June 1994

Building Anthropometry-Based Virtual Human Models

Francisco Azuola
University of Pennsylvania

Norman I. Badler
University of Pennsylvania, badler@seas.upenn.edu

Pei-Hwa Ho
University of Pennsylvania

Ioannis Kakadiaris
University of Pennsylvania

Dimitris Metaxas
University of Pennsylvania

See next page for additional authors

Recommended Citation

Azuola, F., Badler, N. I., Ho, P., Kakadiaris, I., Metaxas, D., & Ting, B. (1994). Building Anthropometry-Based Virtual Human Models. Retrieved from http://repository.upenn.edu/hms/54

Building Anthropometry-Based Virtual Human Models

Abstract

Creating realistic virtual humans requires models that resemble real humans both visually and behaviorally. Physical and behavioral fidelity in human modeling is the focus of research and development work at the University of Pennsylvania's Center for Human Modeling and Simulation. In this article, we briefly describe the Center's human modeling paradigm, *Jack®*, and our research activities, including *Jack's* Spreadsheet Anthropometric Scaling System (SASS) and its Free-Form Deformation model.

Comments

Copyright 1994 IEEE. Reprinted from *Proceedings of the IMAGE VII Conference*, June 1994.

Author(s)

Francisco Azuola, Norman I. Badler, Pei-Hwa Ho, Ioannis Kakadiaris, Dimitris Metaxas, and Bond-Jay Ting

Building Anthropometry-Based Virtual Human Models

Francisco Azuola, Norman I. Badler, Pei-Hwa Ho,
Ioannis Kakadiaris, Dimitri Metaxas, and **Bond-Jay Ting**
Center for Human Modeling and Simulation
Department of Computer and Information Science
University of Pennsylvania
200 South 33rd Street
Philadelphia, PA 19104-6389

Abstract

Creating realistic virtual humans requires models that resemble real humans both visually and behaviorally. Physical and behavioral fidelity in human modeling is the focus of research and development work at the University of Pennsylvania's Center for Human Modeling and Simulation. In this article, we briefly describe the Center's human modeling paradigm, *Jack*®, and our research activities, including *Jack*'s Spreadsheet Anthropometric Scaling System (SASS) and its Free-Form Deformation model.

Introduction

In many computer applications, one can find the need for simulating humans. The process of creating virtual human models introduces a series of issues, such as scaling the virtual body, estimating segmentation and joint centers, and obtaining realistic appearance.

Anthropometry, a branch of physical anthropology, studies measurements of the human body. Such measurements are commonly necessary in the design of man-machine interfaces. Knowledge of body size variability within a particular design population is critical if an item of clothing, personal protective equipment, or work station is to be designed to accommodate its intended users. We discuss a spreadsheet system which handles the problem of generating anthropometrically correct virtual human figures.

Related to the problem of scaling the virtual body is that of obtaining the necessary anthropometric information. As one option, a standardized human model can be created, based on a given statistical population, (e.g., the Army General Forces [5]). Alternatively, a given person's dimensions can be used in the creation of a virtual human model. If the starting point is a set of images of the given individual, the automatic extraction of measurements from the images becomes a major issue. We present a novel approach to perform active shape estimation, segmentation, and joint determination from a set of images of a human in motion.

Finally, we introduce a flexible, virtual body model, based on deformable segments (via free form deformation). Most articulated virtual human models consider the human body segments as rigid bodies, which introduces problems of smoothness and continuity. Using free form deformation we overcome these difficulties, obtaining a virtual model with more realistic appearance.

Jack

The Center's core technology is an interactive, programmable graphics platform named *Jack* [2]. Ongoing research with *Jack* focuses principally on two areas: human behavior simulation and human factors engineering. Human behavior simulation covers physical human behaviors, such as walking, balance, climbing, and grasping. Human factors engineering uses *Jack* as a design tool that supports testing a product for human usability. For example, in automobile driver compartment design, one may be concerned with the driver's visibility or access to controls. In designing clothes, one may be concerned with human body proportions. For surgery, a surgeon might want to know if an incision will be large enough to access and repair a wound.

A *Jack* virtual environment is made up of any number of 3-D figures. Each figure represents an individual object or agent, and is broken down into geometric segments. The segments of a figure are linked by joints. By default, *Jack* represents a segment as a 3-D polyhedra comprised of the points, edges, and faces of its surfaces. However, *Jack* provides a mechanism to integrate other representations for segments, such as parametric descriptions.

