimisations. Once the skeleton has been positioned in space, we optimise the neck and head sub-chain, for which we only use 2 DOF in the clavicle neck joint to model the tilt of the head. The movement of the clavicle left and clavicle right joint on their own does not produce enough variation in the silhouette shape to be optimised individually. Therefore, in the next step, we combine the left clavicle joint with two rotational dimensions of the shoulder joint and optimise the parameters of the left upper arm, a 4 DOF optimisation. Likewise, we then optimise the right upper arm, again 4 DOF. At the end we are left with the left and right lower arm, each modelled with 2 DOF as described in the boundary case. The two 4 DOF upper arm optimisations required a denser sampling of the inertia weight function to correctly locate the optimum region.

Our fitness function is based on silhouette overlap between the model and the original silhouette coming from the video sequence. When optimising joint parameters hierarchically, the joints lower in the hierarchy mislead the silhouette overlap count as they contribute to it despite of not having been optimised yet. We avoid this by deforming the subdivision body model so that at a particular stage of the optimisation only those body parts which are currently optimised or have already been optimised are visible. We also exclude the hands from the model entirely, as the hands in the original sequence exhibit too much articulation and constantly mislead the optimisation. The hands can be very useful as a constraint, however, that is entirely based on the input images and the explicit modelling as such is not necessary. All these modifications of the model are very easy to achieve by simply hard-coding the joint rotation angles for the unwanted joints to hide them inside the model or, rather more directly, simply setting the unwanted limb lengths to zero. The subdivision body model makes all these modifications very easy, as it produces a smooth model with a desirable silhouette even when extremely deformed.

The results of the hierarchical approach were encouraging (see Figure 5 left). The optimisation correctly identified the pose. However, having optimised hierarchically without looking back, another problem creeped in, namely that of error propagation. In our hierarchical approach we rely on the fact that each individual stage will come up with the best possible result and therefore provide a good starting point for the next stage. However, results show that this is not always the case. We address this problem in the next section.

TABLE II
STEPS IN THE HIERARCHICAL OPTIMISATION

TORSO (**1**) Root location 3DOF: $root_x, root_y, root_z$ (**2**) Root orientation 3DOF: $\alpha_x^0, \beta_y^0, \gamma_z^0$	**RIGHT UPPER ARM** (**5**) Clavicle-right orientation + Right shoulder orientation 4DOF: $\alpha_x^5, \gamma_z^5, \alpha_x^6, \gamma_z^6$
NECK & HEAD (**3**) Clavicle-neck orientation 2DOF: α_x^1, γ_z^1	**LEFT LOWER ARM** (**6**) Left shoulder orientation + Left elbow orientation 2DOF: β_y^3, α_x^4
LEFT UPPER ARM (**4**) Clavicle-left orientation + Left shoulder orientation 4DOF: $\alpha_x^2, \gamma_z^2, \alpha_x^3, \gamma_z^3$	**RIGHT LOWER ARM** (**7**) Right shoulder orientation + Right elbow orientation 2DOF: β_y^6, α_x^7

E. Combination of both approaches

We mentioned in Section VI-C that we had reservations about trying to make the PSO find the global best solution in a 20-dimensional space. Our main reason was the time constraint as we expect that a 20-dimensional optimisation would have to spend a significant amount of time globally exploring the search space in an attempt to identify the global optimum region and only then work on converging to its exact position. After having performed hierarchical optimisation, however, the problem has now changed significantly. The hierarchical optimisation without exception correctly identifies the region of the search space containing the global optimum. It does not always manage to converge to its exact position though, mainly for the reasons of error propagation mentioned in the previous section.

In order to more accurately locate the position of the global optimum in search space, we once again use the full optimisation described in Section VI-C, however, this time, the positions of the particles in space are initialised around the result of the hierarchical optimisation, and the initial inertia value is set to a low value, $w = 0.5$, forcing the swarm to explore the space around the provided initial solution. This approach successfully corrects the influence of the error propagation as shown in Figure 5 right. The algorithm was tested on various poses and some of the results are shown in Figures 6 and 7. The model illustrating one of the poses estimated is shown in Figure 8.

VII. DISCUSSION AND FUTURE WORK

We applied the PSO to the problem of body pose estimation and obtained encouraging results. No doubt there are many other ways in which one can modify PSO to achieve even better results, however, for the purpose of this work, the results obtained are a good starting point for the next step on the way towards the target application. Our aim is much more than just individual frame-based body pose estimation. We are working with video sequences where

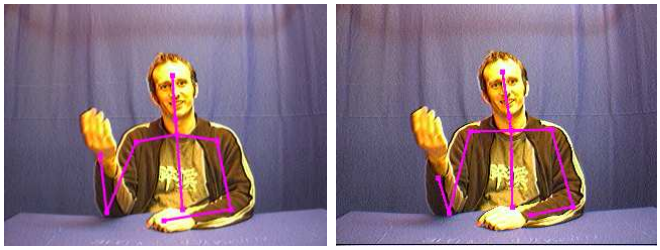

Fig. 5. The left image shows the result under the influence of the error propagation in the hierarchical approach. The right image illustrates how this can be corrected using the combined approach.

