Engineering Letters, 15:2, EL_15_2_08

Forward Kinematics for Virtual Agents

Antonio Benitez R., Guillermo de los Santos T. * and Daniel Vallejo R. †

Abstract—**Realistic motion and natural-looking simulations require a thorough understanding of human movement control strategies. This work presents an initial developed on virtual agents. We describe how the agent was built searching an humanoid representation through a geometric model. Although this research is based on the work of Pettré et al. [16], we are building a solid platform to give total mobility and realism to both, the agent and the virtual environment where our humanoid works. Preliminary results about forward kinematic applied to the actor are presented. This work allow us to keep on working on virtual agents area, to analyze and propose techniques to give the agent total autonomy to its motion.**

Keywords: virtual agents, forward kinematic, model and design of the 3D actors.

1 Introduction

Mobile robots pose a unique set of challenges to artificial intelligence researchers. Such challenges include issues of autonomy, uncertainty (both sensing and control), and reliability, which are all constrained by the discipline that the real world imposes. Planning, sensing, and acting must occur in concert and in context. That is, information processing must satisfy not only the constraints of logical correctness but also some assortment of crosscutting, physical constraints. Particularly interesting among these robots are humanoids, which assume an anthropomorphic (human-like) form [21].

One of the main industrial applications of virtual human modeling and simulation is to make an ergonomic evaluation of the man-machine interface of a product in a computer-aided design (CAD) environment at a very early stage of design (also called digital prototyping). The cost of developing prototypes is typically waived or completely reduced when digital prototyping is implemented. While digital mockups have already made significant impact on manufacturing, the use of digital humans to evaluate a design has not been extensively used. Today, the development of video games on the internet, the development of interactive training systems, the computer-aided production of movies, especially, virtual reality applications, also raise the need for tools that will facilitate the creation and animation of autonomous virtual characters in 3-D worlds. Motion planning techniques will be used to direct digital human characters at the task level and to create highly interactive systems. This motivates us to explore techniques and create fast planners capable of using physics-based models to generate realistic-looking motions.

Towards this objective, it is necessary to simulate a human representation through a virtual environment.

The purpose of this research is to obtain a better understanding of the mathematical modeling of human motion using well-established kinematic theories. Particularly from the field of robotics, we believe there are significant applicable theories and numerical algorithms that, if appropriate, can be tailored to address long standing problems in human modeling and simulation.

During the last few years, a number of robotics techniques have made significant inroads in graphic systems, e.g., dynamic modeling to create physically realistic animation, vision sensing to automatically acquire 3D-models, haptic interaction to ”fee” virtual objects, and motion planning to create collision-free motions. Today the development of video games on the Internet, the development of interactive training systems, the increasing sophistication of websites, and the computer-aided production of movies raise the need for new tools that will facilitate the creation and animation of autonomous virtual characters $ordigitalactors$ in 3-D worlds. Motion planning techniques are used to direct digital actors at the task-level and to create highly interactive systems. This domain will motivate the creation of fast planners capable of using physically-based models to generate realistic-looking motions. It will require planned motions to be seamlessly combined with motions extracted from large libraries of clipped motions. See [12, 11] for preliminary results in motion planning applied to the animation of digital actors.

2 Modeling the Virtual Agent

To establish a systematic method for biomechanically modeling human anatomy, researchers have implemented conventions for representing segmental links and joints. Human anatomy can be represented as a sequence of rigid bodies (links) connected by joints. Of course, this serial

---

[*] Autonomous University Juárez of Tabasco, Academic Division of Informatics and Systems, Cunduacan - Tabasco, México. Tel: +52 (993) 355 81 575 ext. 6733. Email: antonio.benitez@dais.ujat.mx

[†] University of Americas - Puebla, Department of Computation, Electronic and Mecatronic, Cholula - Puebla, México. Email: daniel.vallejo@udlap.mx

linkage could be an arm, a leg, a finger, a wrist, or any other functional mechanism. Joints in the human body vary in shape, function, and form. The complexity offered by each joint must also be modeled, to the extent possible, to enable a correct simulation of the motion. The degree by which a model replicates the actual physical model is called the level of fidelity. Perhaps the most important element of a joint is its function, which may vary according to the joints location and physiology. The physiology becomes important when we discuss the loading conditions of a joint. In terms of kinematics, we shall address the function in terms of the number of degrees of freedom associated with its overall movement.

The human body model has been a complicated task which play designers have found.