Jack's human models exhibit real-time inverse kinematics and constraints, and human behavior functions. Given the desired position for the end of an articulated (connected by joints) set of segments, inverse kinematics determines the appropriate angles for the intermediate joints. The real-time implementation of inverse kinematics makes end-effector motion such as "move arm" or "bend torso" easy to handle. All the angle computations are hidden. Human behavior functions, such as viewing cones, for “seeing” what the agent sees, balance, for automatically adjusting posture based on the agent's center of mass, and collision detection, make the human model more realistic than other systems. *Jack* is described in detail in [2].

Human Models

Our human model is represented by a set of segment geometries connected by joints [10]. Not all joints in the human body are represented by a joint in the model, e.g., the sternoclavicular joint. Joints that do get represented in the model are the synovial joints [2] with one to three degrees of freedom and specified joint limits. Special modules are created to model more complicated joint complexes, such as the shoulder [13, 2] and spine [9, 2]. Joints in the model are defined in such a way that they rotate about the approximate anatomical joint centers in real humans.

Polyhedral Human Models

The polyhedral human model is composed of simple geometries totaling about two thousand polygons for easy manipulation and scaling(Fig. 3). To create a model of a different size we simply change the appropriate scaling factors associated with each segment. Scaling is done linearly, i.e., each segment is stretched or shrunk by the same amount and the scaling in width and thickness is kept the same between neighboring segments to avoid discontinuities.

Realistic Models

We have adopted models from Viewpoint Animation Engineering to create realistic human models(Fig. 11). These models have more than twenty thousand polygons each and thus require faster machines to manipulate. The same linear scaling techniques can be used to create models of different sizes. We are also developing non-uniform scaling techniques to enable us to create models that can reflect different somatotypes, thus a muscular person will have a different appearance from an obese person of the same weight and stature.

SASS

SASS [1], **S**preadsheet **A**nthropometric **S**caling **S**ystem, is a spreadsheet-like system which allows flexible interactive access to all anthropometric variables needed to size a computer-based human figure.

SASS enables the user to create precise computer human models, that can be manipulated using the *Jack* software.

On one hand, one can create a standardized human model, based on a given statistical population, e.g., Army General Forces [5]. These type of models have the desirable property of being generated with statistically smoothed data, which make them ideal for testing purposes, such as sizing of working spaces.

On the other hand, one can use a specific person's dimensions in the creation of a virtual human model. In this case, the necessary dimensions could be extracted, for instance, from a set of images of a particular individual, then processed, and finally mapped into a generic computer-based body, to obtain the proportions of the original (real) person.

Data that may be accessed in **SASS** is organized into four “groups”: segment dimension (“girth”), joint limits, center of mass, and strength, all of which work based on statistical population data. **SASS** creates structural descriptions of virtual human figures based on this data.

The data required for the graphical representation of a realistic human figure falls into three broad categories:

1. General Body Attributes, such as Standard Anthropometric Measurements (SAM's),
2. Body Segment Information, such as the segment name, girth, mass, and
3. Body Joint Information, such as joint name, type, and limits in the range of motion about the joint.

SASS was developed to support the last two categories, i.e., body segment and joint information. For every human figure model there are many segments and joints to be considered. A complete copy of the body segment information must be stored for each segment in the entire body and similarly for body joint information. This results in large data files for each individual figure stored. An anthropometric database comes as the perfect tool for storing and manipulating this data. **SASS** works as a relational spreadsheet that connects together a full set of parameters describing the human body. Additionally, **SASS** provides access to an anthropometric database containing anthropometric data from “real-world” individuals.

SASS supplies the dimensions of each segment for a virtual human figure, based upon population data supplied as input. The human model generated by **SASS** consists of sixty eight segments (body structures), of which sixty seven have a geometrical representation. For each of these segments, there are three dimensions required, namely, length, width, and thickness. Also, the body joint information for the twenty joints considered must be available.

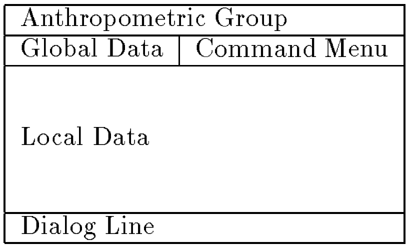


Figure 1: Anthropometric Spreadsheet Layout

The psurf (polyhedral surface) [2] geometry of each segment is scaled by real measurements for a person or percentile measurements for some specifiable population. **SASS** generates "figure files" with the appropriate description of the segment dimensions and joint limits of the virtual human figure.

The **SASS** screen, shown diagrammatically in Fig. 1, is divided into different sections including anthropometric group selection, global data, command menu, local data.

The global data section of the spreadsheet is intended to allow a global view of the current figure parameters. Currently the items considered are: **population, gender, figure model, mass, stature**, and **overall percentile** of any human figure.

