Graphical Models 70 (2008) 87–104

Contents lists available at ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

Full-body performance animation with Sequential Inverse Kinematics

Luis Unzueta [a,*], Manuel Peinado [b], Ronan Boulic [c], Ángel Suescun [a]

[a] Department of Mechanics, CEIT and Tecnun, University of Navarra, Manuel de Lardizabal 15, 20018 San Sebastián, Spain
[b] Escuela Politécnica, University of Alcalá, Spain
[c] Virtual Reality Laboratory, Ecole Polytechnique Fédérale de Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 25 May 2007
Received in revised form 26 February 2008
Accepted 27 March 2008
Available online 8 April 2008

Keywords:
Inverse Kinematics
Motion reconstruction
Human animation

ABSTRACT

In this paper, we present an analytic-iterative Inverse Kinematics (IK) method, called Sequential IK (SIK), that reconstructs 3D human full-body movements in real time. The input data for the reconstruction is the least possible (i.e., the positions of wrists, ankles, head and pelvis) in order to be usable within a low-cost human motion capture system that would track only these six features. The performance of our approach is compared to other well-known IK methods in reconstruction quality and computation time obtaining satisfactory results for both. The paper first describes how we handle the spine and the clavicles before offering a simple joint limit model for ball-and-socket joints and a method to avoid self-collisions induced by the elbow. The second part focuses on the algorithms comparison study.

1. Introduction

In performance animation, the captured movements of human performers are mapped in real time to the skeletons of virtual characters that represent their poses. The applications of this mapping are multiple, like TV production, virtual reality, workspace design, entertainment, human–machine interaction, skills acquisition and rapid prototyping of animations. Most systems addressing performance animation require the use of large sensor or marker sets, making them too cumbersome and expensive for low-cost applications such as home entertainment.

Human motion is typically represented as a series of different configurations of a rigid multibody mechanism consisting of a set of segments connected by joints. Segments correspond to body parts such as the thighs, shanks, upper arms, forearms, etc., and joints correspond to articulations such as hips, knees, shoulders, elbows, etc. These joints are hierarchically ordered and have one or more Degrees of Freedom (DoF) which represent the rotations relative to their parent joints. There is a root joint of which the position and orientation are represented with respect to the absolute coordinate system. A good example of this way of representing humanoid characters is the H-Anim standard [1]. This standard places the root joint at the pelvis, and defines a standard name for each joint, as well as a standard reference, or neutral, posture. In the context of this work we adhere to this standard.

It is often too cumbersome and time-consuming for an animator to manually set all the DoFs of a virtual character. This is solved by Inverse Kinematics (IK) techniques, in which only the positions (or sometimes also the orientations) of certain joints, usually the end-effectors, must be specified by the animator or by the motion capture system. The remaining DoFs are automatically determined according to different criteria that depend on the IK variant one employs. End-effector positions can be modeled as a function of the DoFs, leading to formal definition of the IK problem as $f(\mathbf{q}) = \mathbf{G}$, where $\mathbf{q}$ is the vector of DoFs and $\mathbf{G}$ is a vector that gathers all the desired end-effector positions. This problem is highly under-constrained as $\mathbf{q}$ usually has a much larger dimension than $\mathbf{G}$. In addition, it is a nonlinear problem as f involves complex combinations of trigonometric functions.

[*] Corresponding author. Fax: +34 943 213076.
E-mail address: lunzueta@ceit.es (L. Unzueta).

1524-0703/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.gmod.2008.03.002

Our previous work [2] presented an analytic-iterative IK method focused only on capturing motions of the upper body. The inputs to this method were the positions of hands, head and torso center, estimated with a vision-based markerless motion capture system. In the first part of this paper, we build on this previous work, extending it to the reconstruction of full-body motion in real time. The inputs of the method, which we call Sequential IK (SIK), are now the positions of wrists, ankles, head and pelvis. Several contributions are made:

- Novel methods for posing anatomically-based models of the spine and the clavicles.
- An improvement to the limb reconstruction algorithm.
- A simple and efficient method for modeling biomechanical joint limits based on cubic splines.
- A method to avoid elbow–torso interpenetrations.

In the second part of the paper, we evaluate the performance of SIK by comparing it to other well-known IK methods, with satisfactory results. We conclude that the reduced dimension of the input data and the low computation cost make the system interesting for applications that require low-cost performance animation capabilities. Also, as no pre-recorded motion database is used, the memory footprint of the method is minimal.

2. Related work

There have been different approaches to solve the IK problem that can be distinguished as: analytical, numerical and hybrid methods. Analytical methods find all possible solutions as a function of the lengths of the mechanism, its starting posture and the rotation constraints. Their advantages are their low computational cost and good accuracy compared to other methods and their drawbacks are that they can only be used for low DoF mechanisms and that they are not feasible when the system is ill-posed. Some examples of analytical solutions of multibody mechanisms are the ones proposed by Zoppi [3], Wu et al. [4] and Gan et al. [5].

On the other hand, numerical methods cover those that require a set of iterations to achieve a satisfactory solution. In this case we can find methods such as the work of Zhao and Badler [6] in which they propose to search for a plausible solution by solving a constrained nonlinear optimization. Another numerical approach is that in which the nonlinear problem is linearized using the Jacobian matrix where at each iteration an update of the DoF is obtained. There are different strategies for this update like the Jacobian Transpose method [7,8], the Pseudoinverse method [9], the Damped Least-Squares (DLS) [10,11], the DLS with Single Value Decomposition (SVD) [12,13] and the Selectively Damped Least Squares (SDLS) [14]. A review on these strategies can be found in the work of Buss and Kim [14]. The numerical approach can be enhanced by enforcing priorities to arbitrate the fulfillment of conflicting constraints such as in the Online Motion Retargetting (OMR) of Choi and Ko [15] and the Prioritized IK (PIK) of Baerlocher and Boulic [16]. The main difference between OMR and PIK is that OMR has only two levels of priority while PIK can have any number making the latter more suited to our purpose. The potential of PIK for full-body motion capture was explored by Peinado et al. [17]. Raunhardt and Boulic [18] use PIK for the reconstruction of human spines. They use both equality and inequality constraints to model the coupling behavior of the spine and reduce the search-space, achieving natural spine shapes. An alternative to this approach for reconstructing the spine is the work of Boulic et al. [19]. It is an approximate method for distributing a relative thorax/pelvis orientation on a set of vertebrae according to their anatomic behavior. It is intended to provide fast qualitative results, which suffice in the synthesis of walking animations.

Other numerical methods are based on neural nets and artificial intelligence such as those proposed by Oyama et al. [20] and D'Souza et al. [21] in which the mechanism rearranges learned movements to reach the targets. A related approach is to reconstruct postures based on a database of prerecorded motions, as in the recent works of Grochow et al. [22], Chai and Hodgins [23] and Liu et al. [24]. The drawback of methods that rely on databases is that if desired postures are too distant from those of the database, odd results are obtained. Another approach is the Cyclic Coordinate Descent (CCD) algorithm used by Wang and Chen [25] where the joints of a kinematic chain are rotated one by one starting from the root a certain step-angle reducing the difference between the end-effector's current position and orientation and a full iteration is performed when all the joints have been rotated.

Finally, hybrid methods are those that combine both analytical and numerical algorithms such as those proposed by Tolani et al. [26] for upper and lower limbs, which we will refer to as the TGB (Tolani–Goswami–Badler) method, and for the human full-body reconstruction such as the method proposed by Shin et al. [27]. Kulpa et al. [28] use CCD and TGB to readapt, in real-time, pre-recorded animations to certain constraints such as feet–ground contact. They apply these algorithms separately in the different body parts in which they subdivide the humanoid; the head, the two arms, the two legs and the trunk. Their work can also be used for full-body reconstruction if, instead of using the postures of an animation for their readjustment, a neutral posture such as a standing pose is used as a starting posture. We will refer to this approach as the KMA (Kulpa–Multon–Arnaldi) method. A comparison between analytical and numerical solutions for the reconstruction of the upper body and limbs was made in our previous work [2] where the advantages and disadvantages of both can be observed.

There have been previous studies in which known orientations are used for motion reconstruction such as those of Monheit and Badler [29] and Molet et al. [30] for the spine, Zordan and Hodgins [31] for the upper-body, and Badler et al. [32] and Semwal et al. [33] for the full-body. Nevertheless we are more interested in adjusting the poses only to known positions instead of orientations, since it is more intuitive for an animator to situate them or for a motion capture system to track them. This opens up new possibilities for the simple control of avatars and for markerless human motion capture based in computer vision.

3. Sequential Inverse Kinematics

Most popular motion capture systems reconstruct human poses by tracking several markers placed over the body of the performer. As Liu et al. [24] state, these data exhibit considerable redundancy. They demonstrate that there is a reduced set of markers, the principal markers, which retain the essential information of movement. Depending on the type of movement, some are more important than others. But in general, we could state that the positions of the end-effectors (i.e., the wrists in the case of the arms, the ankles in the case of the legs, and the pelvis and the head in the case of the torso), are important to situate a human pose. There are exceptions like those in which the size of the captured user is different from the virtual character and, as Shin et al. [27] state, we could be more interested in preserving the angles of the original pose instead of the positions of the end-effectors because they are not interacting with the environment. However, as stated in the previous section, we are more interested in end-effector positions as they are more easily specified by an animator and tracked by motion capture systems. Also, using position data opens up the possibility of interacting with a virtual environment.

The main idea of our approach is that the reconstruction is solved sequentially using simple analytic-iterative IK algorithms in different parts of the body in a specific order. No pre-recorded motion database is necessary, thereby avoiding the need for extra memory. First the orientation of the root joint is estimated from the known positions. The configuration of the spine is found using a hybrid IK method that combines this estimated orientation with the positions of the root and head markers. Then the orientations of clavicles are determined with the positions of their corresponding known end-effector positions and the already positioned spine. Finally each of the limbs is situated according to their known end-effector positions with an analytic IK method. Complex biomechanical rotation limits are modeled from only a few known anatomical data to constrain the joint orientations and prevent elbows from penetrating the torso in order to obtain visually plausible human poses. Algorithm 1 shows the general procedure of our approach. Its novel aspects are described in the following sections.

Algorithm 1. Sequential Inverse Kinematics algorithm

```
procedure SIK ($pos_{pelvis}, pos_{head}, pos_{wrists}, pos_{ankles}$)
    Estimate $rot_{pelvis}$ from $pos_{pelvis}$, $pos_{head}$, $pos_{wrists}$ and $pos_{ankles}$
    Reconstruct the spine from $rot_{pelvis}$, $pos_{pelvis}$ and $pos_{head}$
    Reconstruct the legs from $pos_{ankles}$
    Reconstruct the clavicles from $pos_{wrists}$
    Reconstruct the arms from $pos_{wrists}$
end procedure
```

4. Spine reconstruction

The reconstruction starts with the torso and involves two steps. In the first one we estimate the orientation of the pelvis, which we do not know since we only consider the positional information of the end-effectors and pelvis. This differs from the work of Monheit and Badler [29] where the known data are the orientations instead of the positions. In the second step we find a suitable configuration for the spine from the knowledge of both end points.