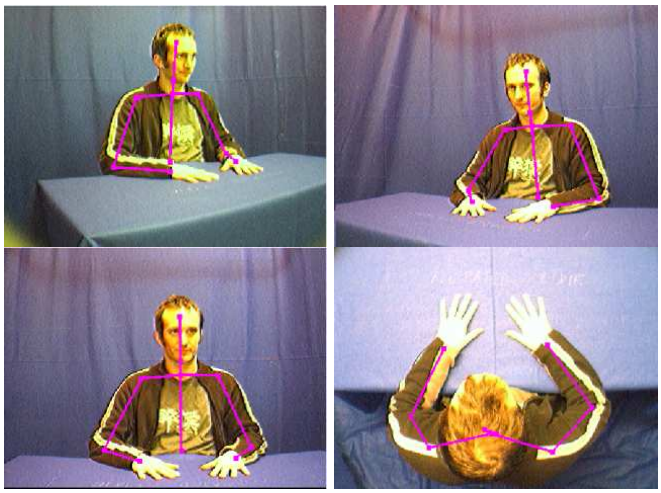

Fig. 6. Combined approach results for second pose, shown in left, right, centre, and top view.

temporal consistency is extremely important. To exploit this, we would like to extend this work to incorporate some form of tracking, possibly with PSO, so that we can produce a smoothly varying pose estimate for the entire video sequence.

The decision to use PSO as the global optimisation method was also made on the basis of knowing that there are ways to parallelise the algorithm, as reported, e.g., in [9], which can significantly increase the execution speed of the optimisation. This is an important factor in our target application. Additionally, the kinematic structure of the skeleton to some extent lends itself to a parallel approach, as it consists of three independent kinematic sub-chains which could be optimised simultaneously.

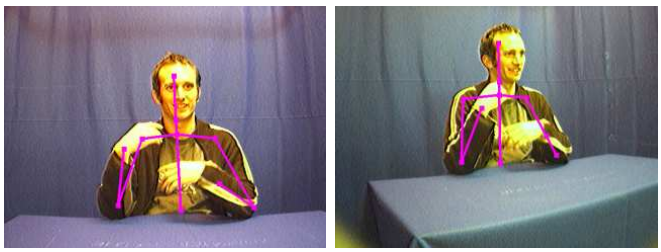

Fig. 7. Combined approach results for third pose, centre and left view.

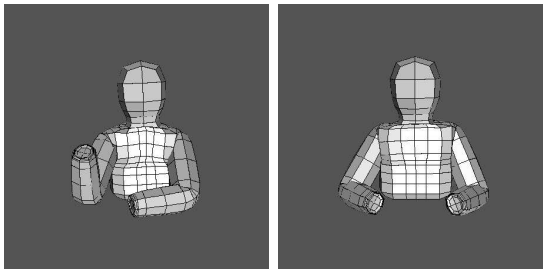

Fig. 8. The body pose illustrated with the subdivision model used in the optimisation.

ACKNOWLEDGMENTS

The authors would like to thank Craig Robertson for bringing PSO to our attention, help with setting up and configuring the camera acquisition studio and time spent discussing this work. We would also like to thank Daniel Clark for help with the test sequence acquisition.

REFERENCES

[1] F. Isgrò, E. Trucco, and O. Schreer, "Three-dimensional image processing in the future of immersive media," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14(3), pp. 288–303, 2004.

[2] S. B. Kang, "A survey of image-based rendering techniques," *Cambridge Research Laboratory Technical Report Series*, vol. 97(4), 1997.

[3] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47(1-3), 2002.

[4] D. Gavrilla, "Visual analysis of human movement: A survey," *Computer Vision and Image Understanding, 1999*, vol. 73, 1999.

[5] T. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, 2001.

[6] R. Plaenkers and P. Fua, "Articulated soft objects for video-based body modeling," in *8th International Conference on Computer Vision, ICCV 2001*, 2001.

[7] J. Carranza, C. Theobalt, M. Magnor, and H. Seidel, "Free-viewpoint video of human actors," *ACM Transactions on Graphics*, vol. 22(3), 2003.

[8] P. R., H. D., N. A., and P. M., "Towards real-time body pose estimation for presenters in meeting environments," in *Proceedings of the 13-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2005*, 2005.

[9] J. F. Schutte, J. A. Reinbolt, B. J. Fregly, R. T. Haftka, and A. D. George, "Parallel global optimization with the particle swarm algorithm," *International Journal for Numerical Methods in Engineering*, vol. 61(13), 2004.

[10] N. Litke, A. Levin, and P. Schroeder, "Fitting subdivision surfaces," in *IEEE Visualization 2001*, 2001, pp. 319–324.

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[12] R. C. Eberhart and Y. H. Shi, "Special issue on particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8(3), 2004.

[13] Y. H. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1998.

[14] E. Catmull and J. Clark, "Recursively generated b-spline surfaces on arbitrary topological meshes," *Computer Aided Design*, vol. 10, 1978.

[15] D. Doo and M. Sabin, "Analysis of the behaviour of recursive division surfaces near extraordinary points," *Computer Aided Design*, vol. 10, 1978.

[16] J. Warren and S. Schaeffer, "A factored approach to subdivision surfaces," *Computer Graphics and Applications*, vol. 24, 2004.

[17] T. Svoboda, D. Martinec, and T. Pajdla, "A convenient multi-camera self-calibration for virtual environments," *PRESENCE: Teleoperators and Virtual Environments*, vol. 14(4), 2005.