The local data section is used for the display of individual segment data and their corresponding percentiles.

Applications

One can find many applications for anthropometric design. In many instances, it is necessary to know body dimensions of the possible users of a product. For example, child cribs must be dimensionally safe to avoid an accident like their heads getting caught between the bars. Automobile design requires a good deal of anthropometry considerations for security and comfort.

SASS enables the user to create precise computer human models, which can be manipulated using *Jack*. As one option, a user can create a standardized human model, based on a given statistical population, (e.g., the Army General Forces [5]). Alternatively, a given person's dimensions can be used in the creation of a virtual human model. Fig. 2 shows the front and side images of a real person. Data is extracted from these images [3], in order to create a virtual human such as that of Fig. 3.

Figures created by **SASS** can be used in multiple situations. For example, consider the task of correctly sizing a driver's seat. Fig. 4 shows three different sizes of humans within such an environment.

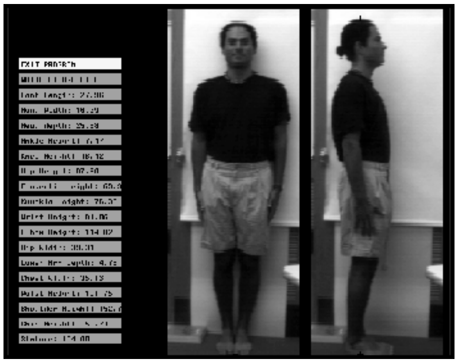


Figure 2: Real Human

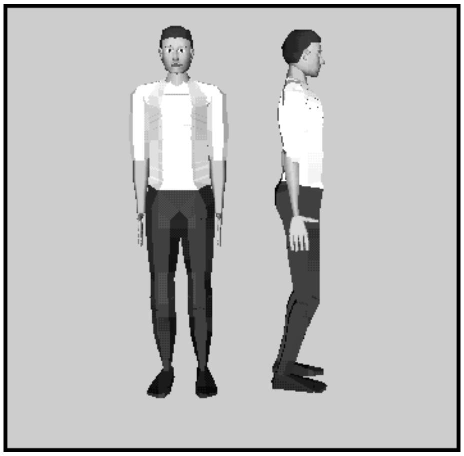


Figure 3: Generated Virtual Human

Active Shape Estimation, Segmentation and Joint Determination of Humans

In this section, we present a novel, active, integrated approach to identify reliably the parts of a moving human and estimate their shape and motion (see [6] for more details). We present the determination of joint locations and some of the experiments demonstrating the technique.

Basic Notions

We developed [12, 7, 8] a physics-based framework which provides deformable models with broad geometric coverage along with robust techniques for inferring shape and motion from noise-corrupted data. Deformable models unify the representation of free-form shape and nonrigid motion.

The positions of points on a model relative to an inertial frame of reference Φ in space are given by $\mathbf{x}(\mathrm{u},t) = (x_1(\mathrm{u},t), x_2(\mathrm{u},t), x_3(\mathrm{u},t))^T$, where ^{T} denotes transposition. We set up a noninertial, model-centered reference frame ϕ and express the position function as

$$\mathbf{x} = \mathbf{c} + \mathbf{R}\mathbf{p} \tag{1}$$

Figure 4: Scaled human figures

where $\mathbf{c}(t)$ is the origin of ϕ at the center of the model and the rotation matrix $\mathbf{R}(t)$ gives the orientation of ϕ relative to Φ. Thus, $\mathbf{p}(\mathrm{u}, t)$ gives the positions of points on the model relative to the model frame. We further express $\mathbf{p} = \mathbf{s} + \mathbf{d}$ as the sum of a global reference shape $\mathbf{s}(\mathrm{u}, t)$ and a local displacement $\mathbf{d}(\mathrm{u}, t)$.

We define the global reference shape as $\mathbf{s} = \mathbf{T}(\mathbf{e}(\mathrm{u}; a_0, a_1, \ldots); b_0, b_1, \ldots)$. Here, a geometric primitive $\mathbf{e}$, defined parametrically in u and parameterized by the variables a_i, is subjected to the *global deformation* $\mathbf{T}$ which depends on the parameters b_i. As an example of global deformations, we consider the case of parameterized piecewise bending deformation that ensures constant curvature along the major axis of bending. The domain of bending is a bounded subspace of the Euclidean space R^3. This domain is partitioned into three non-intersecting zones: the *fixed zone*, the *bending zone* and the *relocation zone*. The fixed zone remains unchanged during bending. The relocation zone is translated and rotated rigidly.