For the first step we extend the approach explained in [2] by also incorporating the known positions of the feet. Taking into account the H-Anim specification [1] for axes definition, we propose to estimate the pelvis orientation as follows:

- *Y* direction defined by the vector that goes from the pelvis to the head which is subsequently normalized
- *X* direction defined by a weighted average of three vectors, the first one from the right wrist to the left one, the second one from the right ankle to the left one, both projected and normalized in the plane whose normal is the *Y* direction, and the third one, the *X* axis of the previous frame's pose
- *Z* calculated as the cross-product of *X* and *Y*

This novel approach to estimate the global orientation of the body is applicable to a broad range of different movements with visually satisfactory results. The weights used for the calculation of the *X* axis (called, respectively, w_1, w_2 and w_3) control the dependency level of the torso's axial rotation with respect to the positions of the upper and lower end-effectors. The third vector is used to increase the *rigidity* of its movements. Once the orientation of the pelvis (root joint) is established, one can proceed to solve the reconstruction of the spine bearing in mind that the objective is to achieve a visually acceptable result, not an exact solution.

As a simple example to visualize the philosophy of this approach, consider a totally straight spine composed of five equally spaced joints and where only the positions of the end-effectors are known. A visually acceptable and reasonable solution for the reconstruction of a human-like spine of this type would be the one shown in Fig. 1 where both end-effectors get closer to each other. In this 2D solution, the root joint or pelvis is rotated by ϕ_{root} and the other joints are rotated in the opposite direction with an equal amplitude ϕ so that the last segment orientation is $-\phi_{root}$. This solution can be represented analytically for n joints with Eqs. (1) and (2). Due to the non-linearity of the latter equation the value of ϕ is obtained iteratively.

$$\phi = \frac{2\phi_{root}}{n-2} \tag{1}$$

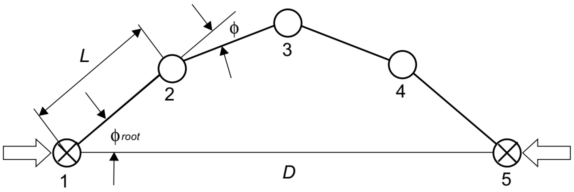

Fig. 1. The readjustment of an equally distributed straight spine with five vertebrae.

$$D = L[\cos\phi_{root} + \cos(\phi_{root} - \phi) + \cos(\phi_{root} - 2\phi) + \cdots \cdots + \cos(\phi_{root} - (n-2)\phi)] \quad (2)$$

Real human spines do not have the same length for all vertebrae and they are not fully extended in their resting posture. Despite these facts experiments show that we are able to obtain visually acceptable postures of n vertebrae spines with the procedure that appears in Algorithm 2 which is based on the aforementioned solution. The plane in which the spine would be bent is that composed of the known (or goal) and current (or previous frame's) positions of the pelvis and the head, i.e., the normal of this plane would be that obtained from the cross product of two vectors; one formed with the known positions of the pelvis and head and the other with their current positions. The plane calculated this way allows rotating the spine in any direction, not only forward–backward. In case of alignment, the previous frame's bending plane is used. In this algorithm, L_{obj} is the distance between the known positions corresponding to the root joint and the head joint. $L_{current}$ is the length measured between these joints of the virtual humanoid for each iteration. $L_{current}$ is intended to reach L_{obj}. Finally, $L_{neutral}$ is the same measure but when the joints are in their neutral or resting posture.

Algorithm 2. Spine reconstruction algorithm

```
procedure SpineReconst ($pos_{pelvis}, pos_{head}, rot_{pelvis}$)
  Pelvis_Joint_Position ← $pos_{pelvis}$
  $L_{obj} \leftarrow |pos_{head} - pos_{pelvis}|$
  if $L_{neutral} > L_{obj}$ then
    if $L_{current} > L_{neutral}$ then
      All_Joint_Orientations ← Identity_Rot
    end if
    Rotate joints till satisfactory $L_{current}$ is
    obtained ▷ (Algorithm 3)
  else
    if Spine not totally stretched then
      Get the corresponding pre-recorded pose
    end if
  end if
  Rotate pelvis to align its X axis with that of
  $rot_{pelvis}$
  Set the spine joints within biomechanical
  limits if necessary
  Rotate pelvis to align the vector that connects
  head and pelvis joints with vector
  $pos_{head} - pos_{pelvis}$
  Translate pelvis along vector $pos_{head} - pos_{pelvis}$
  to distribute the positional error of the end-
  effectors evenly
end procedure
```

In human-like spines there is a curvature on its neutral posture so if our known end-effector positions are too distant the spine must be stretched out completely. The process to calculate these stretching postures is the same as the one presented where the joint rotation limits are those in which their corresponding segments are aligned with the vector that goes from the pelvis to the head. This process could result in excessive computational costs compared with the rest of the posture calculations. This occurs because the stretching postures are close to the singular configuration of the spine, and adjusting the angles is not very efficient for lengthening the spine. Therefore, we use a set of pre-computed stretching postures to speed up this transition and so that there is a unique way of stretching the spine. Then, if after stretching completely the spine we come back to reachable positions, the spine would not recover its original shape again for $L_{current} = L_{neutral}$. For this reason, when $L_{current} > L_{neutral}$ we must set the spine in the neutral posture for further calculations. The iterative process to get a satisfactory value of $L_{current}$ when $L_{neutral} > L_{obj}$ is shown in Algorithm 3.

This method is employed because it is impossible to express in simple terms the explicit equations in the 3D spine for every bending plane. If the step angle is too small the process could become too slow. For this reason the iterative process is done in two phases. First, it is solved by a coarse step angle until it "overshoots" a bit and then we iterate "back the other way" with a refined step angle until it also "overshoots" a tinier bit. An example of the readjustment of a complete human spine can be seen in Fig. 2.

Algorithm 3. Satisfactory $L_{current}$ search algorithm

```
procedure SatisfactoryLengthSearch
($L_{current}, L_{obj}, L_{neutral}$)
  if $L_{current} > L_{obj}$ then
    $sign = +1$
  else
    $sign = -1$
  end if
  while $(sign \times L_{current}) > (sign \times L_{obj})$ do
    Rotate pelvis the corresponding step angle ▷
    ($\approx -sign \times 0.1$ rad)
    Rotate remaining joints according Eq. (1)
    Update $L_{current}$
  end while
  while $(sign \times L_{current}) < (sign \times L_{obj})$ do
    Rotate pelvis the corresponding step angle ▷
    ($\approx sign \times 0.01$ rad)
    Rotate the rest of the joints according to Eq.
    (1)
    Update $L_{current}$
  end while
end procedure
```

Finally, it must be stressed that this methodology focuses on contexts in which only one position is known for the pelvis and one for the head (this is typical of vision-based mocap). As a consequence, there is no reliable means for evaluating the relative twist between the pelvis and the head. For this reason such angular twist is not handled in Algorithms 2 and 3. If we could also measure the pelvis and head orientations we would be able to distribute the relative twist quantity along the spine. At this point it must be remarked that at every iteration the joints are set within biomechanical limits in case it is necessary. Due to

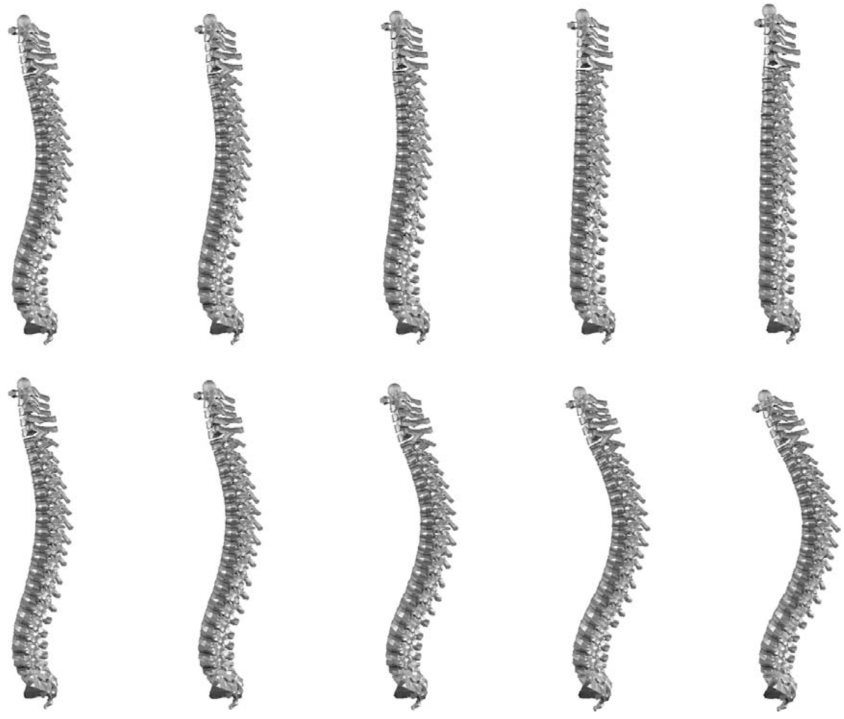

Fig. 2. The resulting readjustment of a complete spine moving on its sagittal plane from the known positions of head and pelvis.

their very limited mobility along the twist axis, the lumbar vertebrae would probably transgress the biomechanical limits at every iteration. So the algorithm could be adapted to take advantage of such a priori knowledge by distributing the twist rotation only in the thoracic and cervical regions. To conclude this discussion, we have observed that the visual influence of this lack of relative twist is not very critical compared to other features of the pose as evaluated in Section 9.

5. Clavicle reconstruction

Once the spine is set the positions of the sternoclavicular joints are also defined. We simplify the *shoulder complex* in a similar manner as Badler et al. [34], i.e., we consider that it is composed of only two joints, which, adhering to H-Anim terminology [1], are called *sternoclavicular* and *shoulder*. These joints are connected by the clavicle segment, resulting in a shoulder model that ignores the scapulothoracic articulation. In Badler's book, the shoulder model of Otani [35] is exploited in order to simulate the observed synergies between the sternoclavicular and the shoulder joints. Because this model expresses the relative distribution of arm elevation and abduction between these joints, it can only be implemented using the concept of joint group.