A wide variety of human body models are available to represent and modify the human body form. Magnenat-Thalmann in [14] used a technique based on coverings to animate Marilyn Monroe, she introduced the *Joint-dependent Local Deformation* concept (JLD), this was used to deform the skin depending of the articulation kind [3], Forsey [9], improve the *B-spline* hierarchic technique to animate 3D characters. Chadwick in [5] proposed a multilevel technique based on FRE-FORM deformation to apply muscular efforts on skeleton. Boulic et al. in [2] proposed an multilevel approximation based on metaballs to construct and animate realistic human bodies. Nedel and Thalmann simulated the muscle's deformation using a mass-means system [19].

2.1 Preparing the sketch

To determine the dimensions of the model, we have used the dimensions given by Milkshape 3D tool [20], though we toke the skeleton provided by the Counter - Strike 1.3 [8]. Taking account the H-ANIM [10] specifications, the origin *(x,y,z)* of the skeleton is placed at the level *(0,0,0)* of the tool.

Our skeleton have 53 joints, although for this work we only are considering 14 joints to move it, in future work we will apply kinematic to every joint on the model.

Next, we draw the sketch, creating the physical appearance of the model. To facilitate the construction of the sketch we used images that Chad Cox tutorial gives [8]. As other tools to build 3D models, MilkShape take account seven visual modes. During this stage, we only used frontal, right profile and ridge modes. On these visual modes, we have drawn the sketch, in Figure 2 a frontal view of the model is presented.

2.2 Reference points, Triangles and Textures

Before and during the modeling it was important to take account the characteristic reference points of the human body. Therefore, our model is supported on CAESAR project [4]. This metrics gives the model an anthropometric basis, since the model is not only an artistic draw, it has to consider the human metrics which define a standard in the 3D modeling.

First, we imported the sketch from Milkshape 3D. Initially, we begin to model the bust, because this was a difficult part to create. The bust was built using a sphere with 12 sides and 6 put into a pile with a 90 degrees turned with respect to X - axis in frontal view. Next we added vertices to the model which were used to create triangles, these were added to build the upper bust.

In the same way the parts of the actor were modeled using triangles to joint the reference points with the sketch of the model. Next, the triangle meshes are grouped to be manipulated and to assign textures. The parts of the model were built as follows: *Bust, Waist, Back, Legs, Arms, Feet and Hands*.

In Figure 1 meshes triangles and textures associated to the agent are showed. Figure 2 presents the sequence used to build the actor, this includes: to place the skeleton on a reference frame, the painted of the sketch and the insertion of reference points, so we can add vertex an triangles to model the actor parts. Finally, we grouped the triangle meshes to apply textures to give more realism the actor.

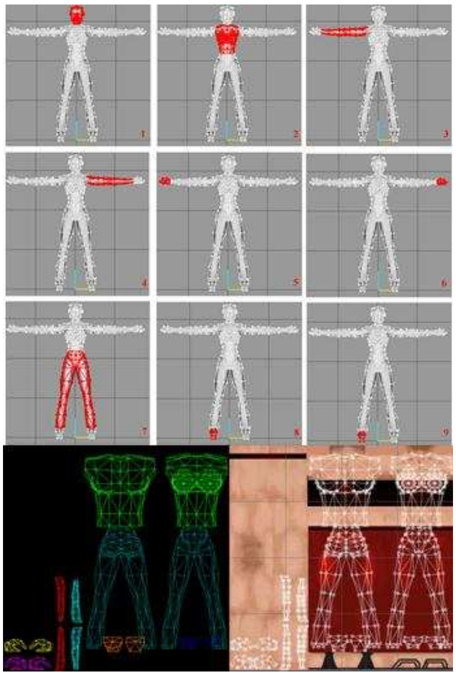


Figure 1: Group of meshes triangles and textures associated.

3 Model Representation

The model of the digital actor, its movement and its personality are built on the skeleton. The skeleton is com-

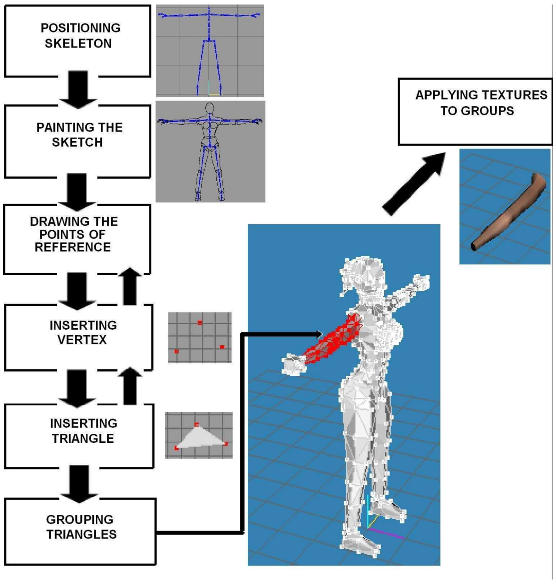


Figure 2: Design of the model.