Through the application of Lagrangian mechanics, we developed a method [7] to systematically convert the geometric parameters of the solid primitive, the global (parameterized) and local (free-form) deformation parameters, and the six degrees of freedom of rigid-body motion into generalized coordinates or dynamic degrees of freedom.

Active Part-Identification Shape and Motion Estimation

Instead of estimating the shape and motion of complex objects under the assumption of prior segmentation, our technique allows active, simultaneous segmentation and fitting. To identify the location of the articulation of the objects, we use a sequence of images which contain different postures of the moving object. When we observe an articulated object in a posture where the articulations are not detectable, we assume initially that the object consists of a single part. We then fit a deformable model to the given time-varying data and recover the relevant model parameters. As the object moves and attains new postures, we decide if and when to replace the initial model with two new models. This decision is based on the error of fit, the rate of change and magnitude of the bending deformation and the continuity of the given data within the bending region.

When the criteria associated with the measures mentioned above are satisfied we decompose the model into two models. We identify the data that correspond to the fixed, bending and relocation zones of the initial model based on the estimated bending parameters b_1 and b_2 and the image projection assumptions (orthographic or perspective). We then initialize the two models to the data that correspond to the fixed and relocation zone respectively. The data though that correspond to the bending region of the initial model are marked as *orphan datapoints* since we are uncertain as to which of the two new models they should be assigned. This is necessary since we do not know in advance the extent of each of the two models. Our goal then is to fit the two new models to the given data. Furthermore, we would like them to fit in a way that allows partial overlap between the two parts. Unlike conventional geometric models, deformable models adapt their shapes and move in response to simulated forces as if they were made of nonrigid materials such as rubber. Since we know to which model the data in the fixed and relocation zones belong we use our previously developed algorithm for assigning forces from datapoints to points on the model. To assign forces from the orphan data to the two models we use a novel algorithm, that allows the weighted assignment of a given orphan datapoint to both deformable models [6]. We compute these weights, whose sum is always equal to one, by minimizing an appropriately selected energy expression. Once we compute all the forces from the datapoints to the two models we estimate the shape and motion of the two new models. An important property of our new force assignment algorithm is that it allows partial overlap between the two models at a joint, whose location we can determine. Once the joint has been identified we initiate a point-to-point constraint. Having determined the location of the joints and using a Kalman Filter to predict the location of each part, we can successfully address the case where parts of the articulated object are occluded during the motion.

Determination of Joint Location

Let's assume that we have estimated the shape and motion of the two parts of an articulated object at times t and $t + \delta t$, by fitting two models m_0 and m_1. We would like then to identify the location of their common joint.

The unknown location of the center of the joint can be expressed using (1) in terms of the parameters of model m_0 at times t and $t + \delta t$ as:

$$\begin{aligned} \mathbf{x}_0(t) &= \mathbf{c}_0(t) + \mathbf{R}_0(t) p_0 \\ \mathbf{x}_0(t+\delta t) &= \mathbf{c}_0(t+\delta t) + \mathbf{R}_0(t+\delta t) p_0, \quad (2) \end{aligned}$$

and with respect to the parameters of model m_1 at times t and $t + \delta t$ as:

$$\mathbf{x}_1(t) = \mathbf{c}_1(t) + \mathbf{R}_1(t) p_1$$

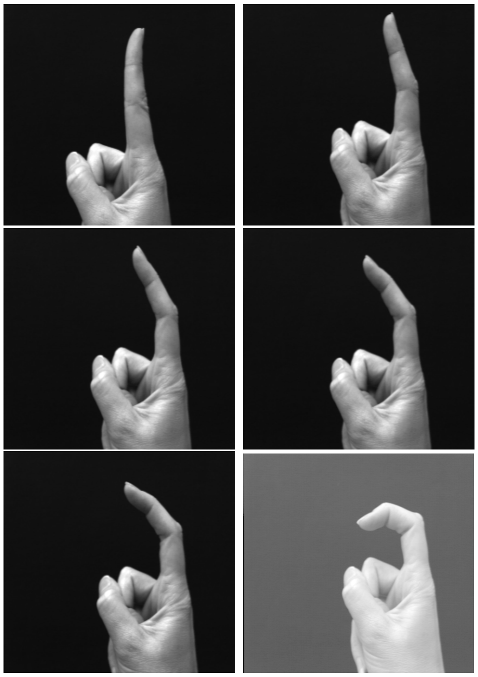


Figure 5: Segmentation, shape and motion estimation of a human finger. A sample of the image sequence.