In our IK solver joint groups are avoided for efficiency reasons, and thus, instead of trying to apply Otani's model we have chosen to apply a heuristic criterion of cost minimization: when the wrist marker can be reached using motion of the shoulder joint alone, the sternoclavicular joint remains in its current configuration. If, on the other hand, after positioning the shoulder joint the target remains unreachable, we change the configuration of the sternoclavicular joint so that the target is reached. This is consistent with the way the *shoulder complex* behaves, as clavicles tend to remain still unless (a) their motion is required for reaching the end-effector's goal, or (b) they are deliberately actuated, which is out of our control given the scarcity of our input data. This approach for the reconstruction of a clavicle is given in Fig. 3.

ST, *SH*, *E* and *W* are the sternoclavicular, shoulder, elbow and wrist joints, L_1 and L_2 are the lengths from *SH* to *W* and from *SH* to W_{obj}, respectively and W_{obj} is the end-effector's known position. θ_1 and θ_2 are the angles between the clavicle segment and the vector that goes from *ST* to *W* and between the clavicle and the vector that goes from *ST* to W_{obj}, respectively. The procedure for readjusting the clavicle is shown in Algorithm 4. An example is given in Fig. 4.

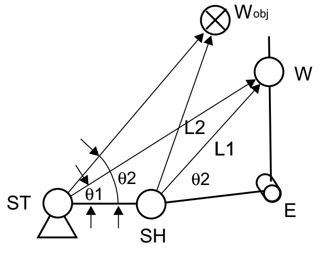

Fig. 3. Diagram of the clavicle readjustment.

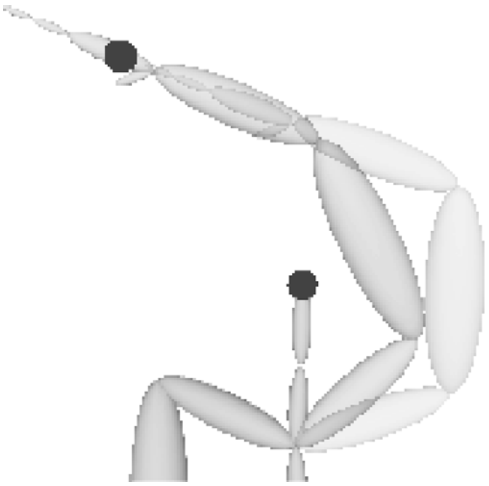

Fig. 4. The resulting readjustment of a clavicle when its corresponding arm cannot reach the known wrist position by itself.

Algorithm 4. Clavicle reconstruction algorithm

```
procedure ClavicleReconst ($W_{obj}$)
  $L_1 \leftarrow |W - SH|$
  $L_2 \leftarrow |W - ST|$
  if $L_2 > L_1$ then
    Calculate $\theta_1$ and $\theta_2$ with the cosine theorem
    if $\theta_2 > \theta_1$ then
      $v \leftarrow$ axis perpendicular to the $ST\_SH\_W_{obj}$ plane
      $\alpha \leftarrow \theta_2 - \theta_1$
      Rotate $ST$ $\alpha$ radians about axis $v$
      Set $ST$ within biomechanical limits if necessary
    end if
  end if
end procedure
```

6. Upper and lower limb reconstruction

Now that the postures of the central parts of the body have been determined, the arms and legs can be adjusted to their corresponding end-effectors. In our previous work [2], we explained how to calculate the ψ and φ angles (left image of Fig. 5) analytically, in a similar way as in the TGB method. Regarding the swivel angle (right image of Fig. 5), we minimized a cost function in order to attract joint A towards the mid-range of its biomechanical Euler angle limits. Using a similar approach, Kang et al. [36] calculated a swivel angle that would minimize the torque. Due to the nonlinearity of its parameterization, these approaches require solving numerically an optimization problem.

To find suitable configurations of the limbs we use Algorithm 5, which exploits biomechanical articular limits, prevents self-collisions and deals with non-reachable end-effector positions. Initially joint A is oriented analytically taking into account biomechanical limits and, in the case of the arms, the fact that the elbow cannot penetrate the torso. Then joint B is oriented, also analytically, to get the least possible error between the limb's end-effector's position and its known one. To avoid the computational burden of performing a numerical optimization to solve for the swivel angle, in our method we always start from the neutral configuration of the limb (i.e., totally stretched with joint A aligned to its parent). This yields reasonable postures, with the additional benefit of making the swivel angle independent of the postures in previous frames.

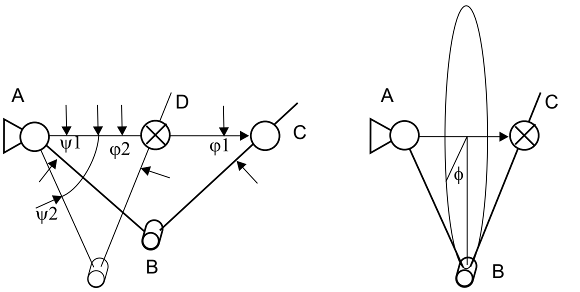

Fig. 5. Diagram of the readjustment of a limb. On the left, the analytically calculable angles, and on the right, the undetermined swivel angle.

Algorithm 5. Limb reconstruction algorithm

```
procedure LimbReconst ($D$)
  Rotate $A$ to align $AC$ and $AD$ vectors
  if $|AB| + |BC| > |AC|$ then
        Rotate $A$ around X axis of $B$ to match with $\psi_2$
  end if
  Set $A$ within biomechanical limits if necessary
  Prevent $B$ interpenetration with the torso (only for the elbow)
  $v \leftarrow$ projection of $BD$ vector in the XZ plane of $B$
  Rotate axially $A$ to align Z axis of $B$ with $v$
  Set $A$ within biomechanical limits if necessary
  Rotate $B$ around its X axis to align the $BC$ and $BD$ vectors
  Set $B$ within biomechanical limits if necessary
end procedure
```

7. Biomechanical limits

Grassia [37] compared different parameterizations of rotation and concluded that, in general, no single parameterization is best. The use of a particular parameterization depends on its performance in the application. In principle, it is possible to limit the rotations regardless of the employed parameterization. For example, it could be possible to limit directly the Euler angles, but the achieved workspace would not be realistic for modeling the behavior of a human joint such as the shoulder. For this reason we propose a method to limit the orientation of the ball-and-socket joints similar to the spherical polygons used by Korein [38] and reviewed by Baerlocher and Boulic [39].

Three types of rotation ranges are distinguished: simple flexion, swing and twist (Fig. 6). Simple flexion occurs only for the elbow and the knee joints. It is not difficult to limit this single DoF. On the other hand limiting swing and twist is not a straightforward task.

The limits of swing movements vary depending on the orientation of the direct child joint. For example, the swing limits of a hip have higher values if the corresponding knee

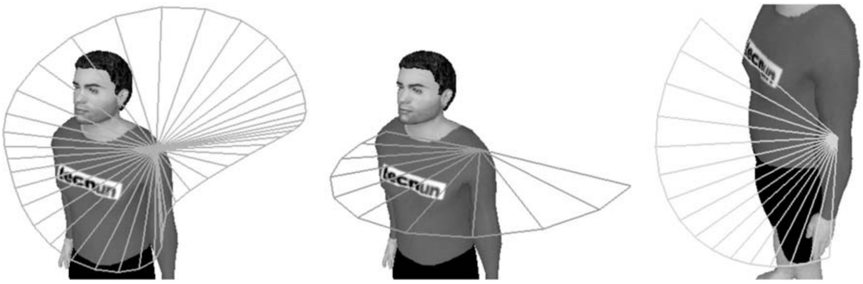

Fig. 6. From left to right: swing and twist limits of a shoulder and flexion limits of an elbow.

is bent instead of stretched. This means that the joint limits are coupled. Our purpose is to get visually pleasing results so these couplings are not considered in order to simplify calculations.

We model swing movements using a spherical parameterization of orientations. The segment that depends directly on the orientation of the joint has a constant size so the radius employed in the spherical coordinates is not taken into account. Only the other two angles, which we will call *circumduction angle* or θ, and *swing amplitude* or ψ, are considered.

The range of θ goes from -180° to +180°, and for each value there is a corresponding biomechanical limit of ψ. In the spherical polygon representation the workspace boundary of the joint is defined by a set of vertices situated on the surface of a unit sphere connected by great arcs which would be the shortest path that bind two points on the sphere. The more vertices we have, the smoother and more realistic the boundary is, but more measurements of the swing limits will be needed. There are some studies of these measurements such as the work of Engin and Chen [40] and Herda et al. [41].

Our alternative is to use a reduced set of biomechanical limits that can be obtained from books such as Kapandji's [42], [43] and [44]; for example the maximal vertical flexion and extension of the shoulder, etc., and apply a cubic spline to these points to get a visually reasonable and smooth limitation of the swing movements (left image of Figs. 6 and 7). The first derivative of the spline at its starting and ending points ($\theta = -180^\circ$ and $\theta = 180^\circ$, respectively) is the same to get a smooth boundary over the entire circumduction movement. Its value is obtained with Eq. (3),

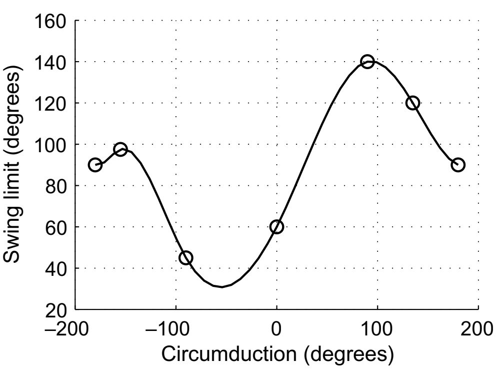

Fig. 7. Modeling of the swing biomechanical limits of a shoulder with a cubic spline.

$$\left(\frac{d\psi}{d\theta}\right)_1 = \left(\frac{d\psi}{d\theta}\right)_n = \frac{1}{2}\left(\frac{\psi_2 - \psi_1}{\theta_2 - \theta_1} + \frac{\psi_n - \psi_{n-1}}{\theta_n - \theta_{n-1}}\right) \tag{3}$$

where the subindices refer to the indices of a vector containing the known θ values (i.e., subindex 1 refers to the starting point and subindex n to the ending point). So to limit the swing amplitude of a joint we only have to calculate its current θ and ψ values. If ψ is above its higher bound, we simply rotate the joint maintaining the same θ, until the maximum value of ψ is reached. In order to limit the twist or axial rotation we need a reference orientation to which the current orientation is compared. We propose to define this reference orientation by, first, considering the orientation of the parent joint as the neutral orientation of the current joint and, second, rotating it with the θ and ψ values corresponding to the current orientation. This way the reference orientation differs from the current one only on the twist rotation, so that the current orientation can be easily set within axial biomechanical limits. These limits vary depending on the swing orientation of the limb as was demonstrated by Wang et al. in [45] for the case of the shoulders. For a visually acceptable result, though, the average values provided in the above mentioned books of Kapandji would be sufficient.