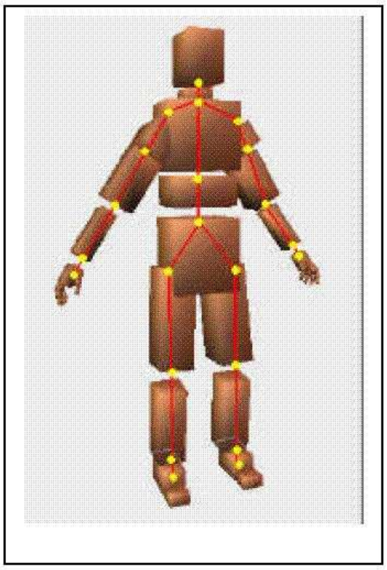


Figure 3: Hierarchic Order.

posed by bones (elements or links) connected through joints or articulations. Links and joints are organized using a tree as a data structure. The skeleton can be used as a invisible armor which is used as a reference frame to calculate the position and orientation of the geometric structure of our digital actor. Besides, a skeleton can be used to animate: rigid movement, facial expressions, fine weaves, smooth weaves (as muscles), fatty parts, mechanic parts, even clothes [17].

3.1 3D Hierarchic Articulated Objects

In a hierarchic character as described in [13], different body parts of the articulated object are separate objects, stored in a hierarchy and joined to each other at pivot points, each referencing child objects attached objects of a lower hierarchic order (Figure 3) .[7] describes how the hierarchic character is set up and how it can be displayed, by recursively traversing the hierarchic data structure starting from the root, passing down the transformation of the parent objects to its children and there concatenating the passed down translation matrix and the local translation matrix. Older 3D games like the original Tomb Raider use this method for character animation. The flexibility and adaptability of this method are its main benefits: Animation data can be generated on the fly using techniques such as Inverse Kinematics (see below), and applied to the model in real-time. Memory usage is also small as the vertex and transformation information for each object contained in the model needs to be stored only once.

3.2 Hierarchical representation

The skeleton topology is represented as a tree structure. The hierarchy consist on selecting a joint as the root when the other joints will be connected. In Figure 4 we present the transition between virtual agent and skeleton representation where joints are assigned to every part of the humanoid, and the transition between the skeleton and the hierarchic tree structure. Figure 5 presents the structure of the hierarchy joints for the skeleton.
Nodes in the tree represent the joints of the skeleton. The root can be selected of arbitrary way, although it is preferred to place this joint near than center of our digital actor. Usually, for digital actors, the root is placed on the spine, so we can joint the pelvis and the torso under the root of the tree.

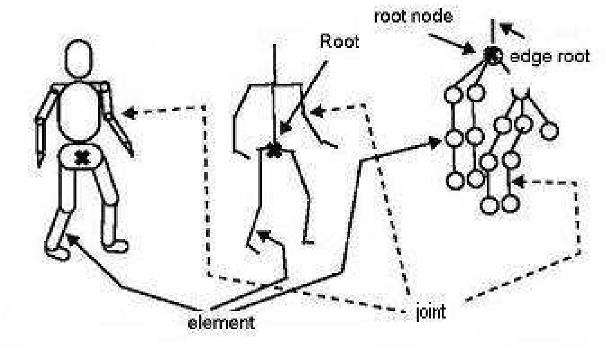


Figure 4: Transition between virtual agent and hierarchic tree representation.

4 Preparing the model to animation

Before we attempting to move the actor, a group of vertices is assigned for every joint on the skeleton. It is important to create a good vertices assignation to avoid deformation on the model when we attempt to move it.

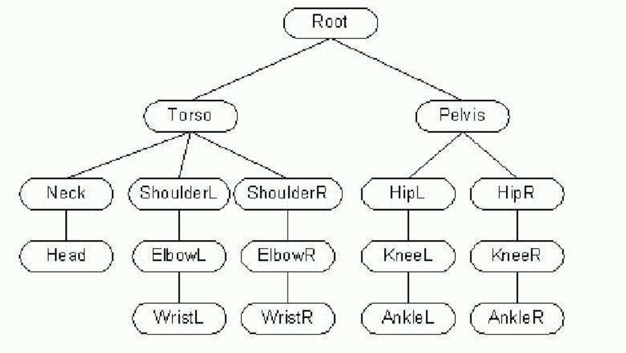


Figure 5: Hierarchical joints structure.

In Figure 6 vertices are assigned to leg and foot joints. In the same way, a vertices group is associated with each joint in the hierarchical structure of the skeleton.

Our model was exported from Milkshape using the MS3D format [6].