$$\mathbf{x}_1(t+\delta t) \quad = \quad \mathbf{c}_1(t+\delta t)+\mathbf{R}_1(t+\delta t)p_1. \quad (3)$$

Under the obvious assumption that

$$\begin{aligned} \mathbf{x}_0(t) &= \mathbf{x}_1(t) \\ \mathbf{x}_0(t+\delta t) &= \mathbf{x}_1(t+\delta t) \end{aligned} \quad (4)$$

and by subtracting the above two equations, we arrive at the following system of equations, with unknowns the locations $\mathbf{p}_0$ and $\mathbf{p}_1$ of the join t with respect to the model reference frames of the two models,

$$\begin{bmatrix} \mathbf{R}_0(t) & -\mathbf{R}_1(t) \\ \mathbf{R}_0(t+\delta t) & -\mathbf{R}_1(t+\delta t) \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \end{bmatrix}$$

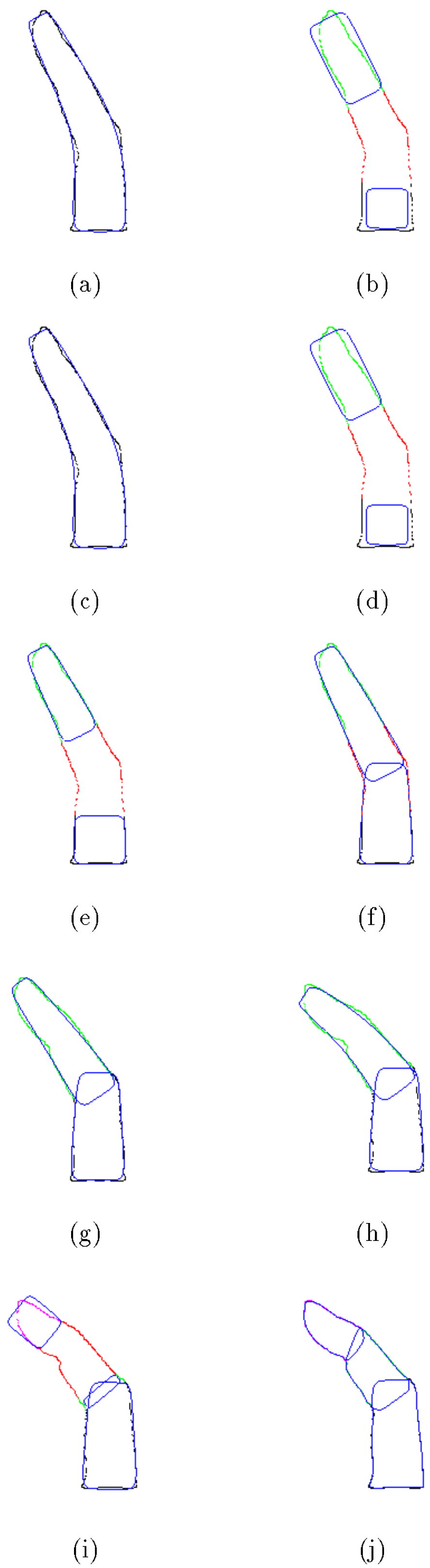
$$= \begin{bmatrix} \mathbf{c}_1(t)-\mathbf{c}_0(t) \\ \mathbf{c}_1(t+\delta t)-\mathbf{c}_0(t+\delta t) \end{bmatrix} \quad (5)$$

which can be easily solved. If, due to noise in the data, the location of the joint varies between frames, then we follow a Kalman filter based approach [8]. In this way we can robustly estimate the locations of the joints of an articulated object.

Experiments

We present experiments demonstrating our integrated approach to segmentation shape and nonrigid motion from motion image data obtained from a single view.

In the first experiment, we use image data obtained from a human finger moving in a plane



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 6: Segmentation, shape and motion estimation of a human finger.