8. Self-collisions avoidance

Biomechanical limits prevent the humanoid from performing many unnatural poses, but self-collisions can still occur. This is a complex problem with many different peculiarities. In this section we present our approach, which prevents one of the most common self-collisions in posture reconstruction; the penetration of the elbows in the torso. First the penetration depth (PD) must be estimated and then the proper response must be applied to get the elbow out of the torso.

There are many studies on PD calculations of colliding figures formed by meshes such as the different body parts of the humanoid that appears in Figs. 6 and 9. Zhang et al. [46] review the different approaches to achieve this objective and present a method to compute the PD taking into account possible rotations throughout the path in order to separate the overlapping objects. Nevertheless, these methods do not take into consideration the limb constraints a multibody system is subject to for repositioning in case of self-collision. The strategy to avoid self-collisions must be integrated within the IK algorithm as it is shown in the works of Kallmann [47] and Peinado et al. [48]. In the present study we integrate a strategy to reorient the upper arms in the SIK method as shown in Algorithm 5.

We focus our interest on a region comprising the upper arm and forearm mesh vertices that are near the elbow joint. For simplicity in Fig. 8 we represent this section as a circle. We conceptualize the meshes of the segments that compose the torso as prismatic bodies, which is not far from reality in most human-like figures. Let p_t be the outer vertex of the colliding torso segment, p_e the deepest vertex of the region of interest of the arm, and P the XZ plane of the spine joint that contains the colliding torso segment. This way we define the depth of the elbow within the torso as the distance from p_t to p_e, in the direction that goes from p_e to p_t projected in P, and that passes through the elbow joint (see Fig. 8 and left image of Fig. 9). We not only obtain the depth value but also two points that represent the deepest point of the elbow and the outer point of the colliding torso segment which define the penetration direction.

Once we have these measurements we are able to make the upper arm amend its orientation while we readjust the arm so that its end-effector matches the known position. We distinguish two cases. In the first one, the arm must be almost or fully stretched to reach the end-effector. Let **v** be vector that goes from the shoulder to the deepest point of the elbow, and **w** the one goes from the shoulder to the outer point of the colliding torso segment. These vectors are assumed to be normalized to length 1. In this case, we solve the elbow collision by rotating the upper arm about the axis defined by the cross product of **v** and **w**. This rotation is done iteratively. The angle applied on each iteration is given by $\arccos(\mathbf{v} \cdot \mathbf{w})$. This value is a lower bound of the one that might be necessary to solve the collision, thus the need for performing several iterations.

In the second case, the one in which the arm is not stretched, we make the upper arm rotate around the swivel angle axis shown in the right image of Fig. 5. Then as in the previous case we rotate iteratively until the elbow exits the torso. In this case it is not easy to determine a lower bound to the optimal angle. We propose rotating an angle which is k times the ratio between the depth and the length of the upper arm. We adjust the proportionality factor k empirically to a value of 1. In Fig. 9 an example of a corrected arm posture following this strategy is given.

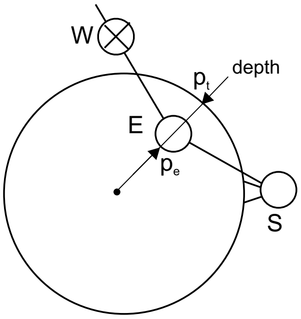

Fig. 8. Diagram of the right elbow penetration depth estimation.

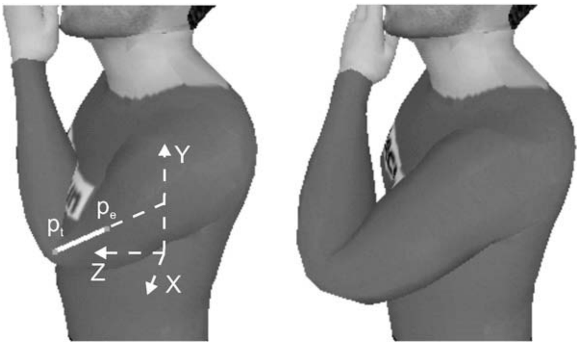

Fig. 9. Left elbow torso penetration: on the left the estimated penetration depth and on the right the penetration avoidance response posture.

9. Experimental results

Finally we compare the SIK method with other well-known approaches, specifically the KMA [28], CCD [25], Jacobian Transpose [7,8], Pseudoinverse [9], DLS [10,11], DLS with SVD [12,13], SDLS [14], PIK [16,17], and the TGB method for anthropomorphic limbs [26] combined with the reconstruction methods presented here for the spine and clavicles. We call the latter TGBSIK. The test consists of reconstructing three sequences with different classes of motions in order to have available different swivel angles of the upper and lower limbs and axial orientations of the pelvis (root joint): the first one with 2783 frames from typical boxing movements (Fig. 10), the second one with 513 frames from a jump kick action (Fig. 11), and the third one with 4510 frames from a playground activity, which involves walking, climbing, hanging and swinging actions (Fig. 12). These animations have been obtained from a marker-based motion capture system and have not been filtered. This way robustness of the algorithms to noisy movements can be analyzed. This noise is more noticeable in the playground animation.

First, the positions of the wrists, ankles, head and root joints are extracted, then the orientation of the root joint is estimated as explained in the spine reconstruction section and finally the different reconstruction methods are applied to a humanoid identical to the original. The weights employed for the estimation of the root joint in the first two animations are $w_1 = 1$, $w_2 = 1$ and $w_3 = 0$, while for the third one are $w_1 = 0.1$, $w_2 = 0.1$ and $w_3 = 1$. This way the reconstructed walking action embedded in the playground animation contains more realistic movements of the torso as its axial *rigidity* is increased with respect to the wrist and ankle vectors' oscillations. In order to make the anthropometry configurable we distinguish the following parameters: foot length, calf length, thigh length, pelvis length, pelvis width, torso length, torso width, neck length, upper arm length, forearm length and

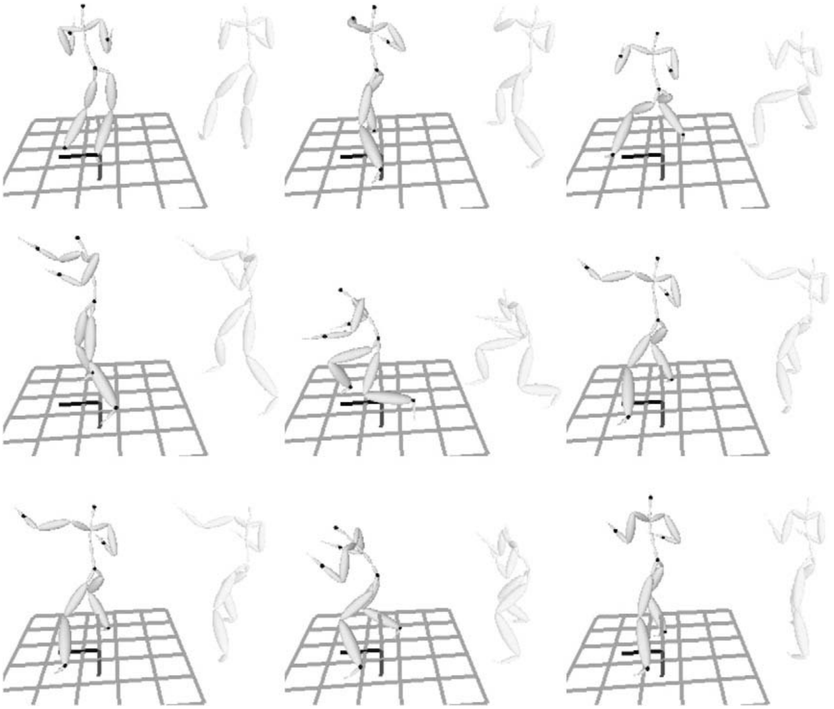

Fig. 10. Samples of the boxing animation: on the left of each sample the SIK reconstructions and on the right the original postures.

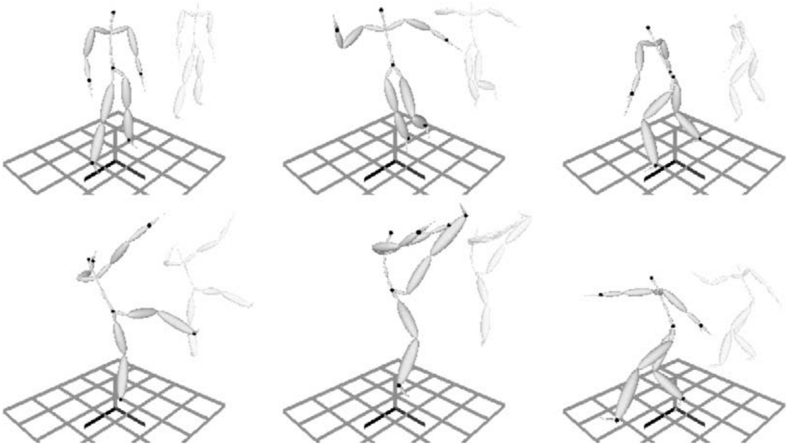

Fig. 11. Samples of the jump kick animation: on the left of each sample the SIK reconstructions and on the right the original postures.

hand length. Based on these parameters we define the height of the humanoid as the sum of the calf, thigh, pelvis, torso and neck lengths and we rescale it for the humanoids of the three sequences so that they are 1.86, 1.82 and 1.83 m tall, respectively. All three humanoids have 34 joints, according to the H-Anim specification [1] (Fig. 13), and for the reconstruction, the relative movements of all the joints except those that derive from wrists and ankles are considered. In this hierarchy *HumanoidRoot*, *sacroiliac* and *vl5* joints are distinguished in order to be compatible with the specification. But in reality, these joints are placed in the same position and they do not have relative motion. These humanoids do not have a mesh that defines their external shape so self-collisions are not considered in these reconstructions.