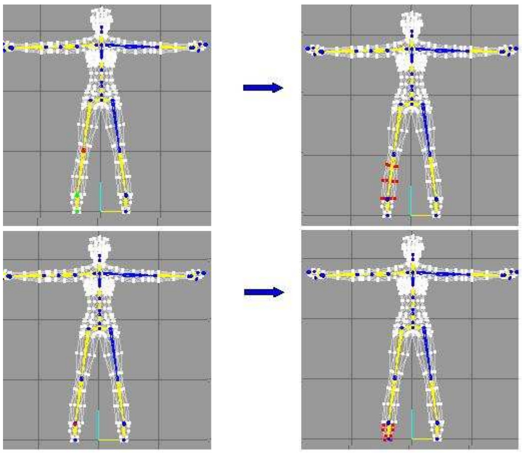


Figure 6: Assigning a vertex group for every joint.

4.1 Denavit-Hartenberg Representation Method

In order to obtain a systematic method for describing the configuration (position and orientation) of each pair of consecutive segmental links, a method was proposed by Denavit and Hartenberg (1955). We shall utilize the method of Denavit and Hartenberg to address human kinematics.

The Denavit-Hartenberg (DH) method was created in the 1950s to systematically represent the relation between two coordinate systems but was only extensively used in the early 1980s with the appearance of computational methods and hardware that enable the necessary calculations. The method is currently used to a great extent in the analysis and control of robotic manipulators. This method has also been successful in addressing human motion, in particular towards a better understanding of the mechanics of human motion.

The method, now referred to as the DH method, is based upon characterizing the configuration of link i with respect to link *(i-1)* by a (4×4) homogeneous transformation matrix representing each links coordinate system. If each pair of consecutive links represented by their associated coordinate system is related via a matrix, then using the matrix chain-rule multiplication, it is possible to relate any of the segmental links (e.g., the hand) with respect to any other segmental link (e.g., the shoulder).

In order to generate the matrix relating any two links, four parameters are needed. The four parameters (depicted in Figure 7) are:

1. θ_i is the joint angle, measured from the x_{i-1} to the x_i axis about the z_{i-1} (right hand rule applies). For a prismatic joint θ_i is a constant. It is basically the angle of rotation of one link with respect to another about the z_{i-1} axis.

2. d_i is the distance from the origin of the coordinate frame $(i-1)$ to the intersection of the z_{i-1} axis with the x_i axis along z_{i-1} axis. For a revolute joint d_i is a constant. It is basically the distance translated by one link with respect to another along the z_{i-1} axis.

3. a_i is the offset distance from the intersection of the z_{i-1} axis with the x_i axis to the origin of the frame i along xi axis (shortest distance between the z_{i-1} and z_i axis).

4. α_i is the offset angle from z_{i-1} axis toz_i axis about the x_i axis (right hand rule).

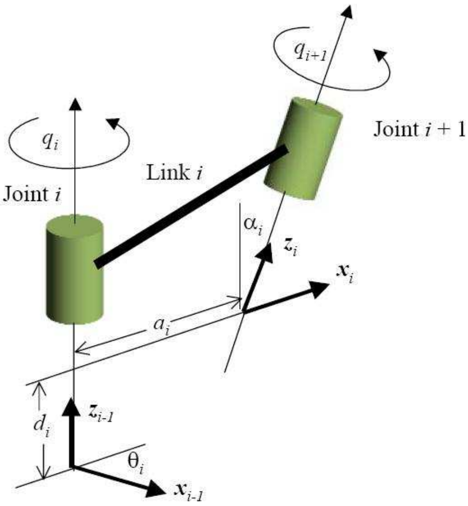


Figure 7: Joint coordinate system convention and its parameters.

Careful attention must be given when the following cases occur:

1. When two consecutive axes are parallel, the common normal between them is not uniquely defined, i.e., the direction of x_i must be perpendicular to both axes, however, the position of x_i is arbitrary.
2. When two consecutive axes intersect, the direction of x_i is arbitrary.

The (4×4) transformation matrix describing a transformation from link $(i-1)$ to link i for joint i is

$$A_i = \begin{bmatrix} C_{\theta_i} & -S_{\theta_i} C_{\alpha_i} & S_{\theta_i} S_{\alpha_i} & a_i C_{\theta_i} \\ S_{\theta_i} & C_{\theta_i} C_{\alpha_i} & -C_{\theta_i S_{\alpha_i}} & a_i S_{\theta_i} \\ 0 & S_{\alpha_i} & C_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The four values for the DH parameters $\theta_i, d_i, a_i, \alpha_i$, are typically entered into a table known as the DH Table. Each row is used to generate the homogeneous transformation matrix. This data set provides complete information about the model in terms of kinematic functionality as well as dimensions.