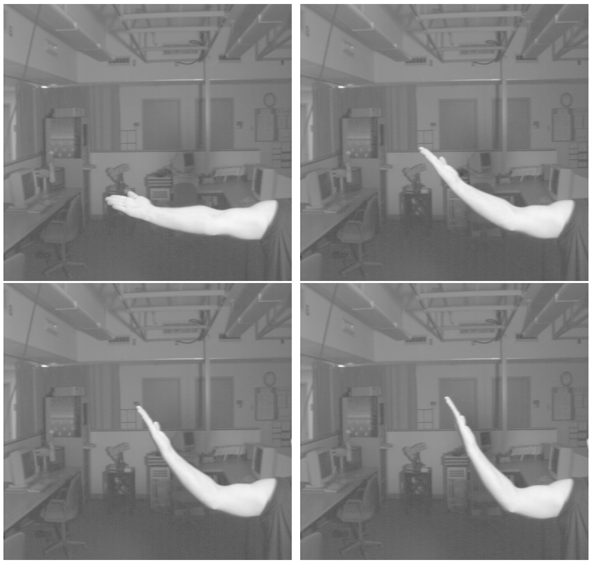
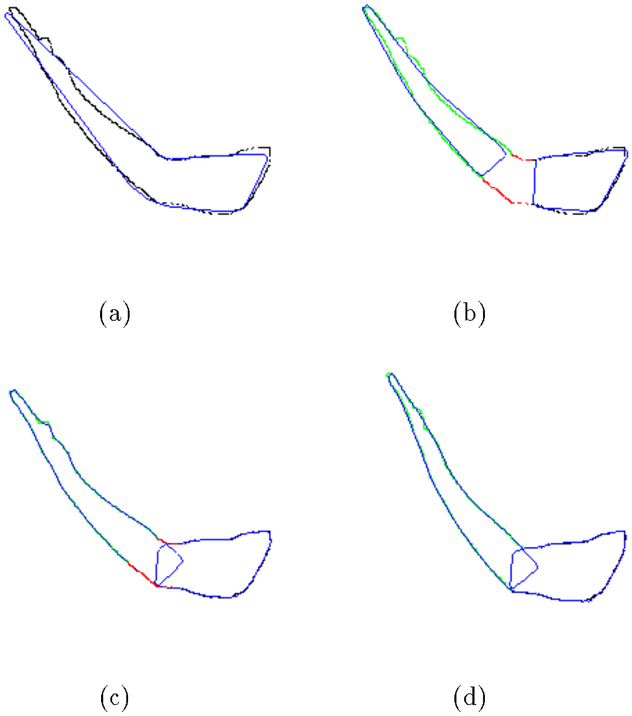


Figure 7: Segmentation, shape and motion estimation of a human arm. A sample of the image sequence.



(a)

(b)

(c)

(d)

Figure 8: Segmentation, shape and motion estimation of a human arm

(Fig. 5). Figs. 6(a-b) shows the models fitted image frames prior to segmentation. Fig. 6(c) shows the model fitted to the image frame where the partitioning criteria are satisfied and the hypothesis that the object is comprised from two parts is generated. Figs. 6(d-f) demonstrate the fitting of the two new models to the image data. Fig. 6(d) shows the initialization of the new models, Fig. 6(e) shows an intermediate step in the fitting process, while Fig. 6(d) shows the finally fitted models. The overlap between the two models allows us to compute robustly the location of the joint over several frames and place a point-to-point constraint between the two models. Fig. 6(f) shows the models fitted to a new frame, while Fig. 6(g) shows the models fitted to the frame where the partitioning criteria are satisfied for the upper model and the hypothesis that the upper model should be replaced by two new models is generated. Fig. 6(i) shows the initialization of the two new models based on our technique, while Fig. 6(j) shows all three models fitted to the given data.

In the second experiment, we use the image data obtained from a observing the planar motion of a human arm (Figs. 7). Since the partitioning criteria are satisfied at Fig. 8(a), we initialize two new models and fit them to the given data (Figs. 8(b). The result of fitting is depicted in Fig. 8(c). Fig. 8(d) depicts the fitted models to a subsequent frame.

Deformable Model

A human body is a combination of rigid body and deformable tissue. An articulated bone structure forms the basis of the human shape. Wrapped around the bone structure, there are different layers of soft tissues such as muscles and fats. Skin, the outermost layer of the human body, is a contiguous deformable tissue which makes the joints between the human body segments continuous and smooth.

Most articulated human models treat the body as a set of rigid object segments based on the underlying bone structure. Although this simplifies the computation, it also creates discontinuity and smoothness problems. There are at least two ways to solve these geometric problems.

One way is to create overlap between segments. The continuity and smoothness of this approach depends on the shape design of the overlap. The advantage of this approach is that every segment remains a rigid body. Although more nodes and faces are introduced to create the overlap, the total computation time is still efficient. This approach works well on long segments such as legs and arms which have large “length vs. cross section ratios”. But, for short segments such as those of the torso, the overlap area is hard to design and the result is seldom attractive.

The second method is to use a deformable model.

In a deformable model, the local coordinate for each node is changed based on joint angles. Direct function control is one way to describe the relationship between each segment node and joint angle. Each node is assigned a function. Each degree of freedom of each related joint is a parameter in the function. This method gives maximum flexibility. Each function can be designed independently from others. This characteristic also makes this approach difficult for modeling. A segment which has two three-degree-of-freedom joints, will create six parameter functions. Computation of these functions also makes this approach the most expensive model. As an alternative, a "two stage indirect control model" is introduced.