In the KMA method a constraint is associated to each known position or marker and their priority goes from top to bottom like this: root marker, head marker, ankle markers and wrist markers. The reconstruction of each frame starts always from the standing pose. First the root joint is placed. Then the CCD algorithm is applied to the spine. The joints are set within biomechanical limits at each step if necessary with the method presented here and the positional error of the end-effectors of the spine

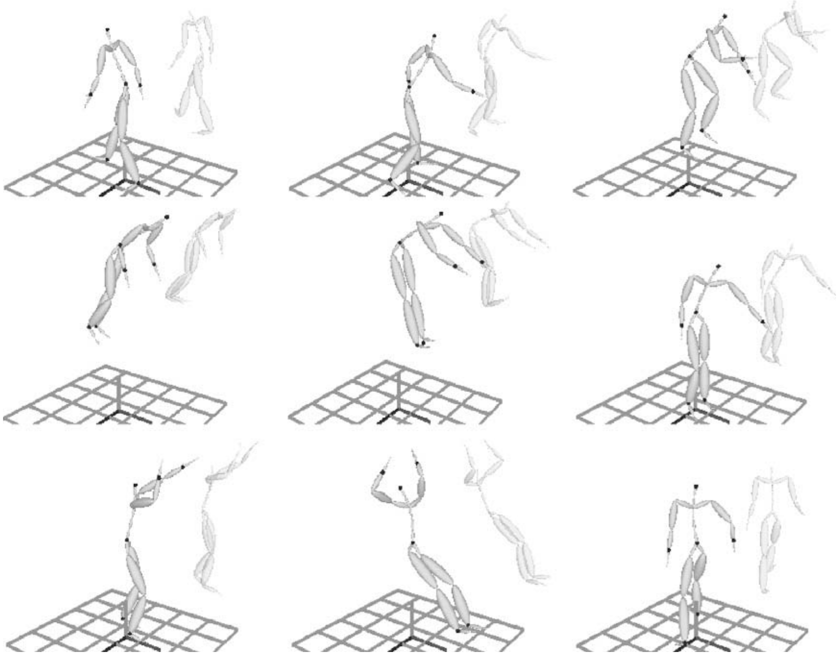

Fig. 12. Samples of the playground animation: on the left of each sample the SIK reconstructions and on the right the original postures.

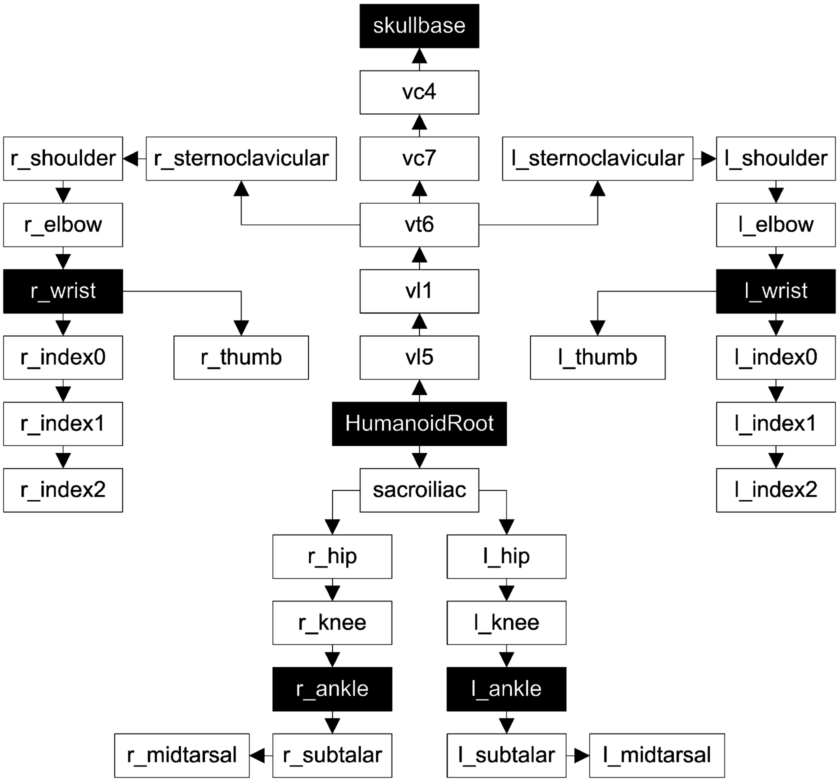

Fig. 13. H-Anim [1] structure of the humanoids of boxing, jump kick and playground animations. Highlighted joints are those whose positions are known.

is evenly distributed as was explained in the spine reconstruction section. The iterative process stops when the difference between the end-effector's known and current positions are less than 1 cm or when that error does not improve from the previous iteration in more than 1 mm or when a maximum number of iterations, which we set at 20, has been reached. The clavicles do not perform any relative movement and the upper and lower limbs are situated using the TGB method with swivel angles equal to zero. Finally the joints of the limbs are set within biomechanical limits if necessary. The references for the swivel angles are:

- For the arms the $-Y$ axis of the estimated pelvis orientation. In case of alignment with the axis that connects the shoulder with the wrist it is changed to the $-Z$ axis of the estimated pelvis orientation.
- For the legs the +*Z* axis of the estimated pelvis orientation. In case of alignment with the axis that connects the hip with the ankle it is changed to the +*Y* axis of the estimated pelvis orientation.

There is a slight difference between the original KMA algorithm and this adaptation regarding the reconstruction of the spine. In the original KMA method the spine is subdivided into three segments (neck, torso and abdomen) to which the CCD algorithm is applied. The positions of the remaining vertebrae are then sampled from a spline that passes through the joints of these three segments. This procedure has been avoided in our test in order to simplify calculations; the reconstruction result does not differ significantly from the original KMA approach.

In the CCD method first the root joint is placed, then the CCD algorithm is applied separately, first to the spine, and then to each clavicle and arm together, and to the legs. The process is the same as in the spine of the KMA method. In this case the starting posture is the one of the previous frame.

For the Jacobian Transpose, Pseudoinverse, DLS, DLS with SVD and SDLS methods, the humanoid is modeled by decomposing the joints in 1 DoF articulations as if they represented the orientation like *XYZ* follower Euler angles. As in CCD, the biomechanical limits are checked at each iteration and the stop criteria is the same. Using these methods, the root is initially placed at its known position and after making each iteration the error in the end-effectors of the spine is redistributed as stated before. The damping factor of the DLS method is set to 75 to get stable reconstructions near singular poses. This factor has been determined experimentally and lower values give the humanoid jerky movements.

With the PIK method, priorities go from top to bottom like this: root marker, ankle markers, wrist markers and head marker. The root position is the most important because the remaining joints depend on it. The priority of ankles and wrists is changeable and the head is the least important in order to obtain the smallest degree of error since the mobility of arms and legs is higher than that of the spine. With this method the joint limits are solved by inequality constraints i.e limits are checked and if any have been transgressed, an equality constraint is added and the iteration is recalculated. Once all constraints have been accomplished, the swivel angles of arms and legs are determined by optimizing a cost function expressed in the joint space in order to attract the obtained posture to a standing pose in which the arms and legs are slightly bent; the elbows backwards and the knees forward. This way one can attain better convergence and a more *relaxed* posture, distant from singular configurations like those in which the limbs are totally stretched.

In the case of the TGBSIK method the swivel angle is optimized by minimizing the equation $f(\phi) = swing^2 + weight \times twist^2$, where ϕ is the swivel angle and both swing and twist values of shoulders and hips are intended to be minimized to obtain the biomechanical configuration as far as possible from the joint limits. We use a weight of 1 in this function so both twist and swing have the same importance. As this function is non-linear and it is impossible to express the explicit equations in simple terms for each posture, we use the same optimization method as in the case of the spine, but in this case we look for the global minimum throughout the entire space (i.e., the whole 2π rotation of ϕ) since there may be more than one local minima. This approach is the same as in our previous work [2] but using the method presented here to model the biomechanical limits instead of Euler angles.

The performance of the IK methods is measured by comparing the differences in the positions and orientations of the joints of the reconstructions with respect to the original sequences, and by measuring the median computation time of one frame on the animations. The joints that derive from wrists and ankles are not considered because they are not controlled by IK algorithms. Therefore 22 joints are used for the error computations. The position error is the Root Mean Square (RMS) value of the sum of the differences between the positions of the reconstructed and the original joints divided by the height of the humanoid in order to have an adimensional value independent from the subject's dimensions, and also divided by the number of joints to get an average error per joint. The orientation error is the RMS value of the sum of exponential map modules of the reconstructed joints orientation relative to the originals divided again by the number of joints. These orientations are computed respect to the parent joint's coordinate system. The position error is more reliable than the orientation error because a little difference in angles of interior joints might lead to considerable visual differences but we also include the latter because focusing only on positions does not provide information about the differences in the orientations of the root and end-effector joints and the twist of the interior joints. RMS values are adequate because they compute the standard deviation of the distribution of the error around the desired value. An alternative to these measurements could be the cloud of points used by Kovar et al. [49]. The median computation time is computed instead of the mean value because it is more robust to outliers.

Fig. 14 shows that the Jacobian Transpose method gives poor results due to the obtained jerky movements and because there is a significant difference between the end-effector positions with respect to their known values. This happens because this method is not appropriate to handle unreachable end-effector positions and because there is more than one target at the same time which is not suitable for this approach. For this reason this method is excluded for the remaining comparisons.

Table 1 shows the RMS values in the reconstruction quality and computation time averaged per joint. The system was implemented using C++, and tested on a 2.66-GHz Pentium 4 with 512 MB RAM.

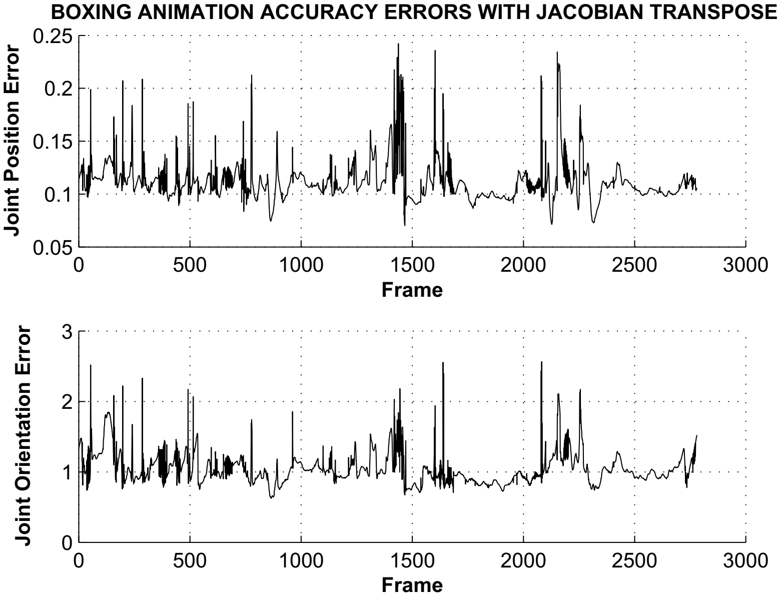

Fig. 14. Boxing animation accuracy errors with Jacobian Transpose.