4.2 Joints

An articulated body is made of a set of segments, connected by joints. The essential feature of a joint is that it permits some degree of relative motion between the two segments it connects. Ideal kinematic joint models are defined in order to formalize this permitted relative motion, called range of motion, characterized by the number of parameters that describe the motion space, and constrained by joint limits. Modeling real joints can be very complex, since the range of motion depends on many factors, especially in the articulations of living organisms and the human in particular [1]. Moreover, joints may be dependent on each other, especially in living organisms. This coupling (of motion and limits) can be integrated directly in the body definition, with the concept of joint group [1], or at the application level, with kinematic constraints resolved by an inverse kinematics engine (for example, the scapulo-thoracic constraint [15]).

The simplest example of joint model is the revolute joint that allows a rotation about an axis axed in both segments it connects, usually within some angular limits. This joint is said to have one degree of freedom (DOF) and, because of its simplicity, is by far the most used joint in robotics. In human modeling, it is a convenient model of the interphalangeal joints of the hands and feet, for example. For more complex articulations such as the shoulder and hip, joint models allowing more degrees of freedom are required. Unfortunately, the accurate kinematic modeling of such articulations is a difficult task. First, a clear mathematical description of the allowed relative motion must be given by a proper parametrization: because of the complex non-Euclidean nature of rotations, this must be done carefully, or one may incur in the problem of singularities. Second, the range of motion should be constrained by some joint limits, to restrict the parameter space to some more realistic subset. For instance, a revolute joint whose configuration is given by the current rotation angle may have some *minimum* and *maximum* values. The situation is more complex with *ball-and-socket* joints, because the boundaries on the three independent parameters are generally coupled.

4.3 Ball-and-Socket Joint

A ball-and-socket joint possesses three rotational degrees of freedom. Hence, it is the most mobile of the purely rotational joints. It allows an axial motion (or twist) of the segment (one DOF), as well as a spherical motion (or swing) that determines its direction (two DOFs). Ball-and-socket joints are used to model articulations such as the human shoulder and hip. A mechanical illustration of this joint is given in Figure 8.

These joints are the most freely moving of all joints. The movements are allowed in all axes and planes: flexion/extension, adduction/abduction, circumduction and rotation. These joints are multiaxial.

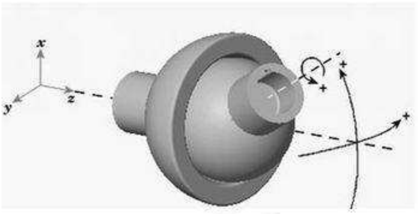


Figure 8: Ball and Socket joint (axial and spherics movement.)

4.4 Forward Kinematics

- In the recursive tree traversal, each joint first computes its local matrix **L** based on the values of its DOFs and some formula representative of the joint type:

  Local matrix $\mathbf{L} = L_{joint}(\phi_1, \phi_2, ..., \phi_n)$

- Then, world matrix **W** is computed by concatenating **L** with the world matrix of the parent joint

  World matrix $\mathbf{W} = L \cdot W_{parent}$

- It is convenient to have a 3D offset vector **T** for every joint which represents its pivot point relative to its parents matrix

$$L_{offset} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_X & T_Y & T_Z & 1 \end{bmatrix} \tag{1}$$

Hence the matrix representation for ball-and-socket is:

$$L_{R_{xyz}}(\theta_x, \theta_y, \theta_z) =$$

$$\begin{bmatrix} C_yC_z & C_yS_z & -S_y & 0 \\ S_xS_yC_z & S_xS_yS_z + C_xC_z & S_xC_y & 0 \\ C_xS_yC_z + S_xS_z & C_xS_yS_z - S_xC_z & C_xC_y & 0 \\ T_X & T_Y & T_Z & 1 \end{bmatrix} \tag{2}$$

Where x, y and z are the rotation angles respect to X, Y and Z axes respectively [18]. T_X, T_Y and T_Z are the offset vector described in previous paragraph.

4.5 DOF Limits

The digital actor move its extremities (shoulder, elbow, hip and knee) based on the next movement ranks:

Shoulder

- Abduction / h-adduction (255 grades)
- Direct Flexion /h-extension (240 grades)
- Extension /horizontal flexion (180 grades)

Elbow

- Flexion / h-extension (160 grades)

Hip

- Flexion / h-extension (150 grades)
- Abduction / adduction (100 grades)
- Left Rotation / right (180 grades)

Knee

- Flexion / h-extension (135 grades)

Figures **??** and Figure 11 shows the parameters used to restrict the movement of the actor.