Two Stage Indirect Control Model

In this approach, we separate articulated motion and local deformation into two control levels. A set of control points is introduced. The lower control level is between segment nodes and control points. For each segment node, there is a simple function of one or more control points. These functions are used to control the segment local deformations. In a higher control level, the motion of each control point is a direct function of the joint angles. The flexibility advantage from the previous approach is traded for computation time and easier modeling.

Free Form Deformation - Lower Control Level

There is more than one way to implement the lower control level. In our case, we choose B-spline based free form deformation. Although free form deformation was originally designed as a tool in modeling an object from a simple primitive [11], it can also be used as an animation tool. In our case, we implement a third order B-spline based free form deformation. There are several nice properties obtained by choosing free form deformation.

- *Continuity and smoothness:* As mentioned above, this control level handles local deformations. Maintaining smoothness and continuity becomes the main goal in this level. As we choose third order B-spline functions, a C^2 continuity is guaranteed [4].
- *Relatively inexpensive in computation time:* Generally speaking, the total number of control points is much smaller than the total number of segment nodes. This means that the total number of functions in the lower control level is much larger than the total number of functions in the higher control level. Simplifying functions in the lower control level will significantly reduce the computation time. A third order B-spline function is simply a second order polynomial function [4].

Control Mesh Design - Higher Control Level

Since the free form deformation is not designed for an articulated model, there are some undesired properties which make control mesh design difficult. In a deformable model, segments are classified into two categories. Due to the different characteristics for both groups, different design criteria are implemented:

- *Articulated segments:* The segments in this group, such as legs and arms, are similar to the long segment in traditional articulated modeling. That is, the distance between joints is large compared to the segment cross section.

  Since the distance between joints is large, the articulated effect limits the bending deformation in a small region. Therefore, when designing a control mesh for an articulated segment, we need more layers of mesh on the joint area. Fig. 9 and Fig. 10 are the example of designing a control mesh. In Fig. 9, a more dense mesh is put on the knee joint position. When the knee is bent in Fig. 10, the bending deformation is limited on the joint area. Other areas only receive stretching (or compression) deformation.
- *Torso-like segments:* This group, including torso and neck segments, has a joint chain. Inside the joint chain, the distance between two consecutive joints is small compared to the cross section.

  Due to this face, the soft tissue has the tendency to smooth out the articulated effect. Therefore, the whole segment receives bending deformation. To design the control mesh, assigning a layer of mesh to each joint is sufficient. As the joint angle changes, the corresponding layers of mesh are rotated. The problem in this group is with the correct joint location. As part of the characteristic of spline curves, the joint position after rotation is slightly off the correct position. This problem will be accumulated when applied to a joint chain, such as the spine. To correct this problem, a correction program is applied after each rotation. Fig. 11 is a deformable upper torso which uses an eighteen layer control mesh to simulate eighteen joints.

Conclusions

Clearly, virtual humans are an important part of many applications. *Jack* provides a flexible environment for the creation and manipulation of virtual humans with both realistic appearance and behavior.

Acknowledgement

This research is partially supported by ARO DAAL03-89-C-0031 including U.S. Army Research Laboratory (Aberdeen), Natick Laboratory; U.S. Air Force DEPTH through Hughes Missile Systems F33615-91-C-0001; Naval Training Systems Center N61339-93-M-0843; Sandia Labs AG-6076; DMSO through the University of Iowa; NASA KSC