Table 1
Reconstruction comparison

IK Methods		KMA	CCD	PsInv	DLS	DLS_{SVD}	*SDLS*	*PIK*	SIK	TGBSIK
Joint average	Boxing	*0.0281*	0.0316	0.0303	0.0325	0.0325	0.0293	*0.0283*	**0.0265**	0.0312
RMS position	Jump kick	0.0256	0.0292	0.0260	0.0291	0.0293	*0.0245*	0.0247	*0.0213*	**0.0208**
Error	Playground	0.0295	0.0275	0.0275	*0.0256*	*0.0257*	0.0265	**0.0240**	0.0261	0.0266
(normalized)	Average	0.0277	0.0294	0.0279	0.0291	0.0292	0.0268	*0.0264*	**0.0246**	*0.0262*
Joint Average	Boxing	*0.2524*	0.4376	0.3157	0.3115	0.3113	0.3060	0.3256	**0.2508**	*0.2937*
RMS orientation	Jump kick	0.3175	0.4651	0.3413	0.3331	0.3336	*0.3119*	0.3447	**0.2849**	*0.2912*
Error	Playground	**0.3452**	0.4606	0.3786	0.3492	0.3495	0.3626	*0.3472*	*0.3469*	0.3475
(radians)	Average	*0.3050*	0.4544	0.3452	0.3313	0.3315	0.3268	0.3392	**0.2942**	*0.3108*
Median	Boxing	*1.11*	*0.87*	17.46	11.45	20.63	15.38	28.56	**0.44**	1.89
Time	Jump kick	*1.02*	*1.02*	22.36	8.31	15.71	17.09	39.54	**0.45**	1.92
Per frame	Playground	*1.13*	*0.78*	14.87	8.28	14.42	12.82	28.40	**0.43**	1.90
(ms)	Average	*1.09*	*0.89*	18.23	9.35	16.92	15.10	32.17	**0.44**	1.90

Highlighted values correspond to the best three of each row (the best in bold font and the other two in italics).

Table 1 shows that there is a considerable difference of computation time between the SIK, KMA, CCD and TGBSIK with respect to others. This demonstrates that the processes to be undertaken by these methods are much faster at obtaining a satisfactory solution. With respect to the quality of the reconstruction, SIK, TGBSIK, PIK and KMA obtain better results. Their main differences, compared with the original sequences, come from the swivel angles of the limbs which is not dramatic since they still get natural poses for the movements considered. In terms of reconstruction quality, CCD presents two main drawbacks. First, the swivel angle of the upper limbs is biomechanically correct but becomes unnatural as the motion progresses. The second drawback is in the motion of the spine, which resembles that of a snake rather than a human spine. This is because bending starts from the head, so joints closer to that end-effector in the hierarchy are moved to a greater extent than those further away. Fig. 15 shows that quality errors tend to increase as the motion advances in the boxing animation. This occurs because bending in the iterations starts from the end-effectors of the chains so the motions of the joints nearest to them are greater.

The comments on CCD for the spine are partly applicable to the KMA approach because the latter is based on the CCD. Nevertheless, the well known “snake-like” artifact is not so noticeable since the KMA reconstruction always exploits the resting posture instead of the previous frame posture. Regarding the limbs, the KMA approach uses the TGB method, but without performing any optimization procedure, unlike TGBSIK. This is why it is faster than TGBSIK.

In the Pseudoinverse, DLS, DLS with SVD and SDLS methods the main drawback comes from the orientation of the clavicles, which tend to bend more than would appear natural. This is because in each iteration the angle of each articulation is updated simultaneously using the same *rigidity* to rotation. The main difference between them comes from the jerkiness when the limbs are near singular postures, which is more noticeable in the play-

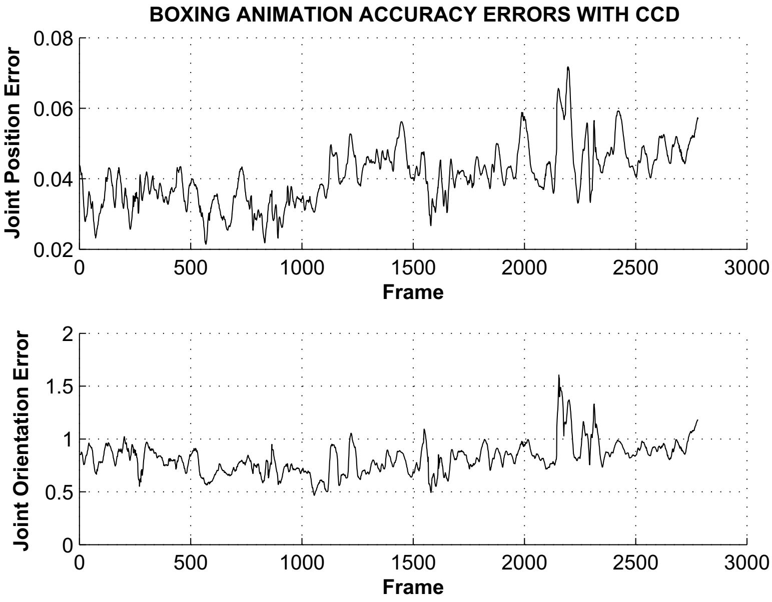

Fig. 15. Boxing animation accuracy errors with CCD.

ground animation due to its inherent noise. Among these, the DLS with SVD and the SDLS methods give the most stable results. The PIK method, which relies on the DLS with SVD, obtains more natural orientations of the clavicles be-

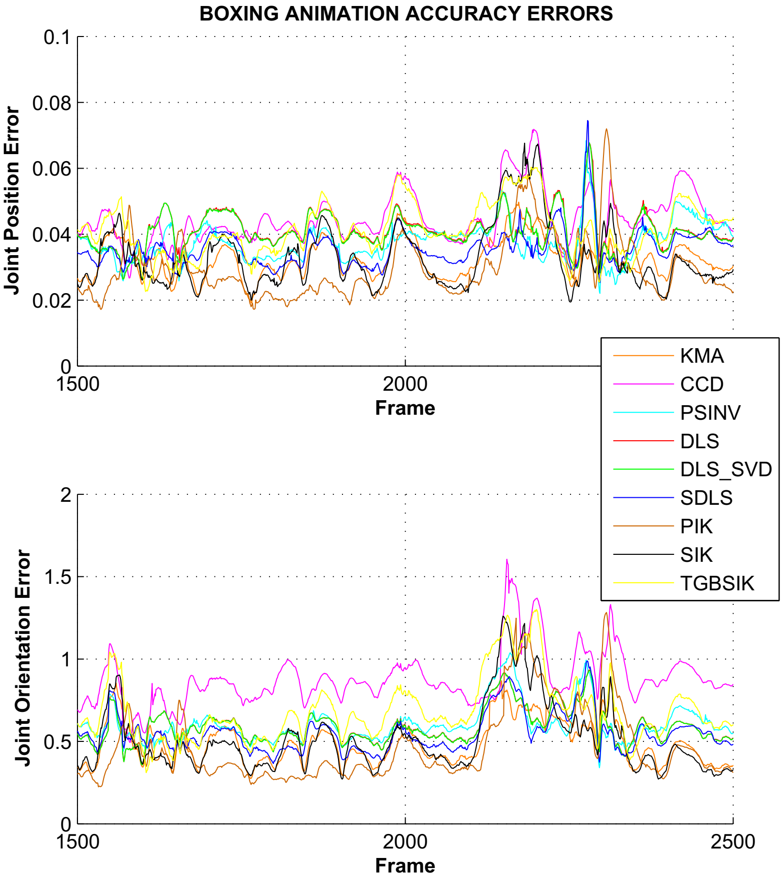

Fig. 16. Boxing animation accuracy errors.

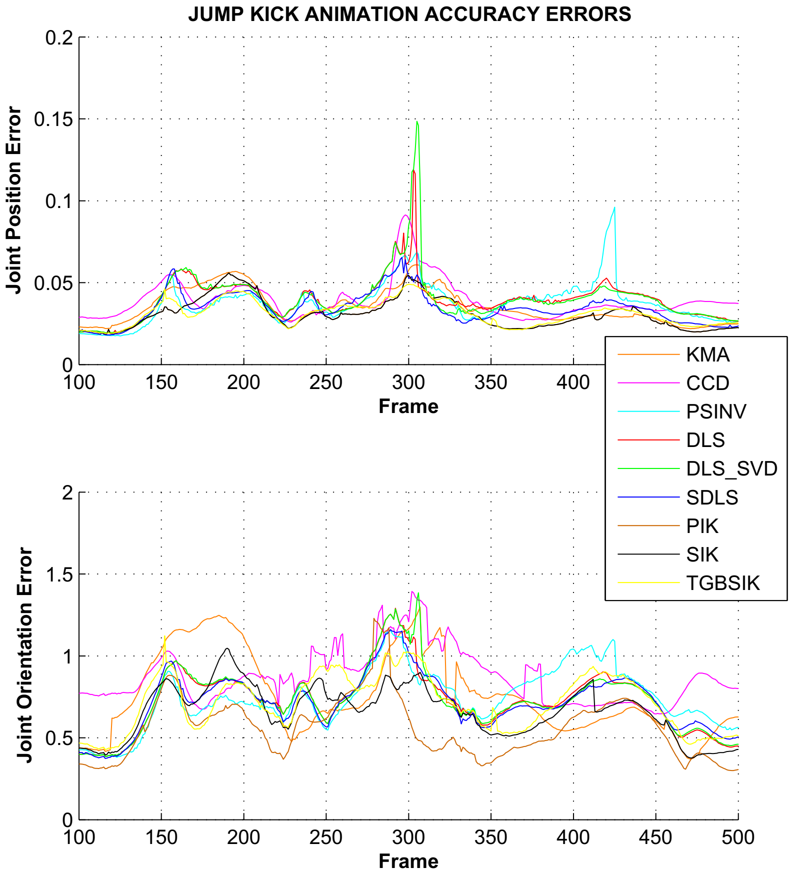

Fig. 17. Jump kick animation accuracy errors.

cause the pose is attracted to the *relaxed* standing posture mentioned above. The TGBSIK method is a bit slower than SIK and KMA in positioning the anthropomorphic limbs because an optimization is being performed. But the reconstruction is also visually acceptable even though the swivel angles are more distant from the original movements. Its main drawback is that there are some frames in which the swivel angle changes abruptly because the minimization based on biomechanical limits does not yield totally smooth results, mainly due to the fact that for a given joint the global minimum is not continuous throughout the workspace. This type of abrupt changes also occur with the KMA method in the playground animation because of its high variety of movements and the discontinuity in the swivel angle references.