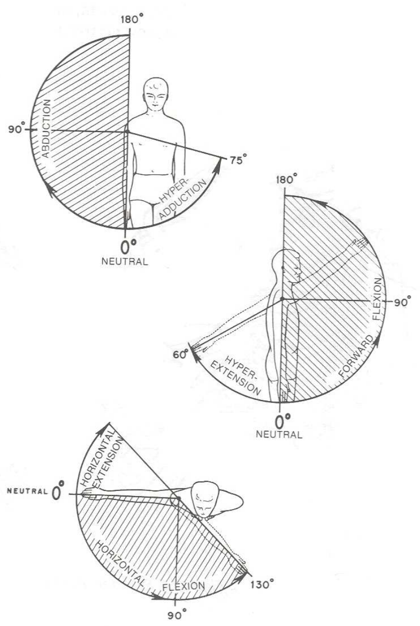


Figure 9: Shoulder range movement.

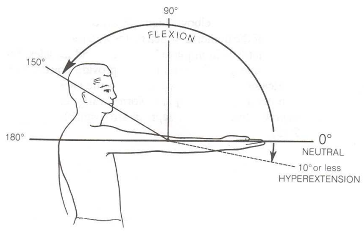


Figure 10: Elbow range movement.

5 Preliminary Results

Initially, forward kinematics was tested and implemented in 2D under MatLab tool, next a 3D implementation was developed in MatLab, and finally a random simulation under a virtual environment in 3D was built to review the DH-Convention. These results are presented un Figure 12, Figure 13 and Figure 14 respectively.

In section 2.1 we said that, our skeleton has 53 joints, although only 14 are being used in this initial work. Figure 15 presents the root, pelvis, hip, knee and ankle joints to be transformed as an example of kinematic chain.

Every articulation was treated as a ball-and-socket joint. So, the transformations used to give an initial movement to the actor was computed using the matrix presented in section 4.4.

It is important to say that, although only fourteen joints were implemented during this initial work, we are working on the total mobility of the actor. Beside, the construction of the model by itself represented an important

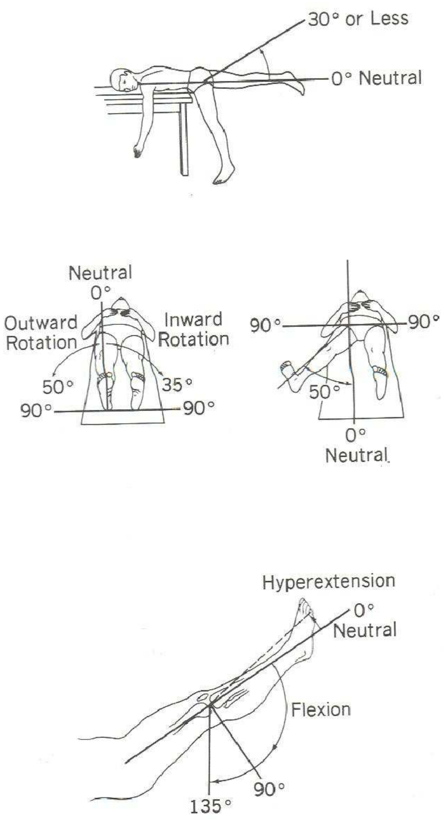
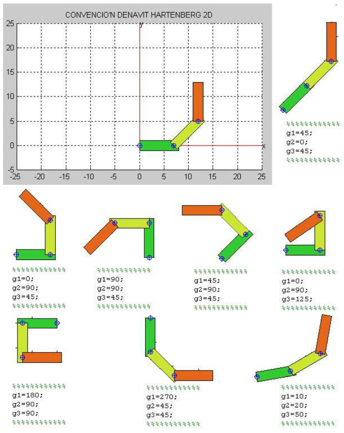


Figure 11: Leg range movement.



```
%%%%%%%%%%%%%
g1=45;
g2=0;
g3=45;
%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%
g1=0;
g2=90;
g3=45;
%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%
g1=90;
g2=90;
g3=45;
%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%
g1=45;
g2=90;
g3=45;
%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%
g1=0;
g2=90;
g3=125;
%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%
g1=180;
g2=90;
g3=90;
%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%
g1=270;
g2=45;
g3=45;
%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%
g1=10;
g2=20;
g3=50;
%%%%%%%%%%%%%
```

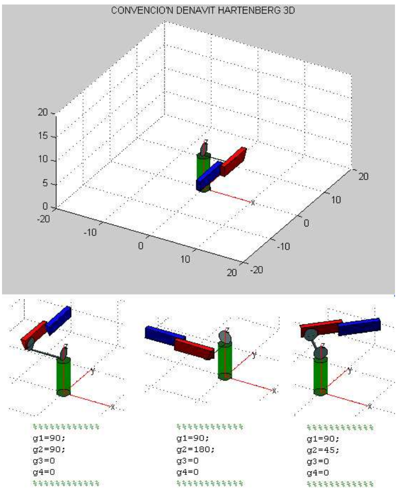
Figure 12: MatLab Simulation 2D kinematics.

work.