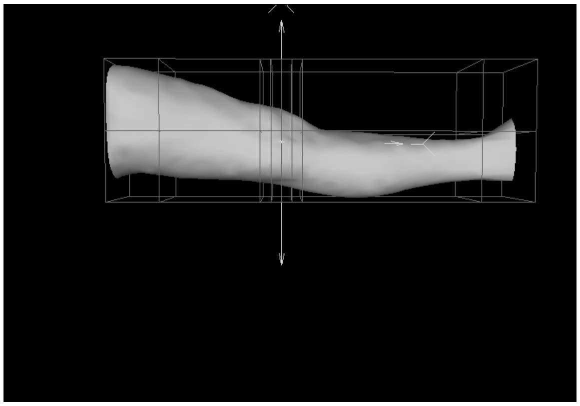


Figure 9: Deformable leg without bending

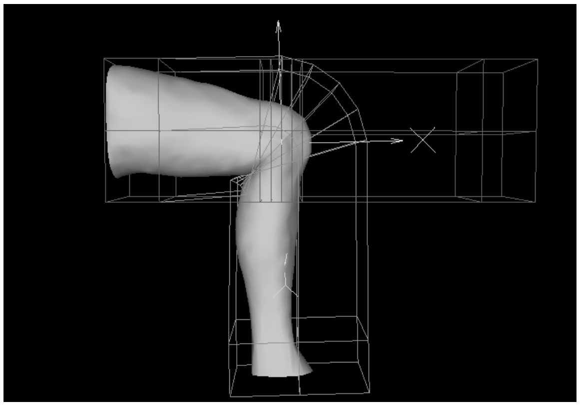


Figure 10: Deformable leg with 90 degree bending

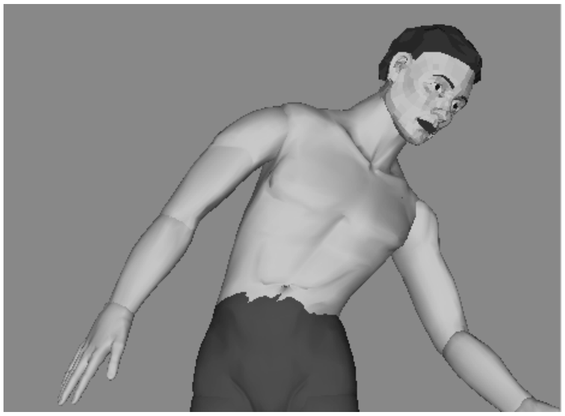


Figure 11: Deformable torso

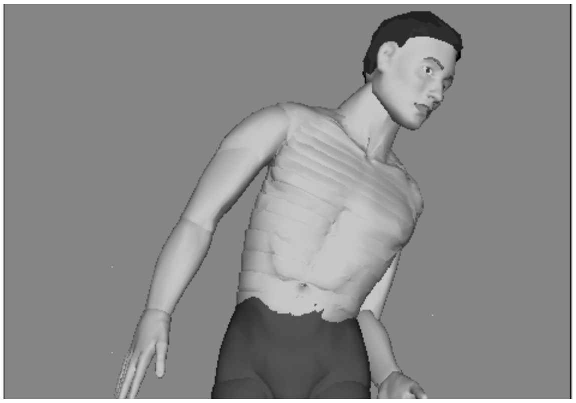


Figure 12: Segmented torso

NAG10-0122; MOCO, Inc.; NSF CISE CDA88-22719, and Instrumentation and Laboratory Improvement Program #USE-9152503.

References

[1] Francisco Azuola, Norman I. Badler, Teo K. Hoon, and Susanna Wei. Sass v.2.1 anthropometric spreadsheet and database for the iris. Technical report, Department of Computer and Information Science, University of Pennsylvania, 1993.

[2] Norman I. Badler, Cary B. Phillips, and Bonnie Lynn Webber. *Simulating Humans, Computer Graphics Animation and Control.* Oxford University Press, New York, 1993.

[3] Wilfred Taylor Dempster. Space requirements of the seated operator. Technical Report WADC TR 55-159, Wright-Patterson Air Force Base, Ohio, July 1955.

[4] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric.* Academic Press, INC., San Diego, CA, 1990.

[5] Claire C. Gordon, Bruce Bradtmiller, Thomas Churchill, Charles E. Clauser, John T. McConville, Ilse Tebbetts, and Robert A. Walker. 1988 anthropometric survey of u.s. army personnel: Methods and summary statistics. Technical Report Natick/TR-89/044, U.S. Army Natick Research, Development and Engineering Center, September 1989.

[6] I.A. Kakadiaris, D. Metaxas, and R. Bajcsy. Active part-decompostion, shape and motion estimation of articulated objects: A physics-based approach. *IEEE Computer Vision and Pattern Recognition Conference*, 1994.

[7] D. Metaxas. *Physics Based Modeling of Nonrigid Objects for Vision and Graphics.* PhD thesis, Department of Computer Science, University of Toronto, Toronto, Canada, 1992.

[8] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics based synthesis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.

[9] Gary Monheit and Norman I. Badler. A kinematic model of the human spine and torso. *IEEE Computer Graphics and Applications*, 11(2), 1991.

[10] Cary B. Phillips. Jack 5 user's guide. Technical Report MS-CIS-91-78, Department of Computer and Information Science, University of Pennsylvania, 1991.

[11] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. *Proceedings of SIGGRAPH '86*, 20(4):151–160, 1986.

[12] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: Deformable superquadrics. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991.

[13] Jianmin Zhao. *Moving Posture Reconstruction from Perspective Projections of Jointed Figure Motion*. PhD thesis, University of Pennsylvania, 1993.