Fig. 16 shows the quality errors in the boxing animation for all the considered IK methods except for the Jacobian Transpose from frames 1500 to 2500, Fig. 17 shows similar results for the jump kick animation from frames 100 to 500, and Fig. 18 the same for the playground animation from frames 1500 to 2500. The maximal errors in the first two cases occur when the humanoid performs a quick turn-around movement, where one of the limbs is totally stretched trying to reach its known end-effector position (a punch in the boxing animation and the kick in the jump kick animation). This maximal value is caused by the differences between the actual pelvis orientation and its estimation, and also because the end-effector is out of range for its corresponding limb. In the third animation maximal errors happens again while the humanoid performs turn-around movements while walking. These facts show that a satisfactory spine positioning is decisive for an appropriate full-body reconstruction quality.

Finally, Figs. 19–21 present excerpts of the reconstructions in which the differences mentioned above can be appreciated.

10. Conclusions and future work

In this paper we have presented an analytic-iterative Inverse Kinematics (IK) method, called Sequential IK (SIK), that reconstructs 3D human full-body movements in real time. The input data for the reconstruction are just the positions of wrists, ankles, head and pelvis, making it applicable to low-cost motion capture environments where these six features are tracked or for an easy interactive control of avatars. No pre-recorded full-body pose database is needed for our approach. We have presented a novel method to estimate the orientation of the root joint from these data applicable to a

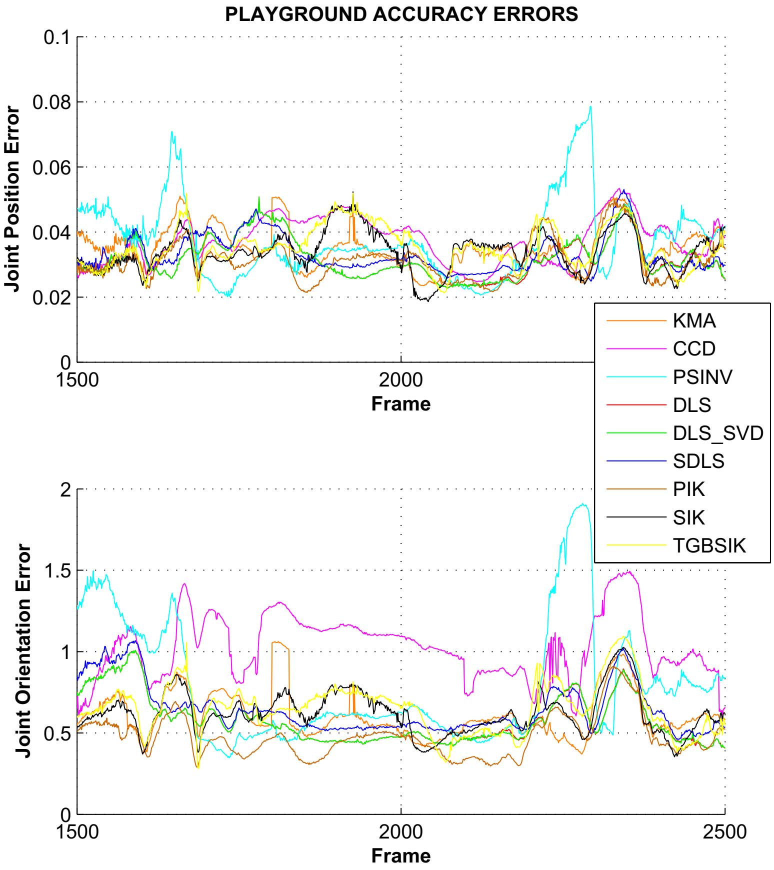

Fig. 18. Playground animation accuracy errors.

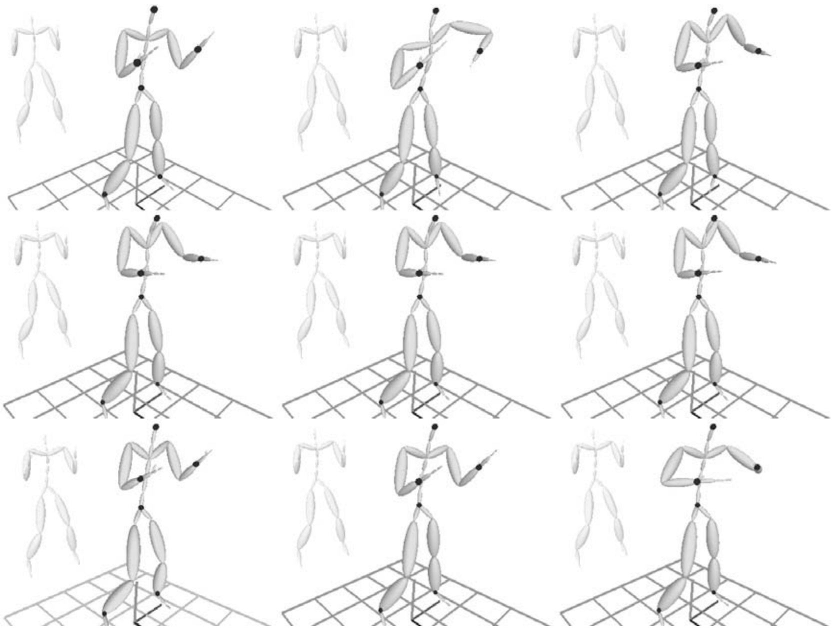

Fig. 19. From left to right and top to bottom: KMA, CCD, Pseudoinverse, DLS, DLS with SVD, SDLS, PIK, SIK and TGBSIK reconstructions in frame 2507 of the boxing animation.

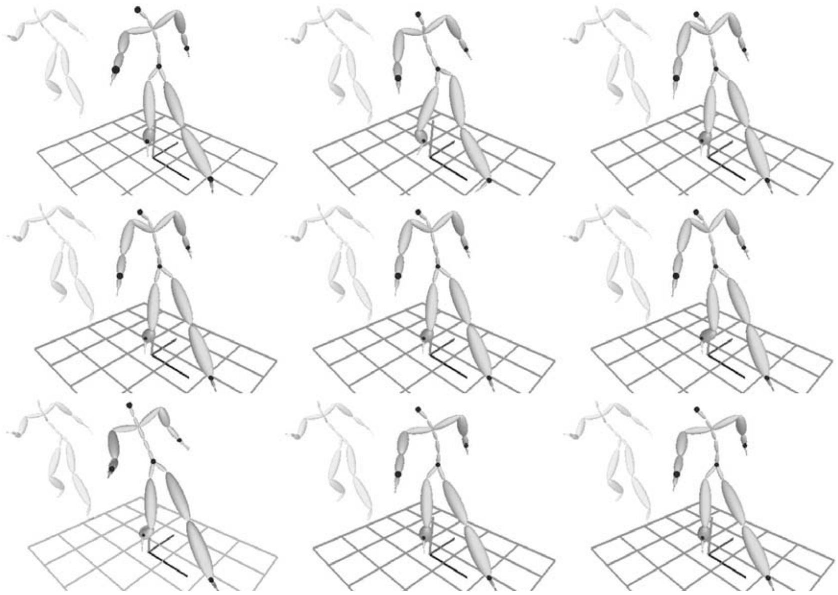

Fig. 20. From left to right and top to botom: KMA, CCD, Pseudoinverse, DLS, DLS with SVD, SDLS, PIK, SIK and TGBSIK reconstructions in frame 327 of the jump kick animation.

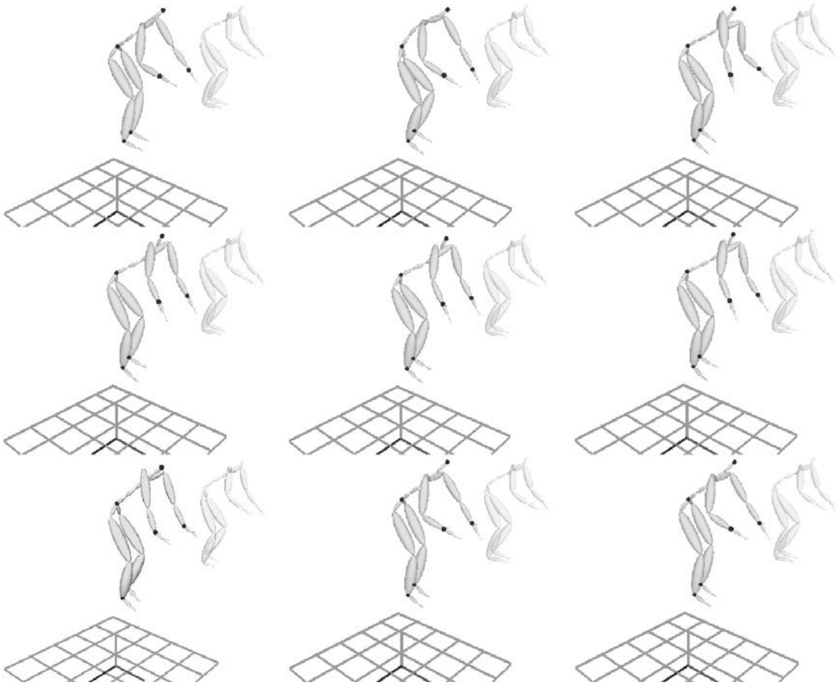

Fig. 21. From left to right and top to botom: KMA, CCD, Pseudoinverse, DLS, DLS with SVD, SDLS, PIK, SIK and TGBSIK reconstructions in frame 675 of the playground animation.

wide range of movements. In SIK poses are reconstructed sequentially, by readjusting first the spine, then the clavicles, and finally each of the upper and lower limbs. We have designed an analytic-iterative reconstruction method of full spines and analytic methods for the clavicles, arms and legs that achieve very fast and visually acceptable results. This gives the computer sufficient time to carry out additional processes in real time, such as computer vision algorithms for markerless motion capture, artificial intelligence algorithms for action recognition or for reconstructing the motion of multiple participants simultaneously.

We have also presented an easy to parametrize method to model realistic biomechanical limits of ball-and-socket joints, using a few anatomical measurements such as adduction, abduction, flexion and extension. Along with it, we have introduced a method to estimate the penetration depth of the elbows within the torso, and a strategy to respond to these self-collisions when they occur.

The performance of the SIK method has been evaluated by reconstructing three different complex animations, quantifying the position and orientation differences between the original sequences and the reconstructions, measuring the computation times and comparing them with other well-known IK methods such as the technique of Kulpa et al. (KMA) [28], Cyclic Coordinate Descent (CCD) [25], Jacobian Transpose [7,8], Pseudoinverse [9], Damped Least Squares (DLS) [10,11], DLS with Single Value Decomposition (SVD) [12,13], Selectively Damped Least Squares (SDLS) [14], the IK technique for anthropomorphic limbs of Tolani et al. (TGB) [26] and the Prioritized IK (PIK) [16,17]. It has been shown that the SIK method is a fast and reliable approach for our purpose, yielding satisfactory results both in terms of quality and computation time.