On the other hand, the present work applies forward kinematics in successful way to move the extremities of the digital actor, that allow us to apply this transformations to move every articulation on the model. Now, we have a digital actor grouped and represented through triangle meshes and built through an humanoid skeleton. Figures 16 to 19 presents different views of the digital actor, which is moving legs and arms.

The left side of Figure 16 a skeleton representation is showed as the fundamental part the actor, on the right side, the skeleton is cover by the triangle meshes giving a human form the agent.

In Figure 17 textures are added the actor to obtain more realism, at time we can see the skeleton incrusted as the base of it. Joints associated to elbow and knee are moved



```
%%%%%%%%%%%%%
g1=90;
g2=90;
g3=0
g4=0
%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%
g1=90;
g2=180;
g3=0
g4=0
%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%
g1=90;
g2=45;
g3=0
g4=0
%%%%%%%%%%%%%
```

Figure 13: MatLab Simulation 3D kinematics.

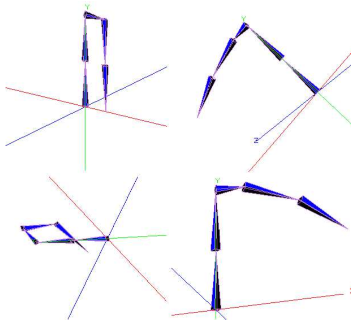


Figure 14: 3D Random Simulation kinematic Chains.

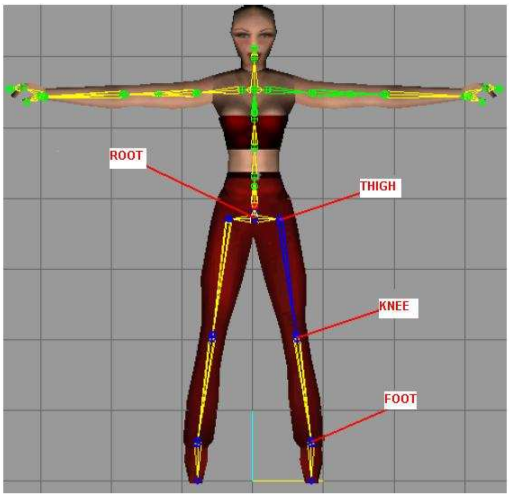


Figure 15: Articulations: Root, Thigh, Knee and Foot.

and showed in Figure 18. Finally the movement of the thigh is initially presented, as we can see in Figure 19, although it is not seem an accurate motion, we are working on it.

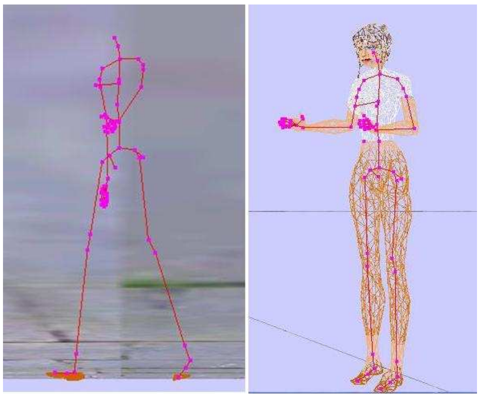


Figure 16: Skeleton and Triangle Meshes Model.

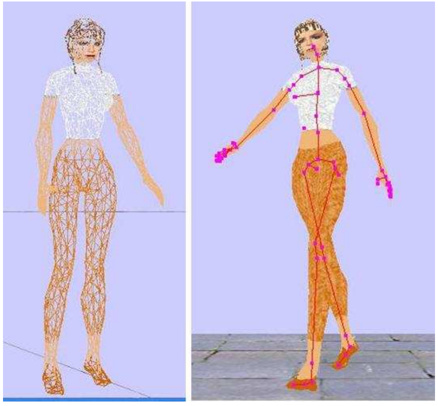


Figure 17: Triangle Meshes using Textures.

Figure 18: Elbow and Knee Flexion.

6 Conclusions and Future Work

The purpose of this research is to obtain a better understanding of the mathematical modeling of human motion using well-established kinematic theories. Particularly from the field of robotics, we believe there are significant applicable theories and numerical algorithms that, if appropriate, can be tailored to address long standing problems in human modeling and simulation.

Figure 19: The actor is waking and inclining.

The work presented in this paper is the result of an initial implementation of the platform needed to develop and test both, motion planning algorithms to give autonomy to the digital actor to plan its own movements in virtual environments, and the design of kinematic model to move any articulated robot. This platform includes: the construction of a digital actor, the implementation of kinematics chains applied to digital actor using *ball-and-socket* joints, and the 3D simulation environment used to place the actor.

These three tools described before, can be improved. Initially, we are interesting to apply motion planing algorithm to give the actor autonomy to plan trajectories in virtual environments, and our second interest is focused to apply digital actors to the WEB, giving gestures and movement on the context of software agents.

Of curse, we can keep on working to move every joint in the actor. Besides, the geometric model of the actor can be improved considerably, and it can be used to another kind of developments. for example: to simulate and predict movement, to build applications where many actors are working an planning in collaboration way, etc.

References

[1] N. Badler, C. Phillips, and B. Webber. Simulating humans, computer graphics animation and control. New York, Oxford University Press, 1993.

[2] R. Boulic, T.K. Capin, Z. Huang, P. Kalra, B. Lintermann, N. Magnenat-Thalmann, L. Moccozet, T. Molet, I.S. Pandzic, K. Saar, A. Schnitt, J. Shen, and D. Thalmann. The humanoid environment for interactive animation of multiple deformable human character. In *ComputerGraphics Forum, Eurographics'95*, 1995.

[3] R. Boulic and O. Renault. 3d hierarchies for animation. In Eds. Wiley N.M. Thalmann, D. Thalmann, editor, *In in New Trends in Animation and Visualization*, pages 59–77, 1991.

[4] CAESAR. Digitally defining the human body. Available at `http://www.sae.org/technicalcommittees/caesarhome.htm`, SAE International, 2003.

[5] John Chadwick, David Haumann, and Richard Parent. Layered construction for deformable animated characters. In *Computer Graphics, vol.23, no 4*, pages 243–252. Proceedings SIGGRAPH, 1989.

[6] Mete Ciragan. Milkshape 3d 1.4.0 file format specification. `http://astronomy.swin.edu.au/~pbourke/geomformats/ms3d/ms3dspec.h`.

[7] Prof. Peter Comninos. "defining hierarchic objects", 3d computer animation. In *National Centre for Computer Animation, Bournemouth University, Bournemouth, UK.*

[8] Chad Cox. Cs girl tutorial. Available at `http://www.dosfx.com/tutorials/csgirl/csgirl001.asp/`.

[9] David Forsey and LiFeng Wang. Multi-resolution surface approximation for animation. Technical Report TR-92-36, 30, 1992.

[10] H-Anim. Information technology computer graphics and image processing humanoid animation (h-anim). Available at `http:http://www.h-anim.org/`.

[11] kuffner. J. and Latombe. J.C. Fast synthetic vision, memory, and learning for virtual humans. In *Proc. Computer Animation'99, IEEE*, pages 118–127, 1999.

[12] Koga. Y. Kondo. K. kuffner. J. and Latombe. J.C. Planning motions with intentions. In *Proc. SIGGRAPH'94, ACM*, pages 395–408, 1994.

[13] Jeff Lander. On creating cool real-time 3d. Available at `http://www.gamasutra.com`.

[14] N. Magnenat-Thalmann and D. Thalmann. Image synthesis and 3d computer animation: a new approach for strategic analysis. In *The Visual Computer, Springer, vol.3*, pages 51–56, 1987.

[15] W. Maurel and D. Thalmann. Human upper limb modeling including scapulo-thoracic constraint and joint sinus cones. pages 203–218. Computers and Graphics, Pergamon Press, 24 2, 2000.

[16] J. Pettré, J.-P. Laumond, and T. Siméon. A 2-stages locomotion planner for digital actors. In *In Eurographics/SIGGRAPH Symposium on Computer Animation*, 2003.

[17] Baerlocher P., Boulic R. *Parametrization and range of motion of the ball-and-socket joint.* Kluwer Academic Publishers, November 2000. Proc. of AVATARS'2000 Conference, Lausanne.

[18] Nick Gebbie, Steve Rotenberg, and Cameron Chrisman. Computer animation (University of California at San Diego). `http://graphics.ucsd.edu/courses/cse169_w05/`, 2005.

[19] Hyewon Seo. *Parameterized Human Body Modeling.* PhD thesis, Geneve University, 2004.

[20] Chumbalum Soft. Milkshape 3d. `http://www.swissquake.ch/chumbalum-soft/md2v/`.

[21] Mark L. Swinson. and David J. Bruemmer. Expanding frontiers of humanoid robotics. In *From the IEEE Intelligent Systems Special Issue on Humanoid Robotics, July/August*, 2000.