Future work will focus on generalizing our root orientation estimation method to all kinds of movements, paying special attention to the weight values. Self-collision avoidance will also be extended taking into account all possible combinations, not only the elbows with the torso. Attention will also be paid to collision avoidance of the humanoid with other objects in a scenario, such as for reaching tasks and to make the feet step on the ground. It will also be interesting to explore the possibility of combining PIK and SIK methods in order to obtain fast reconstruction due to the analytic calculations of SIK and to have the option of adding prioritized constraints like in PIK, which would be helpful in improving reconstruction quality. The usage of databases to improve the reconstruction quality will also be explored taking advantage of the low computation time of SIK. For example, the initial posture could be obtained from the database and then the readjustment process would be carried out by SIK.

Acknowledgments

This work has been subsidized by the European Network of Excellence IST-002114-ENACTIVE Interfaces and by the University of Alcalá under grant PI2005/083. Special thanks to the University of Geneva for providing the 3DS Max human body model that can be seen on Figs. 6 and 9. Also special thanks to Sam Buss for sharing the C++ implementation of the Jacobian Transpose, Pseudoinverse, DLS, DLS with SVD and SDLS IK methods from http://math.ucsd.edu/sbuss/ResearchWeb/ikmethods/index.html. The motion capture data used in this project was obtained from http://mocap.cs.cmu.edu. This database was created with funding from NSF EIA-0196217. Luis Unzueta's contract is funded by the Ministry of Education of Spain within the framework of the Torres Quevedo Program, and cofunded by the European Social Fund. The PIK software has been supported by the Swiss National Science Foundation.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.gmod.2008.03.002.

References

[1] H-Anim specification, Available at: <http://www.hanim.org>.
[2] R. Boulic, J. Varona, L. Unzueta, M. Peinado, Á. Suescun, F.J. Perales, Evaluation of on-line analytic and numeric inverse kinematics approaches driven by partial vision input, Virtual Reality 10 (2006) 48–61.
[3] M. Zoppi, Effective backward kinematics for a 6R painting robot, in: ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference DETC02, Montreal, Canada, 2002.
[4] X. Wu, L. Ma, Z. Chen, Y. Gao, A 12-DOF analytic inverse kinematics solver for human motion control, Journal of Information Computational Science 1 (1) (2004) 137–141.
[5] J.Q. Gan, E. Oyama, E.M. Rosales, H. Hu, A complete analytical solution to the inverse kinematics of the Pioneer 2 robotic arm, Journal of Robotica 23 (1) (2005) 123–129.
[6] J. Zhao, N.I. Badler, Inverse kinematics positioning using nonlinear programming for highly articulated figures, ACM Transactions on Graphics (TOG) 13 (4) (1994) 313–336.
[7] A. Balestrino, G. De Maria, L. Sciavicco, Robust control of robotic manipulators, in: Proceedings of the Ninth IFAC World Congress, vol. 5, 1984, pp. 2435-2440.
[8] W.A. Wolovich, H. Elliot, A computational technique for inverse kinematics, in: Proceedings 23rd IEEE Conference on Decision and Control, vol. 23, 1984, pp. 1359-1363.
[9] D.E. Whitney, Resolved motion rate control of manipulators and human prostheses, IEEE Transactions on Man-Machine Systems 10 (1969) 47–53.
[10] Y. Nakamura, H. Hanafusa, Inverse kinematics solutions with singularity robustness for robot manipulator control, Journal of Dynamic Systems, Measurement, and Control 108 (1986) 163–171.
[11] C.W. Wampler, Manipulator inverse kinematic solutions based on vector formulations and damped least squares methods, IEEE Transactions on Systems, Man, and Cybernetics 16 (1986) 93–101.
[12] A.A. Maciejewski, C.A. Klein, Numerical filtering for the operation of robotic manipulators through kinematically singular configurations, Journal of Robotic Systems 5 (6) (1988) 527–552.
[13] A.A. Maciejewski, Dealing with the ill-conditioned equations of motion for articulated figures, IEEE CGA 10 (3) (1990) 63–71.
[14] S.R. Buss, J.S. Kim, Selectively damped least squares for inverse kinematics, Journal of Graphics tools 10 (3) (2005) 37–49.
[15] K.-J. Choi, H.-S. Ko, On-line motion retargetting, in: Proceedings of the International Pacific Graphics '99, Seoul, Korea, 1999.
[16] P. Baerlocher, R. Boulic, An inverse kinematic architecture enforcing an arbitrary number of strict priority levels, The Visual Computer, International Journal of Computer Graphics 20 (6) (2004) 402–417.
[17] M. Peinado, B. Herbelin, M. Wanderley, B. Le Callennec, R. Boulic, D. Thalmann, D. Méziat, Towards configurable motion capture with prioritized inverse kinematics, in: Proceedings of the third International Workshop on Virtual Rehabilitation (IVWR'04), Lausanne, Switzerland, 2004.
[18] D. Raunhardt, R. Boulic, Exploiting coupled joints, in: Proceedings of GRAPP07, Barcelona, Spain, 2007.
[19] R. Boulic, B. Ulicny, D. Thalmann, Versatile walk engine, Journal of Game Development 1 (1) (2004) 29–52.
[20] E. Oyama, A. Agah, T. Maeda, S. Tachi, Inverse kinematics learning by modular architecture neural networks with performance prediction networks, in: Proceedings of the 2001 IEEE International Conference on Robotics and Automation Seoul, Korea, 2001.
[21] A. D'Souza, S. Vijayakumar, S. Schaal, Learning inverse kinematics, in: Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii, USA, 2001.
[22] K. Grochow, S.L. Martin, A. Hertzmann, Z. Popovic, Style-based inverse kinematics, ACM Transactions on Graphics 23 (3) (2004) 522–531.
[23] J. Chai, J.K. Hodgins, Performance animation from low-dimensional control signals, Proceedings of ACM SIGGRAPH (2005) 686–696.
[24] G. Liu, J. Zhang, W. Wang, L. McMillan, Human motion estimation from a reduced marker set, in: Proceedings of the 2006 symposium on Interactive 3D graphics and games, 2006, pp. 35-42.

[25] L.-C.T. Wang, C.C. Chen, A combined optimization method for solving the inverse kinematics problem of mechanical manipulators, IEEE Transactions on Robotics and Automation 7 (1991) 489–499.

[26] D. Tolani, A. Goswami, N.I. Badler, Real-time inverse kinematics techniques for anthropomorphic limbs, Graphical Models 62 (5) (2000) 353–388.

[27] H.J. Shin, J. Lee, S.Y. Shin, M. Gleicher, Computer puppetry: an importance-based approach, ACM Transactions on Graphics 20 (2) (2001) 67–94.

[28] R. Kulpa, F. Multon, B. Arnaldi, Morphology-independent representation of motions for interactive human-like animation, in: Eurographics, Dublin, Ireland, 2005.

[29] G. Monheit, N.I. Badler, A kinematic model of the human spine and torso, IEEE Computer Graphics and Applications 11 (2) (1991) 29–38.

[30] T. Molet, R. Boulic, D. Thalmann, Human motion capture driven by orientation measurements, Presence 8 (2) (1999) 187–203.

[31] V.B. Zordan, J.K. Hodgins, Tracking and modifying upper-body human motion data with dynamic simulation, in: Proceedings of Computer Animation and Simulation '99, 1999.

[32] N.I. Badler, M. J Hollick, J.P. Granieri, Real-time control of a virtual human using minimal sensors, Presence: Teleoperators and Virtual Environments 2 (1) (1993) 82–86.

[33] S.K. Semwal, R. Hightower, S. Stansfield, Mapping algorithms for real-time control of an avatar using eight sensors, Presence: Teleoperators and Virtual Environments 7 (1) (1998) 1–21.

[34] N.I. Badler, C.B. Phillips, B.L. Webber, Simulating Humans: Computer Graphics, Animation, and Control, Oxford University Press, Oxford, 1993.

[35] E. Otani, Software Tools for Dynamic and Kinematic Modeling of Human Motion, Master Thesis, University of Pennsylvania, Philadelphia, PA, USA, 1989.

[36] T. Kang, S.H. Tillery, J. He, Determining natural arm configuration along reaching trajectory, in: Proceedings of the 25th Annual International Conference of the IEEE EMBS, Cancun, Mexico, 2003.

[37] F.S. Grassia, Practical parameterization of rotations using the exponential map, The Journal of Graphics Tools 3 (3) (1998) 29–48.

[38] J.U. Korein, A Geometric Investigation of Reach, The MIT Press, Cambridge, 1985.

[39] P. Baerlocher, B. Boulic, Parametrization and range of motion of the ball-and-socket joint, Deformable avatars, Kluwer Academic Publishers, Dordrecht, 2001.

[40] A.E. Engin, S.M. Chen, Statistical data base for the biomechanical properties of the human shoulder complex I: Kinematics of the shoulder complex, Journal of Biomechanical Engineering 108 (1986) 215–221.

[41] L. Herda, R. Urtasun, P. Fua, A. Hanson, Automatic determination of shoulder joint limits using quaternion field boundaries, International Journal of Robotics Research 22 (6) (2003) 419–438.

[42] I.A. Kapandji, Physiology of the Joints, The Upper Limb, Vol. 1, Churchill Livingstone, New York, USA, 1982, ISBN 0-443-02504-5.

[43] I.A. Kapandji, Physiology of the Joints, The Lower Limb, Vol. 2, Churchill Livingstone, New York, USA, 1988, ISBN 0-443-03618-7.

[44] I.A. Kapandji, Physiology of the Joints, The Trunk and the Vertebral Column, Vol. 3, Churchill Livingstone, New York, USA, 1974, ISBN 0-443-01209-1.

[45] X.G. Wang, F. Mazet, N. Maia, K. Voinot, J.P. Verriest, M. Fayet, Three-dimensional modelling of the motion range of axial rotation of the upper arm, Journal of Biomechanics 31 (10) (1998) 899–908.

[46] L. Zhang, Y.J. Kim, G. Varadhan, D. Manocha, Generalized Penetration Depth Computation, in: ACM Solid and Physical Modeling Conference (SPM06), 2006, pp. 173-184.

[47] M. Kallmann, Scalable solutions for interactive virtual humans that can manipulate objects, in: Proceedings of the Artificial Intelligence and Interactive Digital Entertainment (AIIDE), Marina del Rey, CA, USA, 2005.

[48] M. Peinado, D. Meziat, R. Boulic, D. Raunhardt, Environment-aware postural control of virtual humans for real-time applications, in: Proceedings of the SAE Conference on Digital Human Modeling for Design and Engineering, ENS a Gerland, Lyon, France, 2006.

[49] L. Kovar, M. Gleicher, F. Pighin, Motion Graphs, ACM SIGGRAPH (2002).